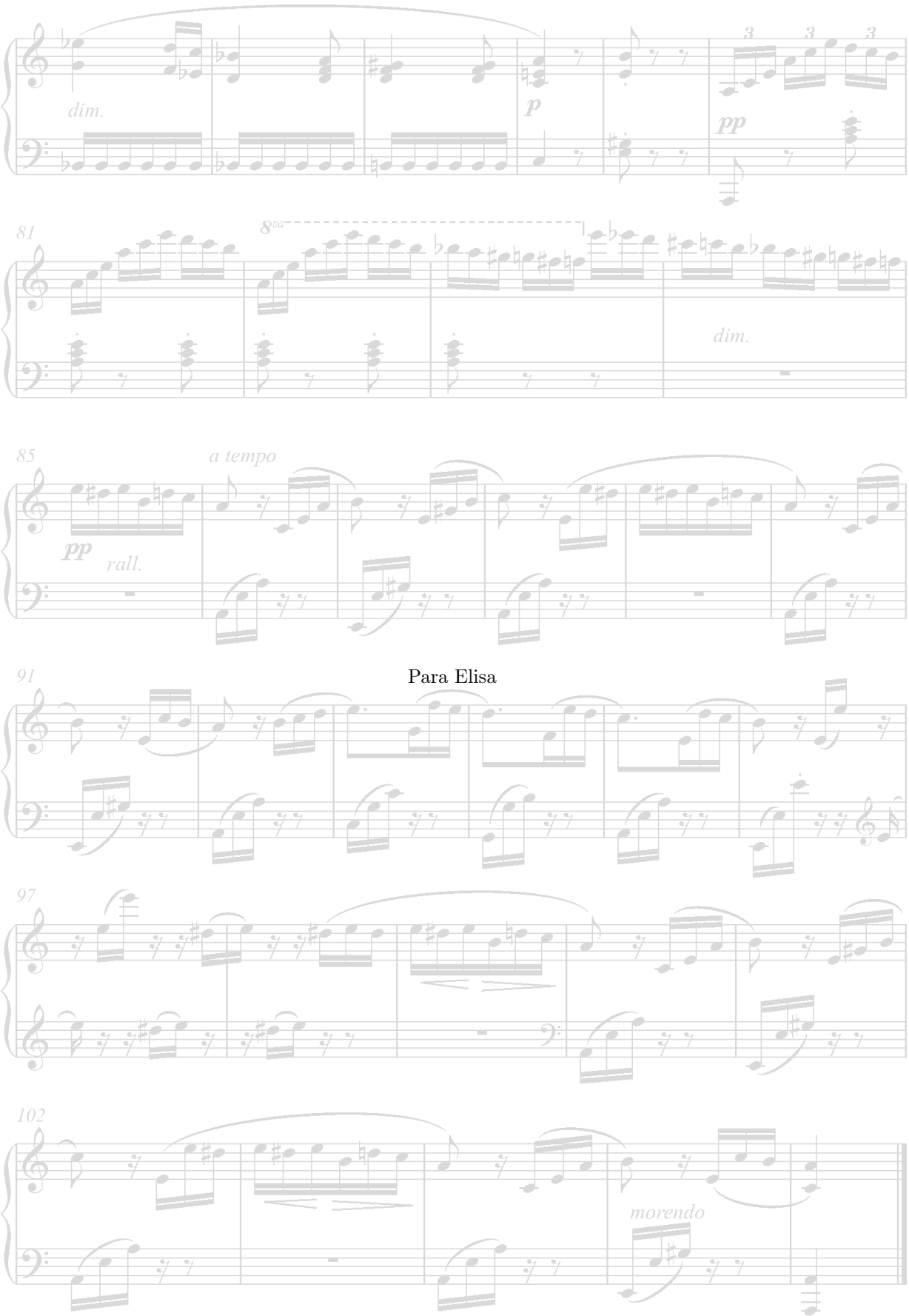
Lógica para Informática y Matemáticas

Camilo Rocha

Email address: me@camilorocha.info



Para Elisa

Última actualización 6 de agosto de 2022.
Versión 0.7

Índice general

Prefacio

Históricamente, la lógica matemática ha sido vista como un objeto de estudio y sus bondades como álgebra del razonamiento han sido relegadas a un segundo plano. Esta disposición no es sorprendente si se tiene en cuenta que las matemáticas avanzaron como ciencia durante siglos sin la ayuda de la lógica matemática. Además, la lógica matemática es un área del conocimiento relativamente nueva al ser comparada con el resto de las matemáticas.

El contexto en cual se desarrollan las matemáticas ha cambiado, sobre todo en esta era digital. Hoy en día la tecnología ha permeado la labor del matemático desde la informática: es posible hacer, por ejemplo, demostraciones asistidas o automáticas en un computador. Para llegar a esta realidad, la informática ha desarrollado nuevos marcos lógicos con una bondad especial: la de servir como álgebras del razonamiento efectivas y gozar de buenas propiedades computacionales, facilitando su mecanización en un computador por medio de inferencias simbólicas. Estos desarrollos también han abierto nuevas posibilidades para contar con nueva notación, demostraciones más legibles y argumentaciones que finalmente resultan en cálculos muy precisos. En este sentido, el rol de la lógica matemática ha cambiado para dejar de ser exclusivamente un objeto de estudio y convertirse en una herramienta efectiva para demostrar teoremas y, en general, para guiar el proceso de razonamiento en muchas áreas del conocimiento.

El propósito de este texto es servir como introducción a la lógica matemática clásica con énfasis en su uso como herramienta de deducción – para documentar y descubrir razonamientos formales– y no solo como objeto de estudio. Una característica particular, y que posiblemente haga de este texto un aporte único, es que el desarrollo del estudio de la lógica se aborda desde tres dimensiones: (i) los *sistemas formales* como marco fundamental de las matemáticas, concebidos inicialmente por

Russell y Whitehead [**11**], y Hilbert y Bernays [**5**]; (ii) la propuesta desde la informática de usar la lógica como herramienta para razonar hecha por Dijkstra y Scholten [**2**], y refinada posteriormente por Gries y Schnider [**3**] y Tourlakis [**10**]; y (iii) resultados recientes en reescritura y lógica computacional [**8, 9**], áreas activas de investigación en métodos formales de la informática.

Dado que el principal énfasis de este texto es en el uso de de la lógica matemática como herramienta para razonar, la dimensión de los sistemas formales es un marco conveniente y riguroso para la formulación de los fundamentos de la lógica. Un sistema formal permite hacer cálculos a partir de una colección de fórmulas distiguidas (llamadas *axiomas*) y unas reglas de manipulación sintáctica (llamadas *reglas de inferencia*). De este modo la lógica, tanto proposicional como de predicados, se presenta como sendos sistemas formales que resultan en juegos de manipulación sintáctica. Claramente, desde la perspectiva de un estudiante que quiere entender qué calcula, es importante asociar nociones de significado a las fórmulas de los sistemas formales lógicos. Para este fin, en este texto se estudian las semánticas proposicional y de predicados, como marcos de referencia conceptual y complementaria al de los sistemas formales lógicos.

El estilo calculatorio de Dijkstra y Scholten ha sido utilizado, principalmente, en el área de métodos formales de la informática durante los últimos 40 años. Su uso en el aula de clase se ha visto concentrado en el mismo planteamiento de la lógica como herramienta de estudio y en las matemáticas discretas durante las últimas dos décadas. El poder de este sistema lógico está relacionado directamente con el hecho de que se privilegia la equivalencia lógica, en lugar de la implicación, como relación para calcular demostraciones. Dado que la equivalencia lógica es un conectivo lógico más fuerte que la implicación lógica, el estilo calculatorio de Dijkstra y Scholten puede también ser usado para cálculos lógicos basados en la implicación, es decir, en este estilo de hacer lógica hay lugar para hacer deducciones con un estilo tradicional en las matemáticas. Sin embargo, a pesar de su aparente potencial académico y didáctico, este estilo de "hacer matemáticas" ha pasado largamente desapercibido por la comunidad académica y, en algunas ocasiones, ha sido objeto de crítica por su formulación. Posiblemente, esto se debe a la tradición en matemáticas en usar el lenguaje coloquial para desarrollar demostraciones, al hecho de que el estilo calculatorio de Dijkstra y Scholten propone algunos cambios en notación e introduce nuevas convenciones para escribir demostraciones. En este texto se materializa el concepto de "demostración" à la Dijkstra y Scholten, y se relaciona directamente con aquel de demostración en un sistema formal. La notación empleada para cuantificar, y para operatorias generalizadas, se presenta como azúcar sintáctico de la notación tradicionalmente utilizada en las matemáticas. De este modo estas notas aclaran al lector que la propuesta de Dijkstra y Scholten desde la informática es en un sistema lógico bien fundamentado.

La lógica computacional se ocupa de mecanizar la lógica matemática y utilizarla como motor de inferencia para razonar acerca de y llevar a cabo computaciones. Recientemente, el autor de este texto, en compañía de otros colaborares, ha estudiado las propiedades computacionales de la lógica de Dijkstra y Sholten como sistema de reescritura de términos, obteniendo resultados inesperados y sorprendentes. Por ejemplo, el sistema formal de la lógica proposicional presentado en este texto puede ser usado como un sistema de reescritura resultando en un procedimiento de decisión para la lógica proposicional. Similarmente, el sistema formal de primer orden presentado en este texto puede ser mecanizado por reescritura, con ayuda de secuentes à la Gentzen, resultando en un procedimiento de semi-decisión para lógica de predicados. Es decir, cada uno de los sistemas lógicos presentados en este texto gozan de las mejores propiedades computacionales posibles (en cada caso). Adicionalmente, relaciones de la lógica de Dijkstra y Scholten con otras teorías como la de anillos Booleanos y la lógica Aristotélica han sido encontradas y documentadas. En este texto se exploran algunas de estas relaciones.

Es abrumadora la cantidad de textos que se han escrito acerca de y sobre la lógica matemática y, recientemente, otro tanto más sobre lógica para informática. Sin embargo, la mayoría de textos en estos temas se escriben para audiencias con cierta maduréz matemática y con el objetivo de estudiar las meta-propiedades de la lógica. Es decir, los textos se centran principal y generalmente en el estudio de la lógica como objeto de estudio, explorando sus posibilidades y limitaciones. Cualquier lector familiarizado con los textos de Hodel [**6**], Hamilton [**4**], y Huth y Ryan [**7**] se dará cuenta de la gran deuda que el autor tiene con ellos gracias al desarrollo de algunas de sus ideas en el presente texto.

Actualmente, hay tres libros que el autor conoce, y que preceden cronológicamente este texto, los cuales se centran en el uso de la lógica matemática como herramienta para calcular demostraciones: uno por Gries y Schnider [**3**], otro por Tourlakis [**10**] y otro por Bohórquez [**1**]. El primero es un texto introductorio que presenta la lógica de Dijkstra y Scholten como una herramienta para demostrar teoremas, en gran medida dejando a un lado aspectos semánticos de la lógica. La segunda parte de ese texto se ocupa, principalmente, en demostrar teoremas de matemáticas discretas que son deseables en la enseñanza de los currículos de computación e informática hoy en día. El segundo, es un texto más avanzado en comparación con el primero y se enfoca, especialmente, en estudiar y presentar los fundamentos matemáticos de la lógica de Dijkstra y Scholten, al igual que la interrelación entre su aparato formal y su semántica. El tercer texto, además de estar escrito en castellano, presenta y extiende las ideas en [**3**] abordando aspectos semánticos de la lógica e incluyendo pruebas de completitud y coherencia, además de numerosos ejemplos que ilustran este enfoque en la corrección y derivación de algoritmos iterativos. El presente texto puede verse como una extensión adicional de los textos anteriormente mencionados, en donde el sistema de Dijkstra y Scholten se

presenta exhibiendo sus propiedades computacionales, acompañado de resultados recientes en investigación.

Este texto se formó a partir de notas de clase escritas para cursos introductorios de lógica enseñados en la Escuela Colombiana de Ingeniería (LCAL y LMAT). La inspiración para escribirlo provino de José Meseguer (tutor de los estudios doctorales del autor en la Universidad de Illinois en Urbana-Champaign) quien, con su ejemplo, "demostró" que no hay una mejor forma de saber acerca de un tema que escribiendo sobre este.

Finalmente, es importante agradecer a Ernesto Acosta, Jaime Bohórquez, Jairo Castrellón, Raúl Chaparro, Wilmer Garzón, Daniela Pérez y Sergio Ramírez porque han leído cuidadosamente este texto, haciendo sugerencias muy útiles sobre la forma y el fondo de su contenido.

Camilo Rocha
Cali, Agosto de 2022

Capítulo 0

Preámbulo

"Desconfíen vuestras mercedes de quien es lector de un solo libro", dijo el Capitán Alatriste.

Arturo Pérez-Reverte
Limpieza de Sangre, 1997

I prioritise story over science, but not at the expense of being really stupid about it.

Alastair Reynolds

Este capítulo contiene material necesario para el resto del texto y debe ser abordado a medida que se necesite. El material en esta sección incluye convenciones y definiciones básicas, y un recuento sucinto acerca de la inducción matemática y los sistemas formales.

0.1. Preludio

Las matemáticas, y en particular la lógica, se desarrollan usando un código común de términos y una metodología. La *terminología* se refiere a aspectos linguísticos que se supone son conocidos y están claramente entendidos. Es en este sentido en el cual los textos de matemáticas adoptan algunos términos como "definición", "proposición" y "teorema", entre otros, y los usan frecuentemente. La *metodología* permite abordar el estudio de un tema de manera sistemática: primero, definiendo unívocamente el objeto de estudio; segundo, estableciendo propiedades sobre el objeto de estudio; tercero, y de manera opcional, utilizando efectivamente el objeto de estudio para resolver un problema de interés. Por objeto de estudio puede entederse, por ejemplo, "cómo demostrar que un algoritmo es correcto", "un

sistema numérico" o "una teoría de la relatividad". Esta sección explica cuáles son el código común de términos y la metodología que sigue este texto.

Siempre, en matemáticas, lo más importante son las definiciones: sin definiciones, o sin entenderlas, no es posible hacer matemáticas; mucho menos pretender que sean de alguna utilidad. El objetivo de una *definición* es "capturar" unívocamente un objeto matemático por medio de sus propiedades básicas y establecer notación para referirse a este. Una definición en este texto hace eso: definir objetos matemáticos y establecer notación para referirse a ellos.

Definición 0.1

El *conjunto de números naturales* es $\{0, 1, 2, \ldots\}$ y se denota como $\mathbb{N}$. Un número n es un *número natural* si y solo si $n \in \mathbb{N}$.

Primero que todo, note que las definiciones vienen identificadas con un código. Así no es necesario referirse a ellas como en "la definición anterior" (puede haber más de una definición antes del texto que la referencia y entonces esta indicación sería ambigua), sino más bien referenciándola directamente como en "la Definición 0.1". En este último caso la palabra "definición" es un nombre propio y por eso su inicial se escribe en mayúscula. Por convención, cuando el nombre propio sea en plural, no se escribe con mayúsculas.

Los tipos de letra son claves para entender una definición. Por ejemplo, en la Definición 0.1 hay dos tipos de letra. Los sustantivos "conjunto de números naturales" y "número natural" están en letra cursiva, lo cual no es un capricho: estos dos son los nombres de los objetos que están siendo definidos. Esto quiere decir, por ejemplo, que en este texto los números naturales son esos y no otros. Como convención se adopta el símbolo $\mathbb{N}$ para representar el conjunto de números naturales. En la segunda parte de la Definición 0.1 la expresión "n" es una *variable matemática* y por eso también aparece en letra cursiva.

En algunas ocasiones se hacen suposiciones de notación y convención. Por ejemplo, en la Definición 0.1 el autor supone conocimiento del lector acerca de qué es un número, de preferir la notación arábica sobre la romana para escribirlos y de entender el significado del símbolo '$\in$'. Es posible tratar de definir todas estas suposiciones formalmente; sin embargo, sería impráctico en el caso de este texto porque desviaría al lector de su objetivo principal.

Una vez un objeto está (unívocamente) definido, es posible entonces proceder a identificar y estudiar sus propiedades. Es en este momento, al haber definido el objeto de interés, que se pueden postular y demostrar propiedades de manera precisa.

Teorema 0.2

Sea n un número. Si $n \in \mathbb{N}$, entonces $(n+1) \in \mathbb{N}$.

El Teorema 0.2 (¡sí!, también tiene nombre) establece una propiedad inductiva de los números naturales que por lo pronto sirve únicamente de ejemplo. La propiedad enunciada en el Teorema 0.2 indica que el *sucesor* (i.e., el siguiente) de cualquier número natural es un número natural. Claramente, esto debe estar acompañado de una demostración rigurosa. En este texto las demostraciones se presentan en un párrafo que inicia con el sustativo "Demostración" en letra negrilla y termina con un cuadro vacío indicando el fin de la demostración, de la siguiente forma:

Demostración. ... □

A diferencia de otras áreas de las matemáticas como el cálculo, el álgebra, etc., el estudio de la lógica se ocupa *simultáneamente* de dos roles de la lógica: como *objeto* de estudio y como *herramienta* para razonar. En este texto se estudian diferentes lógicas y cada una de ellas se define como un *sistema formal* (Sección 0.3). Entonces, dependiendo del ámbito, el tipo de propiedades que se pueden establecer son dos: (i) propiedades *acerca* del sistema formal y (ii) propiedades *dentro* del sistema formal. La lógica, como sistema formal en el caso (i), es un objeto de estudio y las propiedades demostradas son *meta-teoremas* (i.e., propiedades de la lógica como sistema formal). La lógica, como herramienta en el caso (ii), es empleada para obtener resultados dentro del sistema formal y las propiedades demostradas son *teoremas* (i.e., propiedades particulares de las expresiones de la lógica dentro del sistema formal). La convención que sigue el texto es clasificar con el nombre de *Metateorema* a cada meta-teorema y con el nombre de *Teorema* a cada teorema. El texto también usa los nombres *Lema* y *Corolario* para referirse a meta-teoremas y teoremas 'secundarios': un lema es un resultado intermedio que antecede a un resultado importante, mientras que un corolario es un caso particular de un meta-teorema o teorema. La intención de usar estos términos para referirse a resultados, tanto al nivel meta-teórico como del objeto de estudio, es facilitar la lectura del texto; dependiendo del contexto en el cual aparezcan, el lector podrá inferir en cuál de los dos roles se enmarcan dichos resultados.

Finalmente, el texto hace uso de "notas" para resaltar una idea, introducir notación o simplemente llamar la atención del lector de aspectos que complementan el contenido primordial del texto.

Nota 0.3

Para estudiar lógica es necesario entender muy bien las definiciones y hacer demostraciones.

Ejercicios

1. Investigue y enuncie las convenciones notacionales para identificar cada uno de los siguientes conjuntos, explicando la relación que hay entre cada uno de ellos:
   a) Números enteros.
   b) Números racionales.
   c) Números reales.
   d) Números complejos.
2. Investigue y enuncie la definición para cada uno de los siguientes objetos matemáticos:
   a) Número algebraico.
   b) Número irracional.
   c) Número trascendental.
3. Investigue y enuncie la definición para cada uno de los siguientes objetos matemáticos:
   a) Conjunto.
   b) Conjunto finito.
   c) Conjunto infinito.
4. Investigue y enuncie la definición para cada uno de los siguientes objetos matemáticos:
   a) Función parcial.
   b) Función total.
   c) Función inyectiva.
   d) Función sobreyectiva.
   e) Función biyectiva.
5. Investigue y enuncie la definición para cada uno de los siguientes objetos matemáticos:
   a) Relación binaria.
   b) Relación binaria reflexiva.
   c) Relación binaria irreflexiva.
   d) Relación binaria simétrica.
   e) Relación binaria antisimétrica.
   f) Relación binaria asimétrica.

g) Relación binaria transitiva.

6. Investigue y enuncie la definición para cada uno de los siguientes objetos matemáticos:
   a) Monoide algebraico.
   b) Grupo algebraico.
   c) Retículo algebraico.
   d) Espacio métrico.
   e) Espacio topológico.

7. Investigue y enuncie los siguientes teoremas (sin demostración):
   a) Teorema fundamental de la aritmética.
   b) Teorema fundamental del álgebra.
   c) Teorema fundamental del cálculo.

8. Para cada una de las siguientes abreviaciones, investigue su origen en latín, su significado en castellano y formule un ejemplo de su uso:
   a) e.g.
   b) et al.
   c) etc.
   d) i.e.
   e) ibid.
   f) Ph.D.
   g) Q.E.D.
   h) v.gr.
   i) viz.
   j) vs.

9. Para cada una de las siguientes expresiones en latín, investigue su siginificado en castellano y explique cómo puede ser usada en matemáticas:
   a) *a fortiori*.
   b) *a priori*.
   c) *ad absurdum*.
   d) *ad infinitum*.

10. Investigue acerca del orígen en griego antíguo de la palabra "lema" y su significado en castellano. Además, enuncie tres lemas importantes de las matemáticas.

11. Investigue acerca del orígen en griego antíguo de la palabra "corolario" y su significado en castellano. Además, enuncie un corolario importante de las matemáticas.

0.2. Inducción

Hay una anécdota acerca del matemático alemán Carl Friedrich Gauss quien, a los 8 años de edad, no prestaba mucha atención en clase. A modo de castigo, una vez su profesor de matemáticas le pidió sumar los números naturales de 0 a 100. La leyenda dice que Gauss respondió correctamente 5050 después de un par de segundos de formulada la pregunta; esto enfureció al profesor. ¿Cómo pudo el "pequeño" Gauss hacer el cálculo con tanta rapidez? Posiblemente Gauss sabía que la *ecuación de los números triangulares*

$$0 + 1 + 2 + 3 + 4 + \cdots + n = \frac{n \cdot (n+1)}{2}$$

es cierta para todo número natural n. Entonces, al tomar $n = 100$, Gauss pudo calcular fácilmente la cantidad consultada por el profesor de la siguiente manera:

$$0 + 1 + 2 + 3 + 4 + \cdots + 100 = \frac{100 \cdot 101}{2} = 5050.$$

Sin embargo, para que un cálculo como el anterior sea siempre correcto es necesario demostrar que la ecuación es cierta para todos los valores de n (algo que posiblemente Gauss ya había hecho). Tenga presente, en general, que si una ecuación es cieta para algunos valores de n, esto no la hace correcta para todos los valores de n: en matemáticas no es posible demostrar una afirmación con base en ejemplos. Aún más, en el caso de los números naturales, es imposible pretender demostrar uno a uno, para cada número, una propiedad porque hay una cantidad infinita de ellos.

El principio de inducción matemática (finita) es una herramienta fundamental para obtener resultados en matemáticas, incluyendo la lógica. Este principio permite, por ejemplo, establecer la veracidad de la ecuación de los números triangulares para cualquier valor de $n \in \mathbb{N}$, a pesar de que el conjunto de los números naturales es enorme. De manera general, el principio de inducción matemática permite demostrar que *cualquier* número natural (o una gran cantidad de ellos) tiene una propiedad dada.

Nota 0.4

Se escribe $M(k)$ para indicar que $k \in \mathbb{N}$ tiene la propiedad M. Por ejemplo, $M(5)$ indica que 5 tiene la propiedad M.

Metateorema 0.5

Considere una propiedad M sobre los números naturales y los siguientes dos casos:

1. **Caso base:** el número natural 0 satisface la propiedad M, (i.e., hay una demostración de $M(0)$).
2. **Caso inductivo:** si n es un número natural para el cual *se supone* M, entonces *se puede demostrar* que $n+1$ tiene la propiedad M (i.e., se tiene una demostración de que $M(n)$ implica $M(n+1)$).

El *principio de inducción matemática* indica que si las condiciones (1) y (2) son ciertas, entonces todo $n \in \mathbb{N}$ tiene la propiedad M.

El principio de inducción es una propiedad inherente de los números naturales. Ahora bien, ¿por qué tiene sentido el principio de inducción? Antes de presentar una demostracíon, considere la siguiente reflexión. Suponga que las condiciones (1) y (2) son ciertas para una propiedad M sobre los números naturales, y sea k un número natural (i.e., $k \in \mathbb{N}$). Si $k = 0$, entonces k tiene la propiedad M por el caso base (i.e., $M(0)$ es cierto). De lo contrario, se puede usar el caso inductivo aplicado a $n = 0$, para concluir que $1 = 1 + 0$ tiene la propiedad M (i.e., $M(1)$ es cierto). El caso inductivo se puede aplicar de la misma forma con $n = 1$ suponiendo $M(1)$ y entonces concluyendo $M(2)$. Este proceso se puede repetir hasta concluir $M(k)$: es como un "efecto dominó" en donde k es "la primera ficha en caer" o "alguna otra ficha que cae" porque está conectada causalmente con la primera ficha en caer.

A contiuación se presenta una demostración del principio de inducción matemática.

Demostración. Suponga que las condiciones (1) y (2) son ciertas para M. Sin embargo, para llegar a una contradicción, suponga que M no es cierta para todos los $n \in \mathbb{N}$. Es decir, hay un $k \in \mathbb{N}$ que no tiene la propiedad M. Sin pérdida de generalidad suponga que dicho k es el más pequeño (i.e., el mínimo) entre todos los números en $\mathbb{N}$ que no tienen la propiedad M. Hay dos casos: $k = 0$ o $k > 0$. Pero k no puede ser 0 porque, por la Condición (1), 0 cumple la propiedad M. Luego, debe ser $k > 0$. Con $k > 0$, se tiene que $k - 1 \in \mathbb{N}$. Además, como k es el número natural más pequeño que no tiene la propiedad M, se sabe que $k - 1$ tiene la propiedad M. Dado que $k - 1$ tiene la propiedad M se concluye, por la Condición (2), que k tiene la propiedad M. Sin embargo, esto contradice el hecho de que k no tiene la propiedad M. En conclusión, si las condiciones (1) y (2) son ciertas para M, necesariamente todos los números naturales tienen la propiedad M. □

Nota 0.6

La suposición $M(n)$ que se hace en el caso inductivo del Metateorema 0.5 es denominada *hipótesis inductiva*.

Ejemplo 0.1

La ecuación de los números triangulares es cierta para todo número natural n. Es decir,

$$0 + 1 + 2 + 3 + 4 + \cdots + n = \frac{n \cdot (n+1)}{2}$$

para $n \in \mathbb{N}$.

Demostración. Primero que todo, se identifica claramente la propiedad que se quiere demostrar. Sea $G(n)$ una abreviación para la ecuación $0 + 1 + 2 + 3 + 4 + \cdots + n = \frac{n\cdot(n+1)}{2}$, lo cual se escribe así:

$$G(n) : 0 + 1 + 2 + 3 + 4 + \cdots + n = \frac{n \cdot (n+1)}{2}.$$

Note que n es el parámetro de G así como n es el parámetro en la ecuación de los números triangulares. El objetivo es demostrar $G(n)$ para $n \in \mathbb{N}$. Por el principio de inducción matemática (Metateorema 0.5) basta con demostrar las condiciones (1) y (2) para G.

Caso base: se quiere demostrar que 0 tiene la propiedad G. Note que $G(0)$ abrevia la ecuación $0 = \frac{0\cdot(0+1)}{2}$, que es directamente una identidad. Entonces 0 tiene la propiedad G, i.e., $G(0)$.

Caso inductivo: se quiere demostrar que si n tiene la propiedad G, entonces $(n+1)$ tiene la propiedad $G(n+1)$ (i.e., $G(n)$ implica $G(n+1)$). Se supone como hipótesis inductiva $G(n)$ (i.e., que $n \geq 1$ tiene la propiedad G) y con esta información se busca una demostración de $G(n+1)$. La expresión $G(n+1)$ abrevia la ecuación

$$0 + \cdots + n + (n+1) = \frac{(n+1) \cdot ((n+1)+1)}{2}.$$

Por la hipótesis inductiva (i.e., suponiendo $G(n)$), la parte izquierda de la igualdad se puede escribir como $\frac{n\cdot(n+1)}{2} + (n+1)$. Factorizando $(n+1)$, esta expresión es igual a $\frac{(n+1)\cdot(n+2)}{2}$, que a su vez es igual a $\frac{(n+1)\cdot((n+1)+1)}{2}$. Entonces, si n tiene la propiedad G, necesariamente $(n+1)$ tiene la propiedad G. Es decir, $G(n)$ implica $G(n+1)$.

Dado que los casos (1) y (2) son ciertos para G, por el principio de inducción matemática (Metateorema 0.5), se sigue que todo $n \in \mathbb{N}$ tiene la propiedad G. $\square$

El principio de inducción matemática tiene muchas variantes. Por ejemplo, es posible contar con una versión en la cual el caso base no sea necesariamente $n = 0$ (Ejercicio 0.2.3). A continuación se presenta un ejemplo del uso de la inducción matemática para demostrar una desigualdad cuyo caso base es $n = 1$.

Ejemplo 0.2

La siguiente desigualdad es cierta para todos los números naturales $n \geq 1$:

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^n} < 1.$$

Demostración. Inicialmente se identifica la propiedad que se desea demostrar. Sea $H(n)$ una abreviación para la desigualdad $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^n} < 1$, es decir,

$$H(n) : \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^n} < 1.$$

Por el principio de inducción matemática (Metateorema 0.5) basta con demostrar las condiciones (1) y (2) para H.

Caso base: se quiere demostrar que 1 tiene la propiedad H, es decir, $H(1)$. En este caso, $H(1)$ abrevia la propiedad $\frac{1}{2} < 1$, la cual es verdadera. Luego, 1 tiene la propiedad H.

Caso inductivo: se quiere demostrar que $n + 1$ tiene la propiedad H si n tiene la propiedad H (i.e., $H(n + 1)$ es consecuencia de $H(n)$). Entonces se supone $H(n)$ como hipótesis inductiva y se procede a buscar una demostración para $H(n + 1)$. La expresión $H(n + 1)$ abrevia la desigualdad

$$H(n+1) : \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^n} + \frac{1}{2^{n+1}} < 1.$$

Al suponer la hipótesis inductiva $H(n)$ se tiene que la desigualdad

$$\frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots + \frac{1}{2^{n+1}} < \frac{1}{2}$$

es cierta porque resulta de multiplicar por $\frac{1}{2}$ cada uno de los dos lados de la desigualdad en $H(n)$. Ahora, sumando $\frac{1}{2}$ a cada uno de los dos lados de esta

última desigualdad, se obtiene

$$\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\cdots+\frac{1}{2^{n+1}}<\frac{1}{2}+\frac{1}{2}=1.$$

Es decir, $H(n+1)$. En conclusión, si n tiene la propiedad H, entonces $n+1$ también tiene la propiedad H.

Dado que las condiciones (1) y (2) son ciertas para H, por el principio de inducción matemática (Metateorema 0.5), se sigue que todo $n \in \mathbb{N}$ con $n \geq 1$ tiene la propiedad H. □

Como se verá en algunas secciones de este texto, el principio de inducción matemática se puede adaptar para hacer demostraciones sobre estructuras diferentes a los números naturales como, por ejemplo, fórmulas y demostraciones en un sistema formal. En cada variante del principio de inducción matemática que se use, lo más importante es identificar las condiciones base (pueden ser más de uno) y los casos inductivos (los cuales también pueden ser más de uno). En la práctica, en muchas ocasiones, es posible y natural reducir un principio de inducción matemática al principio de inducción matemática para números naturales estudiado en esta sección. Esto hace del principio de inducción matemática para números naturales una herramienta efectiva en las matemáticas.

En algunas ocasiones hay más de una posibilidad en el enunciado de un problema para decidir cómo hacer inducción. Se presenta a continuación un ejemplo que ilustra cómo escribir claramente la propiedad que se desea demostrar facilita el uso del principio de inducción matemática.

Ejemplo 0.3

La siguiente ecuación es cierta para todos los números naturales n y r, con $r \neq 1$:

$$r^0+r^1+\cdots+r^n=\frac{1-r^{n+1}}{1-r}.$$

Demostración. Considere la propiedad I sobre los números naturales definida por

$$I(n): r^0+r^1+\cdots+r^n=\frac{1-r^{n+1}}{1-r}.$$

El objetivo es demostrar $I(n)$ para $n \in \mathbb{N}$.

Caso base: Se quiere demostrar $I(0)$, que abrevia $r^0 = \frac{1-r^{0+1}}{1-r}$. Por un lado, $r^0 = 1$. Por el otro y como $r \neq 0$, se tiene que $\frac{1-r^{0+1}}{1-r} = 1$. Es decir, 0 tiene la propiedad I.

Caso inductivo: Se usa como hipótesis inductiva $I(n)$ y se demuestra $I(n+1)$. Es decir, el objetivo es demostrar

$$r^0 + r^1 + \cdots + r^n + r^{n+1} = \frac{1 - r^{n+2}}{1 - r}.$$

Considere el siguiente cálculo, en el cual se supone $r \neq 1$:

$$\begin{aligned}
r^0 + r^1 + \cdots + r^n + r^{n+1} &= \frac{1 - r^{n+1}}{1 - r} + r^{n+1} && \text{(por la hipótesis inductiva)} \\
&= \frac{1 - r^{n+1}}{1 - r} + \frac{(1 - r) \cdot r^{n+1}}{1 - r} && \text{(aritmética)} \\
&= \frac{1 - r^{n+1} + r^{n+1} - r \cdot r^{n+1}}{1 - r} && \text{(aritmética)} \\
&= \frac{1 - r^{n+2}}{1 - r} && \text{(aritmética).}
\end{aligned}$$

Es decir, al suponer $I(n)$ se puede demostrar $I(n+1)$.

Por el principio de inducción matemática (Metateorema 0.5) se obtiene que todo $n \in \mathbb{N}$ tiene la propiedad I. □

El Ejemplo 0.3 presenta un caso particular de una propiedad de la *serie geométrica* (ver Ejercicio 0.2.13). Este ejemplo, además de ilustrar el uso del principio de inducción matemática, es interesante porque hay dos variables involucradas en el enunciado del problema, es decir, n y r. En la demostración incluida como parte de este ejemplo se optó por hacer inducción sobre n y no sobre r (note que la propiedad I es definida con parámetro n). Como la única suposición que se hace sobre r es que no puede ser 1, entonces la demostración en el Ejemplo 0.3 justifica que la propiedad dada es cierta para cualesquiera números naturales n y r siempre y cuando $r \neq 1$. Es más, como la única condición sobre r es esa, esta propiedad es cierta para r siendo, por ejemplo, un número real.

Antes de finalizar la sección se presenta un ejemplo clásico sobre el uso del principio de inducción matemática para establecer propiedades de los números de Fibonacci.

Ejemplo 0.4

Los *números de Fibonacci* se definen para todo $n \in \mathbb{N}$ por:

$$\begin{aligned} F(0) &= 0 \\ F(1) &= 1 \\ F(n+2) &= F(n) + F(n+1) \quad \text{, para } n \geq 0. \end{aligned}$$

Demuestre que la siguiente ecuación es cierta para todo $n \in \mathbb{N}$:

$$F(0) + F(1) + \cdots + F(n) = F(n+2) - 1.$$

Demostración. El objetivo es demostar, para cualquier $n \in \mathbb{N}$, la propiedad $J(n)$:

$$J(n) : F(0) + F(1) + \cdots + F(n) = F(n+2) - 1.$$

Casos base: para demostrar $J(0)$ (i.e., $F(0) = F(2) - 1$), considere el siguiente cálculo:

$$\begin{aligned} F(0) &= 0 && \text{(definición de } F \text{ par } n = 0) \\ &= 1 - 1 && \text{(aritmética)} \\ &= F(0) + F(1) - 1 && \text{(definición de } F \text{ para } n = 0, 1) \\ &= F(2) - 1 && \text{(definición de } F \text{ para } n = 2). \end{aligned}$$

La demostración de $J(1)$ se propone como ejercicio para el lector.

Caso inductivo: se supone $J(n)$ y se demuestra $J(n+1)$:

$$\begin{aligned} F(0) + F(1) + \cdots + F(n) + F(n+1) &= (F(n+2) - 1) + F(n+1) && \text{(por la hipótesis inductiva)} \\ &= (F(n+1) + F(n+2)) - 1 && \text{(aritmética)} \\ &= F(n+3) - 1 && \text{(definición de } F \text{ para } n \geq 0). \end{aligned}$$

Por el principio de inducción matemática (Metateorema 0.5) se obtiene que todo $n \in \mathbb{N}$ tiene la propiedad J. □

Note que la demostración en el Ejemplo 0.4 consta de dos casos base. Esto se debe a que la definición de la función F indica, en el caso general, que el valor $F(n)$ corresponde a la suma de los *dos* valores inmediatamente anteriores. Por esa misma razón, se plantean dos casos base para la definición de F, más precisamente, $F(0)$ y $F(1)$.

Ejercicios

1. Investigue sobre Carl Friedrich Gauss y describa sus principales aportes a las matemáticas.
2. Investigue y explique por medio de dibujos por qué la ecuación usada por el "pequeño" Gauss, enunciada al inicio de esta sección, recibe el nombre de ecuación de los números triangulares.
3. Considere una propiedad M sobre los números naturales y un número $m \in \mathbb{N}$. Además, considere las siguientes dos condiciones:
   (1) El número m tiene la propiedad M (i.e., $M(m)$).
   (2) Si el número natural $n \geq m$ tiene la propiedad M, entonces $m+1$ tiene la propiedad M (i.e., $M(n)$ implica $M(n+1)$).
   Demuestre que si las condiciones (1) y (2) son ciertas, entonces todo número natural $n \geq m$ tiene la propiedad M.
4. Use el principio de inducción matemática, indicando cada uno de los casos y el uso de la hipótesis inductiva, para demostrar
   $$A(n) : (2 \cdot 1 - 1) + (2 \cdot 2 - 1) + (2 \cdot 3 - 1) + \cdots + (2 \cdot n - 1) = n^2$$
   para todo $n \in \mathbb{N}$ con $n \geq 1$.
5. Explique por medio de dibujos por qué la ecuación del Ejercicio 4 es conocida como la ecuación de los números cuadrados.
6. Use el principio de inducción matemática, indicando cada uno de los casos y el uso de la hipótesis inductiva, para demostrar
   $$B(n) : 0^2 + 1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n \cdot (n+1) \cdot (2 \cdot n + 1)}{6}$$
   para todo $n \in \mathbb{N}$.
7. Use el principio de inducción matemática para demostrar
   $$C(n) : 8 + 13 + 18 + 23 + \cdots + (3 + 5n) = \frac{5n^2 + 11n}{2}$$
   para todo $n \in \mathbb{N}$ con $n \geq 1$. ¿Qué sucede cuando $n = 0$?
8. Demuestre que $2^n \geq n + 12$ para todo número natural $n \geq 4$. En este problema el caso base es $n = 4$. ¿Es cierta esta propiedad para $n < 4$? Justifique su respuesta.
9. Use el principio de inducción matemática para demostrar
   $$D(n) : (1 + 2 + \cdots + n)^2 = 1^3 + 2^3 + \cdots + n^3.$$
10. Use el principio de inducción matemática para demostrar
    $$(-1)^1 1^2 + (-1)^2 2^2 + \cdots + (-1)^n n^2 = \frac{(-1)^n n(n+1)}{2}.$$

11. Use el principio de inducción matemática para demostrar
$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{n\cdot(n+1)}=\frac{n}{n+1}.$$
12. Use el principio de inducción matemática para demostrar
$$\left(1-\frac{1}{2^2}\right)\left(1-\frac{1}{3^2}\right)\cdots\left(1-\frac{1}{n^2}\right)=\frac{n+1}{2n}.$$
13. Use el principio de inducción matemática para demostrar la siguiente propiedad de la *serie geométrica*:
$$ar^0+ar^1+\cdots+ar^n=\frac{a\left(1-r^{n+1}\right)}{1-r}$$
para todo $a,n,r\in\mathbb{N}$ con $r\neq 1$.
14. Complete el Ejemplo 0.4 con la demostración del segundo caso base.
15. Investigue y explique el significado asociado a la secuencia $F(0),F(1),\ldots$ por Fibonacci en el libro Liber Abaci (1202).
16. Demuestre que la función F de Fibonacci satisface las siguientes igualdades:
    a) $F(1)+F(3)+\cdots+F(2(n-1)+1)=F(2n)$, para $n\geq 1$.
    b) $F(0)+F(2)+\cdots+F(2n)=F(2n+1)-1$, para $n\geq 0$.
17. Demuestre que la función F de Fibonacci satisface, para $n\in\mathbb{N}$, la siguiente igualdad:
$$F(0)^2+F(1)^2+\cdots+F(n)^2=F(n)F(n+1).$$
18. Demuestre que la función F de Fibonacci satisface, para $n\in\mathbb{N}$, la siguiente igualdad:
$$F(n)^2-F(n+1)F(n-1)=(-1)^{n+1}.$$
19. El producto entre dos matrices de dimensión 2×2 se define como:
$$\begin{pmatrix}a_{00}&a_{01}\\a_{10}&a_{11}\end{pmatrix}\times\begin{pmatrix}b_{00}&b_{01}\\b_{10}&b_{11}\end{pmatrix}=\begin{pmatrix}a_{00}b_{00}+a_{01}b_{10}&a_{00}b_{01}+a_{01}b_{11}\\a_{10}b_{00}+a_{11}b_{10}&a_{10}b_{01}+a_{11}b_{11}\end{pmatrix}.$$
Dada una matriz A de 2×2, se define A^n para $n\geq 1$ de la siguiente manera:
$$A^1=A$$
$$A^{n+1}=A\times A^{n-1}\quad,\ n\geq 2.$$
Usando el principio de inducción matemática demuestre, para $n\geq 1$, la siguiente igualdad relacionada con la función F de Fibonacci:
$$\begin{pmatrix}1&1\\1&0\end{pmatrix}^n=\begin{pmatrix}F(n+1)&F(n)\\F(n)&F(n-1)\end{pmatrix}.$$
20. Demuestre que la función F de Fibonacci, para $n\geq 1$, satisface
$$F(n)\geq\left(\frac{3}{2}\right)^{n-2}.$$

21. Considere la función *factorial* definida de la siguiente manera para cualquier $n \in \mathbb{N}$:

$$0! = 1,$$
$$n! = n \cdot (n-1)! \quad \text{, si } n > 0.$$

Proponga un caso base $m \in \mathbb{N}$ y demuestre por el principio de inducción matemática la desigualdad

$$2^n < n! \quad \text{, para } n \geq m.$$

¿Cuál es el mínimo m que satisface esta desigualdad?

22. El *coeficiente binomial* $\binom{n}{k}$ se define para $0 \leq k \leq n$ de la siguiente manera:

$$\binom{n}{0} = 1$$
$$\binom{n}{n} = 1$$
$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad \text{, para } 1 \leq k \leq n.$$

Demuestre para n y k números naturales:
a) $\binom{n}{k} = \binom{n}{n-k}$, con $0 \leq k \leq n$.
b) $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$, con $1 \leq k \leq n$.

23. Demuestre para $n, k \in \mathbb{N}$ con $0 \leq k \leq n$:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

24. Demuestre para $n \in \mathbb{N}$:

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n.$$

25. Demuestre para $n \in \mathbb{N}$:

$$0\binom{n}{0} + 1\binom{n}{1} + \cdots + n\binom{n}{n} = n2^{n-1}.$$

26. Demuestre para $n \in \mathbb{N}$:

$$\binom{n}{0}^2 + \binom{n}{1}^2 + \cdots + \binom{n}{n}^2 = \binom{2n}{n}.$$

27. Sean $n, x \in \mathbb{N}$. Demuestre que si $1 + x > 0$ y $n \geq 1$, entonces $(1+x)^n \geq 1 + nx$.

28. Sea $h(n) = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$. Demuestre, para todo $n \geq 1$ que

$$h(1) + h(2) + h(3) + \cdots + h(n) = (n+1) \cdot h(n) - n.$$

29. Sea f una función definida para todo $n \in \mathbb{N}$ por:

$$f(0) = 0,$$
$$f(1) = 1,$$
$$f(n) = f(f(n-1)) + f(n - f(n-1)) \quad \text{, si } n > 1.$$

Calcule $f(10000)$.

30. Proponga una fórmula para calcular la siguiente suma y demuestre que dicha fórmula es correcta:

$$0 \cdot 1 + 1 \cdot 2 + 2 \cdot 3 + \cdots + n(n+1).$$

31. Sean k y l números naturales. Se dice que k es *divisible* por l si y solo si existe un $p \in \mathbb{N}$ tal que $k = p \cdot l$ (en este caso, p se llama el *testigo* de la divisibilidad de k entre l). Por ejemplo, 15 es divisible por 3 con testigo 5 porque $15 = 5 \cdot 3$. Use el principio de inducción matemática para demostrar que $11^n - 4^n$ es divisible por 7 para todo $n \in \mathbb{N}$.

32. Demuestre que $x^n - 1$ es divisible por $x - 1$ para todos $n, x \in \mathbb{N}$.

33. Demuestre que las siguientes afirmaciones son ciertas para cualquier $n \in \mathbb{N}$:
    a) $n^3 - n$ es divisible por 3.
    b) $n^5 - n$ es divisible por 5.

34. Demuestre o refute: $n^4 - n$ es divisible por 4, para todo $n \in \mathbb{N}$.

35. Demuestre usando el principio de inducción matemática que un número natural en representación decimal es divisible por 3 si y solo si la suma de sus dígitos es divisible por 3.

36. Demuestre usando el principio de inducción matemática que:
    a) El cuadrado de un número natural impar es de la forma $8m+1$, para algún $m \in \mathbb{N}$.
    b) La cuarta potencia de un número natural impar es de la forma $16m + 1$, para algún $m \in \mathbb{N}$.

37. Suponga que una oficina postal vende estampillas de \$2 y \$3. Demuestre que cualquier cantidad de dinero $n \geq$ \$2 puede ser pagada con estampillas de estas denominaciones. Ayuda: use inducción matemática sobre n. En el caso inductivo considere dos posibilidades: primero, que n puede pagarse usando únicamente estampillas de \$2; segundo, que n puede pagarse usando al menos una estampilla de \$3.

38. Demuestre que dada una cantidad ilimitada de monedas de \$6, \$10 y \$15, se puede obtener cualquier suma de dinero mayor que \$29.

39. Este ejercicio ilustra por qué el caso base es importante en el principio de inducción matemática. Considere la siguiente afirmación:

    'El número $n^2 + 5n + 1$ es par para todo $n \in \mathbb{N}$.'

a) Demuestre el caso inductivo para esta afirmación.
b) Demuestre que el caso base falla para esta afirmación.
c) Concluya que la afirmación es falsa.
d) Use inducción para demostrar que $n^2 + 5 \cdot n + 1$ es impar para todo $n \in \mathbb{N}$.

40. Considere un tablero de ajedrez de $2^n \times 2^n$ en el cual ha sido eliminada arbitrariamente una de sus celdas ($n \geq 1$). Demuestre que dicho tablero puede ser cubierto, sin omitir celdas ni con sobrelapamientos, con fichas en forma de L que cubren exactamente 3 celdas.

41. El *principio del palomar*, en su versión más sencilla, corresponde a la siguiente afirmación para $n \geq 1$: si $n + 1$ canicas se distribuyen en n cajas, entonces al menos una caja contiene más de una canica. Demuestre por el principio de inducción matemática esta formulación del principio del palomar.

42. Considere un grupo de $n \in \mathbb{N}$ parejas. Usando el principio del palomar (ver Ejercicio 41) determine cuántas de las $2n$ personas del grupo deben ser seleccionadas para garantizar que se ha seleccionado al menos una pareja?

43. De la colección $1, 2, \ldots, 200$ son seleccionados 101 números. Demuestre que entre los números seleccionados hay dos tales que uno divide al otro.

44. Considere un conjunto C de $n+1$ números tomados del conjunto $\{1, 2, \ldots, 2n\}$, con $n \geq 1$. Demuestre que en C hay dos números tales que uno divide al otro.

45. Un maestro de ajedrez tiene 11 semanas de preparación para un torneo. Para ello decide jugar al menos una partida todos los días. Sin embargo, para no cansarse más de la cuenta decide no jugar más de 12 partidas en una semana calendario. Demuestre que hay una sucesión (consecutiva) de días en los cuales el maestro de ajedrez juega *exactamente* 21 partidas.

0.3. Sistemas formales

Un sistema formal es el marco matemático apropiado para definir y aclarar conceptos fundamentales como *proposición matemática*, *axioma*, *demostración* y *teorema*. Antes que todo, se introduce la terminología comúnmente usada en el estudio de sistemas formales.

Definición 0.7

Sea S un conjunto de símbolos. Una *expresión en* S (o *palabra en* S) es una secuencia finita (posiblemente vacía) de símbolos de S.

Por ejemplo, si $S = \{a, b, c\}$, entonces *baca* y *cba* son expresiones en S.

En la descripción de un lenguaje formal primero se especifica el conjunto de símbolos S y luego se seleccionan ciertas expresiones construídas con base en los símbolos que se designan como *fórmulas*. No necesariamente toda expresión es una fórmula del sistema formal. Por ejemplo, si S es el conjunto de letras del alfabeto castellano con sus acentos (i.e., $S = \{a, b, c, \ldots, \tilde{n}, \ldots, z\}$) entonces *niño* y *qwerty* son expresiones en S. Si se declaran como fórmulas aquellas expresiones que aparecen en el *diccionario de la lengua española* (http://rae.es), entonces *niño* es una fórmula pero *qwerty* no.

Definición 0.8

Un *sistema formal* tiene tres componentes:

- un *lenguaje formal* que consiste en un conjunto de símbolos y fórmulas construídas a partir de los símbolos,
- un conjunto de *axiomas* que corresponde a algunas fórmulas,
- unas *reglas de inferencia* que permiten obtener una fórmula a partir de un conjunto de fórmulas.

Para el lenguaje del sistema formal debe siempre existir un algoritmo para decidir (i.e., que responda *si* o *no* en un número finito de pasos) si una expresión arbitraria es o no una de sus fórmulas. Es importante recalcar que todos los lenguajes estudiados en este texto tienen esta propiedad.

Nota 0.9

A lo largo de este texto las letras griegas minúsculas $\phi, \psi, \alpha, \beta, \gamma, \delta, \ldots$ (incluyendo sus versiones primadas o con subíndices) se usan para denotar fórmulas de un sistema formal, mientras que las letras griegas mayúsculas $\Phi, \Psi, \Gamma, \Delta, \ldots$ se usan para denotar conjuntos de fórmulas.

Los *axiomas* de un sistema formal son algunas fórmulas especialmente seleccionadas. En muchos casos el lenguaje formal se escoge con cierta interpretación inicial en mente y los axiomas entonces son algunas fórmulas "obviamente ciertas". Un sistema formal debe contar con un algoritmo que decida si una de sus fórmulas es o no un axioma.

Una *regla de inferencia* es un mecanismo que permite obtener una fórmula (llamada *conclusión*) a partir de un conjunto finito de fórmulas (llamadas *premisas*). Suponga que $\phi_0, \phi_1, \ldots, \phi_k, \phi$ son fórmulas. Una regla de inferencia con premisas $\phi_0, \phi_1, \ldots, \phi_k$ (es común que $k = 0, 1, 2$) y conclusión ϕ se escribe esquemáticamente

como

$$\frac{\phi_0 \qquad \phi_1 \qquad \cdots \qquad \phi_k}{\phi}.$$

En cualquier sistema formal hay una cantidad finita de reglas de inferencia.

Ejemplo 0.5

A continuación se define el sistema formal ADD:

Lenguaje: los símbolos son '+', '=' y '|'. Una fórmula es una expresión de la forma $x + y = z$, en donde x, y y z son secuencias no vacías en las cuales únicamente aparece el símbolo '|'.

Axiomas: el único axioma es la fórmula $| + | = ||$.

Reglas de inferencia: hay dos reglas de inferencia

$$\frac{x + y = z}{x| + y = z|}\ \text{R1} \qquad \frac{x + y = z}{y + x = z}\ \text{R2}.$$

Un sistema formal usualmente se propone con una *interpretación* o intuición en mente. Es decir, el lenguaje y los axiomas son seleccionados para estudiar un tema específico como la lógica proposicional, la aritmética, la teoría de conjuntos, etc. Sin embargo, es importante aclarar que una interpretación no hace parte de un sistema formal. Aquellos conceptos de un sistema formal que se definen sin hacer referencia a una interpretación son llamados elementos *sintácticos*. A su vez, conceptos que dependen del significado o interpretación de los símbolos son llamados elementos *semánticos*. La interacción entre los conceptos sintácticos y semánticos es fundamental en el estudio de la lógica matemática vista como un sistema formal. Por ejemplo, en la lógica proposicional existe el concepto sintáctico de *teorema* y el semántico de *tautología*. Un objetivo frecuente en el estudio de la lógica proposicional es demostrar que una fórmula es un teorema si y solo si es una tautología; en este texto, este tipo de propiedades se demuetra, cuando llegue el momento, tanto para la lógica proposicional como para la lógica de predicados.

Al contar con el concepto de sistema formal, ahora es posible dar definiciones precisas de lo que son una *demostración* y un *teorema*.

Definición 0.10

Sea F un sistema formal. Una *demostración* en F es una secuencia finita de fórmulas $\phi_0, \phi_1, \ldots, \phi_n$ de F tal que, para cada $0 \leq k \leq n$, una de las siguientes condiciones es cierta:

1. ϕ_k es un axioma de F, o
2. $k > 0$ y ϕ_k es la conclusión de una regla de inferencia de F cuyas premisas aparecen en la secuencia $\phi_0, \ldots \phi_{k-1}$.

Si $\phi_0, \phi_1, \ldots, \phi_n$ es una demostración en F, entonces se dice que ϕ_n es un *teorema* de F, lo cual se escribe como

$$\vdash_{\mathsf{F}} \phi_n.$$

Las definiciones de demostración y teorema carecen de semántica alguna y es así como estos dos conceptos son netamente elementos sintácticos de cualquier sistema formal.

Nota 0.11

Una demostración $\phi_0, \phi_1, \ldots, \phi_n$ en un sistema formal F se puede diagramar esquemáticamente de la siguiente forma

$$\begin{array}{lll} 0. & \phi_0 & (\text{explicación}_0) \\ 1. & \phi_1 & (\text{explicación}_1) \\ \ldots & & \\ n. & \phi_n & (\text{explicación}_n). \end{array}$$

en la cual *explicación*$_i$ es un texto explicando cómo se obtiene la fórmula ϕ_i a partir de $\phi_0, \ldots, \phi_{i-1}$. Adicionalmente, cuando el sistema formal bajo discusión esté claramente identificado, se puede obviar el subíndice y escribir $\vdash \phi$ en lugar de $\vdash_{\mathsf{F}} \phi$ para cualquier teorema ϕ de F.

A continuación, en el Ejemplo 0.6, se muestra cómo el formato de demostración puede ser usado en el sistema ADD. En particular, es importante notar el papel importante que juegan las explicaciones en cada uno de los renglones de una demostración para que un lector pueda seguirla y entender por qué es correcta.

Ejemplo 0.6

La fórmula $||+|||=|||||$ es un teorema del sistema formal ADD:

0. $|+|=||$ (axioma de ADD)
1. $||+|=|||$ ($R1$ con premisa 0)
2. $|||+|=||||$ ($R1$ con premisa 1)
3. $|+|||=||||$ ($R2$ con premisa 2)
4. $||+|||=|||||$ ($R1$ con premisa 3).

En conclusión $\vdash_{\mathsf{ADD}} ||+|||=|||||$.

La noción de demostración formal es clave en el desarrollo de las matemáticas. De hecho, los sistemas formales fueron inicialmente planteados por Hilbert con el objetivo de estudiar y analizar las demostraciones matemáticas. Así como la geometría estudia objetos en un espacio geométrico y la aritmética estudia los números, *la teoría de demostraciones* estudia las demostraciones. Los sistemas formales son el marco dentro del cual se lleva a cabo la teoría de demostraciones.

Finalmente, es conveniente incluir notación para referirse a aquellas fórmulas de un sistema formal que no hacen parte de su conjunto de teoremas.

Nota 0.12

Para F un sistema formal y ϕ una fórmula de F, la expresión

$$\nvdash_{\mathsf{F}} \phi$$

se usa para denotar que ϕ *no* es teorema de F.

Unas palabras de precaución sobre el significado asociado a la relación $\nvdash_{\mathsf{F}}$ para cualquier sistema formal F. Para poder afirmar que una fórmula ϕ no es teorema de F (i.e., para afirmar $\nvdash_{\mathsf{F}} \phi$) hay que justificar que *no* existe demostración alguna de ϕ en F. En particular, no es suficiente con desfallecer en el intento de encontrar una demostración para ϕ, sino que es necesario "demostrar" que no existe tal demostración. Para este propósito se requieren, generalmente, técnicas de razonamiento externas al sistema formal, es decir, técnicas para razonar acerca del sistema formal como objeto de estudio (e.g., metateoremas del sistema formal).

Por ejemplo y de acuerdo con la interpretación intuitiva de lo que representa el sistema formal ADD, sería deseable que la fórmula $||+||=|||$ no sea uno de sus teoremas. En el Ejemplo 0.7 de la siguiente sección se establece un metateorema garantizando que la cantidad de palotes a la izquierda y a la derecha de '=' coincide

en cualquier teorema de ADD. Gracias a dicho metateorema, se puede concluir que $\nvdash_{\mathsf{ADD}} || + || = |||$ porque la cantidad de palotes a la izquierda y a la derecha de '=' no coincide: son 4 y 3, respectivamente. La técnica utilizada en el Ejemplo 0.7 es la inducción matemática sobre demostraciones de un sistema formal, el cual es el propósito de la Sección 0.4.

Ejercicios

1. Liste todos los teoremas del siguiente sistema formal:
   Lenguaje: hay tres símbolos 'a', 'b' y 'c'; toda expresión es una fórmula.
   Axiomas: *cabcba*.
   Reglas de inferencia: hay tres reglas de inferencia:
   a) si la fórmula comienza con a, agregue *cac* a la derecha y luego elimine los primeros tres símbolos de la expresión resultante;
   b) si la fórmula comienza con b, agregue *bab* a la derecha y luego elimine los primeros tres símbolos de la expresión resultante;
   c) si la fórmula comienza con c, agregue *ca* a la derecha y luego elimine los primeros tres símbolos de la expresión resultante.

2. Liste todos los teoremas del siguiente sistema formal:
   Lenguaje: hay tres símbolos 'a', 'b' y 'c'; toda expresión es una fórmula.
   Axiomas: *abccba*.
   Reglas de inferencia: hay tres reglas de inferencia:
   a) si la fórmula comienza con a, agregue *bab* a la derecha y luego elimine los primeros tres símbolos de la expresión resultante;
   b) si la fórmula comienza con b, agregue *abba* a la derecha y luego elimine los primeros tres símbolos de la expresión resultante;
   c) si la fórmula comienza con c, agregue *ca* a la derecha y luego elimine los primeros tres símbolos de la expresión resultante.

3. Justifique por qué las siguientes fórmulas no son teoremas de ADD:
   a) $| + | = |$
   b) $|| + | = ||||$
   c) $| + || = ||||$

4. Proponga un sistema formal MULT que sea similar a ADD pero cuyos teoremas sean ecuaciones ciertas de la multiplicación de los números naturales positivos. Demuestre que la fórmula $|| \times ||| = ||||||$ es un teorema de MULT.

5. Investigue acerca de las siguientes expresiones y explique brevemente su significado. Ilustre cada una de ellas con un ejemplo.
   a) Problema de decisión.
   b) Algoritmo/procedimiento de decisión.

c) Problema decidible.
d) Problema indecidible.

0.4. Inducción sobre demostraciones

En ocasiones es necesario demostrar que el conjunto de teoremas de un sistema formal F satisface una propiedad Q. Esta sección estudia la inducción sobre la longitud de una demostración, un principio de inducción derivado del *principio de inducción matemática* (Metateorema 0.5), sumamente útil para demostrar propiedades sobre teoremas de un sistema formal y extensivamente empleado a lo largo de este texto para tal fin.

Suponga que $\vdash_{\mathsf{F}} \phi$, en donde ϕ es una fórmula de F. Entonces, por la definición de teorema, necesariamente existe al menos una demostración de ϕ con $n \geq 1$ renglones (aquí, n es una variable sobre los números naturales). Dicha demostración puede verse de la siguiente forma:

$$\begin{array}{rll} 0. & \phi_0 & (\cdots) \\ 1. & \phi_1 & (\cdots) \\ \cdots & & \\ n-1. & \phi_{n-1} & (\cdots) \end{array}$$

en la cual cada expresión ϕ_i $(0 \leq i \leq n-1)$ es una fórmula de F y la última fórmula ϕ_{n-1} es en realidad ϕ (de lo contrario la secuencia de fórmulas no sería una demostración de ϕ).

Nota 0.13

Fíjese que la fórmula ϕ que se usa como objeto de la demostracíon, aparte de ser un teorema, es cualquier fórmula del sistema formal. Ninguna otra suposición se ha hecho sobre la forma o estructura interna de ϕ. Lo mismo sucede con los ϕ_i. Entonces, en caso tal de lograr el objetivo de demostrar que ϕ tiene la propiedad Q, necesariamente se logra demostrar que *cualquier* teorema de F tiene la propiedad Q. Este tipo de comportamiento se conoce comúnmente como el *principio de universalidad*.

Por un momento suponga que la propiedad Q satisface las siguientes dos condiciones:

1. cada axioma de F tiene la propiedad Q,
2. cada regla de inferencia de F es tal que si sus premisas tienen la propiedad Q, entonces necesariamente su conclusión tiene la propiedad Q.

Bajo estas dos suposiciones, el objetivo es demostrar $Q(\phi)$ (i.e., que el teorema ϕ tiene la propiedad Q). Al analizar la demostración $\phi_0, \ldots, \phi_{n-1}$ de ϕ bajo estos dos supuestos, se tiene que la fórmula ϕ_0 es necesariamente un axioma de F dado que aparece en la primera línea de la demostración. Por la suposición (1), entonces se concluye que ϕ_0 tiene la propiedad Q. Ahora considere la segunda fórmula ϕ_1. Si esta fórmula es también un axioma, entonces tiene la propiedad Q por la suposición (1). De lo contrario, ϕ_1 es la conclusión de una regla de inferencia cuya única premisa puede ser ϕ_0. Ya se conoce que la fórmula ϕ_0 tiene la propiedad Q. Entonces, por la suposición (2), se concluye que ϕ_1 también tiene la propiedad Q. Al continuar de esta forma, se llegará a la conclusión de que la fórmula ϕ_{n-1} (la misma ϕ) tiene la propiedad Q, i.e., $Q(\phi)$.

El párrafo anterior hace un recuento informal de una técnica de demostración importante llamada *inducción sobre teoremas* o *inducción sobre la longitud de una demostración*. A continuación, en el Metateorema 0.14, se presenta una demostración formal (i.e., rigurosa) de este resultado.

Metateorema 0.14

Sea F un sistema formal y Q una propiedad sobre las fórmulas de F.

Para demostrar que todo teorema de F tiene la propiedad Q, basta con:

1. demostrar que cada axioma de F tiene la propiedad Q
2. demostrar que cada regla de inferencia de F es tal que si cada una de sus premisas tiene la propiedad Q, entonces su conclusión tiene la propiedad Q.

Demostración. Para $n \in \mathbb{N}$, sea $S(n)$ la propiedad "todo teorema de F con una demostración de máximo n renglones tiene la propiedad Q". El objetivo es entonces demostrar que todo $n \geq 1$ tiene la propiedad S. De este modo, se habrá demostrado que todo teorema de F tiene la propiedad Q (¿por qué?). La demostración procede por inducción matemática sobre n, suponiendo que (1) y (2) son ciertos.

Caso base: En este caso $n = 1$ y el objetivo es demostrar que todo teorema de F con una demostración de máximo un renglón tiene la propiedad Q. Dado que hay un solo renglón en la demostración, necesariamente dicho teorema es un axioma (porque aparece en el primer renglón de la demostración); por la suposición (1), se concluye que el teorema tiene la propiedad Q.

Caso inductivo: Suponga que $n \geq 1$ tiene la propiedad S; el objetivo es demostrar que $n+1$ tiene la propiedad S. Sea ϕ un teorema de F con una demostración de $n+1$ pasos; algo de la forma $\phi_0, \phi_1, \ldots, \phi_{n-1}, \phi$, en donde cada ϕ_i es una fórmula de F. Hay dos casos sobre ϕ. Si ϕ es un axioma, entonces por la suposición (1), ϕ tiene la propiedad Q. Si no, ϕ es la conclusión de una regla de

inferencia con premisas en $\phi_0, \phi_1, \ldots, \phi_{n-1}$. Note que cualquier premisa tiene una demostración de máximo n líneas (la misma demostración de ϕ sirve como testigo). Entonces, por la hipótesis inductiva se obtiene que cada premisa tiene la propiedad Q. Por la suposición (2), dado que ϕ es la conclusión de una regla de inferencia cuyas premisas todas tienen la propiedad Q, necesariamente ϕ tiene la propiedad Q. En cualquiera de los dos casos, ϕ tiene la propiedad Q y consecuentemente $n+1$ tiene la propiedad S. Es decir, $S(n)$ implica $S(n+1)$.

Por el principio de inducción matemática todo $n \geq 1$ tiene la propiedad $S(n)$. Es decir, todo teorema de F tiene la propiedad Q. □

No debe ser una sorpresa "descubrir" que la inducción sobre teoremas comúnmente recibe el nombre de "demostración por inducción sobre el número de pasos de una demostración". Su uso se ilustra con el siguiente ejemplo.

Ejemplo 0.7

Considere el sistema ADD. Una fórmula $x + y = z$ de ADD se llama *balanceada* si y solo si el número de ocurrencias de '|' en x e y es igual al número de ocurrencias de '|' en z. Por ejemplo, $|| + ||| = |||||$ es balanceada pero $|| + | = |$ no lo es. El objetivo es demostrar que todo teorema de ADD es una fórmula balanceada (o, alternativamente, que si ϕ es un teorema de ADD, entonces ϕ es balanceada).

Para este fin, sea usa el principio de inducción sobre teoremas (Metateorema 0.14). En este ejemplo, Q es la propiedad de que una fórmula sea balanceada. El único axioma de ADD es $| + | = ||$; esta fórmula es balanceada dado que la cantidad de '|' a la izquierda y a la derecha de '=' es 2. Hay dos reglas de inferencia. En este ejemplo se razona sobre $R1$ y se deja al lector completar la demostración para $R2$. La premisa de $R1$ es $x + y = z$; sean $i, j, k \in \mathbb{N}$ la cantidad de ocurrencias de '|' en x, y y z, respectivamente. Como hipótesis se tiene que $i + j = k$. En la conclusión de $R1$, el número de ocurrencias de '|' en $x|$ es $i + 1$ y en $z|$ es $k + 1$. Entonces $(i + 1) + j = (k + 1)$ y la conclusión $x| + y = z|$ es balanceada, como se esperaba.

En el Ejemplo 0.7 se demuestra (usando el principio de inducción sobre teoremas) que si ϕ es un teorema de ADD, entonces ϕ es balanceada. Como se justificará en el desarrollo de este texto, esta conclusión es equivalente a la siguiente afirmación: si una fórmula de ADD no es balanceada, entonces no es uno de sus teoremas. Esto permite concluir directamente, por ejemplo, que la fórmula $|| + || = |||$ no es teorema de ADD.

Nota 0.15

Cuando se desee demostrar que una fórmula ϕ no hace parte del conjunto de teoremas de un sistema formal, posiblemente sea útil encontrar una propiedad que cumplen todos los teoremas de dicho sistema formal, pero que no cumple ϕ. Con base en esta propiedad universal de todos los teoremas, se puede justificar que ϕ no es uno de ellos.

Hay una forma alternativa para presentar la demostración del Ejemplo 0.7, usando como principio subyacente el mismo de la inducción sobre teoremas. El objetivo es contar con una demostración más algebraica, en la cual la manipulación simbólica permita ahorrar en prosa y, a su vez, haga los pasos deductivos posiblemente más claros para algunos lectores. Con este propósito, se enriquecerá el lenguaje con el que hasta ahora se cuenta.

Ejemplo 0.8

Considere el sistema ADD. Se propone la función P, definida para cualquier expresión ϕ de ADD, de la siguiente manera:

$P(\phi)$**:** cantidad de palotes en ϕ.

Por ejemplo, $P(|||+|) = 4$, $P(|+|=|||) = 5$, $P(=) = 0$ y $P(+) = 0$. Note que para cualquier fórmula $x + y = z$ de ADD se tiene que

$$P(x + y = z) = P(x) + P(y) + P(z).$$

Nota 0.16

Es importante resaltar que en el Ejemplo 0.8 los símbolos + y = aparecen con dos significados distintos. Por una parte, estos símbolos hacen parte del sistema formal ADD y no tienen nada que ver con la suma e igualdad de números, respectivamente. Por otra parte, los símbolos + y = en la definición de P si corresponden a la suma e igualdad de números naturales, respectivamente. En particular, la expresión que aparece al final del Ejemplo 0.8 puede ser escrita, con más precisión y evitando ambigüedades, anotando cada operador con un sufijo que indique el sistema formal al cual pertenece; por ejemplo:

$$P(x +_{\mathsf{ADD}} y =_{\mathsf{ADD}} z) =_{\mathbb{N}} P(x) +_{\mathbb{N}} P(y) +_{\mathbb{N}} P(z).$$

En general, este nivel de detalle se se evita en la escritura de expresiones para facilitar su lectura. Sin embargo, en algunas partes del texto se apelará a este tipo de detalles dado que evitan posibles confusiones al lector.

Con base en la meta-notación introducida en el Ejemplo 0.8 (i.e., P no hace parte del sistema formal), a continuación se presenta una demostración alternativa a la del Ejemplo 0.7.

Ejemplo 0.9

Considere el sistema ADD. Una fórmula $x + y = z$ de ADD se llama *balanceada* si y solo si $P(x) + P(y) = P(z)$. El objetivo es demostrar que todo teorema de ADD es una fórmula balanceada.

Para este fin, sea usa el principio de inducción sobre teoremas (Metateorema 0.14). En este ejemplo, Q es la propiedad definida para cualquier fórmula $x + y = z$ de la siguiente manera:

$$Q(x + y = z) : P(x) + P(y) = P(z).$$

El único axioma de ADD es $| + | = ||$; esta fórmula es balanceada dado que:

$$P(|) + P(|) = 1 + 1 = 2 = P(||).$$

Hay dos reglas de inferencia. En este ejemplo se razona sobre $R1$ y se deja al lector completar la demostración para $R2$. La premisa de $R1$ es $x + y = z$ y entonces la hipótesis es $P(x) + P(y) = P(z)$. Como $x| + y = z|$ es la conclusión de $R1$, el objetivo es demostrar que esta fórmula tiene la propiedad Q:

$$\begin{aligned} P(x|) + P(y) &= P(x) + 1 + P(y) && \text{(definición de } P) \\ &= (P(x) + P(y)) + 1 && \text{(aritmética)} \\ &= P(z) + 1 && \text{(suposición: } P(x) + P(y) = P(z)) \\ &= P(z|) && \text{(definición de } P). \end{aligned}$$

Es decir, si $x + y = z$ es balanceada, también lo es $x| + y = z|$.

Al final, la diferencia primordial entre las demostraciones de los ejemplos 0.7 y 0.9 radica en la forma en la cual se presentan las argumentaciones. La del Ejemplo 0.7 sigue una tradición más cercana a lo comúnmente usado en matemáticas, mientras que la del Ejemplo 0.9 está más alineada con lo propuesto y preferido

en este texto: es muchas ocasiones, en aras de la claridad, es posible y deseable 'calcular' una demostración.

Ejercicios

1. Complete el Ejemplo 0.7 con el caso para la regla $R2$ de ADD.
2. Complete el Ejemplo 0.9 con el caso para la regla $R2$ de ADD.
3. Demuestre por qué las siguientes fórmulas no son teoremas de ADD:
   a) $|+|=|$
   b) $||+|=||||$
   c) $|+||=||||$
4. En el Ejemplo 0.7 se demuestra que si una fórmula es teorema de ADD, entonces dicha fórmula es balanceada. Demuestre la otra dirección de esta afirmación: si una fórmula de ADD es balanceada, entonces dicha fórmula es teorema de ADD.
5. Considere el sistema formal PR cuyas fórmulas son cadenas (i.e., secuencias) de paréntesis. El lenguaje tiene dos símbolos: '(' y ')'. Cualquier expresión es una fórmula. El único axioma es (). Sean ϕ y ψ dos fórmulas de PR; hay tres reglas de inferencia:

$$\frac{\phi}{(\phi)}\ \textsc{Add} \qquad \frac{\phi}{\phi\phi}\ \textsc{Double} \qquad \frac{\phi()\psi}{\phi\psi}\ \textsc{Omit}.$$

   a) Demuestre que las siguientes fórmulas son teoremas de PR:
      1) ((())())
      2) (())()()()
      3) ()((())())
   b) Demuestre que todo teorema de PR tiene la propiedad de que la cantidad de paréntesis izquierdos es igual a la cantidad de paréntesis derechos.
6. Considere el sistema formal PR′ cuyas fórmulas, al igual que en el sistema PR (Ejercicio 5), son cadenas de paréntesis bien formadas. De hecho, en PR′, el lenguaje, el conjunto de fórmulas y el conjunto de axiomas son los mismos de PR. Las reglas de inferencia de PR′ son las siguientes:

$$\frac{\phi}{(\phi)}\ \textsc{Add'} \qquad \frac{\phi \qquad \psi}{\phi\psi}\ \textsc{Join}.$$

   a) Demuestre que cada una de las fórmulas en el Ejercicio 5*a* son teoremas de PR′.
   b) ¿Son todos los teoremas de PR teoremas de PR′? Justifique su respuesta.
   c) ¿Son todos los teoremas de PR′ teoremas de PR? Justifique su respuesta.

7. Sea F un sistema formal. Considere una función f que asigna a cada fórmula de F un valor en el conjunto $\{0, 1\}$ y que satisface las siguientes condiciones:
   a) Si ϕ es un axioma de F, entonces $f(\phi) = 0$.
   b) Si $\phi_0, \ldots, \phi_n$ son las premisas de una regla de inferencia de F con conclusión ϕ y $f(\phi_0) = f(\phi_1) = \cdots = f(\phi_n) = 0$, entonces $f(\phi) = 0$.

   Demuestre que si $\vdash_{\mathsf{F}} \psi$, entonces $f(\psi) = 0$.

8. Considere el sistema formal EVEN cuyos teoremas representan números naturales pares distintos a cero. El único símbolo del sistema formal es '|', cualquier expresión es una fórmula y el único axioma es ||. Hay una regla de inferencia:

   $$\frac{\phi}{\phi||}\ \textsc{Pile}.$$

   a) Demuestre que cualquier teorema de EVEN tiene una cantidad par de palotes.
   b) Demuestre que cualquier fórmula con una cantidad par de palotes es un teorema de EVEN.
   c) Demuestre que EVEN es decidible.
   d) Cambie el axioma de EVEN por uno nuevo, resultando en un sistema formal ODD, de tal manera que los teoremas de ODD representen exactamente los números naturales impares.

9. Considere el sistema formal 2POW descrito a continuación: hay dos símbolos 'a' y 'b', cualquier expresión es una fórmula, el único axioma es ab y hay una regla de inferencia:

   $$\frac{\phi}{a\phi b}\ \textsc{Extend}.$$

   a) Demuestre para $n \geq 1$, $a^n b^n$ es un teorema de 2POW (a^n denota la cadena de n apariciones de a).
   b) Sea ϕ un teorema de 2POW. Demuestre que hay un $n \geq 1$ tal que ϕ es $a^n b^n$.
   c) Demuestre que 2POW es decidible.

10. A continuación se describe el sistema formal PAL cuyos teoremas son todas los palíndromes sobre 3 letras. El lenguaje tiene tres símbolos 'a', 'b' y 'c', cualquier expresión es una fórmula y hay seis axiomas: a, b, c, aa, bb y cc. El sistema cuenta con tres reglas de inferencia en las cuales ϕ denota cualquier expresión de PAL:

    $$\frac{\phi}{a\phi a}\ \textsc{Add}\ a \qquad \frac{\phi}{b\phi b}\ \textsc{Add}\ b \qquad \frac{\phi}{c\phi c}\ \textsc{Add}\ c.$$

    a) Demuestre que las siguientes fórmulas son teoremas de PAL:
       1) *abccccba*

2) *abcccba*

b) Demuestre que todo teorema de PAL es palíndrome.

c) Demuestre que todo palíndrome sobre los símbolos 'a', 'b' y 'c' es teorema de PAL.

11. A continuación se describe el sistema formal MIU el cual aparece en el libro *Gödel, Escher, Bach* de D. Hofstadter (1973). Hay tres símbolos 'M', 'I' y 'U', cualquier expresión es una fórmula y hay un único axioma MI. El sistema tiene 4 reglas de inferencia, que se presentan a continuación. En ellas, ϕ y ψ representan cualquier expresión (i.e., secuencia finita de símbolos, posiblemente vacía):

$$\frac{\phi I}{\phi IU}\ \text{R1} \qquad \frac{M\phi}{M\phi\phi}\ \text{R2} \qquad \frac{\phi III\psi}{\phi U\psi}\ \text{R3} \qquad \frac{\phi UU\psi}{\phi\psi}\ \text{R4.}$$

a) Demuestre que las siguientes fórmulas son teoremas de MIU:

1) MIU

2) $MIUIUIUIU$

3) $MUIIIIU$

b) Demuestre que la cantidad de apariciones de I en un teorema de MIU nunca es múltiplo de 3.

c) ¿Es MU un teorema de MIU?

12. Considere el sistema formal COFFEE en el cual hay dos símbolos, '$\circ$' y '$\bullet$'. Cualquier expresión es una fórmula y hay un único axioma $\circ$. El sistema tiene 4 reglas de inferencia, las cuales se presentan a continuación. En ellas, ϕ y ψ representan cualquier expresión (i.e., secuencia finita de símbolos, posiblemente vacía):

$$\frac{\phi\bullet\psi}{\phi\bullet\circ\psi}\ \text{R1} \qquad \frac{\phi\bullet\psi}{\phi\circ\bullet\psi}\ \text{R2} \qquad \frac{\phi\circ\psi}{\phi\circ\circ\psi}\ \text{R3} \qquad \frac{\phi\circ\psi}{\phi\bullet\bullet\psi}\ \text{R4.}$$

a) Demuestre que las siguientes fórmulas son teoremas de COFFEE:

1) $\circ\circ\circ\circ$

2) $\circ\circ\bullet\circ\bullet$

3) $\circ\circ\circ\bullet\bullet\circ\bullet\circ\bullet$

b) Demuestre que la cantidad de apariciones de $\bullet$ en un teorema de COFFEE es par.

13. Considere el sistema formal C$\bullet$FFEE que resulta de COFFEE cambiando el axioma $\circ$ por $\bullet$. La reglas de inferencia de C$\bullet$FFEE son las mismas que en COFFEE. Demuestre o refute:

a) La cantidad de apariciones de $\bullet$ en un teorema de C$\bullet$FFEE es par.

b) La cantidad de apariciones de $\bullet$ en un teorema de C$\bullet$FFEE es impar.

c) La cantidad de apariciones de $\circ$ en un teorema de C$\bullet$FFEE es impar.

d) La cantidad de apariciones de $\circ$ en un teorema de C$\bullet$FFEE es par.

14. Considere el sistema formal MCOFFEE en el cual hay dos símbolos, '$\circ$' y '$\bullet$'. Cualquier expresión es una fórmula y hay un único axioma $\circ$. El sistema tiene 4 reglas de inferencia, las cuales se presentan a continuación. En ellas, ϕ y ψ representan cualquier expresión (i.e., secuencia finita de símbolos, posiblemente vacía):

$$\frac{\phi \circ\circ\circ \bullet\psi}{\phi \bullet \circ\psi}\ \text{R1} \qquad \frac{\phi \bullet \psi}{\phi \circ \bullet\psi}\ \text{R2} \qquad \frac{\phi \circ\circ\circ \circ\psi}{\phi \bullet \bullet\psi}\ \text{R3} \qquad \frac{\phi \circ \psi}{\phi \circ \circ\psi}\ \text{R4.}$$

   a) Demuestre que las siguientes fórmulas son teoremas de MCOFFEE:
      1) $\circ\circ\circ\bullet\bullet$
      2) $\circ\bullet\bullet\circ$
      3) $\circ\circ\circ\bullet\bullet\circ\bullet\circ\bullet$

   b) Demuestre que la cantidad de apariciones de $\bullet$ en un teorema de MCOFFEE es par.

Parte 1

Lógica proposicional

Capítulo 1

Lenguaje y especificación

> The limits of my language means the limits of my world.
>
> Ludwig Wittgenstein

El propósito de la lógica en informática es desarrollar lenguajes para especificar (i.e., modelar) situaciones reales a la cuales se enfrenta un profesional cuando especifica, analiza, diseña, construye, prueba, verifica, etc. software, de tal modo que sea posible razonar formalmente acerca de ellas. En este contexto, *razonar* significa construir argumentaciones rigurosas sobre las situaciones de tal forma que (i) estos sean válidos, (ii) se puedan defender totalmente y, en algunos casos, (iii) puedan ser ejecutados en una máquina.

Considere la argumentación en el Ejemplo 1.1.

Ejemplo 1.1

Si el tren llega tarde y no hay taxis en la estación, entonces Juan llegará tarde a su reunión. Juan no llega tarde a su reunión. El tren llegó tarde. *Consecuentemente*, había taxis en la estación.

La argumentación en el Ejemplo 1.1 es 'justificable' dado que si las frases primera y tercera se analizan en conjunto, estas indican que si no hay taxis, entonces Juan llega tarde a la reunión. La segunda frase indica que Juan no llegó tarde a la reunión: esto quiere decir que necesariamente había taxis en la estación. Informalmente, una argumentación es correcta cuando la frase después de la palabra 'consecuentemente' *debe* ser cierta si las frases que le preceden lo son.

Ahora considere la argumentación en el Ejemplo 1.2.

Ejemplo 1.2

Si está lloviendo y Juana no tiene una sombrilla a su alcance, entonces ella se empapará. Juana no se empapó. Está lloviendo. *Consecuentemente*, Juana tiene una sombrilla a su alcance.

La argumentación en el Ejemplo 1.2 es intuitivamente cierta. Al examinarla con cuidado, se puede observar que ¡tiene la misma estructura de la argumentación en el Ejemplo 1.1! Lo que ha sucedido de un ejemplo a otro es que algunas frases y fragmentos han sido sustituidos:

Ejemplo 1.1	**Ejemplo 1.2**
el tren está tarde	está lloviendo
hay taxis en la estación	Juana tiene una sombrilla a su alcance
Juan llega tarde a la reunión	Juana se empapa

Unificando, se puede concluir que la veracidad de las argumentaciones en los ejemplos 1.1 y 1.2 puede ser analizada sin referirse a protagonistas específicos como Juan, Juana, trenes, taxis, sombrillas o lluvia. Esta unificación puede lograrse de la siguiente forma:

> Si p y no q, entonces r.
>
> No r.
>
> p.
>
> Consecuentemente, q.

Como moraleja, es importante entender que al desarrollar una lógica la principal preocupación no es saber el significado particular de cada frase sino más bien aprender sobre su estructura. Obviamente, al *aplicar* un razonamiento, como en los casos anteriores, el significado de cada frase es importante.

Este capítulo se ocupa de desarrollar el lenguaje de la lógica proposicional como lenguaje para especificar. El siguiente capítulo explica cuándo una especificación es 'cierta' y cuándo una argumentación es 'válida'.

1.1. Proposiciones

Para razonar rigurosamente es indispensable desarrollar un lenguaje que sirva el propósito de especificar afirmaciones de forma tal que su estructura interna sea

evidente y pueda ser analizada sistemáticamente. El lenguaje de la lógica proposicional está basado en *proposiciones* o *sentencias declarativas*, i.e., afirmaciones o frases con un valor de verdad asociado.

Ejemplo 1.3

Las siguientes frases son ejemplos de proposiciones:

1. La suma de los números 3 y 5 es 8.
2. Juana reaccionó agresivamente ante las acusaciones de Juan.
3. Todo número natural $n > 2$ par puede expresarse como la suma de dos números primos.
4. A todo marciano le gusta el calentado con fríjoles.
5. Such a good actress, hiding all her pain, trading in memories for fortune and fame.
6. Albert Camus était un écrivain français.
7. Die Würde des Menschen ist unantastbar.
8. jIyajbe'.

Todas estas frases son declarativas dado que cada una de ellas es 'verdadera' o 'falsa'. La frase (1) puede ser verificada con aritmética básica (y suponiendo tácitamente una representación arábica en base diez de los números naturales). La frase (2) es un poco problemática. Para determinar su valor de verdad es necesario saber quiénes son Juana y Juan, y posiblemente tener un recuento fehaciente sobre la situación allí descrita. En principio, si alguien estuvo en la escena, entonces pudo haber detectado la reacción 'agresiva' de Juana. La frase (3) es conocida como la *conjetura de Goldbach* y a simple vista parece trivial. Claramente una afirmación acerca de *todos* los números naturales pares mayores que 2 es verdadera o falsa. Al día de hoy, nadie sabe si la conjetura de Goldbach es verdadera o falsa. La frase (4) parece un poco tonta: afirma que *si* los marcianos existen y gustan comer calentado, entonces o lo prefieren con fríjoles o sin fríjoles. Esta frase tiene asociado un valor de verdad independientemente de si existan o no los marcianos; en este sentido es una proposición. Et alors, qu'est-ce qu'on pense des phrases (5), (6), (7) et (8)? Las frases (5), (6), (7) y (8) son también proposiciones; esto lo puede corroborar si puede leer un poco de, respectivamente, inglés, francés, alemán o klingon.

Nota 1.1

En este texto se considera como proposición a una frase que tenga un valor de verdad asociado, independientemente de si este valor corresponde o no a la realidad. Además, una proposición puede estar escrita en cualquier lenguaje, natural o artifical.

Ejemplo 1.4

Ejemplos de frases que no son proposiciones:

- ¿Puede pedir un domicilio?
- En sus marcas, listos, fuera.
- Que la fuerza lo acompañe.

Las lógicas estudiadas en este texto son *simbólicas* por naturaleza y su desarrollo sigue una metodología que se describe a continuación. Inicialmente, se especifica un conjunto suficientemente grande de proposiciones del Castellano como cadenas de símbolos. Estas cadenas son cortas pero codifican las proposiciones. Posteriormente, estas proposiciones son analizadas sistemáticamente por medio de su significado matemático o de un sistema de inferencia. En algunas ocasiones, labores de validación de especificaciones son mecanizadas, aprovechando la facilidad y velocidad de una máquina para manipular expresiones simbólicamente.

Ejercicios

1. Los siguientes párrafos han sido tomados de Wikipedia. De estos párrafos, liste tres proposiciones y tres frases que no sean proposiciones.

   > La lógica matemática estudia los sistemas formales en relación con el modo en el que codifican conceptos intuitivos de objetos matemáticos como conjuntos, números, demostraciones y computación. La lógica estudia las reglas de deducción formales, las capacidades expresivas de los diferentes lenguajes formales y las propiedades metalógicas de los mismos.
   > En un nivel elemental, la lógica proporciona reglas y técnicas para determinar si es o no válido un argumento dado dentro de un determinado sistema formal. En un nivel avanzado, la lógica matemática se ocupa de la posibilidad de axiomatizar las teorías matemáticas, de clasificar su capacidad

> expresiva, y desarrollar métodos computacionales útiles en sistemas formales. La teoría de la demostración y la matemática inversa son dos de los razonamientos más recientes de la lógica matemática abstracta. Debe señalarse que la lógica matemática se ocupa de sistemas formales que pueden no ser equivalentes en todos sus aspectos, por lo que la lógica matemática no es método de descubrir verdades del mundo físico real, sino sólo una fuente posible de modelos lógicos aplicables a teorías científicas, muy especialmente a la matemática convencional.
> La lógica matemática no se encarga por otra parte del concepto de razonamiento humano general o del proceso creativo de construcción de demostraciones matemáticas mediante argumentos rigurosos pero hechas usando lenguaje informal con algunos signos o diagramas, sino sólo de demostraciones y razonamientos que pueden ser completamente formalizados en todos sus aspectos.

2. Los siguientes párrafos han sido tomados de Wikipedia. De estos párrafos, liste tres proposiciones y tres frases que no sean proposiciones.

> Una especificación formal usa notación matemática para describir de manera precisa las propiedades que un sistema de información debe tener, sin preocuparse por la forma de obtener dichas propiedades. Describe lo que el sistema debe hacer sin decir cómo se va a hacer.
> Esta abstracción hace que las especificaciones formales sean útiles en el proceso de desarrollar un sistema, porque permiten responder preguntas acerca de lo que el sistema hace con confianza, sin la necesidad de tratar con una gran cantidad de información no relevante que se encuentra en el código de programa del sistema en un lenguaje de programación cualquiera, o especular sobre el significado de frases en un impreciso Pseudocódigo.
> Una especificación formal puede servir como un punto de referencia fiable tanto para quienes se dedican a investigar sobre los requerimientos del cliente que solicita el sistema, como para aquellos que desarrollan los programas para satisfacer esos requerimientos, y también para los que redactan manuales de instrucciones para el sistema. Debido a que es independiente del código del programa, las especificaciones formales de un sistema pueden ser elaboradas a principios de su desarrollo; y puede ser un medio valioso para promover un entendimiento común entre todos los interesados en el sistema.

3. Considere los siguientes tres argumentos:
   a) Si está soleado, entonces es de día. Está soleado. Por lo tanto, es de día.
   b) Si no es martes, entonces es lunes. No es martes. Por lo tanto, es lunes.
   c) Todos los planetas giran alrededor del Sol. Marte es un planeta. Por lo tanto, Marte gira alrededor del Sol.

   Identifique dos argumentos que tengan la misma estructura.

4. Investigue quién inventó el Klingon y cómo se hizo popular. Además presente tres proposiciones en Klingon con su respectiva traducción al Castellano.

1.2. Lenguaje formal

El lenguaje de la lógica proposicional se describe formalmente en dos fases. Primero, se define el conjunto de símbolos del lenguaje. Segundo, se define el conjunto de fórmulas que no serán otra cosa que las proposiciones de la lógica.

Definición 1.2

Los *símbolos* del lenguaje de la lógica proposicional son:

- Una colección infinita $\mathcal{V}$ de *variables proposicionales*

  $$p_0, p_1, p_2, \ldots$$

- Paréntesis izquierdo '(' y paréntesis derecho ')'.
- Una colección de *conectivos lógicos*

  $$\mathit{true}, \mathit{false}, \neg, \equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow .$$

Las variables proposicionales $p_0, p_1, \ldots$ en $\mathcal{V}$ representan proposiciones *atómicas* o *indivisibles*, i.e., proposiciones cuya estructura interna no exhibe ningún conectivo lógico. Al ser infinita, la colección de variables proposicionales garantiza que siempre haya símbolos suficientes para especificar cualquier proposición. Los paréntesis izquierdo y derecho se usan para puntuar y agrupar proposiciones. Los conectivos lógicos reciben los siguientes nombres y tienen las siguientes interpretaciones intuitivas:

Símbolo	**Nombre**	**Interpretación**
true	verdadero	verdad
false	falso	falsedad
$\neg$	negación	no $\cdots$
$\equiv$	equivalencia	$\cdots$ si y solo si $\cdots$
$\not\equiv$	discrepancia	$\cdots$ discrepa de $\cdots$
$\vee$	disyunción	$\cdots$ o $\cdots$
$\wedge$	conjunción	$\cdots$ y $\cdots$
$\rightarrow$	implicación	si $\cdots$, entonces $\cdots$
$\leftarrow$	consecuencia	$\cdots$ si $\cdots$

Verdadero y falso son *constantes* (sin argumentos), la negación es un conectivo *unario* (un argumento) y los demás conectivos son *binarios* (dos argumentos). Los paréntesis y los conectivos lógicos son también llamados *símbolos lógicos*.

Nota 1.3

En las especificaciones y traducciones del Castellano al lenguaje de la lógica proposicional, los símbolos lógicos siempre se interpretarán de la misma forma.

En general, cada una de las variables proposicionales es una fórmula de la lógica proposicional (o proposición), al igual que las constantes *true* y *false*. Además, una proposición puede obtenerse negando una proposición o usando un conectivo lógico binario para componer dos proposiciones. Los paréntesis se usan para "puntuar" las proposiciones y evitar, por ejemplo, que una proposición pueda construirse de más de una forma. Formalmente, las fórmulas de la lógica proposicional (o proposiciones) corresponden a las cadenas formadas con los símbolos de la lógica proposicional de acuerdo como se describe en la Definición 1.2.

Definición 1.4

Las *fórmulas de la lógica proposicional* (o *proposiciones*) son aquellas cadenas obtenidas usando una cantidad finita de veces las siguientes reglas de construcción:

1. Cada variable proposicional es una proposición.
2. *true* y *false* son proposiciones.
3. Si ϕ es una proposición, entonces $(\neg\phi)$ es una proposición.
4. Si ϕ y ψ son proposiciones, entonces:
   a) $(\phi \equiv \psi)$ es una proposición
   b) $(\phi \not\equiv \psi)$ es una proposición
   c) $(\phi \vee \psi)$ es una proposición
   d) $(\phi \wedge \psi)$ es una proposición
   e) $(\phi \rightarrow \psi)$ es una proposición
   f) $(\phi \leftarrow \psi)$ es una proposición

La expresión $\mathcal{T}(\mathcal{V})$ denota el conjunto de proposiciones con variables proposicionales en $\mathcal{V}$.

De acuerdo con la Definición 1.4, una proposición es una variable proposicional, una constante o una fórmula construída a partir de unas proposiciones más sencillas y los conectivos lógicos. Note que $\mathcal{V} \subseteq \mathcal{T}(\mathcal{V})$ porque cada variable proposicional es una proposición. La proposición $(\neg\phi)$ es llamada *negación* de ϕ. La proposición

$(\phi \equiv \psi)$ es la *equivalencia* de ϕ y ψ, mientras que la proposición $(\phi \not\equiv \psi)$ es la *discrepancia* de ϕ y ψ. La proposición $(\phi \vee \psi)$ es una *disyunción* con *disyuntos* ϕ y ψ. La proposición $(\phi \wedge \psi)$ es una *conjunción* con *conyuntos* ϕ y ψ. La proposición $(\phi \rightarrow \psi)$ es una *implicación* con *antecedente* ϕ y *consecuente* ψ. La proposición $(\phi \leftarrow \psi)$ es una *consecuencia* con *consecuente* ϕ y *antecedente* ψ. Finalmente, note que $\mathcal{V}$ es un subconjunto propio de $\mathcal{T}(\mathcal{V})$, escrito $\mathcal{V} \subset \mathcal{T}(\mathcal{V})$.

Ejemplo 1.5

Cada una de las siguientes expresiones es una proposición:

- p_4
- *false*
- $((\neg p_1) \equiv p_2)$
- $(\neg(p_1 \vee p_2))$
- $((p_1 \rightarrow p_2) \wedge (\neg p_6))$

Por ejemplo, para justificar la última expresión: las variables proposicionales p_1, p_2, p_6 son proposiciones; entonces $(p_1 \rightarrow p_2)$ y $(\neg p_6)$ son proposiciones, y consecuentemente $((p_1 \rightarrow p_2) \wedge (\neg p_6))$ es una proposición.

Una forma alternativa para determinar si una expresión es una proposición, es usar el *principio de inversión*. Este principio indica que el proceso de construcción de proposiciones *siempre* se puede invertir porque dada una proposición, es posible saber cuál de las reglas de construcción (Definición 1.4) fue la última usada para construírla. En particular, se puede determinar si una expresión es una proposición aplicando recurrentemente el principio de inversión sobre sus subexpresiones tratando de usar como mecanismo de 'división' la última regla de construcción aplicada. Si este proceso es exitoso para todas las subexpresiones, entonces la expresión inicial es una proposición; de lo contrario, no lo es. Note que el principio de inversión también permite determinar si una expresión *no* es una proposición, algo tan importante como determinar si lo es. El siguiente ejemplo ilustra la aplicación del principio de inversión para detectar que una expresión no es una proposición.

Ejemplo 1.6

Considere la expresión $(\neg)() \vee p_0 p_1 \rightarrow$ que únicamente usa los símbolos del lenguaje de la lógica proposicional. Note la mención del símbolo de negación $\neg$ en la expresión, el cual, por las reglas de construcción de proposiciones, no puede estar a

la derecha de una proposición. Entonces debe estar a la izquierda de una proposición. Sin embargo, ninguna proposición inicia con el símbolo de paréntesis derecho ‘)’ dado que la única forma de usar el paréntesis derecho es justo después de una proposición (¡revise esto para cada una de las reglas de construcción!) y $\neg$ no es una proposición. Entonces, la expresión $(\neg)() \vee p_0 p_1 \rightarrow$ no es una proposición.

La naturaleza inductiva de la Definición 1.4 directamente abre las puertas a la mecanización algorítmica de aspectos relacionados con la sintaxis de las proposiciones. Por ejemplo, es natural pensar en el diseño recurrente de un algoritmo que dada una expresión en el lenguaje de la lógica proposicional, determine si esta es una proposición o no.

Las definiciones inductivas, como la Definición 1.4, son tan frecuentes en informática que existe una notación especialmente diseñada para describirlas de manera abreviada. Esta notación es comúnmente conocida con el nombre de *notación Backus-Naur* (BNF, del inglés *Backus-Naur form*). En BNF, la Definición 1.4 puede escribirse compactamente como: BNF

$$\begin{aligned}\phi ::= {} & p \mid \mathit{true} \mid \mathit{false} \mid (\neg\phi) \mid (\phi \equiv \phi) \mid (\phi \not\equiv \phi) \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \\ & \mid (\phi \rightarrow \phi) \mid (\phi \leftarrow \phi)\end{aligned}$$

en donde p representa cualquier variable proposicional y cada mención de ϕ a la derecha de ‘::=’ representa cualquier proposición que ya ha sido construída independientemente de otras menciones de ϕ. La barra ‘|’ se usa para separar los diferentes casos de construcción y se lee como ‘o’.

Nota 1.5

Además, se adoptan algunas convenciones para escribir proposiciones. Primero, las letras minúsculas $p, q, r, \ldots$ se prefieren sobre $p_0, p_1, p_2, \ldots$, a pesar de que estas últimas son más precisas. Segundo, siguiendo la tradición del uso de letras griegas en matemáticas, las letras griegas minúsculas $\phi, \psi, \tau, \gamma, \delta, \ldots$ se usan para denotar proposiciones y las mayúsculas $\Phi, \Psi, \Gamma, \Delta, \ldots$ se usan para denotar conjuntos de proposiciones.

Ejemplo 1.7

Usando las convenciones en la Nota 1.5, la argumentación en el Ejemplo 1.1, puede especificarse con la siguiente secuencia de cuatro proposiciones

$$((p \wedge (\neg q)) \rightarrow r),\ (\neg r),\ p,\ q$$

en donde las variables proposicionales simbolizan:

p: el tren está tarde

q: hay taxis en la estación

r: Juan llega tarde a su reunión

Ejercicios

1. Justifique por qué cada una de las siguientes expresiones es una proposición:
   a) p
   b) $(true \equiv false)$
   c) $(q \wedge (\neg q))$
   d) $(p \vee (p \equiv (\neg q)))$
   e) $(\neg(\neg(\neg(p \rightarrow r))))$
   f) $((q \wedge (\neg q)) \leftarrow (\neg(\neg(\neg(p \rightarrow r)))))$

2. Justifique por qué las siguientes expresiones no son proposiciones:
   a) $(p\vee)$
   b) $(true)$
   c) $\neg p$
   d) $p \vee q \vee r$
   e) $(p \vee q) \vee r$
   f) $((p \equiv q) \wedge (r))$

3. Use el lenguaje de la lógica proposicional para especificar las siguientes proposiciones:
   a) Un número natural es par si y solo si no es impar.
   b) Si el sol brilla hoy, entonces no brilla mañana.
   c) Juan estaba celoso o estaba de mal genio.
   d) Si una petición ocurre, entonces eventualmente será atendida o el proceso de horarios se bloqueará.
   e) Hoy lloverá o hará sol, pero no las dos.
   f) Sin zapatos o camisa no hay servicio en el restaurante.
   g) Mi hermana quiere un gato blanco y negro.

h) Mi pareja ni raja ni presta el hacha.

4. Considere la siguiente especificación:
   h: El cuarteto interpretará a Haydn.
   m: El cuarteto interpretará a Mozart.
   Con base en la especificación anterior, traduzca del lenguaje de la lógica proposicional al Castellano cada una de las siguientes proposiciones procurando que dicha traducción sea lo más cercana posible al lenguaje cotidiano:
   a) $(h \vee m)$
   b) $(h \equiv (\neg m))$
   c) $(\neg(h \equiv m))$
   d) $(\neg(h \wedge m))$
   e) $(((\neg(h \wedge m)) \wedge (\neg h)) \rightarrow m)$
5. Use el lenguaje de la lógica proposicional para especificar cada una de las siguientes argumentaciones, indicando claramente en cada caso la especificación de las variables proposicionales:
   a) Si Pedro entiende matemáticas, entonces puede entender lógica. Pedro no entiende lógica. Consecuentemente, Pedro no entiende matemáticas.
   b) Si llueve o cae nieve, entonces no hay electricidad. Llueve. Entonces, no habrá electricidad.
   c) Si llueve o cae nieve, entonces no hay electricidad. Hay electricidad. Entonces no nevó.
   d) Si $\sin x$ es diferenciable, entonces $\sin x$ es contínua. Si $\sin x$ es contínua, entonces $\sin x$ es diferenciable. La función $\sin x$ es diferenciable. Consecuentemente, la función $\sin x$ es integrable.
   e) Si Gödel fuera presidente, entonces el Congreso presentaría leyes razonables. Gödel no es presidente. Por lo tanto, el Congreso no presenta leyes razonables.
   f) Si llueve, entonces no hay picnic. Si cae nieve, entonces no hay picinic. Llueve o cae nieve. Por lo tanto, no hay picnic.

1.3. Árboles de sintaxis

El uso excesivo de paréntesis es tedioso para los humanos. La razón por la cual los paréntesis son necesarios en las proposiciones es debido a que, a pesar de que comúnmente sean escritas como líneas de texto, las proposiciones en realidad tienen forma de ¡árbol! Los paréntesis son los signos de puntuación que permiten escribir proposiciones linealmente. La Figura 1 presenta el árbol de sintaxis correspondiente a la proposición $(((p \rightarrow q) \wedge (\neg \mathit{true})) \equiv (r \vee q))$.

Note que los paréntesis son innecesarios en un árbol de sintaxis porque su estructura arborescente elimina cualquier ambigueadad posible al representar una

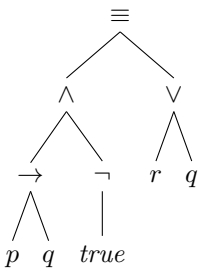


Figura 1. Árbol de sintaxis de $(((p \to q) \wedge (\neg true)) \equiv (r \vee q))$.

proposición. Al escribir una proposición linealmente, su estructura arborescente se conserva insertando paréntesis que agrupan subproposiciones que corresponden a subárboles.

Un árbol de sintaxis tiene como *raíz* (i.e., primer símbolo de arriba a abajo) una variable proposicional o un conectivo lógico. En el primer caso, la variable proposicional es el único símbolo que aparece en el árbol. En el caso en que la raíz sea el conectivo lógico '$\neg$', entonces la raíz tiene exactamente *un* subárbol. En cualquier otro caso, la raíz tiene exactamente *dos* subárboles. En cualquiera de estas situaciones, los subárboles se comportan tal y como se acaba de describir el comportamiento de la raíz (este es un ejemplo de una estructura definida inductivamente).

Pensar acerca de proposiciones usando su representación en árbol de sintaxis puede ser útil para entender nociones como, por ejemplo, la de subproposición (o, equivalentemente, la de subfórmula).

Definición 1.6

Sea ϕ una proposición. Una *subproposición de* ϕ es una proposición que corresponde a uno de los subárboles en el árbol de sintaxis de ϕ.

Ejemplo 1.8

Las subproposiciones de $(((p \to q) \wedge (\neg true)) \equiv (r \vee q))$ se listan a continuación:

p	q	r
$true$	$(p \to q)$	$(\neg true)$
$(r \vee q)$	$((p \to q) \wedge (\neg true))$	
$(((p \to q) \wedge (\neg true)) \equiv (r \vee q))$.		

Hay árboles de símbolos que no son árboles de sintaxis por la misma razón que hay expresiones en el lenguaje de la lógica proposicional que no son proposiciones.

Ejemplo 1.9

Considere el siguiente árbol de símbolos:

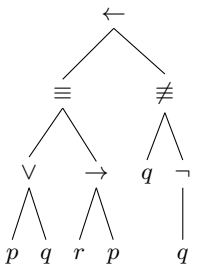
Este árbol no es un árbol de sintaxis porque una discrepancia aplica sobre dos proposiciones y no sobre una sola.

Ejercicios

1. Dibuje el árbol de sintaxis para cada una de las siguientes proposiciones:
   a) p
   b) true
   c) $(p \equiv r)$
   d) $((p \wedge (\neg q)) \rightarrow r)$
   e) $((p \wedge q) \vee ((\neg p) \wedge (\neg q)))$
   f) $(p \rightarrow (q \rightarrow p))$
   g) $((p \vee r) \leftarrow (p \not\equiv q))$
   h) $(\neg((\mathit{false} \wedge (r \leftarrow (p \vee s))) \equiv (\neg((p \rightarrow q) \vee (r \wedge (\neg r))))))$
2. Liste todas las subproposiciones de cada una de las siguientes proposiciones:
   a) p
   b) $(p \equiv r)$
   c) $((p \wedge (\neg q)) \rightarrow r)$
   d) $((p \wedge q) \vee ((\neg p) \wedge (\neg q)))$
   e) $(p \rightarrow (q \rightarrow p))$
   f) $((p \vee r) \leftarrow (p \wedge q))$
   g) $(\neg((r \wedge (r \leftarrow (p \vee s))) \equiv (\neg((p \rightarrow q) \vee (r \wedge (\neg r))))))$
3. Dibuje árbol de sintaxis para cada una de los siguientes casos:
   a) Una proposición que es una negación de una equivalencia.
   b) Una proposición que es una disyunción cuyos disyuntos ambos son conjunciones.
   c) Una proposición que es una conjunción de conjunciones.
   d) Una proposición que es una implicación cuyo antecedente es una negación y consecuente es una equivalencia.

e) Una proposición que es una consecuencia cuyo antecedente es una disyunción y consecuente una discrepancia.

4. Escriba la proposición correspondiente al siguiente árbol de sintaxis:



5. En cada uno de los siguientes casos, dibuje un árbol de sintaxis que no represente una proposición y que satisfaga las condiciones dadas:
   a) Al extenderlo el árbol resultante represente una proposición.
   b) Sea patológicamente mal formado, i.e., no hay forma de extenderlo con subárboles de tal modo que el árbol obtenido represente una proposición.

1.4. Inducción sobre proposiciones

Para demostrar que toda proposición (o una gran cantidad de ellas) tiene una propiedad dada se emplea el principio de *inducción sobre proposiciones*. Este principio de inducción se presenta en el Metateorema 1.7; su demostración se obtiene directamente del principio de inducción matemática sobre los números naturales (Metateorema 0.5). La inducción sobre proposiciones a veces recibe el nombre de *inducción sobre el número de conectivos de la proposición* o *inducción sobre la complejidad de una proposición*.

Metateorema 1.7

Sea Q una propiedad sobre proposiciones. Para demostrar que toda proposición ϕ tiene la propiedad Q, basta con demostrar:

1. Cada una de las variables proposicionales tiene la propiedad Q.
2. Cada una de las constantes *true* y *false* tiene la propiedad Q.
3. Si ϕ es de la forma $(\neg\psi)$ y ψ tiene la propiedad Q, entonces ϕ tiene la propiedad Q.
4. Si ϕ es de la forma $(\psi \equiv \tau)$, $(\psi \not\equiv \tau)$ $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \to \tau)$ o $(\psi \leftarrow \tau)$, y ψ y τ tienen la propiedad Q, entonces ϕ tiene la propiedad Q.

Demostración. Considere la siguiente propiedad para $n \in \mathbb{N}$:

$$S(n) : \text{Cualquier proposición con a lo sumo } n \text{ conectivos lógicos tiene la propiedad } Q.$$

La demostración procede por inducción matemática sobre $n \in \mathbb{N}$ (Metateorema 0.5), suponiendo que (1), (2), (3) y (4) son ciertos.

Caso base: Se debe demostrar $S(0)$, es decir, que cualquier proposición con 0 conectivos lógicos tiene la propiedad Q. Las únicas proposiciones con 0 conectivos lógicos son las variables proposicionales que, por la suposición (1), tienen la propiedad Q.

Caso inductivo: Se supone $S(n)$ $(n \geq 0)$ y se demuestra $S(n+1)$. Suponga que ϕ tiene $n+1$ conectivos lógicos. Bajo esta suposición ϕ no puede ser una variable proposicional; entonces, hay 9 casos: que ϕ sea de la forma *true*, *false*, $(\neg\psi)$, $(\psi \equiv \tau)$, $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \to \tau)$ o $(\psi \leftarrow \tau)$. Considere el caso en que ϕ sea de la forma $(\psi \equiv \tau)$. Tanto ψ como τ tienen una cantidad de conectivos lógicos estrictamente menor que la cantidad de conectivos lógicos en ϕ. Entonces, por la propiedad (4), al suponer $S(n)$, ϕ tiene la propiedad Q. Los demás 8 casos son similares y se proponen como ejercicios para el lector.

Entonces, toda proposición tiene la propiedad Q. □

En el Metateorema 1.7, los numerales (1) y (2) corresponden a los casos base de una demostración por inducción, mientras que los numerales (3) y (4) corresponden a los casos inductivos; la suposición de que ψ o τ tienen la propiedad Q son las *hipótesis inductivas*.

Ejemplo 1.10

Se desea demostrar que toda proposición tiene la misma cantidad de paréntesis izquierdos y derechos. Para una proposición ϕ, se definen las funciones L y R de la siguiente manera:

$$L(\phi) : \text{número de paréntesis izquierdos en } \phi,$$
$$R(\phi) : \text{número de paréntesis derechos en } \phi.$$

Para cualquier proposición ϕ, el objetivo es demostrar la propiedad P:

$$P(\phi) : L(\phi) = R(\phi).$$

La demostración procede por inducción sobre ϕ.

Caso base: Si ϕ es una variable proposicional, entonces ϕ no tiene paréntesis alguno y se tiene que $L(\phi) = 0 = R(\phi)$. Lo mismo sucede si ϕ es una constante.

Caso inductivo: Hay 7 casos: que ϕ sea de la forma $(\neg\psi)$, $(\psi \equiv \tau)$, $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \rightarrow \tau)$ o $(\psi \leftarrow \tau)$. Suponga que ϕ es de la forma $(\psi \equiv \tau)$. Por la hipótesis inductiva, se sabe que $L(\psi) = R(\psi)$ y $L(\tau) = R(\tau)$. Note que:

$$\begin{aligned} L(\phi) &= L((\psi \equiv \tau)) \\ &= 1 + L(\psi) + L(\tau) \\ &= 1 + R(\psi) + R(\tau) \\ &= R((\psi \equiv \tau)) = R(\phi). \end{aligned}$$

En cualquiera de las dos situaciones $L(\phi) = R(\phi)$, como se esperaba. Los casos en que ϕ es de la forma $(\neg\psi)$, $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \rightarrow \tau)$ o $(\psi \leftarrow \tau)$ son similares y se proponen como ejercicios para el lector.

Ejercicios

1. Complete la demostración del Metateorema 1.7 con los casos en que ϕ sea de las forma *true*, *false*, $(\neg\psi)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \rightarrow \tau)$ o $(\psi \leftarrow \tau)$.
2. Complete el Ejemplo 1.10 con los casos en que ϕ sea de la forma $(\neg\psi)$, $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \rightarrow \tau)$ o $(\psi \leftarrow \tau)$.
3. Proponga definiciones inductivas para las funciones L y R en el Ejemplo 1.10.
4. Proponga definiciones inductivas para las funciones *Con* y *Con'* tales que, para cualquier proposición φ:

   Con(φ) : cantidad de símbolos (i.e., variables proposicionales, conectivos lógicos y paréntesis) en φ.

   Con'(φ) : cantidad de símbolos distintos a paréntesis en φ.
5. Demuestre que cualquier proposición que no sea una variable proposicional ni una constante inicia con un paréntesis izquierdo y termina con un paréntesis derecho.
6. Demuestre que cualquier proposición que no menciona variables proposicionales tiene al menos una constante.
7. Demuestre que cualquier fórmula con al menos una mención de $\equiv$ y que no menciona constantes, es tal que menciona al menos dos variables proposicionales (no necesariamente distintas).
8. Sea *Prop*$(\neg, \vee)$ el conjunto de proposiciones que tienen a $\neg$ y $\vee$ como únicos conectivos lógicos. Demuestre que si $\phi \in$ *Prop*$(\neg, \vee)$, entonces ϕ es de la forma

p, $(\neg p)$, $(\neg(\neg\psi))$, $(\psi \vee \tau)$ o $(\neg(\psi \vee \tau))$, en donde ψ y τ son proposiciones en $Prop(\neg, \vee)$.

9. Sea $Prop(\neg, \vee)$ el conjunto de proposiciones que tienen a $\neg$ y $\vee$ como únicos conectivos lógicos, y sea $\Gamma \subseteq Prop(\neg, \vee)$. Suponga que Γ satisface las siguientes cuatro condiciones:
   a) no existe una variable proposicional p tal que $\{p, (\neg p)\} \subseteq \Gamma$
   b) si $(\neg(\neg\phi)) \in \Gamma$, entonces $\phi \in \Gamma$
   c) si $(\neg(\phi \vee \psi)) \in \Gamma$, entonces $(\neg\phi) \in \Gamma$ y $(\neg\psi) \in \Gamma$
   d) si $(\phi \vee \psi) \in \Gamma$, entonces $\phi \in \Gamma$ o $\psi \in \Gamma$

   Demuestre que no existe $\tau \in Prop(\neg, \vee)$ tal que $\{\tau, (\neg\tau)\} \subseteq \Gamma$.

Capítulo 2

Semántica

> " The best lies about me are the ones I told."
>
> Patrick Rothfuss
> *The Name of the Wind*, 2007

En el Capítulo 1 se introdujo el lenguaje de la lógica proposicional como un primer paso hacia la definición de esta lógica como un sistema formal. Dicha definición se dió de la mano de interpretaciones intuitivas y coloquiales del significado de los conectivos lógicos.

Este capítulo aborda el estudio formal de la *semántica* de la lógica proposicional: semántica se refiere al significado matemático de las proposiciones. El estudio de la semántica proposicional está basado en funciones Booleanas que, dada una asignación de valores de verdad a las variables proposicionales de una proposición, asignan un único significado a dicha proposición. Las valuaciones se presentan como una herramienta complementaria a las funciones Booleanas, con la cual es posible razonar con la misma efectividad pero con la ventaja de que, en algunas ocasiones, permite determinar el significado de una proposición sin conocer completamente su estructura sintáctica.

La semántica proposicional es una herramienta efectiva para establecer relaciones importantes entre proposiciones. Este capítulo muestra cómo la semántica proposicional permite analizar y clasificar las proposiciones de manera general e independientemente del aparato deductivo que se elija como sistema formal proposicional. Es decir, dependiendo de la semántica de una proposición, esta puede ser clasificada y estudiada sistemáticamente sin necesidad usar un aparto deductivo.

Finalmente, la semántica proposicional es usada para determinar cuándo una argumentación, como las de los ejemplos 1.1 y 1.2 en el Capítulo 1, es correcta o no.

También se aborda, como caso de estudio, la especificación y la solución de cierta clase de acertijos lógicos usando como herramienta la semántica proposicional.

2.1. Funciones Booleanas

Una proposición expresa un hecho acerca del mundo, real o imaginario, de alguno 'abstracto' como lo es un modelo de computador o simplemente se refiere a una idea o un sentimiento. En cualquier escenario, una proposición tiene un único valor de verdad (así este no se conozca): es *verdadera* o es *falsa*. De esta forma, en el lenguaje formal de la lógica proposicional, cada variable proposicional puede ser verdadera o falsa, pero no ambas al tiempo. Igualmente, una proposición, construída a partir de variables proposicionales y conectivos lógicos, puede ser verdadera o falsa. El objetivo de esta sección es justificar cómo el valor de verdad de una proposición está unívocamente determinado por el valor de verdad de sus variables proposicionales y por su estructura sintáctica.

Definición 2.1

El conjunto de *valores de verdad*, o *valores Booleanos*, se denota como $\mathbb{B}$ y está definido por $\mathbb{B} = \{\texttt{F}, \texttt{T}\}$.

El símbolo `F` representa el valor 'falso' (i.e., falsedad) y el símbolo `T` el valor 'verdadero' (i.e., veracidad).

Nota 2.2

Note que los valores Booleanos `F` y `T` son distintos a las constantes Booleanas *false* y *true*, respectivamente, las cuales son conectivos lógicos del lenguaje formal de la lógica proposicional.

Nota 2.3

El conjunto $\mathbb{B}$ recibe el nombre de Booleano en honor al matemático, lógico y filósofo inglés George Boole (1815-1864), quien fue el precursor de la lógica proposicional al escribir y publicar el libro "Las Leyes del Pensamiento" (en inglés *The Laws of Thought*) a finales del siglo XIX.

Dado que la *interpretación* de un conectivo lógico depende exclusivamente del valor de verdad de sus operandos, es natural asociar una *función* Booleana a cada uno de los conectivos lógicos.

Nota 2.4

Una *función* es una regla que asigna a cada elemento de un primer conjunto, llamado *dominio*, un único elemento de un segundo conjunto, llamado *rango*. Una función f con dominio A y rango B se denota como $f : A \to B$. La *composición* de las funciones $f : A \to B$ y $g : B \to C$ es la función $g \circ f : A \to C$ definida para cualquier $a \in A$ por $(g \circ f)(a) = g(f(a))$.

Una función representa una relación entre un conjunto de valores de entrada y un conjunto de valores de salida de forma tal que cada valor de entrada está relacionada con un *único* valor de salida.

Definición 2.5

Sea $n \in \mathbb{N}$. Una *función Booleana de n parámetros* (o *función Booleana n-aria*) es una función $H : \mathbb{B}^n \to \mathbb{B}$.

Dado que los conectivos lógicos del lenguaje formal de la lógica proposicional son constantes, unarios y binarios, hay un interés particular en funciones Booleanas de 0, 1 y 2 parámetros. En la Definición 2.5, este tipo de funciones corresponden, respectivamente, a los casos $n = 0, 1, 2$.

Nota 2.6

Una *tabla de valores* (o *tabla de verdad*) es una representación tabular de una función Booleana.

Como se verá más adelante, una tabla de verdad puede ser usada de una manera más general para representar cálculos con funciones Booleanas.

Ejemplo 2.1

Sea $H : \mathbb{B}^2 \to \mathbb{B}$ la función Booleana definida por

$$H(\mathtt{F},\mathtt{F}) = H(\mathtt{F},\mathtt{T}) = \mathtt{F} \qquad \text{y} \qquad H(\mathtt{T},\mathtt{F}) = H(\mathtt{T},\mathtt{T}) = \mathtt{T}.$$

La función H es una función Booleana binaria dado que $n = 2$. La siguiente tabla de verdad representa a H:

F	F	F
F	T	F
T	F	T
T	T	T

Las primeras dos columnas de la tabla sistemáticamente listan todos los elementos del conjunto $\mathbb{B}^2$ (i.e., los valores de entrada); la tercera columna muestra el valor de H para cada una de las parejas de las primeras dos columnas (i.e., los valores de salida).

El significado de cada uno de los operadores lógicos del lenguaje de la lógica proposicional se hace preciso asociando una función Booleana a cada uno de ellos.

Definición 2.7

La función H_{true} define la *interpretación de true* y la función H_{false} define la *interpretación de false*:

$$H_{true}() = \mathtt{T}$$
$$H_{false}() = \mathtt{F}$$

Dado que *true* y *false* son constantes en el lenguaje de la lógica proposicional, es natural que las funciones que definen sus interpretaciones sean funciones Booleanas constantes. La interpretación de *true* es el valor Booleano `T` de verdad, mientras que la interpretación de *false* es el valor Booleano `F` de falsedad.

Definición 2.8

La función $H_\neg$ define la *interpretación de la negación*:

p	$(\neg p)$
F	T
T	F

$$H_\neg(\mathtt{F}) = \mathtt{T}$$
$$H_\neg(\mathtt{T}) = \mathtt{F}$$

La fórmula $(\neg p)$ es verdadera cuando p es falsa y es falsa cuando p es verdadera. Las etiquetas en la parte superior de las columnas en la tabla de verdad de la negación

(Definición 2.8) no son oficialmente parte de la tabla de verdad: en general, los encabezados se usan con el propósito de enfatizar o resaltar el efecto que tiene una función Booleana sobre *cualquier* variable proposicional (no solamente sobre p).

Definición 2.9

La función $H_{\equiv}$ define la *interpretación de la equivalencia*:

p	q	$(p \equiv q)$
F	F	T
F	T	F
T	F	F
T	T	T

$$H_{\equiv}(\mathtt{F},\mathtt{F}) = H_{\equiv}(\mathtt{T},\mathtt{T}) = \mathtt{T}$$
$$H_{\equiv}(\mathtt{F},\mathtt{T}) = H_{\equiv}(\mathtt{T},\mathtt{F}) = \mathtt{F}$$

Una equivalencia $(p \equiv q)$ es verdadera cuando p y q tienen el mismo valor de verdad y es falsa cuando los valores de verdad son distintos (i.e., opuestos).

Definición 2.10

La función $H_{\not\equiv}$ define la *interpretación de la discrepancia*:

p	q	$(p \not\equiv q)$
F	F	F
F	T	T
T	F	T
T	T	F

$$H_{\not\equiv}(\mathtt{F},\mathtt{F}) = H_{\not\equiv}(\mathtt{T},\mathtt{T}) = \mathtt{F}$$
$$H_{\not\equiv}(\mathtt{F},\mathtt{T}) = H_{\not\equiv}(\mathtt{T},\mathtt{F}) = \mathtt{T}$$

Una discrepancia $(p \not\equiv q)$ es verdadera cuando los valores de verdad de p y q discrepan y es falsa cuando los valores de verdad son iguales.

Definición 2.11

La función $H_{\vee}$ define la *interpretación de la disyunción*:

p	q	$(p \vee q)$
F	F	F
F	T	T
T	F	T
T	T	T

$$H_{\vee}(\mathtt{F},\mathtt{F}) = \mathtt{F}$$
$$H_{\vee}(\mathtt{F},\mathtt{T}) = H_{\vee}(\mathtt{T},\mathtt{F}) = H_{\vee}(\mathtt{T},\mathtt{T}) = \mathtt{T}$$

Una disyunción $(p \vee q)$ es falsa únicamente cuando los valores de verdad de p y q son ambos falsos; de cualquier otra forma $(p \vee q)$ es verdadera.

Definición 2.12

La función $H_\wedge$ define la *interpretación de la conjunción*:

p	q	$(p \wedge q)$
F	F	F
F	T	F
T	F	F
T	T	T

$$H_\wedge(\mathtt{F},\mathtt{F}) = H_\wedge(\mathtt{F},\mathtt{T}) = H_\wedge(\mathtt{T},\mathtt{F}) = \mathtt{F}$$
$$H_\wedge(\mathtt{T},\mathtt{T}) = \mathtt{T}$$

Una conjunción $(p \wedge q)$ es verdadera únicamente cuando los valores de verdad de p y q son ambos verdaderos; de cualquier otra forma $(p \wedge q)$ es falsa.

Definición 2.13

La función $H_\to$ define la *interpretación de la implicación*:

p	q	$(p \to q)$
F	F	T
F	T	T
T	F	F
T	T	T

$$H_\to(\mathtt{F},\mathtt{F}) = H_\to(\mathtt{F},\mathtt{T}) = H_\to(\mathtt{T},\mathtt{T}) = \mathtt{T}$$
$$H_\to(\mathtt{T},\mathtt{F}) = \mathtt{F}$$

Una implicación $(p \to q)$ es falsa únicamente cuando el valor del antecedente p es verdadero y el valor del consecuente q es falso (renglón 3 de la tabla de verdad en la Definición 2.13). De esta forma, la interpretación de la implicación usada en este texto sigue la tradición del uso de la implicación en las matemáticas: una expresión de la forma 'si ϕ, entonces ψ' es verdadera *excepto* cuando el antecedente ϕ es verdadero y el consecuente ψ es falso.

Nota 2.14

Una expresión de la forma 'si ϕ, entonces ψ' puede ser analizada de la siguiente manera:

1. Suponiendo que la proposición ϕ es verdadera.
2. Procediendo a demostrar que la proposición ψ es *necesariamente* verdadera.

Una demostración de este estilo se llama 'demostración por suposición del antecedente' y su objetivo es mostrar que si el antecedente en una implicación es verdadero, es el último renglón (i.e., cuarto renglón) de la tabla de verdad de la implicación el que aplica y no el tercer renglón (Definición 2.13). Note que se puede ignorar convenientemente el caso en que el antecedente ϕ es falso (primero y segundo renglón de la tabla de verdad de la implicación) porque, de ser así, la implicación es verdadera sin importar el valor de verdad del consecuente.

Definición 2.15

La función $H_{\leftarrow}$ define la *interpretación de la consecuencia*:

p	q	$(p \leftarrow q)$
F	F	T
F	T	F
T	F	T
T	T	T

$$H_{\leftarrow}(\mathtt{F},\mathtt{F}) = H_{\leftarrow}(\mathtt{T},\mathtt{F}) = H_{\leftarrow}(\mathtt{T},\mathtt{T}) = \mathtt{T}$$
$$H_{\leftarrow}(\mathtt{F},\mathtt{T}) = \mathtt{F}$$

Una consecuencia $(p \leftarrow q)$ es falsa únicamente cuando el valor del antecedente q es verdadero y el valor del consecuente p es falso (segundo renglón de la tabla de verdad en la Definición 2.15). En este sentido, es conveniente pensar que la consecuencia es una implicación escrita de derecha a izquierda.

Un uso frecuente de las funciones Booleanas y de las tablas de verdad es en el análisis de proposiciones. La pregunta es: ¿cómo hacerlo? Lo realmente interesante de la forma como se ha definido la semántica de los conectivos lógicos usando funciones Booleanas, es que dicha semántica es inherentemente *composicional* y puede ser *mecanizable*. Es composicional porque, por ejemplo, para conocer el valor de verdad de una equivalencia $(\phi \equiv \psi)$ conociéndo los valores de verdad de cada uno de los renglones de ϕ y ψ, basta con usar la función $H_{\equiv}$ para obtener el valor de verdad de cada uno de los renglones de $(\phi \equiv \psi)$. Es mecanizable porque, por ejemplo, recorriendo el árbol de sintaxis de $(\phi \equiv \psi)$ desde las hojas (que corresponden a variables proposicionales y a las constantes Booleanas) hacia la raíz, se calculan incrementalmente los valores de ϕ y ψ para, finalmente, aplicar la función $H_{\equiv}$ con los valores obtenidos. Este análisis arrojará valores únicos para cada combinación de valores de verdad de las variables proposicionales (¿por qué?).

En el Ejemplo 2.2, observe que las primeras dos columnas sistemáticamente listan todas las posibles combinaciones de los valores de verdad de p y q; la última

columna muestra el valor de verdad de $((p \to q) \equiv ((\neg p) \vee q))$ para cada una de estas combinaciones. Las columnas 3, 4 y 5 muestran cálculos intermedios que dependen de las funciones Booleanas definidas en esta sección. Por ejemplo, la columna tres, cuyo encabezado es $(p \to q)$, se obtiene usando la función Booleana $H_{\to}$ con parámetros en las columnas 1 y 2.

Ejemplo 2.2

La proposición

$$((p \to q) \equiv ((\neg p) \vee q))$$

resulta en `T` para cualquier combinación de valores de verdad de p y q. La tabla de verdad que aparece a continuación se usa para justificar este hecho:

p	q	$(p \to q)$	$(\neg p)$	$((\neg p) \vee q)$	$((p \to q) \equiv ((\neg p) \vee q))$
F	F	T	T	T	T
F	T	T	T	T	T
T	F	F	F	F	T
T	T	T	F	T	T

En general, para una proposición ϕ que menciona variables proposicionales en la lista $p_0, p_1, \ldots, p_{n-1}$, es posible analizar el valor de verdad de ϕ usando las funciones Booleanas $H_{true}, H_{false}, H_{\neg}, H_{\equiv}, H_{\not\equiv}, H_{\vee}, H_{\wedge}, H_{\to}, H_{\leftarrow}$ al asignar valores de verdad a cada una de estas variables proposicionales. La tabla de verdad de una proposición como ϕ, que menciona n variables proposicionales, tiene exactamente 2^n renglones, uno para cada combinación de $p_0, p_1, \ldots, p_{n-1}$. En la práctica, esto limita la efectividad del uso de las tablas de verdad porque para valores de n relativamente grandes, dibujar la tabla de verdad es humanamente imposible. Por ejemplo, para diez variables proposicionales (i.e., $n = 10$), la tabla de verdad tiene $2^{10} = 1024$ renglones. Por estas limitaciones, las tablas de verdad son útiles para valores de n relativamente pequeños.

Ejercicios

1. Liste todas las funciones Booleanas unarias.
2. Determine el número de funciones Booleanas binarias y describa dos que sean diferentes a $H_{\equiv}$, $H_{\not\equiv}$, $H_{\vee}$, $H_{\wedge}$, $H_{\to}$ y $H_{\leftarrow}$.
3. Demuestre que $H_{\not\equiv} = H_{\neg} \circ H_{\equiv}$.
4. Dibuje la tabla de verdad para cada una de las siguientes proposiciones:
   a) $(\mathit{true} \not\equiv \mathit{false})$

b) $(p \equiv (\neg q))$
c) $(p \vee (\neg q))$
d) $(p \wedge (\neg q))$
e) $(\mathit{false} \to p)$
f) $((p \to q) \equiv (q \leftarrow p))$
g) $(\neg(q \wedge (\neg p)))$
h) $(\neg(\mathit{true} \wedge \mathit{false}))$
i) $((p \equiv q) \equiv (q \equiv p))$
j) $((p \wedge q) \to p)$
k) $(p \to (p \vee q))$
l) $((p \wedge (p \equiv q)) \equiv (p \wedge q))$
m) $((p \equiv (q \equiv r)) \equiv ((p \equiv q) \equiv r))$
n) $((p \vee (q \vee r)) \equiv ((p \vee q) \vee r))$
ñ) $((p \wedge (q \wedge r)) \equiv ((p \wedge q) \wedge r))$

5. ¿Tienen las proposiciones $(p \wedge q)$ y $((p \vee q) \equiv (p \equiv q))$ el mismo significado? Justifique su respuesta.

6. Justifique que la implicación no es asociativa, es decir, que las proposiciones $(p \to (q \to r))$ y $((p \to q) \to r)$ no tienen el mismo significado.

7. Dibuje la tabla de verdad de la proposición que se encuentra en la Figura 1 (Sección 1.3).

8. Una proposición se llama *tautología* si y solo si todos los renglones de su tabla de verdad resultan en `T` y se llama *contradicción* si y solo si todos los renglones resultan en `F`. Proponga una proposición que no sea tautología ni contradicción y dibuje su tabla de verdad.

9. Una relación binaria sobre $\mathbb{B}$ es un subconjunto del conjunto

   $$\mathbb{B}^2 = \{(\mathtt{T},\mathtt{T}), (\mathtt{T},\mathtt{F}), (\mathtt{F},\mathtt{T}), (\mathtt{F},\mathtt{F})\}.$$

   A un conectivo lógico binario $\otimes$ de la lógica proposicional puede asociarse una relación binaria $R_\otimes$ de la siguiente manera:

   $$R_\otimes = \{(x, y) \in \mathbb{B}^2 \mid H_\otimes(x, y) = \mathtt{T}\}.$$

   En otras palabras, $R_\otimes$ es el conjunto de parejas en $\mathbb{B}^2$ que hacen verdaero a $\otimes$.
   a) Escriba las relaciones $R_\equiv$, $R_{\not\equiv}$, $R_\vee$, $R_\wedge$, $R_\to$ y $R_\leftarrow$.
   b) Investigue y escriba definiciones que permitan identificar cuándo una relación binaria es:
   - asociativa
   - conmutativa
   - reflexiva
   - irreflexiva
   - asimétrica
   - antisimétrica

- idempotente
- transitiva

c) Clasifique las relaciones $R_{\equiv}$, $R_{\not\equiv}$, $R_{\vee}$, $R_{\wedge}$, $R_{\rightarrow}$ y $R_{\leftarrow}$ de acuerdo con la lista de propiedades del numeral anterior.

10. Considere un conectivo lógico binario $\star$ cuya interpretación está dada por la función Booleana $H_\star : \mathbb{B}^2 \to \mathbb{B}$ definida de la siguiente manera:

$$H_\star(\mathtt{F},\mathtt{F}) = \mathtt{T} \qquad \text{y} \qquad H_\star(\mathtt{F},\mathtt{T}) = H_\star(\mathtt{T},\mathtt{F}) = H_\star(\mathtt{T},\mathtt{T}) = \mathtt{F}.$$

a) Proponga una proposición que defina $\star$ en términos de los conectivos lógicos $\{\mathit{true}, \mathit{false}, \neg, \equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$.

b) Encuentre una proposición que únicamente mencione la variable proposicional p y el conectivo $\star$, y que tenga la misma tabla de verdad de $(\neg p)$.

c) Encuentre una proposición que únicamente mencione las variables proposicionales p, q y el conectivo $\star$, y que tenga la misma tabla de verdad de $(p \wedge q)$.

d) Justifique o refute:

1) $\star$ es asociativo
2) $\star$ es conmutativo
3) $\star$ es reflexivo
4) $\star$ es irreflexivo
5) $\star$ es asimétrico
6) $\star$ es antisimétrico
7) $\star$ es idempotente
8) $\star$ es transitivo

11. Considere un conectivo lógico binario $\oplus$ cuya interpretación está dada por la función Booleana $H_\oplus : \mathbb{B}^2 \to \mathbb{B}$ definida de la siguiente manera:

$$H_\oplus(\mathtt{F},\mathtt{F}) = H_\oplus(\mathtt{F},\mathtt{T}) = H_\oplus(\mathtt{T},\mathtt{F}) = \mathtt{T} \qquad \text{y} \qquad H_\oplus(\mathtt{T},\mathtt{T}) = \mathtt{F}.$$

a) Proponga una proposición que defina $\oplus$ en términos de los conectivos lógicos $\{\mathit{true}, \mathit{false}, \neg, \equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$.

b) Encuentre una proposición que únicamente mencione la variable proposicional p y el conectivo $\oplus$, y que tenga la misma tabla de verdad de $(\neg p)$.

c) Encuentre una proposición que únicamente mencione las variables proposicionales p, q y el conectivo $\oplus$, y que tenga la misma tabla de verdad de $(p \vee q)$.

d) Justifique o refute:

1) $\oplus$ es asociativo
2) $\oplus$ es conmutativo
3) $\oplus$ es reflexivo
4) $\oplus$ es irreflexivo
5) $\oplus$ es asimétrico

6) $\oplus$ es antisimétrico
7) $\oplus$ es idempotente
8) $\oplus$ es transitivo

12. Considere un conectivo lógico binario $\odot$ cuya interpretación está dada por la función Booleana $H_{\odot} : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ definida de la siguiente manera:

$$H_{\odot}(\mathtt{F},\mathtt{F}) = H_{\odot}(\mathtt{T},\mathtt{F}) = H_{\odot}(\mathtt{T},\mathtt{T}) = \mathtt{F} \qquad \text{y} \qquad H_{\odot}(\mathtt{F},\mathtt{T}) = \mathtt{T}.$$

a) Proponga una proposición que defina $\odot$ en términos de los conectivos lógicos $\{true, false, \neg, \equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$.
b) Justifique o refute:
1) $\odot$ es asociativo
2) $\odot$ es conmutativo
3) $\odot$ es reflexivo
4) $\odot$ es irreflexivo
5) $\odot$ es asimétrico
6) $\odot$ es antisimétrico
7) $\odot$ es idempotente
8) $\odot$ es transitivo

13. Considere cuatro cartas que muestran la siguiente información:

Cada carta tiene una letra en un lado y un número en el otro. Especifique qué cartas deben ser destapadas para establecer el valor de verdad de la siguiente afirmación: "si una carta tiene una vocal en uno de sus lados, entonces esa carta tiene un número par en su lado opuesto".

14. Suponga que Juana quiere ir de compras con sus amigas. Para poder ir de compras, ella debe realizar al menos una de las siguientes tareas hogareñas (las demás las hará su hermano):
Opción 1: podar el césped (p)
Opción 2: lavar y secar los platos (l) y doblar las toallas de la cocina (t)
Opción 3: limpiar el polvo (d)
Opción 4: fregar los pisos (f)
Opción 5: hacer mercado (h) y recoger la ropa de la lavandería (r)
a) Especifique cada una de las opciones de Juana usando los símbolos indicados en cada proposición.
b) Suponga que Juana se rehusa a limpiar el polvo, fregar el piso y podar el césped. Juana también está cansada de lavar y secar los platos, pero ha decidido doblar las toallas de la cocina. Ella también decide hacer mercado pero olvida recoger la ropa de la lavandería. ¿Puede ir Juana de compras con sus amigas? Justifique su respuesta.

15. Esta historia es acerca de una caravana que atraviesa el desierto del Sahara. Una noche, para dormir, levantaron carpas. Los principales protagonistas de la historia son a, b y c. La persona a no soporta a c y por ello decide matarlo vertiendo veneno en la cantimplora de c. De manera independiente, b decide matar a c y entonces, sin saber que a ha vertido veneno en la cantimplora de c, b perfora un hueco en la cantimplora de c con el objetivo de que pierda su bebida. Como resultado de estas acciones, c muere un par de días después a causa de la sed. La pregunta es ¿quién es el asesino, a o b?

    De acuerdo con una explicación, b es el asesino dado que c nunca tomó de la bebida envenenada: de esta forma, c habría muerto así a no hubiera envenenado su bebida. De acuerdo con una explicación contraria, a es el asesino porque las acciones de b no tienen un efecto directo en el resultado de la situación: una vez a envenenó la bebida, el destino de c era claro aún si b no hubiera perforado el agujero en la cantimplora. ¿Cuál de las dos explicaciones es correcta?

16. Demuestre, usando el principio de inducción matemática para $n \in \mathbb{N}$, que hay 2^{2^n} funciones Booleanas n-arias.

2.2. Valuaciones

Las valuaciones ofrecen un mecanismo complementario al de las funciones Booleanas (Sección 2.1) para analizar el significado de las proposiciones. Una valuación es una función que asocia un valor de verdad a una proposición dependiendo del valor de verdad de cada una de las variables proposicionales que la conforman. Formalmente, una valuación es una función sobre las variables proposicionales que se extiende unívocamente a proposiciones con ayuda de las funciones Booleanas. Es así como una valuación es una herramienta que simplifica el uso de las funciones Booleanas para estudiar el significado de las proposiciones. En algunas ocasiones, y a diferencia de las funciones Booleanas, las valuaciones permiten además determinar el significado de una proposición sin conocer completamente su estructura sintáctica.

Definición 2.16

Una *valuación* es una función que asigna valores en $\mathbb{B}$ a cada una de las variables proposicionales. Formalmente, una valuación es una función

$$\mathbf{v} : \mathcal{V} \to \mathbb{B}$$

con dominio en las variables proposicionales $\{p_0, p_1, \ldots\}$ y rango en los valores Booleanos $\{\mathtt{F}, \mathtt{T}\}$.

Una valuación asigna valores de verdad a *cada* variable proposicional.

Nota 2.17

Se usarán letras minúsculas en negrilla para denotar valuaciones.

El siguiente paso es asociar un valor de verdad a una proposición con base en una valuación. Es decir, el objetivo es extender una valuación que asocia valores de verdad a las variables proposicionales a una función que asocia valores de verdad a cualquier proposición. Esta extensión se formula inductivamente sobre la complejidad de las proposiciones con ayuda de las funciones Booleanas en la Definición 2.18.

Definición 2.18

Sea $\mathbf{v}$ una valuación. La *extensión de* $\mathbf{v}$, denotada como $\overline{\mathbf{v}}$, se define inductivamente para toda proposición de la siguiente forma:

1. $\overline{\mathbf{v}}(p) = \mathbf{v}(p)$
2. $\overline{\mathbf{v}}(\mathit{true}) = H_{\mathit{true}}()$
3. $\overline{\mathbf{v}}(\mathit{false}) = H_{\mathit{false}}()$
4. $\overline{\mathbf{v}}((\neg\psi)) = H_{\neg}(\overline{\mathbf{v}}(\psi))$
5. $\overline{\mathbf{v}}((\psi \equiv \tau)) = H_{\equiv}(\overline{\mathbf{v}}(\psi), \overline{\mathbf{v}}(\tau))$
6. $\overline{\mathbf{v}}((\psi \not\equiv \tau)) = H_{\not\equiv}(\overline{\mathbf{v}}(\psi), \overline{\mathbf{v}}(\tau))$
7. $\overline{\mathbf{v}}((\psi \vee \tau)) = H_{\vee}(\overline{\mathbf{v}}(\psi), \overline{\mathbf{v}}(\tau))$
8. $\overline{\mathbf{v}}((\psi \wedge \tau)) = H_{\wedge}(\overline{\mathbf{v}}(\psi), \overline{\mathbf{v}}(\tau))$
9. $\overline{\mathbf{v}}((\psi \rightarrow \tau)) = H_{\rightarrow}(\overline{\mathbf{v}}(\psi), \overline{\mathbf{v}}(\tau))$
10. $\overline{\mathbf{v}}((\psi \leftarrow \tau)) = H_{\leftarrow}(\overline{\mathbf{v}}(\psi), \overline{\mathbf{v}}(\tau))$

En la Definición 2.18 se establece que una valuación y su extensión coinciden en las variables proposicionales (caso (1)). Los demás casos en esta definición corresponden a formulaciones inductivas que se obtienen a partir del significado dado por las funciones Booleanas a cada uno de los conectivos lógicos.

Ejemplo 2.3

Considere una valuación $\mathbf{v}$ tal que $\mathbf{v}(p) = \mathtt{T}$ y $\mathbf{v}(q) = \mathtt{F}$. De acuerdo con la Definición 2.18, se tiene que:

$$\begin{aligned}
\overline{\mathbf{v}}(((\neg p) \to q)) &= H_{\to}(\overline{\mathbf{v}}((\neg p)), \overline{\mathbf{v}}(q)) \\
&= H_{\to}(\overline{\mathbf{v}}((\neg p)), \mathtt{F}) && (\overline{\mathbf{v}}(q) = \mathbf{v}(q) = \mathtt{F}) \\
&= H_{\to}(H_{\neg}(\overline{\mathbf{v}}(p)), \mathtt{F}) \\
&= H_{\to}(H_{\neg}(\mathtt{T}), \mathtt{F}) && (\overline{\mathbf{v}}(p) = \mathbf{v}(p) = \mathtt{T}) \\
&= H_{\to}(\mathtt{F}, \mathtt{F}) \\
&= \mathtt{T}.
\end{aligned}$$

El Metateorema 2.19, que se presenta a continuación, justifica la afirmación hecha inicialmente en esta sección acerca de que una valuación (en realidad, su extensión), al igual que las funciones Booleanas, permite asociar un valor de verdad a una proposición.

Metateorema 2.19

Si $\mathbf{v}$ es una valuación, entonces $\overline{\mathbf{v}}$ es una función con dominio en el conjunto de proposiciones $\mathcal{T}(\mathcal{V})$ y rango en los Booleanos $\mathbb{B}$ (i.e., $\overline{\mathbf{v}} : \mathcal{T}(\mathcal{V}) \to \mathbb{B}$).

Demostración. Por la Definición 2.18, es directo que $\overline{\mathbf{v}}$ asigna al menos un valor de verdad en $\mathbb{B}$ a cualquier proposición. Basta entonces con demostrar que $\overline{\mathbf{v}}$ asigna un único valor a cada proposición. Considere la siguiente propiedad F para cualquier proposición γ:

$F(\gamma)$: "$\overline{\mathbf{v}}$ asigna un único valor a γ".

La demostración de que toda proposición ϕ tiene la propiedad F se sigue del principio de inducción sobre proposiciones (Metateorema 1.7):

Caso base: Si ϕ es una variable proposicional, por ejemplo p, entonces $\overline{\mathbf{v}}(p) = \mathbf{v}(p)$ y la propiedad F vale porque $\mathbf{v}$ es una función. Por su parte, si ϕ es una constante, entonces $\overline{\mathbf{v}}(\mathit{true}) = H_{\mathit{true}}()$ y $\overline{\mathbf{v}}(\mathit{false}) = H_{\mathit{false}}()$ y la propiedad F vale porque $H_{\mathit{true}}()$ y $H_{\mathit{false}}()$ son funciones.

Caso inductivo: Hay 7 casos: que ϕ sea de la forma $(\neg\psi)$, $(\psi \equiv \tau)$, $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \to \tau)$ o $(\psi \leftarrow \tau)$. Si ϕ es de la forma $(\neg\psi)$ y ψ tiene la propiedad F, entonces $\overline{\mathbf{v}}((\neg\psi)) = H_{\neg}(\overline{\mathbf{v}}(\psi))$ tiene la propiedad F porque $H_{\neg}$ es una función y $\overline{\mathbf{v}}$ asigna un único valor a ψ (hipotesis inductiva). Los casos

en que ϕ es de la forma $(\psi \equiv \tau)$, $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \to \tau)$ o $(\psi \leftarrow \tau)$ son similares y se proponen como ejercicios para el lector.

En cualquiera de los dos casos $\overline{\mathbf{v}}$ asigna un único valor a ϕ. En conclusión, $\overline{\mathbf{v}}$ es una función con dominio en el conjunto de proposiciones y rango $\mathbb{B}$. □

Nota 2.20

El Metateorema 2.19 aporta un resultado poderoso para el análisis semántico de las proposiciones: si $\mathbf{v}$ es una valuación y ϕ una proposición, entonces $\overline{\mathbf{v}}(\phi) = \mathtt{T}$ o $\overline{\mathbf{v}}(\phi) = \mathtt{F}$ (pero no los dos). Como se verá al final de esta sección, este resultado es clave para estudiar el significado de una proposición sin conocer el detalle de su estructura sintáctica.

A pesar de que las valuaciones y las funciones Booleanas son herramientas similares para el análisis del significado de las proposiciones, hay una diferencia estructural entre ellas. Suponga por un momento que ϕ es una proposición en donde aparecen las variables $p_0, \ldots, p_{n-1}$. Si se dibuja una tabla de verdad para ϕ, entonces se asignan valores de verdad a las variables de esta lista, ignorando cualquier otra variable. Sin embargo, al razonar con valuaciones que asignan valores de verdad a cualquier variable proposicional, hay más de una valuación que corresponde a un mismo renglón de la tabla de verdad de ϕ. Entonces surge la pregunta: ¿cómo escoger 'la' valuación indicada de modo tal que los valores asignados a variables no mencionadas en ϕ no afecten el valor de verdad de ϕ? La respuesta, como se enuncia a continuación, es que el valor de verdad de una proposición bajo una valuación no se ve afectado por variables proposicionales que no aparecen en ella. Es decir, para determinar el valor de verdad de ϕ con respecto a una valuación, es suficiente con considerar únicamente el valor de verdad asignado por dicha valuación a las variables proposicionales en ϕ.

Metateorema 2.21

Sea ϕ una proposición y $\mathbf{v}, \mathbf{w}$ valuaciones. Si $\mathbf{v}$ y $\mathbf{w}$ coinciden en todas las variables proposicionales que aparecen en ϕ, entonces

$$\overline{\mathbf{v}}(\phi) = \overline{\mathbf{w}}(\phi).$$

Demostración. Considere la siguiente propiedad M para cualquier proposición γ:

$$M(\gamma) : \overline{\mathbf{v}}(\gamma) = \overline{\mathbf{w}}(\gamma).$$

El objetivo es demostrar que si $\mathbf{v}$ y $\mathbf{w}$ coinciden en todas las variables proposicionales que aparecen en ϕ, entonces ϕ tiene la propiedad M. Suponga que $\mathbf{v}$ y $\mathbf{w}$ coinciden en todas las variables proposicionales que aparecen en ϕ. La demostración de que ϕ tiene la propiedad M se sigue del principio de inducción sobre proposiciones (Metateorema 1.7):

Caso base: Si ϕ es una variable proposicional, por ejemplo p, entonces

$$\overline{\mathbf{v}}(p) = \mathbf{v}(p) = \mathbf{w}(p) = \overline{\mathbf{w}}(p),$$

dado que $\mathbf{v}$ y $\mathbf{w}$ coinciden en cualquier variable proposicional en ϕ.

Por su parte, si ϕ es una constante, entonces:

$$\overline{\mathbf{v}}(\mathit{true}) = H_{\mathit{true}}() = \overline{\mathbf{w}}(\mathit{true}),$$
$$\overline{\mathbf{v}}(\mathit{false}) = H_{\mathit{false}}() = \overline{\mathbf{w}}(\mathit{false}).$$

Caso inductivo: Hay 7 casos: que ϕ sea de la forma $(\neg\psi)$, $(\psi \equiv \tau)$, $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \to \tau)$ o $(\psi \leftarrow \tau)$. Si ϕ es de la forma $(\neg\psi)$ y ψ tiene la propiedad M, entonces:

$$\begin{aligned}\overline{\mathbf{v}}((\neg\psi)) &= H_{\neg}(\overline{\mathbf{v}}(\psi)) \\ &= H_{\neg}(\overline{\mathbf{w}}(\psi)) && \text{(hipótesis inductiva)} \\ &= \overline{\mathbf{w}}((\neg\psi)).\end{aligned}$$

Si ϕ es de la forma $(\psi \equiv \tau)$ y tanto ψ como τ tienen la propiedad M, entonces:

$$\begin{aligned}\overline{\mathbf{v}}((\psi \equiv \tau)) &= H_{\equiv}(\overline{\mathbf{v}}(\psi), \overline{\mathbf{v}}(\tau)) \\ &= H_{\equiv}(\overline{\mathbf{w}}(\psi), \overline{\mathbf{w}}(\tau)) && \text{(hipótesis inductiva dos veces)} \\ &= \overline{\mathbf{w}}((\psi \equiv \tau)).\end{aligned}$$

Los casos en que ϕ es de la forma $(\psi \not\equiv \tau)$, $(\psi \vee \tau)$, $(\psi \wedge \tau)$, $(\psi \to \tau)$ o $(\psi \leftarrow \tau)$ son similares y se proponen como ejercicios para el lector.

En cualquiera de los tres casos $\overline{\mathbf{v}}(\phi) = \overline{\mathbf{w}}(\phi)$, como se desea. □

Nota 2.22

Para simplificar la escritura de la extensión $\overline{\mathbf{v}}$ de una valuación $\mathbf{v}$, se adopta la convención de referirse a dicha extensión como una valuación y denotarla como $\mathbf{v}$. Es decir, intencionalmente se evita distinguir entre una valuación y su extensión.

Además, dada una lista $p_0, p_1, \ldots, p_{n-1}$ de variables proposicionales, se usará la expresión

$$\{p_0 \mapsto \mathit{val}_0, p_1 \mapsto \mathit{val}_1, \ldots, p_{n-1} \mapsto \mathit{val}_{n-1}, \ldots\}$$

para denotar una valuación que asigna los valores Booleanos val_0 a p_0, val_1 a p_1, $\cdots$ y val_{n-1} a p_{n-1}. Con esta notación se está ignorando intencionalmente el valor asignado a aquellas variables proposicionales que no aparecen en la lista dada.

Las valuaciones pueden ser usadas para simplificar (aún más) razonamientos y cálculos observando las propiedades de los conectivos lógicos. En algunas ocasiones, estas observaciones bastan para analizar el significado de las proposiciones sin conocer su estructura sintactica. El Metateorema 2.23 incluye algunas de estas observaciones.

Metateorema 2.23

Sean ϕ y ψ proposiciones, y $\mathbf{v}$ una valuación de ϕ y ψ. Entonces:

1. $\mathbf{v}((\neg\phi)) = \mathtt{F}$ si y solo si $\mathbf{v}(\phi) = \mathtt{T}$; $\mathbf{v}((\neg\phi)) = \mathtt{T}$ si y solo si $\mathbf{v}(\phi) = \mathtt{F}$.
2. $\mathbf{v}((\phi \equiv \psi)) = \mathtt{T}$ si y solo si $\mathbf{v}(\phi) = \mathbf{v}(\psi)$; de lo contrario $\mathbf{v}((\phi \equiv \psi)) = \mathtt{F}$.
3. $\mathbf{v}((\phi \not\equiv \psi)) = \mathtt{T}$ si y solo si $\mathbf{v}(\phi) \neq \mathbf{v}(\psi)$; de lo contrario $\mathbf{v}((\phi \not\equiv \psi)) = \mathtt{F}$.
4. $\mathbf{v}((\phi \vee \psi)) = \mathtt{F}$ si y solo si $\mathbf{v}(\phi) = \mathbf{v}(\psi) = \mathtt{F}$; de lo contrario $\mathbf{v}((\phi \vee \psi)) = \mathtt{T}$.
5. $\mathbf{v}((\phi \wedge \psi)) = \mathtt{T}$ si y solo si $\mathbf{v}(\phi) = \mathbf{v}(\psi) = \mathtt{T}$; de lo contrario $\mathbf{v}((\phi \wedge \psi)) = \mathtt{F}$.
6. $\mathbf{v}((\phi \to \psi)) = \mathtt{F}$ si y solo si $\mathbf{v}(\phi) = \mathtt{T}$ y $\mathbf{v}(\psi) = \mathtt{F}$; de lo contrario $\mathbf{v}((\phi \to \psi)) = \mathtt{T}$.
7. $\mathbf{v}((\phi \leftarrow \psi)) = \mathtt{F}$ si y solo si $\mathbf{v}(\phi) = \mathtt{F}$ y $\mathbf{v}(\psi) = \mathtt{T}$; de lo contrario $\mathbf{v}((\phi \leftarrow \psi)) = \mathtt{T}$.

Demostración. Para el caso (1) note que $\mathbf{v}((\neg\phi)) = H_{\neg}(\mathbf{v}(\phi))$. Si $\mathbf{v}(\phi) = \mathtt{T}$, entonces $\mathbf{v}((\neg\phi)) = \mathtt{F}$; si $\mathbf{v}(\phi) = \mathtt{F}$, entonces $\mathbf{v}((\neg\phi)) = \mathtt{T}$. En cualquiera de los dos casos se tiene que $\mathbf{v}(\phi) \neq \mathbf{v}((\neg\phi))$. Para el caso (2) note que $\mathbf{v}((\phi \equiv \psi)) = H_{\equiv}(\mathbf{v}(\phi), \mathbf{v}(\psi))$. Si $\mathbf{v}(\phi) = \mathbf{v}(\psi)$, entonces $H_{\equiv}(\mathbf{v}(\phi), \mathbf{v}(\psi)) = \mathtt{T}$ y consecuentemente $\mathbf{v}((\phi \equiv \psi)) = \mathtt{T}$. De lo contrario $\mathbf{v}(\phi) \neq \mathbf{v}(\psi)$, obteniendo $H_{\equiv}(\mathbf{v}(\phi), \mathbf{v}(\psi)) = \mathtt{F}$ y consecuentemente $\mathbf{v}((\phi \equiv \psi)) = \mathtt{F}$. Los casos (3)-(7) son similares y se proponen como ejercicios para el lector. □

El Metateorema 2.23 indica que el símbolo $\neg$ se interpreta como un *no*, el símbolo $\equiv$ se interpreta como un *igual*, el símbolo $\not\equiv$ se interpreta como un *diferente*, el símbolo $\vee$ se interpreta como un *o*, el símbolo $\wedge$ se interpreta como un *y*, el símbolo $\to$ se interpreta como un *condicional* y el símbolo $\leftarrow$ se interpreta como una *consecuencia*.

Algunas expresiones que aparecen en el Metateorema 2.23 usan el castellano y no exclusivamente símbolos del sistema formal proposicional. La expresión "si

y solo si" que aparece en el Metateorema 2.23 es una metaequivalencia: tiene dos operandos y es verdad únicamente cuando los dos operandos coinciden en sus valores de verdad. Note que su interpreación es similar a la de la equivalencia lógica. Sin embargo, no es el mismo operador: el "si y solo si" tiene como operandos expresiones que no hacen parte del lenguaje de la lógica proposicional.

Nota 2.24

El operador "si y solo si" en algunas ocasiones se abrevia como "sii".

A continuación se presentan ejemplos que ilustran el uso del Metateorema 2.23. En el Ejemplo 2.4 se analiza la semántica de una proposición de la cual se conoce en detalle su estructura sintáctica. En el Ejemplo 2.5 se hace el mismo ejercicio con una proposición que tiene la misma estructura sintáctica, pero de manera más general (i.e., desconociendo algunos detalles de su sintaxis). En este sentido, el resultado presentado en el Ejemplo 2.5 subsume aquel presentado en el Ejemplo 2.4.

Ejemplo 2.4

Se demostrará que $\mathbf{v}(((p \to q) \equiv ((\neg p) \vee q))) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$ (el objetivo es el mismo que en el Ejemplo 2.2). Por el Metateorema 2.23 se tiene:

$$\begin{aligned} \mathbf{v}((p \to q)) = \mathtt{F} \text{ sii } & \mathbf{v}(p) = \mathtt{T} \text{ y } \mathbf{v}(q) = \mathtt{F} && \text{(caso } \to\text{)} \\ \text{sii } & \mathbf{v}((\neg p)) = \mathtt{F} \text{ y } \mathbf{v}(q) = \mathtt{F} && \text{(caso } \neg\text{)} \\ \text{sii } & \mathbf{v}(((\neg p) \vee q)) = \mathtt{F} && \text{(caso } \vee\text{).} \end{aligned}$$

De estos cálculos se concluye que $\mathbf{v}((p \to q)) = \mathbf{v}(((\neg p) \vee q))$ y, por el mismo Metateorema 2.23, se obtiene $\mathbf{v}(((p \to q) \equiv ((\neg p) \vee q))) = \mathtt{T}$ (caso $\equiv$).

En el Ejemplo 2.4 es importante notar que con base en una valuación cualquiera se establece cómo dos proposiciones concretas, para las cuales se conoce en detalle la estructura, tienen el mismo valor Booleano. Aplicando los planteamientos del Metateorema 2.23, se puede seguir el mismo procedimiento del Ejemplo 2.4 para proposiciones sin conocer completamente su estructura concreta. El Ejemplo 2.5 aborda esta idea para cualesquiera proposiciones ϕ y ψ, en lugar de las variables proposicionales p y q del Ejemplo 2.4.

Ejemplo 2.5

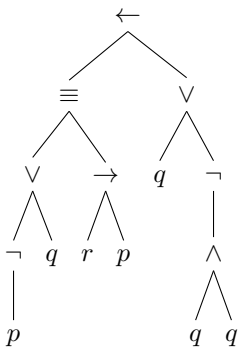
Se demostrará que $\mathbf{v}(((\phi \to \psi) \equiv ((\neg\phi) \vee \psi))) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$ y proposiciones ϕ y ψ. Por el Metateorema 2.23 se tiene:

$$\begin{aligned} \mathbf{v}((\phi \to \psi)) = \mathtt{F} \text{ sii } & \mathbf{v}(\phi) = \mathtt{T} \text{ y } \mathbf{v}(\psi) = \mathtt{F} && \text{(caso } \to\text{)} \\ \text{sii } & \mathbf{v}((\neg\phi)) = \mathtt{F} \text{ y } \mathbf{v}(\psi) = \mathtt{F} && \text{(caso } \neg\text{)} \\ \text{sii } & \mathbf{v}(((\neg\phi) \vee \psi)) = \mathtt{F} && \text{(caso } \vee\text{).} \end{aligned}$$

De estos cálculos se concluye que sin importar qué valuación se escoja ni qué variables proposicionales aparecen en ϕ y ψ, las proposiciones $(\phi \to \psi)$ y $((\neg\phi) \vee \psi)$ tienen el mismo valor de verdad, es decir, significan lo mismo.

Ejercicios

1. Considere la proposición ϕ representada por el siguiente árbol de sintaxis:



   a) Proponga una valuación $\mathbf{v}$ tal que $\mathbf{v}(\phi) = \mathtt{T}$.
   b) Proponga una valuación $\mathbf{w}$ tal que $\mathbf{w}(\phi) = \mathtt{F}$.

2. Considere valuaciones $\mathbf{v}$ y $\mathbf{w}$ tales que

$$\mathbf{v} = \{p \mapsto \mathtt{T}, q \mapsto \mathtt{F}, r \mapsto \mathtt{F}, \ldots\} \qquad \text{y} \qquad \mathbf{w} = \{p \mapsto \mathtt{T}, q \mapsto \mathtt{F}, r \mapsto \mathtt{T}, \ldots\}.$$

   Demuestre $\mathbf{v}((p \equiv (\neg q))) = \mathbf{w}((p \equiv (\neg q)))$.

3. Complete la demostración del caso inductivo para el Metateorema 2.19.
4. Complete la demostración del Metateorema 2.23 con los casos (3)-(7).
5. Demuestre que $\mathbf{v}(\phi) \neq \mathbf{v}((\neg\phi))$ para cualquier valuación $\mathbf{v}$.
6. Demuestre que $\mathbf{v}((\phi \equiv \phi)) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$.
7. Demuestre que $\mathbf{v}((\phi \equiv (\neg\phi))) = \mathtt{F}$ para cualquier valuación $\mathbf{v}$.
8. Demuestre que $\mathbf{v}((\phi \vee (\neg\phi))) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$.

9. Demuestre que $\mathbf{v}((\phi \wedge (\neg\phi))) = \mathtt{F}$ para cualquier valuación $\mathbf{v}$.

2.3. Clasificación de proposiciones

Como se observó en la Sección 2.2, el valor de verdad de una proposición es relativo a una valuación (o renglón en su tabla de verdad) dado que bajo dos valuaciones distintas este valor puede ser distinto. Esta sección presenta terminología que permite clasificar las proposiciones de acuerdo con el comportamiento de sus valores de verdad con respecto a *todas* las valuaciones (o renglones en sus tabla de verdad). Esta clasificación ayudará, por ejemplo, a analizar sistemáticamente algunos aspectos de la semántica proposicional y a justificar algunas decisiones de diseño que se toman en el sistema formal proposicional que se definirá en capítulos posteriores.

Considere inicialmente la proposición

"el agua moja"

que es verdadera por virtud de un hecho de la naturaleza; pueda que esta proposición sea falsa en otro mundo o realidad. A su vez, la proposición

"el agua moja o no moja"

es cierta por virtud de su estructura interna y, en particular, por el significado de la disyunción y la negación. Esta segunda proposición es cierta en cualquier mundo o realidad, y para el objeto de la clasificación propuesta en esta sección, corresponde a una tautología, i.e., a una proposición cuyo valor de verdad es verdadero con respecto a *cualquier* valuación.

Definición 2.25

Una proposición ϕ es una *tautología*, escrito $\models \phi$, si y solo si $\mathbf{v}(\phi) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$.

Una proposición ϕ es una tautología si para cualquiera de sus valuaciones, sin importar qué valores asigne a las variables proposicionales, el valor de verdad de ϕ es verdadero. Por ejemplo, las proposiciones $(p \equiv p)$, $(p \vee (\neg p))$ y $((p \to q) \equiv ((\neg p) \vee q))$ son tautologías, mientras que $(p \vee q)$ no lo es. Una tautología es cierta por virtud de su estructura interna y de sus conectivos lógicos, y resulta entonces que su interpretación es independiente de la interpretación de sus variables. Análogamente, una proposición es una tautología cuando todos los renglones de su tabla de verdad resultan en $\mathtt{T}$.

Ejemplo 2.6

A continuación se muestra que, para cualquier proposición ϕ, se tiene

$$\models (\phi \vee (\neg\phi)).$$

Sea $\mathbf{v}$ una valuación; considere los siguientes cálculos:

$$\begin{aligned}\mathbf{v}((\phi \vee (\neg\phi))) = \mathtt{T} \text{ sii } & \mathbf{v}(\phi) = \mathtt{T} \text{ o } \mathbf{v}((\neg\phi)) = \mathtt{T} \\ \text{sii } & \mathbf{v}(\phi) = \mathtt{T} \text{ o } \mathbf{v}(\phi) = \mathtt{F},\end{aligned}$$

lo cual es trivialmente cierto dado que el rango de $\mathbf{v}$ es $\mathbb{B}$ (Metateorema 2.19). Dado que $\mathbf{v}$ es cualquier valuación, se concluye $\models (\phi \vee (\neg\phi))$.

Fíjese que, a simple vista, la tarea de determinar si una proposición es una tautología parece humanamente imposible porque hay una cantidad infinita de valuaciones. Sin embargo, y gracias al Metateorema 2.21, para determinar si una proposición es una tautología basta con fijarse únicamente en una cantidad finita de valuaciones, i.e., una por cada renglón en la tabla de verdad de la proposición dada, y esto puede hacerse mecánicamente.

Metateorema 2.26

Hay un algoritmo que, dada cualquier proposición de la lógica proposicional, decide si esta es una tautología o no.

Demostración. El algoritmo requerido construye la tabla de verdad para la proposición dada y responde afirmativamente sii todos los renglones resultan en `T` (de lo contrario, responde negativamente). □

Nota 2.27

Gracias al Metateorema 2.26, se dice que la lógica proposicional es *decidible*.

Hay proposiciones que no son tautologías. Por ejemplo, la proposición p evalúa a `T` bajo una valuación $\{p \mapsto \mathtt{T}, \ldots\}$, pero evalúa a `F` bajo una valuación $\{p \mapsto \mathtt{F}, \ldots\}$. La Definición 2.28 presenta dos conceptos adicionales al de tatutología para clasificar proposiciones, completando así la clasificación propuesta en esta sección.

Definición 2.28

Sea ϕ una proposición. Se dice que:

1. ϕ es *satisfacible* si y solo si hay una valuación $\mathbf{v}$ tal que $\mathbf{v}(\phi) = \mathtt{T}$.
2. ϕ es *insatisfacible* (o una *contradicción*) si y solo si cualquier valuación $\mathbf{v}$ es tal que $\mathbf{v}(\phi) = \mathtt{F}$.

Note que si una proposición es una tautología, entonces es satisfacible. Además, una proposición es una tautología siempre y cuando su negación sea una contradicción (y viceversa). Algunas de estas relaciones se proponen como ejercicios para el lector.

Ejemplo 2.7

Para justificar que $(p \vee (\neg q))$ es satisfacible, basta con encontrar una valuación $\mathbf{v}$ tal que $\mathbf{v}((p \vee (\neg q))) = \mathtt{T}$. Note que

$$\mathbf{v} = \{p \mapsto \mathtt{T}, q \mapsto \mathtt{F}, \ldots\}$$

es tal que $\mathbf{v}((p \vee (\neg q))) = \mathtt{T}$. Dado que esta proposición es satisfacible, es imposible que sea una contradicción.

Antes de continuar, la Nota 2.29 introduce notación para expresar cuándo una proposición no es una tautología. Sin embargo, como se explica a continuación, hay que ser cuidadosos con esta nueva notación y así evitar su uso incorrecto.

Nota 2.29

Para cualquier proposición ϕ, la expresión

$$\not\models \phi$$

denota que ϕ *no* es una tautología.

Note que si una proposición ϕ no es una tautología, esto no indica que ϕ sea insatisfacible. Es decir, es incorrecto usar $\not\models \phi$ para denotar que la proposición ϕ es insatisfacible.

Por razones similares a las usadas para justificar por qué es posible determinar mecánicamente si una proposición es una tautología, también es posible determinar mecánicamente si una proposición es satisfacible o insatisfacible.

Metateorema 2.30

Las siguientes dos afirmaciones son ciertas:

1. Hay un algoritmo que, dada cualquier proposición de la lógica proposicional, decide si esta es satisfacible o no.
2. Hay un algoritmo que, dada cualquier proposición de la lógica proposicional, decide si esta es insatisfacible o no.

Demostración. La demostración de la existencia de cada uno de los dos algoritmos puede, por ejemplo, usar el algoritmo construído en la demostración del Metateorema 2.26. El detalle de los dos algoritmos se propone como ejercicio para el lector. □

Finalmente, se usa la noción de tautología para definir los conceptos de "equivalencia semántica" e "implicación semántica", comúnmente usados en informática y matemáticas.

Definición 2.31

Sean ϕ y ψ proposiciones. Se dice que:

1. ϕ y ψ son *lógicamente equivalentes* si y solo si $\models (\phi \equiv \psi)$.
2. ϕ *implica lógicamente* a ψ si y solo si $\models (\phi \rightarrow \psi)$.

Dos proposiciones son lógicamente equivalentes si sus valores de verdad coinciden para cualquier valuación. Una proposición implica lógicamente a otra siempre y cuando si la primera es verdadera bajo una valuación, entonces la segunda necesariamente es verdadera bajo esa misma valuación.

La implicación lógica no es una relación simétrica: puede que ϕ implique lógicamente a ψ, pero no necesariamente ψ debe implicar ϕ. En este sentido, la equivalencia lógica es un concepto más fuerte que la implicación lógica porque, a diferencia de la implicación lógica, la equivalencia lógica es simétrica.

Ejemplo 2.8

Las proposiciones $(p \equiv q)$ y $(q \equiv p)$ son lógicamente equivalentes, al igual que las proposiciones $(p \equiv (q \equiv r))$ y $((p \equiv q) \equiv r)$. La proposición $(p \wedge q)$ implica lógicamente a p. Sin embargo, las proposiciones p y q no son lógicamente equivalentes ni p implica lógicamente a $(p \wedge q)$ (¿por qué?).

Ejercicios

1. Demuestre que las siguientes proposiciones son tautologías para cualesquiera proposiciones ϕ, ψ, τ:
   a) $((\phi \equiv (\psi \equiv \tau)) \equiv ((\phi \equiv \psi) \equiv \tau))$.
   b) $((\phi \equiv \psi) \equiv (\psi \equiv \phi))$.
   c) $((\phi \equiv \mathit{true}) \equiv \phi)$.
   d) $((\phi \vee (\psi \vee \tau)) \equiv ((\phi \vee \psi) \vee \tau))$.
   e) $((\phi \vee \psi) \equiv (\psi \vee \phi))$.
   f) $((\phi \vee \mathit{false}) \equiv \phi)$.
   g) $((\phi \vee \phi) \equiv \phi)$.
   h) $((\phi \vee (\psi \equiv \tau)) \equiv ((\phi \vee \psi) \equiv (\phi \vee \tau)))$.
   i) $((\phi \wedge (\psi \wedge \tau)) \equiv ((\phi \wedge \psi) \wedge \tau))$.
   j) $((\phi \wedge \psi) \equiv (\psi \wedge \phi))$.
   k) $(\neg(\phi \wedge (\neg\phi)))$.
   l) $(\phi \to (\psi \to \phi))$.
   m) $((\phi \to (\psi \to \tau)) \equiv ((\phi \to \psi) \to (\phi \to \tau)))$.
   n) $((\phi \to \psi) \equiv ((\neg\psi) \to (\neg\phi)))$.
2. Demuestre que las siguientes proposiciones son satisfacibles pero no tautologías:
   a) $(p \equiv q)$.
   b) $((\neg p) \vee q)$.
   c) $((\neg p) \vee (p \wedge q))$.
   d) $(\neg(p \wedge (\neg q)))$.
   e) $((\neg(p \vee q)) \to p)$.
3. Proponga 3 proposiciones que sean contradicciones.
4. De acuerdo con la demostración del Metateorema 2.26, hay un procedimiento (o función), por ejemplo `taut`, que para cualquier proposición ϕ la invocación `taut(`ϕ`)` determina correctamente si ϕ es una tautología o no. Usando *únicamente* este procedimiento como oráculo, diseñe:
   a) un procedimiento `sat` que para cualquier proposición ϕ la invocación `sat(`ϕ`)` determine correctamente si ϕ es satisfacible o no.

   b) un procedimiento `unsat` que para cualquier proposición ϕ la invocación `unsat(`ϕ`)` determine correctamente si ϕ es insatisfacible o no.

5. Investigue y explique en qué consiste cada uno de los siguientes problemas de la lógica computacional:
   a) SAT
   b) 3SAT

6. En cada uno de los siguientes casos, determine si existe una proposición ϕ que sea una tautología y explique su respuesta:
   a) ϕ tiene a *true* como único conectivo lógico.
   b) ϕ tiene a *false* como único conectivo lógico.
   c) ϕ tiene a $\equiv$ como único conectivo lógico.
   d) ϕ tiene a $\not\equiv$ como único conectivo lógico.
   e) ϕ tiene a $\vee$ como único conectivo lógico.
   f) ϕ tiene a $\wedge$ como único conectivo lógico.
   g) ϕ tiene a $\rightarrow$ como único conectivo lógico.
   h) ϕ tiene a $\leftarrow$ como único conectivo lógico.

7. Demuestre para cualesquiera proposiciones ϕ, ψ, τ:
   a) $\models (\phi \equiv \psi)$ si y solo si $\models (\phi \rightarrow \psi)$ y $\models (\phi \leftarrow \psi)$.
   b) Si $\models (\phi \equiv \psi)$ y $\models (\psi \equiv \tau)$, entonces $\models (\phi \equiv \tau)$.
   c) Si $\models (\phi \equiv \psi)$ y $\models \phi$, entonces $\models \psi$.

8. Demuestre para cualesquiera proposiciones ϕ, ψ, τ:
   a) Si $\models (\phi \rightarrow \psi)$ y $\models (\psi \rightarrow \tau)$, entonces $\models (\phi \rightarrow \tau)$.
   b) Si $\models (\phi \rightarrow \psi)$ y $\models \phi$, entonces $\models \psi$.
   c) Si $\models (\phi \rightarrow \psi)$ y $\models (\psi \equiv \tau)$, entonces $\models (\phi \rightarrow \tau)$.
   d) Si $\models (\phi \equiv \psi)$ y $\models (\psi \rightarrow \tau)$, entonces $\models (\phi \rightarrow \tau)$.

9. Demuestre para cualquier proposición ϕ:
   a) ϕ es insatisfacible si y solo si $\models (\phi \equiv \mathit{false})$.
   b) ϕ es insatisfacible si y solo si $\models (\neg\phi)$.

10. Demuestre o refute para cualesquiera proposiciones ϕ y ψ:
    a) $\models (\phi \equiv \psi)$ si y solo si $\models \phi$ y $\models \psi$.
    b) $\models (\phi \wedge \psi)$ si y solo si $\models \phi$ y $\models \psi$.

11. Demuestre o refute para cualquier proposición ϕ:
    a) Si $\models \phi$, entonces ϕ es satisfacible.
    b) Si ϕ es satisfacible, entonces $\models \phi$.

12. Demuestre o refute para cualquier proposición ϕ:
    a) Si $\not\models \phi$, entonces ϕ es insatisfacible.
    b) Si ϕ es insatisfacible, entonces $\not\models \phi$.

2.4. Consecuencia tautológica

Esta sección introduce el concepto de consecuencia tautológica como herramienta para asociar una semántica precisa al significado de una argumentación, como aquellas en los ejemplos 1.1 y 1.2 al inicio del Capítulo 1. El concepto de consecuencia tautológica puede entenderse como una generalización del concepto de tautología (Sección 2.3) en donde el valor de verdad de una proposición ahora se hace relativo a un conjunto de proposiciones dado.

Para el propósito de presentar el concepto de consecuencia tautológica, es necesario generalizar los conceptos de satisfacibilidad e insatisfacibilidad para que ahora sean relativos a un conjunto de proposiciones.

Definición 2.32

Sea Γ un conjunto de proposiciones. Se dice que:

1. Γ es *satisfacible* si y solo si hay una valuación $\mathbf{v}$ tal que $\mathbf{v}(\phi) = \mathtt{T}$ para toda $\phi \in \Gamma$; en este caso también se dice que $\mathbf{v}$ *satisface* a Γ.
2. Γ es *insatisfacible* si y solo si no hay una valuación $\mathbf{v}$ tal que $\mathbf{v}(\phi) = \mathtt{T}$ para toda $\phi \in \Gamma$.

Hay una forma alternativa de plantear cuándo un conjunto de proposiciones Γ es insatisfacible. En particular, esta propiedad se puede formular de manera positiva: Γ es insatisfacible si y solo si para cualquier valuación $\mathbf{v}$ hay al menos una proposición $\phi \in \Gamma$ tal que $\mathbf{v}(\phi) = \mathtt{F}$. La correspondencia entre esta definición alternativa y la propuesta en la Definición 2.32 se propone como ejercicio de esta sección.

Ejemplo 2.9

El conjunto $\{p, q\}$ es satisfacible porque la valuación $\{p \mapsto \mathit{true}, q \mapsto \mathit{true}\}$ evalúa cada una de sus proposiciones a $\mathtt{T}$. El conjunto $\{p, (\neg p)\}$ es insatisfacible porque ninguna valuación puede hacer $\mathtt{T}$ las proposiciones p y $(\neg p)$ al tiempo.

Como se adelantó al inicio de esta sección, el concepto de consecuencia tautológica puede entenderse como una generalización del concepto de tautología en donde el valor de verdad de una proposición ϕ se hace relativo a un conjunto de proposiciones Γ. En una consecuencia tautológica, el conjunto Γ juega el papel de "filtro" de tal manera que únicamente se tengan en cuenta las valuaciones que satisfacen Γ cuando se analice el valor de verdad de ϕ. Esto permite ser más generales en la tarea de asignar un valor de verdad a una proposición.

Definición 2.33

Sea ϕ una proposición y Γ un conjunto de proposiciones. Se dice que ϕ es una *consecuencia tautológica* de Γ (alternativamente, Γ *tautológicamente implica* ϕ, escrito $\Gamma \models \phi$), si y solo si cualquier valuación que satisface a Γ también satisface a ϕ.

En una consecuencia tautológica $\Gamma \models \phi$ (Definición 2.33), cuando Γ es el conjunto vacío (i.e., $\Gamma = \{\}$), entonces la proposición ϕ es necesariamente una tautología y, convenientemente, $\{\} \models \phi$ se abrevia como $\models \phi$.

Note que hay dos casos en los cuales se puede establecer trivialmente $\Gamma \models \phi$. Uno de ellos cuando el conjunto de proposiciones Γ es insatisfacible porque no existe valuación alguna que falsifique ϕ y, a la vez, satisfaga Γ. El segundo cuando ϕ es una tautología porque sin importar si una valuación satisface o no a Γ, esta valuación satisface a ϕ.

Metateorema 2.34

Sean ϕ una proposición y Γ un conjunto de proposiciones:

1. Si Γ es insatisfacible, entonces $\Gamma \models \phi$.
2. Si $\models \phi$, entonces $\Gamma \models \phi$.

Demostración. Las demostraciones se proponen como ejercicio para el lector. □

En el caso general cuando Γ es satisfacible y ϕ no es una tautología, para establecer $\Gamma \models \phi$ se debe tomar cualquier valuación que satisfaga a Γ y demostrar que esta satisface a ϕ. De alguna forma, el conjunto Γ puede verse como un mecanismo que permite filtrar valuaciones y fijarse únicamente en algunas de ellas (precisamente en aquellas que satisfacen a Γ) para determinar el valor de verdad de ϕ (relativo a Γ).

Nota 2.35

Para establecer $\Gamma \models \phi$ cuando Γ es satisfacible y ϕ no es una tautología, se puede proceder de cualquiera de las siguientes maneras:

1. Suponer que hay una valuación $\mathbf{v}$ que satisface a Γ y demostrar que $\mathbf{v}(\phi) = \mathtt{T}$.
2. Demostrar que cualquier valuación $\mathbf{v}$ tal que $\mathbf{v}(\phi) = \mathtt{F}$ es tal que esta no satisface a Γ.

Ejemplo 2.10

Sea $\Gamma = \{(p \equiv q), p\}$. Note que $\Gamma \models q$ porque si $\mathbf{v}$ es tal que $\mathbf{v}((p \equiv q)) = \mathtt{T}$ y $\mathbf{v}(p) = \mathtt{T}$, por la función $H_{\equiv}$ se sigue $\mathbf{v}(q) = \mathtt{T}$. Otra forma de proceder para llegar a la misma conclusión es tratar de falsificar la conscuencia lógica: encontrar una valuación $\mathbf{v}$ tal que $\mathbf{v}(q) = \mathtt{F}$ y que $\mathbf{v}((p \equiv q)) = \mathbf{v}(p) = \mathtt{T}$. Pero note que si $\mathbf{v}(p) = \mathtt{T}$, entonces es imposible tener $\mathbf{v}((p \equiv q)) = \mathtt{T}$ y $\mathbf{v}(q) = \mathtt{F}$. En este caso también se tiene $\Gamma \models q$.

Así como no toda proposición es una tautología, no se puede esperar que una proposición sea consecuencia tautológica de un conjunto de proposiciones arbitrario.

Nota 2.36

Para ϕ una proposición y Γ un conjunto de proposiciones, la expresión

$$\Gamma \not\models \phi$$

denota que ϕ *no* es una consecuencia tautológica de Γ (o que Γ *no* implica tautológicamente ϕ). Esto quiere decir que hay al menos una valuación $\mathbf{v}$ que cumple $\mathbf{v}(\phi) = \mathtt{F}$ y que satisface a Γ.

A diferencia de una demostración de $\Gamma \models \phi$ que involucra a *toda* valuación, para demostrar que ϕ no es una consecuencia tautológica de Γ (i.e., $\Gamma \not\models \phi$) basta con encontrar al menos *una* valuación que satisfaga Γ y que no satisfaga a ϕ. En otras palabras, basta con encontrar un testigo de la afirmación "no toda valuación que satisface Γ es tal que también satisface a ϕ", lo cual corresponde a la negación de la definición de consecuencia tautológica (Definición 2.33), como se propone en la Nota 2.36.

Nota 2.37

Para justificar $\Gamma \not\models \phi$ cuando Γ es satisfacible y ϕ no es una tautología, se puede proceder de cualquiera de las siguientes maneras:

1. Proponer una valuación $\mathbf{v}$ que satisfaga a Γ y establecer $\mathbf{v}(\phi) = \mathtt{F}$.
2. Proponer una valuación $\mathbf{v}$ tal que $\mathbf{v}(\phi) = \mathtt{F}$ y establecer que $\mathbf{v}$ satisface a Γ.

Ejemplo 2.11

Sea $\Gamma = \{(p \vee q), p\}$. Note que q no es una consecuencia tautológica de Γ. Tome, por ejemplo, $\mathbf{v} = \{p \mapsto \mathtt{T}, q \mapsto \mathtt{F}, \ldots\}$ y note que $\mathbf{v}$ satsiface Γ pero $\mathbf{v}(q) = \mathtt{F}$. Luego q no es una consecuencia tautológica de Γ, es decir, $\{(p \vee q), p\} \not\models q$.

El Metateorema 2.38 establece una relación clara entre los conceptos de tautología y consecuencia tautológica para casos en los cuales el conjunto de proposiciones Γ que acompaña la consecuencia tautológica es *finito*. Para el caso en el cual Γ puede ser infinito, hay un resultado similar pero que está fuera del alcance de esta sección.

Metateorema 2.38

Sea ϕ una proposición y $\Gamma = \{\phi_1, \phi_2, \ldots, \phi_n\}$ un conjunto de proposiciones, para algún $n \in \mathbb{N}$. Las siguientes dos afirmaciones son equivalentes:

1. $\models (\phi_1 \to (\phi_2 \to (\cdots \to (\phi_n \to \phi) \cdots)))$
2. $\Gamma \models \phi$

Demostración. Se puede obtener una demostración por inducción sobre $n \in \mathbb{N}$, la cual se propone como ejercicio para el lector. □

Hay una observación importante acerca del Metateorema 2.38. Note que la proposición en la expresión (1) es una de varias posibles dado que el orden de las proposiciones en Γ puede ser establecido de muchas formas. La moraleja que deja esta sitaución, es que en algunas ocasiones puede ser provechoso ser oportunista y construír la proposición en la expresión (1) ordenando las proposiciones en Γ como mejor convenga para resolver el problema dado.

Ejercicios

1. Demuestre que un conjunto de proposiciones Γ es insatisfacible si y solo si para cualquier valuación $\mathbf{v}$ hay al menos una proposición $\phi \in \Gamma$ tal que $\mathbf{v}(\phi) = \mathtt{F}$.
2. Demuestre para cualesquiera proposiciones ϕ, ψ, τ:
   a) $\{\phi\} \models \phi$.
   b) $\{(\neg(\phi \equiv \phi))\} \models \psi$.
   c) $\{\phi\} \models (\phi \vee \psi)$.
   d) $\{(\phi \vee \psi), ((\neg\phi) \vee \tau)\} \models (\psi \vee \tau)$.

3. Demuestre el Metateorema 2.34.1.
4. Demuestre el Metateorema 2.34.2.
5. Demuestre el Metateorema 2.38.
6. Sean Γ un conjunto de proposiciones y ϕ, ψ proposiciones. Demuestre o refute:
   a) $\Gamma \models (\phi \vee \psi)$ si y solo si $\Gamma \models \phi$ o $\Gamma \models \psi$.
   b) $\Gamma \models (\phi \wedge \psi)$ si y solo si $\Gamma \models \phi$ y $\Gamma \models \psi$.
   c) $\Gamma \models (\phi \rightarrow \psi)$ si y solo si $\Gamma \not\models \phi$ o $\Gamma \models \psi$.
   d) $\Gamma \not\models (\phi \vee \psi)$ si y solo si $\Gamma \not\models \phi$ y $\Gamma \not\models \psi$.
   e) $\Gamma \not\models (\phi \wedge \psi)$ si y solo si $\Gamma \not\models \phi$ o $\Gamma \not\models \psi$.
   f) $\Gamma \not\models (\phi \rightarrow \psi)$ si y solo si $\Gamma \models \phi$ y $\Gamma \not\models \psi$.
7. Sean Γ y Δ conjuntos de proposiciones, y ϕ, ψ proposiciones. Demuestre:
   a) Si $\models \phi$, entonces $\Gamma \models \phi$.
   b) Si $\Gamma \models \phi$ y $\Gamma \subseteq \Delta$, entonces $\Delta \models \phi$.
   c) $\Gamma \cup \{\phi\} \models \psi$ si y solo si $\Gamma \models (\phi \rightarrow \psi)$.
   d) Si $\Gamma \cup \{\phi\} \models \psi$ y $\Gamma \cup \{(\neg\phi)\} \models \psi$, entonces $\Gamma \models \psi$.
8. Sea Γ un conjunto de proposiciones. Demuestre que Γ es insatisfacible si y solo si $\Gamma \models \mathit{false}$.
9. Sean Γ y Δ conjuntos de proposiciones, y ϕ una proposición. Demuestre que si $\Gamma \models \phi$ y $\Delta \models (\neg\phi)$, entonces $\Gamma \cup \Delta$ es insatisfacible.
10. Sean Γ y Δ conjuntos de proposiciones, y ϕ, ψ proposiciones. Demuestre o refute:
    a) Si $\Gamma \not\models \phi$ y $\Gamma \subset \Delta$, entonces $\Delta \not\models \phi$.
    b) Si $\Delta \not\models \phi$ y $\Gamma \subset \Delta$, entonces $\Gamma \not\models \phi$.
11. Considere el Ejercicio 5 en la Sección 1.2. Especifique cada una de las argumentaciones y determine cuáles de las argumentaciones son válidas o inválidas. Justifique su respuesta.

2.5. Argumentaciones

Habiendo definido el concepto de consecuencia tautológica en la Sección 2.4, es posible asociar una semántica precisa a las argumentaciones.

Ejemplo 2.12

Considere el Ejemplo 1.1:

Si el tren llega tarde y no hay taxis en la estación, entonces Juan llegará tarde a su reunión. Juan no llega tarde a su reunión. El tren llegó tarde. Consecuentemente, había taxis en la estación.

Usando la simbolización:

p**:** el tren llega tarde

q**:** hay taxis en la estación

r**:** Juan llega tarde a la reunión

el Ejemplo 1.1 puede especificarse (o ser traducido) al lenguaje de la lógica proposicional con la siguiente lista de proposiciones:

$$((p \wedge (\neg q)) \rightarrow r), (\neg r), p, q.$$

La lista de proposiciones obtenida al final del Ejemplo 2.12 recibe el nombre de *forma de argumentación* (o *argumentación*).

Definición 2.39

Una *argumentación* es una secuencia no vacía de proposiciones. En una argumentación, todas las proposiciones con excepción de la última reciben el nombre de *hipótesis*, mientras que la última proposición recibe el nombre de *conclusión*.

Finalmente, se define cuándo una argumentación es *válida* o *inválida*, es decir, se establece una semántica formal para las argumentaciones.

Definición 2.40

Sea $\phi_1, \ldots, \phi_n, \phi$ una argumentación:

1. Se dice que $\phi_1, \ldots, \phi_n, \phi$ es *válida* si y solo si ϕ es una consecuencia tautológica de $\{\phi_1, \ldots, \phi_n\}$, i.e., $\{\phi_1, \ldots, \phi_n\} \models \phi$.
2. De lo contrario se dice que $\phi_1, \ldots, \phi_n, \phi$ es *inválida*, i.e., cuando $\{\phi_1, \ldots, \phi_n\} \not\models \phi$.

Ejemplo 2.13

Se demuestra que la siguiente argumentación, correspondiente al Ejemplo 2.12, es válida:

$$((p \wedge (\neg q)) \rightarrow r), (\neg r), p, q.$$

Por la Definición 2.40, se debe demostrar

$$\{((p \wedge (\neg q)) \rightarrow r), (\neg r), p\} \models q.$$

Sea $\mathbf{v}$ una valuación tal que:

$$(1)\ \mathbf{v}(((p \wedge (\neg q)) \rightarrow r)) = \mathtt{T}, \qquad (2)\ \mathbf{v}((\neg r)) = \mathtt{T}, \qquad (3)\ \mathbf{v}(p) = \mathtt{T}.$$

El objetivo es demostrar $\mathbf{v}(q) = \mathtt{T}$. De (2) se tiene que $\mathbf{v}(r) = \mathtt{F}$. Esto, junto con (1), indica que $\mathbf{v}((p \wedge (\neg q))) = \mathtt{F}$. Por (3), se tiene necesariamente que $\mathbf{v}((\neg q)) = \mathtt{F}$, es decir, $\mathbf{v}(q) = \mathtt{T}$ y se concluye que la argumentación es válida.

Una demostración alternativa a la presentada en el Ejemplo 2.13, usando el Metateorema 2.38, se propone como ejercicio para el lector.

Ejemplo 2.14

Considere la siguiente argumentación:

$$(\neg(p \wedge q)), (\neg p), q.$$

Para demostrar que es inválida, basta con encontrar una valuación que satisfaga a $\{(\neg(p \wedge q)), (\neg p)\}$ y que falsifique a q. Basta con tomar $\mathbf{v} = \{p \mapsto \mathtt{F}, q \mapsto \mathtt{F}\}$ y notar que:

$$\mathbf{v}((\neg(p \wedge q))) = \mathbf{v}((\neg p)) = \mathtt{T} \qquad \text{pero} \qquad \mathbf{v}(q) = \mathtt{F}.$$

Ejercicios

1. Proponga una demostración del Ejemplo 2.13 que use directamente el Metateorema 2.38.
2. Demuestre que cada una de las siguientes argumentaciones son válidas:
   a) $(\phi \equiv \psi), (\neg\phi), (\neg\psi)$.
   b) $(\phi \rightarrow \psi), (\neg\psi), (\neg\phi)$.
   c) $(\phi \vee \psi), (\neg\phi), \psi$.
   d) $(\phi \rightarrow \psi), (\phi \rightarrow (\neg\psi)), (\neg\phi)$.

3. Demuestre que cada una de las siguientes argumentaciones es inválida:
   a) $(p \to q), q, p$.
   b) $(p \to q), (\neg p), (\neg q)$.
4. ¿Si una argumentación $\phi_1, \ldots, \phi_n, \phi$ es inválida, entonces la argumentación

   $$\phi_1, \ldots, \phi_n, (\neg\phi)$$

   es válida? Justifique su respuesta.

2.6. La isla de caballeros y escuderos

Hay una amplia variedad de acertijos lógicos (i.e., adivinanzas) relativas a una isla en la cual ciertos habitantes llamados *caballeros* dicen siempre la verdad y otros llamados *escuderos* siempre mienten. En dicha isla cada habitante es caballero o escudero; de ahí su nombre: *la isla de caballeros y escuderos*. Esta sección presenta: (i) una forma para especificar acertijos relacionados con la isla de caballeros y escuderos, y (ii) tres técnicas basadas en la semántica proposicional para analizar estas especificaciones. Los metodos presentados en esta sección pueder ser utilizados para analizar otros tipos de acertijos.

Nota 2.41

Las isla de caballeros y escuderos (en inglés, *the island of knights and knaves*) fue presentada inicialmente por Raymond Smullyan en su libro "¿Cómo se llama este libro?" (en inglés, *What is the name of this book?*) en 1978. Smullyan fue un prolífico matemático y lógico Estadounidense.

2.6.1. Especificación. La especificación de un acertijo de la isla de caballeros y escuderos consiste en establecer proposiciones que representen afirmaciones hechas por los habitantes de la isla. Por convención, los habitantes de la isla son identificados con las letras mayúsculas $A, B, \ldots, Z$.

Ejemplo 2.15

Suponga que un turista está en presencia de dos habitantes de la isla llamados A y B, respectivamente. A dice: "al menos uno de nosotros es escudero". ¿Puede el turista determinar las naturalezas de A y B?

El Ejemplo 2.15 presenta un acertijo prototípico de la isla de caballeros y escuderos. Este acertijo involucra a dos "isleños" llamados A y B, y pregunta por la posibilidad

de determinar sus naturalezas. Dado que únicamente hay dos posibles naturalezas para cada habitante de la isla, la naturaleza de una habitante se modela con una variable proposicional.

Nota 2.42

Si la variable proposicional p modela la naturaleza del isleño P, la asignación $p \mapsto \texttt{T}$ indica que P dice la verdad (i.e., P es caballero) y la asignación $p \mapsto \texttt{F}$ indica que P miente (i.e., P es escudero).

Bajo la convención en la Nota 2.42, especificar afirmaciones de la forma "el habitante P es caballero" y "el habitante P es escudero" es directo: la primera afirmación se especifica con la proposición p y la segunda afirmación con la proposición $(\neg p)$. Para el primer caso, la justificación es que si la afirmación "el habitante P es caballero" es cierta, entonces p debe ser asignado `T`; conversamente, si p es asignado `T`, entonces la afirmación "P es caballero" debe ser cierta. Para el segundo caso, la justificación es que si la afirmación "el habitante P es escudero", entonces p debe ser asignado `F`; conversamente, si p es asignado `F`, entonces la afirmación "P es escudero" debe ser cierta. Note que por convención, la naturaleza de una habitante de la isla está siendo modelada con la variable proposicional correspondiente a la versión minúscula de su nombre.

Ejemplo 2.16

Considere el Ejemplo 2.15. Lo dicho por A puede ser especificado con la proposición:

$$((\neg a) \vee (\neg b)).$$

Fíjese que la proposición $((\neg a) \vee (\neg b))$ en el Ejemplo 2.16 es cierta únicamente cuando al menos una de las variables a y b es asignada `F`, i.e., cuando al menos uno entre A y B es escudero. Esto coincide con la afirmación hecha inicialmente por A en el Ejemplo 2.15.

Hasta ahora se ha logrado establecer cómo especificar con proposiciones las afirmaciones hechas por los habitantes de la isla. Sin embargo, aún no se ha establecido cómo relacionar cada una de estas proposiciones con la naturaleza del isleño que la emite. Para avanzar en esta dirección, observe que hay una relación biunívoca entre la naturaleza de quien enuncia una afirmación y la veracidad de dicha afirmación. Por ejemplo, si un caballero hace una afirmación, entonces dicha afirmación es cierta. Análogamente, si dicha afirmación es cierta, entonces quien la

enunció es caballero. En general, se puede observar que la veracidad de una afirmación es *lógicamente equivalente* a la naturaleza de quien la enuncia. Fíjese que de esta forma es imposible que un caballero mienta o que un escudero diga la verdad.

Ejemplo 2.17

Considere el Ejemplo 2.15. Este acertijo puede ser especificado por la siguiente proposición:

$$(a \equiv ((\neg a) \vee (\neg b))).$$

En el Ejemplo 2.17 se establece que

> A dice: "al menos uno de nosotros es escudero"

se especifica con la proposición

$$(a \equiv ((\neg a) \vee (\neg b))).$$

Observe cómo la estructura de la afirmación es similar a la de la proposición: el fragmento "A dice: ..." es especificado con el patrón "$a \equiv \ldots$". Esto corresponde al hecho de que la naturaleza de A es consistente con el valor de verdad de lo que dice, como se explicó anteriormente.

2.6.2. Análisis por casos. En el caso del Ejemplo 2.15, el objetivo es determinar, de ser posible, las naturalezas de A y B con base en la afirmación hecha por A. Dado que el Ejemplo 2.17 presenta una proposición que formaliza la afirmación hecha por A, basta entonces con analizar esta proposición. En este caso, el interés es por encontrar valuaciones de a y b que hagan cierta dicha proposición, y así determinar si A y B pueden ser clasificados unívocamente.

Ejemplo 2.18

El *análisis por casos* del acertijo del Ejemplo 2.15 se hace sobre la proposición propuesta en el Ejemplo 2.17:

$$(a \equiv ((\neg a) \vee (\neg b))).$$

Caso $a \mapsto$ `T`**:** Para que el lado derecho de la equivalencia sea cierto, se requiere $b \mapsto$ `F`, de lo contrario la proposición no evaluaría a `T`. Entonces, en el caso en que A sea caballero, el habitante B necesariamente es escudero.

Caso $a \mapsto$ `F`**:** En este caso, el lado izquierdo de la equivalencia evalúa a `F`, mientras que el lado derecho, independientemente del valor de b, evalúa a `T`. Es decir,

en este caso no es posible que la proposición sea cierta. Consecuentemente, es imposible que A sea escudero.

Se concluye que A es caballero y B es escudero.

El análisis en el Ejemplo 2.18 se hace por casos sobre la naturaleza de A. Inicialmente, se explora la posibilidad de que A sea caballero, concluyendo entonces que B es escudero. Posteriormente, se explora la posibilidad de que A sea escudero, concluyendo que esta situación es imposible. Resumiendo, solo hay una posibilidad y entonces las naturalezas de A y B se pueden determinar unívocamente: A es caballero y B es escudero.

2.6.3. Análisis con tablas de verdad. Hay una forma alternativa de analizar este tipo de acertijos. En particular, note que en el análisis por casos del Ejemplo 2.18 explora *todas* las posibles valuaciones de la proposición que especifica el acertijo. Otra forma de analizar el acertijo es por medio de tablas de verdad.

Ejemplo 2.19

El *análisis por tabla de verdad* del acertijo del Ejemplo 2.15 se hace sobre la proposición propuesta en el Ejemplo 2.17:

$$(a \equiv ((\neg a) \vee (\neg b))).$$

La tabla de verdad de la proposición es la siguiente:

a	b	$((\neg a) \vee (\neg b))$	$(a \equiv ((\neg a) \vee (\neg b)))$
F	F	T	F
F	T	T	F
T	F	T	T
T	T	F	F

El *único* renglón de la tabla de verdad que hace la proposición cierta corresponde a $a \mapsto \mathtt{T}$ y $b \mapsto \mathtt{F}$. Entonces, A es caballero y B es escuedero.

2.6.4. Análisis con argumentaciones. Finalmente, se presenta como tercera opción, una forma de analizar acertijos de la isla de caballeros y escuderos con ayuda de argumentaciones. Esta forma de análisis, consiste en especificar un acertijo como una argumentación de la forma

$$\phi_1, \ldots, \phi_n, \phi,$$

en donde $\phi_1, \ldots, \phi_n$ son proposiciones que especifican información suministrada en el enunciado del acertijo y ϕ es una proposición incógnita. Con este enfoque, el objetivo es 'despejar' ϕ buscando proposiciones que suministren información sobre la naturaleza de los habitantes de la isla involucrados en el acertijo. Por ejemplo, si $\phi_1, \ldots, \phi_n$ son las hipótesis de la argumentación y se quiere averigüar si el habitante P *puede* ser caballero, basta con encontrar una valuación $\mathbf{v}$ tal que $\mathbf{v}(p) = \mathtt{T}$ y $\mathbf{v}$ satisfaga $\{\phi_1, \ldots, \phi_n\}$ (i.e., basta con determinar si $\{\phi_1, \ldots, \phi_n, p\}$ es satisfacible). Coversamente, si se quiere averigüar si el habitante P *puede* ser escudero, basta con encontrar una valuación $\mathbf{w}$ tal que $\mathbf{w}(p) = \mathtt{F}$ y $\mathbf{w}$ satisfaga $\{\phi_1, \ldots, \phi_n\}$ (i.e., basta con determinar si $\{\phi_1, \ldots, \phi_n, (\neg p)\}$ es satisfacible). En el primer caso, de ser factible, se establece que la situación descrita por el acertijo es posible cuando P es caballero. Similarmente y de ser factible, en el segundo caso se establece que la situación descrita por el acertijo es posible cuando P es escudero. Luego, si *exactamente* uno de los dos casos es factible para P, entonces se ha determinado unívocamente la naturaleza de P.

Ejemplo 2.20

En esta ocasión, tres habitantes de la isla, llamados A, B y C, están siendo entrevistados. Se emiten las siguientes afirmaciones:

A **dice:** B es caballero.

B **dice:** Si A es caballero, entonces también lo es C.

El objetivo es determinar las naturalezas de A, B y C.

Solución. Inicialmente se especifican las afirmaciones hechas por A y B con las siguientes proposiciones:

$$(a \equiv b)$$
$$(b \equiv (a \to c))$$

Para saber si A puede ser caballero, se trata de encontrar una valuación $\mathbf{v}$ tal que $\mathbf{v}(a) = \mathtt{T}$ y $\mathbf{v}$ satisfaga las hipótesis $\{(a \equiv b), (b \equiv (a \to c))\}$. En este caso, $\mathbf{v} = \{a \mapsto \mathtt{T}, b \mapsto \mathtt{T}, c \mapsto \mathtt{T}, \ldots\}$ sirve como testigo. Entonces, A puede ser caballero y, además, también B y C pueden ser caballeros.

Para saber si A puede ser escudero, se busca una valuación $\mathbf{w}$ tal que $\mathbf{w}(a) = \mathtt{F}$ y $\mathbf{w}$ satisfaga las hipótesis. Note que dicha valuación no existe porque, por la primera suposición, necesariamente $\mathbf{w}(b) = \mathtt{F}$ y por la segunda suposición, independientemente del valor de c, se tiene $H_{\equiv}(\mathtt{F}, \mathtt{T}) = \mathtt{T}$, lo cual es imposible. Entonces, A no puede ser escudero.

Se ha establecido que A es caballero. Por la primera suposición, las naturalezas de A y B son las mismas, entonces B también es caballero. Dado que A y B son caballeros, de la segunda suposición se tiene que C no puede ser escudero, es decir, C es caballero. En conclusión, A, B y C son caballeros.

Ejercicios

1. Explique por qué en el Ejemplo 2.18 es suficiente hacer análisis por casos sobre a, y no es necesario hacer análisis por casos sobre a y b.
2. Suponga que un habitante de la isla llamado A dice: "soy un escudero o B es un caballero". Determine la naturaleza de A y B.
3. Suponga que un habitante de la isla llamado A dice: "soy un escudero y B no lo es". Determine la naturaleza de A y B.
4. Suponga que un turista está en presencia de dos habitantes de la isla llamados A y B. A dice : "nosotros tenemos la misma naturaleza". ¿Pueden determinarse las naturalezas de A y B? Justifique su respuesta.
5. Suponga que un turista está en presencia de dos habitantes de la isla llamados A y B. A dice : "al menos uno de nosotros es caballero". ¿Pueden determinarse las naturalezas de A y B? Justifique su respuesta.
6. Proponga una afirmación que puede ser hecha por cualquier habitante de la isla, sin importar si este es caballero o escudero. Explique su respuesta.
7. Proponga una afirmación que no puede ser hecha por un habitante de la isla, sin importar si este es caballero o escudero. Explique su respuesta.
8. Suponga que las variables proposicionales a y b representan la naturaleza de dos habitantes de la isla llamados A y B. Invente un acertijo que corresponda a la siguiente especificación, y determine la naturaleza de A y B:

   $(a \equiv (\neg b))$,

   $(b \equiv (a \wedge b))$.
9. Resuelva el acertijo del Ejemplo 2.20 por análisis de casos.
10. Resuelva el acertijo del Ejemplo 2.20 por tablas de verdad.
11. Hace muchos años, tres habitantes de la isla, llamados A, B y C, estaban en un jardín. Un turista pasó por allí y le preguntó a A: "¿eres caballero o escudero?". El habitante A respondió, pero tan confusamente que el turista no entendió la respuesta. Entonces el turista preguntó a B: "¿Qué dijo A?"; a lo

cual B respondió: "A dijo que es escudero". En ese momento C intervino con la siguiente afirmación: "No le crea a B porque está mintiendo". Determine la naturaleza de B y C.

12. Al resolver el Ejercicio 11 es claro que el habitante C no cumple ningún papel importante en el acertijo. Desde el momento en que B habló se puede saber que estaba mintiendo sin el testimonio de C. Considere la siguiente variante de ese acertijo.
    El turista le pregunta a A: "¿cuántos caballeros hay entre Ustedes?". El habitante A respondió, de nuevo, confusamente y el turista no entendió la respuesta. Entonces el turista pregunta a B: "¿Qué dijo A?"; a lo cual B responde: "A dijo que hay al menos un caballero entre nosotros". Y por su parte C dice: "No le crea a B porque está mintiendo". Determine la naturaleza de B y C.

13. En esta ocasión hay tres habitantes de la isla llamados A, B y C:
    A **dice:** Todos nosotros somos escuderos.
    B **dice:** Exactamente uno de nosotros es caballero.
    Determine la naturaleza de A, B y C.

14. Tres habitantes de la isla llamados A, B y C están reunidos:
    A **dice:** Todos nosotros somos escuderos.
    B **dice:** Exactamente uno de nosotros es escudero.
    ¿Puede determinarse la naturaleza de B y C? Justifique su respuesta.

15. De nuevo, tres habitantes de la isla llamados A, B y C están reunidos:
    A **dice:** B es escudero.
    B **dice:** A y C son del mismo tipo.
    Determine la naturaleza de C.

16. Tres habitantes de la isla llamados A, B y C están reunidos. A dice: "B y C son de la misma naturaleza". Alguien pregunta entonces a C: "¿Son A y B de la misma naturaleza?". Determine, justificando su respuesta, qué responde C.

17. Diseñe un acertijo que involucre a un habitante de la isla de caballeros y escuderos, y que permita determinar que es caballero.

18. Diseñe un acertijo que involucre a un habitante de la isla de caballeros y escuderos, y que permita determinar que es escudero.

19. Diseñe un acertijo que involucre a dos habitantes de la isla de caballeros y escuderos, y que permita determinar que ambos son caballeros.

20. Diseñe un acertijo que involucre a dos habitantes de la isla de caballeros y escuderos, y que permita determinar que ambos son escuderos.

21. Diseñe un acertijo que involucre a dos habitantes de la isla de caballeros y escuderos, y que permita determinar que son de diferente naturaleza, pero sin saber qué son cada uno de ellos.

22. Diseñe un acertijo que involucre a tres habitantes de la isla de caballeros y escuderos, y que permita determinar que todos son escuderos.

23. Hace muchos años, algunos de los habitantes de la isla de caballeros y escuderos eran hombres lobo, los cuales tenian la fea costumbre de transformarse en la noche y devorar a la gente. Considere la siguiente situación en la cual un turista se encontró con tres habitantes de la isla, llamados A, B, C:

    A dice: Yo soy hombre lobo.

    B dice: Yo soy hombre lobo.

    C dice: A lo sumo uno de nosotros es caballero.

    Suponiendo que exactamente uno de A, B, C es hombre lobo, haga una clasificación completa de sus naturalezas. *Ayuda:* note que al menos uno entre A y B está mintiendo.

24. Suponga que Γ es un conjunto de proposiciones que especifica información dada acerca de un acertijo de la isla de caballeros y escuderos. Además, suponga que la variable proposicional a modela la naturaleza de un habitante A de la isla. Demuestre o refute:

    a) Si A es caballero, entonces $\Gamma \models a$.

    b) Si $\Gamma \models a$, entonces A es caballero.

Capítulo 3

Herramientas proposicionales

Este capítulo desarrolla algunas herramientas que permiten la manipulación, construcción y análisis de proposiciones, además de propiedades de expresividad de algunos conectivos lógicos. La finalidad de este capítulo es brindar un compendio de herramientas útiles para los capítulos posteriores, específicamente para la formulación del sistema formal para la lógica proposicional de Dijkstra y Scholten en el Capítulo 4.

3.1. Sustitución textual

La sustitución textual es un mecanismo que permite usar la estructura sintáctica de una proposición para obtener otra proposín. Por ejemplo, a partir de la proposicion $(p \equiv q)$ se puede obtener la proposición $((p \wedge r) \equiv q)$ al sustituir en la primera proposición la variable proposicional p por la proposición $(p \wedge r)$. Esta sección presenta el concepto de sustitución textual y lo ilustra con algunos ejemplos.

La sustitución textual se define con base en el concepto de *sustitución*.

Definición 3.1

Una *sustitución* es una función $F : \mathcal{V} \to \mathcal{T}(\mathcal{V})$ distinta a la identidad en una cantidad finita de elementos del dominio.

Una sustitución F es una función que asocia una proposición $F(p)$ a cualquier variable proposicional p.

Nota 3.2

Cualquier sustitución F es tal que $F(p) \neq p$ para una cantidad finita de variables p. Por ello, una sustitución siempre puede ser escrita como un conjunto finito de la forma

$$\{q_0 \mapsto \phi_0, q_1 \mapsto \phi_1, \ldots, q_n \mapsto \phi_n\}$$

indicando que la proposición ϕ_i está asociada a la variable proposicional q_i ($0 \leq i \leq n$) y cualquier otra variable que no aparece en la lista $q_0, \ldots, q_n$ está asociada a sí misma. Note que bajo esta convención una sustitución puede ser escrita en más de una forma. Sin embargo, esta situación no traerá problemas en la definición de sustitución textual.

Como se explicó inicialmente, una sustitución F puede ser aplicada a una proposición ϕ con el propósito de obtener una proposición $F(\phi)$ en la cual algunas variables proposicionales p en ϕ son *sustituídas* por $F(p)$. La aplicación de una sustitución a una proposición recibe el nombre de *sustitución textual*.

Definición 3.3

Sea ϕ una proposición y F una sustitución. La *sustitución textual* de F en ϕ, denotada como $\overline{F}(\phi)$, se define inductivamente para toda subproposición de ϕ de la siguiente forma:

1. $\overline{F}(p) = F(p)$
2. $\overline{F}(\mathit{true}) = \mathit{true}$
3. $\overline{F}(\mathit{false}) = \mathit{false}$
4. $\overline{F}((\neg\psi)) = (\neg\overline{F}(\psi))$
5. $\overline{F}((\psi \otimes \tau)) = (\overline{F}(\psi) \otimes \overline{F}(\tau))$, si $\otimes \in \{\equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$.

De acuerdo con la Definición 3.3, aplicar una sustitución F a una proposición ϕ resulta en una proposición $\overline{F}(\phi)$ similar a ϕ pero en la cual algunas variables proposicionales han sido cambiadas. En particular, el caso (1) indica explícitamente cómo una variable proposicional p que aparece en ϕ es reemplazada por $F(p)$. Para este caso, note que si p no hace parte de las variables "afectadas" por F, entonces $F(p) = p$, lo cual, para efectos prácticos, resulta en no sustituír la variable p. Una sustitución textual no afecta a las constantes *true* y *false* (casos (2) y (3)). Los demás casos en la Definición 3.3 corresponden a definiciones inductivas que dependen de los demás conectivos lógicos.

Ejemplo 3.1

Considere la sustitución $F = \{p \mapsto (p \wedge \mathit{false}), r \mapsto \mathit{true}, s \mapsto q\}$:

$$\begin{aligned}
\overline{F}(((p \to q) \not\equiv (\neg r))) &= (\overline{F}((p \to q)) \not\equiv \overline{F}((\neg r))) && \text{(caso } \not\equiv\text{)} \\
&= ((\overline{F}(p) \to \overline{F}(q)) \not\equiv \overline{F}((\neg r))) && \text{(caso } \to\text{)} \\
&= (((p \wedge \mathit{false}) \to \overline{F}(q)) \not\equiv \overline{F}((\neg r))) && \text{(caso } p \in \{p, r, s\}\text{)} \\
&= (((p \wedge \mathit{false}) \to q) \not\equiv \overline{F}((\neg r))) && \text{(caso } q \notin \{p, r, s\}\text{)} \\
&= (((p \wedge \mathit{false}) \to q) \not\equiv (\neg \overline{F}(r))) && \text{(caso } \neg\text{)} \\
&= (((p \wedge \mathit{false}) \to q) \not\equiv (\neg \mathit{true})) && \text{(caso } r \in \{p, r, s\}\text{).}
\end{aligned}$$

Note que como $q \notin \{p, r, s\}$ se tiene por convención que $F(q) = q$.

La extensión $\overline{F}$ de una sustitución F es una función del conjunto de proposiciones en si mismo.

Metateorema 3.4

Si F es una sustitución, entonces $\overline{F}$ es una función con dominio y rango en el conjunto de proposiciones (i.e., $\overline{F} : \mathcal{T}(\mathcal{V}) \to \mathcal{T}(\mathcal{V})$).

Demostración. Esta demostración se propone como ejercicio para el lector y se sugiere usar como guía la demostración del Metateorema 2.19. □

Nota 3.5

Para simplificar la escritura de la extensión $\overline{F}$ de una sustitución F, se adopta la convención de referirse a dicha extensión como una sustitución y denotarla como F. También, si ϕ es una proposición y F es una sustitución $\{q_0 \mapsto \phi_0, \ldots, q_n \mapsto \phi_n\}$, entonces $F(\phi)$ puede escribirse como

$$\phi[q_0, \ldots, q_n := \phi_0, \ldots, \phi_n]\,.$$

Observe que esta notación permite definir sustituciones anónimamente, lo cual en ciertas ocasiones puede ser conveniente para escribir concisamente.

Ejercicios

1. Considere la sustitución $F = \{p \mapsto (p \equiv q), q \mapsto (r \to s), r \mapsto \mathit{false}\}$. Proponga tres formas distintas de escribir F, sin tener en cuenta el orden en que se escriben los elementos de la sustitución.
2. Considere la sustitución $F = \{p \mapsto (p \equiv q), q \mapsto (r \to s), r \mapsto \mathit{false}\}$. Determine la proposición correspondiente a la sustitución textual de F en cada una de las siguientes proposiciones:
   a) p
   b) $(p \equiv r)$
   c) $((p \wedge (\neg q)) \to r)$
   d) $((p \wedge q) \vee ((\neg p) \wedge (\neg q)))$
   e) $(p \to (q \to p))$
   f) $((p \vee r) \leftarrow (p \wedge q))$
   g) $(\neg((r \wedge (r \leftarrow (p \vee s))) \equiv (\neg((p \to q) \vee (r \wedge (\neg r))))))$
3. Para cada una de las siguientes proposiciones encuentre una sustitución F tal que la proposición resultante de la sustitución textual bajo F sea una tautología:
   a) p
   b) $(p \equiv r)$
   c) $((p \wedge (\neg q)) \to r)$
   d) $((p \wedge q) \vee ((\neg p) \wedge (\neg q)))$
   e) $(p \to (q \to p))$
   f) $((p \vee r) \leftarrow (p \wedge q))$
   g) $(\neg((r \wedge (r \leftarrow (p \vee s))) \equiv (\neg((p \to q) \vee (r \wedge (\neg r))))))$
4. Demuestre el Metateorema 3.4.
5. Para cada uno de los siguientes casos encuentre proposiciones concretas ϕ, ψ, τ tales que:
   a) $(\phi[q := \tau])[p := \psi] \neq (\phi[p := \psi])[q := \tau]$.
   b) $\phi[p, q := \psi, \tau] \neq (\phi[p := \psi])[q := \tau]$.
6. Sean p, q, r variables proposicionales distintas y ϕ, ψ, τ proposiciones tales que r no aparece en ϕ ni en ψ. Demuestre que si $\gamma = \psi[q := r]$, entonces:

   $$\phi[p, q := \psi, \tau] = \phi[p := \gamma][q := \tau][r := q]\,.$$

7. Sea ϕ una proposición y F una sustitución. Demuestre que cualquier sustitución textual $F(\phi)$ puede ser escrita como $F_k(\cdots F_0(\phi)\cdots)$ en donde cada uno de los F_i $(0 \leq i \leq k)$ es una sustitución igual a la identidad excepto por un elemento del dominio. Ayuda: proceda por inducción matemática sobre la cantidad de variables proposicionales p tales que $F(p) \neq p$, el cual es finito por definición, y use el Ejercicio 6.

8. Sea φ una proposición y F una sustitución tal que: (i) F es un renombramiento de variables (i.e. $F : \mathcal{V} \to \mathcal{V}$) y (ii) F es inyectiva (i.e., $F(p) = F(q)$ si $p = q$, para cualesquiera variables $p, q \in \mathcal{V}$). Demuestre:
   a) φ es satisfacible sii $F(\varphi)$ es satisfacible.
   b) φ es una tautología sii $F(\varphi)$ es una tautología.
   c) φ es una contradicción sii $F(\varphi)$ es una contradicción.

3.2. Instanciación de variables proposicionales

La instanciación proposicional consiste en la sustitución textual de una variable proposicional en una proposición. Como tal, la instanciación proposicional preserva algunas propiedades de la proposición original como, por ejemplo, el hecho de ser tautología. Por ello, la instanciación proposicional puede ser una herramienta útil en la tarea, por ejemplo, de analizar semánticamente las proposiciones.

Antes de presentar el resultado principal de la sección es necesario presentar un resultado técnico que relaciona valuaciones y sustituciones. El Lema 3.6 presenta este resultado que consiste, intuitivamente, en establecer que el efecto semántico de una sustitución textual sobre una proposición puede ser capturado también por medio de una valuación sin alterar la proposición dada, y viceversa.

Lema 3.6

Sean p una variable proposicional, ϕ, ψ proposiciones y $\mathbf{v}$ una valuación. Hay una valuación $\mathbf{w}$ tal que $\mathbf{v}(\phi[p := \psi]) = \mathbf{w}(\phi)$.

Demostración. Considere la valuación $\mathbf{w}$ definida para cualquier variable proposicional q de la siguiente manera:

$$\mathbf{w}(q) = \begin{cases} \mathbf{v}(\psi) & \text{, si } q = p \\ \mathbf{v}(q) & \text{, si } q \neq p. \end{cases}$$

Considere la siguiente propiedad S sobre proposiciones:

$$S(\phi) : \mathbf{v}(\phi[p := \psi]) = \mathbf{w}(\phi).$$

Se procede a demostrar $S(\phi)$ por inducción sobre la complejidad de ϕ.

Caso base: Si ϕ es una variable proposicional, esta puede ser p o no serlo. Si ϕ es p note que $\mathbf{v}(p[p := \psi]) = \mathbf{v}(\psi) = \mathbf{w}(p)$. Si ϕ no es p, por ejemplo q, note que $\mathbf{v}(q[p := \psi]) = \mathbf{v}(q) = \mathbf{w}(q)$. Los casos en que ϕ es *true* o *false* se proponen como ejercicio para el lector.

Caso inductivo: Suponga que ϕ_0 y ϕ_1 tienen la propiedad S (i.e., ϕ_0 y ϕ_1 son tales que $\mathbf{v}(\phi_0[p := \psi]) = \mathbf{w}(\phi_0)$ y $\mathbf{v}(\phi_1[p := \psi]) = \mathbf{w}(\phi_1)$).

- Si ϕ es de la forma $(\neg\phi_0)$:

$$\begin{aligned}
\mathbf{v}((\neg\phi_0)[p := \psi]) &= \mathbf{v}((\neg\phi_0[p := \psi])) && \text{(sustitución textual)}\\
&= H_{\neg}(\mathbf{v}(\phi_0[p := \psi])) && \text{(aplicación } \mathbf{v})\\
&= H_{\neg}(\mathbf{w}(\phi_0)) && \text{(hipótesis inductiva)}\\
&= \mathbf{w}((\neg\phi_0)) && \text{(aplicación de } \mathbf{w}).
\end{aligned}$$

- Si ϕ es de la forma $(\phi_0 \otimes \phi_1)$ para $\otimes \in \{\equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$:

$$\begin{aligned}
\mathbf{v}((\phi_0 \otimes \phi_1)[p := \psi]) &= \mathbf{v}((\phi_0[p := \psi] \otimes \phi_1[p := \psi])) && \text{(sustitución textual)}\\
&= H_{\otimes}(\mathbf{v}(\phi_0[p := \psi]), \mathbf{v}(\phi_1[p := \psi])) && \text{(aplicación de } \mathbf{v})\\
&= H_{\otimes}(\mathbf{w}(\phi_0), \mathbf{w}(\phi_1)) && \text{(hipótesis inductivas)}\\
&= \mathbf{w}((\phi_0 \otimes \phi_1)) && \text{(aplicación de } \mathbf{w}).
\end{aligned}$$

Por el principio de inducción matemática para proposiciones (Metateorema 1.7) se concluye que cualquier proposición ϕ tiene la propiedad S. Luego, la valuación $\mathbf{w}$ es tal que satisface la condición del enunciado. □

Una propiedad interesante de la instanciación proposicional es que preserva la validez proposicional, es decir, cualquier instancia de una tautología también es una tautología. Esta propiedad semántica de la lógica proposicional se presenta en el Metateorema 3.7.

Metateorema 3.7

Sean p una variable proposicional y ϕ, ψ proposiciones. Si $\models \phi$, entonces $\models \phi[p := \psi]$.

Demostración. Suponga $\models \phi$, i.e., ϕ es una tautología. Si $\phi[p := \psi]$ no es una tautología, entonces hay una valuación $\mathbf{v}$ tal que $\mathbf{v}(\phi[p := \psi]) = \mathtt{F}$. Por el Lema 3.6, hay una valuación $\mathbf{w}$ tal que $\mathbf{w}(\phi) = \mathtt{F}$, pero esto no es posible porque se supuso que ϕ es tautología. Por tanto, no hay tal valuación $\mathbf{v}$ que falsifique a $\phi[p := \psi]$ cuando ϕ es tautología, i.e., $\phi[p := \psi]$ es una tautología cuando ϕ lo es. □

El Metateorema 3.7 brinda la posibilidad de analizar cualquier proposición por medio de tablas de verdad, algo que hasta este momento no es posible. En particular, para saber si una proposición τ es tautología, basta con encontrar una proposición ϕ *concreta* (i.e., que mencione únicamente variables proposicionales y conectivos lógicos) tal que τ sea $\phi[p := \psi]$ (i.e., $\tau = \phi[p := \psi]$) para alguna proposición ψ. Dado que ϕ es una proposición concreta, es posible analizarla, por ejemplo, con tablas de verdad.

Ejemplo 3.2

Considere la proposición $(\phi \vee (\neg\phi))$. Como $(p \vee (\neg p))$ es tautología y se tiene que $(\phi \vee (\neg\phi)) = (p \vee (\neg p))[p := \phi]$, por el Metateorema 3.7 se concluye $\models (\phi \vee (\neg\phi))$.

El Metateorema 3.7 puede ser utilizado una y otra vez sobre una misma proposición, como se ilustra en el Ejemplo 3.3.

Ejemplo 3.3

Considere la proposición $((\phi \to \psi) \equiv ((\neg\phi) \vee \psi))$. En este ejemplo se establece que esta proposición es una tautología, de forma alternativa a lo presentado en el Ejemplo 2.5. Tome dos variables proposicionales p y q tales que: q sea distinta a p y no aparezca en ϕ ni en ψ. Note entonces que

$$((\phi \to \psi) \equiv ((\neg\phi) \vee \psi)) = ((p \to q) \equiv ((\neg p) \vee q))[p := \phi][q := \psi]\,.$$

Por los ejemplos 2.2 y 2.4 se sabe que

$$\models ((p \to q) \equiv ((\neg p) \vee q)).$$

Consecuentemente, por el Metateorema 3.7

$$\models ((p \to q) \equiv ((\neg p) \vee q))[p := \phi]$$

y finalmente

$$\models ((p \to q) \equiv ((\neg p) \vee q))[p := \phi][q := \psi]\,.$$

Es decir, $\models ((\phi \to \psi) \equiv ((\neg\phi) \vee \psi))$.

Cuando se usa el Metateorema 3.7 una y otra vez sobre una misma proposición, es importante ser cuidadoso al escoger las variables que se instancian porque es posible cometer errores muy fácilmente con las sustituciones textuales (ver Ejercicio 3.2.8).

Ejercicios

1. Proponga una proposición ϕ tal que $\phi[p := \psi] = \phi$ para cualesquiera variable proposicional p y proposición ψ.
2. Complete el caso base en la demostración del Lema 3.6 para los casos en que ϕ sea *true* y sea *false*.
3. Justifique por qué $\models (p \vee (\neg p))$.

4. Sean p una variable proposicional, ϕ, ψ proposiciones y Γ un conjunto de proposiciones. Demuestre o refute:
   a) Si $\Gamma \models \phi$, entonces $\Gamma \models \phi[p := \psi]$.
   b) Si $\Gamma \models \phi$, entonces $\Delta \models \phi[p := \psi]$, en donde $\Delta = \{\gamma[p := \psi] \mid \gamma \in \Gamma\}$.
5. Sean p una variable proposicional y ϕ, ψ proposiciones. Demuestre o refute:
   a) Si ϕ es satisfacible, entonces $\phi[p := \psi]$ es satisfacible.
   b) Si ϕ es insatisfacible, entonces $\phi[p := \psi]$ es insatisfacible.
   c) Si $\phi[p := \psi]$ es satisfacible, entonces ϕ es satisfacible.
   d) Si $\phi[p := \psi]$ es insatisfacible, entonces ϕ es insatisfacible.
6. Sean $p_0, \ldots, p_n$ variables proposicionales, $\phi, \psi_0, \ldots, \psi_n$ proposiciones y $\mathbf{v}$ una valuación. Demuestre que hay una valuación $\mathbf{w}$ tal que

   $$\mathbf{v}(\phi[p_0, \ldots, p_n := \psi_0, \ldots, \psi_n]) = \mathbf{w}(\psi).$$

   Ayuda: use el Ejercicio 3.1.6.
7. Sean $p_0, \ldots, p_n$ variables proposicionales, $\phi, \psi_0, \ldots, \psi_n$ proposiciones. Demuestre: si $\models \phi$, entonces $\models \phi[p_0, \ldots, p_n := \psi_0, \ldots, \psi_n]$. Ayuda: use el Ejercicio 6.
8. Considere la siguiente afirmación hecha en el Ejemplo 3.3:

   > Tome dos variables proposicionales p y q tales que: q sea distinta a p y esta no aparezca en ϕ ni en ψ.

   Con base en esta afirmación:
   a) Explique por qué es posible encontrar variables proposicionales p y q bajo las condiciones dadas.
   b) Suponga que p y q son tales que satisfacen las condiciones en la afirmación anterior, excepto que q puede aparecer en ϕ o en ψ. Explique por qué, en cualquiera de estos casos, la siguiente igualdad puede fallar:

   $$((\phi \to \psi) \equiv ((\neg\phi) \vee \psi)) = ((p \to q) \equiv ((\neg p) \vee q))[p := \phi][q := \psi]\,.$$

3.3. Reemplazo de 'iguales por iguales'

Esta sección presenta dos nociones complementarias de reemplazo de 'iguales por iguales' para la lógica proposicional, como herramientas para analizar el significado de las proposiciones. En la ciencia, la *igualdad* (palabra que proviene del griego *equālis*) es una relación entre dos cantidades o, de manera más abstracta, entre dos expresiones afirmando que estas representan el mismo objeto. Visto de manera más general, la noción de igualdad puede ser caracterizada como lo propone Leibniz al afirmar que dos expresiones son iguales siempre y cuando estas tengan exactamente las mismas propiedades. Esta última noción permite, por ejemplo, que dos expresiones sintácticamente distintas puedan ser 'iguales' de otras formas. En la lógica proposicional, dado que las expresiones son proposiciones y la relación de

igualdad es la equivalencia, el reemplazo de 'iguales por iguales' está directamente relacionado con la noción de equivalencia lógica: dos proposiciones son lógicamente equivalentes siempre y cuando una pueda ser sustituida por la otra en una tercera proposición sin alterar el valor de verdad de dicha proposición.

3.3.1. Ecuanimidad. La primera forma de reemplazar 'iguales por iguales' está basada en el hecho de que cuando una proposición es verdadera y esta es indistinguible (con respecto a la igualdad lógica) de una segunda proposición, entonces debe ser correcto afirmar que esta segunda proposición también es cierta. El Metateorema 3.8 presenta esta primera noción de reemplazo de 'iguales por iguales' en el caso general cuando la igualdad entre proposiciones es relativa a un conjunto de proposiciones dado.

Metateorema 3.8

Sean Γ un conjunto de proposiciones y ϕ, ψ proposiciones. Si $\Gamma \models \psi$ y $\Gamma \models (\psi \equiv \phi)$, entonces $\Gamma \models \phi$.

Demostración. Suponga $\Gamma \models \psi$ y $\Gamma \models (\psi \equiv \phi)$. Sea $\mathbf{v}$ una valuación. Si $\mathbf{v}$ no satisface Γ, entonces la demostración es trivialmente cierta. Ahora, si $\mathbf{v}$ satisface Γ, entonces se debe demostrar $\mathbf{v}(\phi) = \mathtt{T}$. Si $\mathbf{v}$ satisface Γ, se tiene que $\mathbf{v}(\psi) = \mathtt{T}$ y $\mathbf{v}((\psi \equiv \phi)) = \mathtt{T}$ por las suposiciones. Note que $\mathbf{v}((\psi \equiv \phi)) = \mathtt{T}$ sii $\mathbf{v}(\psi) = \mathbf{v}(\phi)$ y como $\mathbf{v}(\psi) = \mathtt{T}$, se concluye $\mathbf{v}(\phi) = \mathtt{T}$. Luego, ϕ es consecuencia tautológica de Γ bajo las suposiciones dadas. □

A continuación se presenta un ejemplo ilustrando el uso del Metateorema 3.8.

Ejemplo 3.4

Considere dos proposiciones ϕ y ψ. Note que:

1. $\{(\neg\phi)\} \models ((\neg\phi) \vee \psi)$.
2. $\models (((\neg\phi) \vee \psi) \equiv (\phi \to \psi))$.

De (2) y del Metateorema 2.34.2, se sigue $\{(\neg\phi)\} \models (((\neg\phi) \vee \psi) \equiv (\phi \to \psi))$. Entonces, del Metateorema 3.8 se sigue $\{(\neg\phi)\} \models (\phi \to \psi)$ lo cual indica, intuitivamente, que cualquier implicación es cierta cuando su antecedente no lo es.

En el caso particular cuando Γ es el conjunto vacío en el Metateorema 3.8, la noción de consecuencia tautológica se especializa a aquella de tautología. Note que si una proposición es una tautología, entonces cualquier proposición lógicamente equivalente a ella necesariamente es una tautología.

Corolario 3.9

Sean ϕ y ψ proposiciones. Si $\models \psi$ y $\models (\psi \equiv \phi)$, entonces $\models \phi$.

Demostración. La demostración se propone como ejercicio para el lector. □

3.3.2. Leibniz. La segunda noción de reemplazo de 'iguales por iguales' está directamente relacionada con el planteamiento de Leibniz acerca de la igualdad resumido al inicio de esta sección. Esta noción consiste en afirmar que si dos proposiciones son iguales lógicamente, entonces cada una de ellas puede ser sustituida en una tercera proposición resultando en dos proposiciones que son iguales lógicamente. El Metateorema 3.10 presenta esta segunda noción de reemplazo de 'iguales por iguales'.

Metateorema 3.10

Sean Γ un conjunto de proposiciones, ϕ, ψ, τ proposiciones y p una variable proposicional. Si $\Gamma \models (\psi \equiv \tau)$, entonces $\Gamma \models (\phi[p := \psi] \equiv \phi[p := \tau])$.

Demostración. Suponga $\Gamma \models (\psi \equiv \tau)$. Sea $\mathbf{v}$ una valuación y considere la siguiente propiedad S sobre proposiciones ϕ:

$$S(\phi) : \mathbf{v}(\phi[p := \psi]) = \mathbf{v}(\phi[p := \tau]).$$

Suponiendo que $\mathbf{v}$ satisface Γ, se procede a demostrar $S(\phi)$ por inducción sobre la complejidad de ϕ.

Caso base: Se propone como ejercicio para el lector.

Caso inductivo: Suponga que ϕ_0 y ϕ_1 tienen la propiedad S.

- Si ϕ es de la forma $(\neg\phi_0)[p := \psi]$:

$$\begin{aligned} \mathbf{v}((\neg\phi_0)[p := \psi]) &= H_\neg(\mathbf{v}(\phi_0[p := \psi])) && \text{(aplicación de } \mathbf{v}) \\ &= H_\neg(\mathbf{v}(\phi_0[p := \tau])) && \text{(hipótesis inductiva)} \\ &= \mathbf{v}((\neg\phi_0)[p := \tau]) && \text{(aplicación de } \mathbf{v}). \end{aligned}$$

- El caso en que ϕ es de la forma $(\phi_0 \otimes \phi_1)$, con $\otimes \in \{\equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$, se propone como ejercicio para el lector.

Por el principio de inducción matemática para proposiciones (Metateorema 1.7) se concluye que cualquier proposición ϕ tiene la propiedad S. Luego, si $\Gamma \models (\psi \equiv \tau)$, entonces $\Gamma \models (\phi[p := \psi] \equiv \phi[p := \tau])$. □

A continuación se ilustra el uso del Metateorema 3.10 con un ejemplo.

Ejemplo 3.5

Considere dos proposiciones ψ y τ. Se desea demostrar que si $\models (\psi \equiv \tau)$, entonces $\models ((\neg\psi) \equiv (\neg\tau))$. Esta conclusión se obtiene directamente del Metateorema 3.10 tomando $\Gamma = \{\}$ y $\phi = (\neg p)$ para alguna variable proposicional p: si $\models (\psi \equiv \tau)$, entonces $\models ((\neg p)[p := \psi] \equiv (\neg p)[p := \tau])$ o, alternativamente, $\models ((\neg\psi) \equiv (\neg\tau))$.

El Corolario 3.11 presenta la noción de reemplazo de 'iguales por iguales' de Leibniz en el caso cuando el conjunto de suposiciones Γ en 3.10 es vacío.

Corolario 3.11

Sean ϕ, ψ, τ proposiciones y p una variable proposicional. Si $\models (\psi \equiv \tau)$, entonces $\models (\phi[p := \psi] \equiv \phi[p := \tau])$.

Demostración. La demostración se propone como ejercicio para el lector. □

Ejercicios

1. Justifique por qué las afirmaciones (1) y (2) en el Ejemplo 3.4 son correctas.
2. Demuestre el Corolario 3.9 usando el Metateorema 3.8 como resultado auxiliar.
3. Demuestre el Corolario 3.9 sin usar el Metateorema 3.8 como resultado auxiliar.
4. Complete el caso base en la demostración del Metateorema 3.10.
5. Complete el caso inductivo en la demostración del Metateorema 3.10.
6. Demuestre el Corolario 3.11 usando el Metateorema 3.10 como resultado auxiliar.
7. Demuestre el Corolario 3.11 sin usar el Metateorema 3.10 como resultado auxiliar.
8. Sean Γ un conjunto de proposiciones, $\phi, \psi_0, \ldots, \psi_n, \tau_0, \ldots, \tau_n$ proposiciones y $p_0, \ldots, p_n$ variables proposicionales. Si $\Gamma \models (\psi_i \equiv \tau_i)$ para $0 \leq i \leq n$, entonces
$$\Gamma \models (\phi[p_0, \ldots, p_n := \psi_0, \ldots, \psi_n] \equiv \phi[p_0, \ldots, p_n := \tau_0, \ldots, \tau_n]).$$
9. Sean p una variable proposicional y ϕ, ψ, τ proposiciones. Demuestre o refute: si $\models (\phi[p := \psi] \equiv \phi[p := \tau])$, entonces $\models (\psi \equiv \tau)$.

Capítulo 4

El sistema de Dijkstra y Scholten para proposiciones

> La única forma de rectificar nuestros razonamientos es haciéndolos tan tangibles como aquellos de los Matemáticos, de forma tal que sea posible encontrar nuestros errores cuanto antes y cuando haya disputas entre personas podamos simplemente decir: *calculemus*, sin mayor reparo, para determinar quién está en lo correcto.
>
> G. W. Leibniz
> *The Art of Discovery*, 1685

Gottfried W. von Leibniz (1646-1716) tenía el sueño de contar con un lenguaje universal y formal, capaz de expresar conceptos matemáticos, científicos y metafísicos. Su idea era utilizar este lenguaje, denominado *caracteristica universalis*, dentro del marco de un cálculo lógico específicamente diseñado para efectuar *cualquier* inferencia lógica. Esta visión formal y mecánica del razonamiento, en la cual era posible "calcular" cualquier teorema de las matemáticas, solo se conoció a finales del siglo XIX y fue desvirtuada (i.e., es imposible que dicho cálculo exista) a mediados del siglo XX por Kurt F. Gödel (1906-1978), con su famoso "segundo teorema de completitud". A pesar de ello, el objetivo de Leibniz de "algebrizar" el razonamiento es considerado uno de los principales pilares de la lógica moderna y de la computación.

Históricamente, la lógica matemática ha sido vista como un objeto de estudio y sus bondades como álgebra del razonamiento han sido relegadas a un segundo plano por los matemáticos. Esto no es sorprendente porque las matemáticas avanzaron durante siglos sin la ayuda de la lógica matemática y, además, esta es un

área del conocimiento relativamente nueva cuando se compara con el resto de las matemáticas. Sin embargo, esta situación ha cambiado en la era digital en la cual la tecnología ha permeado el quehacer de la humanidad y ahora es posible hacer, por ejemplo, demostraciones asistidas o automáticas en un computador. Para hacer esto posible y principalmente desde la informática, se han desarrollado nuevas lógicas que pueden ser mecanizadas en un computador. Es así como el sueño de Leibniz, a pesar de que nunca se podrá llevar a cabo en su totalidad, tiene más relevancia que nunca en nuestra época.

Este capítulo presenta el *sistema proposicional de Dijkstra y Scholten* (o *sistema formal* DS), un sistema formal para la lógica proposicional diseñado por los científicos holandeses Edsger W. Dijkstra (1930-2002) y Carel S. Scholten (1925-2009) para potenciar el razonamiento basado en la equivalencia lógica. Este sistema lógico, junto con su versión más general de primer orden, ha sido utilizado extensivamente en el diseño y en la verificación de algoritmos durante las últimas cuatro décadas, y su mecanización es un tema activo de investigación. El razonamiento basado en la equivalencia lógica, en contraste con el tradicional basado en la implicación lógica, y un formato propio de deducción hacen de DS un sistema efectivo y versátil en la práctica de la demostración (i.e., cálculo) de teoremas. Este sistema lógico es un aporte de la informática y su diseño basado en la equivalencia, junto con su formato de demostración, son suficientes para distinguirlo de cualquier otro sistema deductivo de la lógica proposicional.

4.1. El sistema formal DS

El lenguaje formal del sistema DS corresponde al lenguaje de la lógica proposicional propuesto en la Definición 1.2 (Sección 1.2). Este lenguaje incluye una cantidad infinita de variables proposicionales $p_0, p_1, \ldots$, usa paréntesis para puntuar expresiones y cuenta con nueve conectivos lógicos: dos constantes (*true* y *false*), un conectivo unario ($\neg$) y seis conectivos binarios ($\equiv$, $\not\equiv$, $\vee$, $\wedge$, $\rightarrow$ y $\leftarrow$). Las fórmulas del sistema DS son llamadas proposiciones y siguen la Definición 1.4 (Sección 1.2) o, equivalentemente, la siguiente BNF:

$$\begin{aligned}\phi ::= \; & p \mid \mathit{true} \mid \mathit{false} \mid (\neg\phi) \mid (\phi \equiv \phi) \mid (\phi \not\equiv \phi) \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \\ & \mid (\phi \rightarrow \phi) \mid (\phi \leftarrow \phi)\end{aligned}$$

en donde p representa cualquier variable proposicional y ϕ cualquier proposición.

Nota 4.1

Recuerde, de la Definición 1.4, que la expresión $\mathcal{V}$ denota el conjunto de variables proposicionales de DS y la expresión $\mathcal{T}(\mathcal{V})$ el conjunto de proposiciones de DS. También recuerde que $\mathcal{V}$ es un subconjunto propio de $\mathcal{T}(\mathcal{V})$, escrito $\mathcal{V} \subsetneq \mathcal{T}(\mathcal{V})$.

A continuación se define el conjunto de axiomas de DS.

Definición 4.2

Sean ϕ, ψ, τ proposiciones de DS. El *conjunto de axiomas* de DS está dado por el siguiente esquema axiomático:

$(Ax1)$**:** $((\phi \equiv (\psi \equiv \tau)) \equiv ((\phi \equiv \psi) \equiv \tau))$

$(Ax2)$**:** $((\phi \equiv \psi) \equiv (\psi \equiv \phi))$

$(Ax3)$**:** $((\phi \equiv \mathit{true}) \equiv \phi)$

$(Ax4)$**:** $((\phi \vee (\psi \vee \tau)) \equiv ((\phi \vee \psi) \vee \tau))$

$(Ax5)$**:** $((\phi \vee \psi) \equiv (\psi \vee \phi))$

$(Ax6)$**:** $((\phi \vee \mathit{false}) \equiv \phi)$

$(Ax7)$**:** $((\phi \vee \phi) \equiv \phi)$

$(Ax8)$**:** $((\phi \vee (\psi \equiv \tau)) \equiv ((\phi \vee \psi) \equiv (\phi \vee \tau)))$

De acuerdo con la Definición 4.2, la equivalencia es asociativa ($Ax1$), conmutativa ($Ax2$) y tiene elemento identidad *true* ($Ax3$). Análogamente, la disyunción es asociativa ($Ax4$), conmutativa ($Ax5$) y tiene elemento identidad *false* ($Ax6$). Además, la disyunción es idempotente ($Ax7$) y distribuye sobre la equivalencia ($Ax8$). La definición de los axiomas de DS omite axiomas que mencionen explícitamente negaciones, discrepancias, conjunciones, implicaciones y consecuencias. Esto es algo deliberado: como se aprenderá en las siguientes secciones, estos operadores pueden "definirse" usando los operadores *true*, *false*, $\equiv$ y $\vee$, los únicos hasta ahora mecionados por los axiomas de DS.

Nota 4.3

Note que los axiomas ($Ax1$-$Ax8$) involucran proposiciones arbitrarias, es decir, cada uno de ellos representa una cantidad infinita de axiomas concretos. Este enfoque es

equivalente a presentar una definición usando únicamente tres variables proposicionales, por ejemplo p, q, r, en lugar de ϕ, ψ, τ para luego extender dicha definición permitiendo que cualquier instancia bajo sustitución textual de un axioma sea un axioma.

A continuación se define el conjunto de reglas de inferencia de DS.

Definición 4.4

Sean p una variable proposicional y ϕ, ψ, τ proposiciones. Las *reglas de inferencia* de DS son:

$$\frac{\psi \qquad (\psi \equiv \phi)}{\phi}\ \textsc{Ecuanimidad} \qquad \frac{(\psi \equiv \tau)}{(\phi[p := \psi] \equiv \phi[p := \tau])}\ \textsc{Leibniz}.$$

El sistema DS cuenta con dos reglas de inferencia. La regla Ecuanimidad representa el hecho de que si hay un teorema en DS, digamos ψ, y se puede demostrar en DS que ψ es equivalente a ϕ, entonces necesariamente ϕ es un teorema de DS. La regla Leibniz determina que al hacer cambio de iguales por iguales en una proposición, las proposiciones resultantes son equivalentes. Para indicar el cambio de iguales por iguales, la regla Leibniz usa el concepto de sustitución textual (Definición 3.3). En el caso de esta regla, fíjese en que la variable proposicional p puede no ser parte de ϕ: en ese caso cada una de las fórmulas $\phi[p := \psi]$ y $\phi[p := \tau]$ en la conclusión de la regla son trivialmente ϕ.

Habiendo completado la definición del sistema DS, la definición de demostración en DS y de la relación $\vdash_{\mathsf{DS}}$ de demostrabiliad en DS son heredadas automáticamente de la Definición 0.10 (Sección 0.3).

Nota 4.5

Una *demostración* de una proposición ϕ en DS es una secuencia no vacía de proposiciones tales que ϕ es la última proposición en la secuencia y para cualquier proposición ψ en la secuencia, al menos una de las siguientes dos condiciones es cierta: ψ es un axioma o es la conclusión de una regla de inferencia cuyas premisas

aparecen en la secuencia antes que ψ. Una proposición ϕ *es teorema de* DS, escrito $\vdash_{\mathsf{DS}} \phi$, sii hay una demostración de ϕ en DS.

A continuación se presentan algunos teoremas de DS relacionados con la equivalencia lógica.

Teorema 4.6

Para cualquier proposición ϕ:

1. $\vdash_{\mathsf{DS}} \mathit{true}$.
2. $\vdash_{\mathsf{DS}} ((\phi \equiv \phi) \equiv \mathit{true})$.
3. $\vdash_{\mathsf{DS}} (\phi \equiv \phi)$.

Demostración. A continuación se presenta una demostración para (1):

1.	$(((\mathit{true} \equiv \mathit{true}) \equiv \mathit{true}) \equiv (\mathit{true} \equiv \mathit{true}))$	$(Ax3)$
2.	$((\mathit{true} \equiv \mathit{true}) \equiv \mathit{true})$	$(Ax3)$
3.	$(\mathit{true} \equiv \mathit{true})$	(Ecuanimidad 2 y 1)
4.	true	(Ecuanimidad 3 y 2).

A continuación se presenta una demostración para (2):

1.	$((\phi \equiv \mathit{true}) \equiv \phi)$	$(Ax3)$
2.	$(((\phi \equiv \mathit{true}) \equiv \phi) \equiv (\phi \equiv (\phi \equiv \mathit{true})))$	$(Ax2)$
3.	$(\phi \equiv (\phi \equiv \mathit{true}))$	(Ecuanimidad 1 y 2)
4.	$((\phi \equiv (\phi \equiv \mathit{true})) \equiv ((\phi \equiv \phi) \equiv \mathit{true}))$	$(Ax1)$
5.	$((\phi \equiv \phi) \equiv \mathit{true})$	(Ecuanimidad 3 y 4).

La demostración de (3) se propone como ejercicio para el lector. □

El Teorema 4.6.1 establece un hecho altamente deseable de un sistema lógico: que la verdad es teorema. El Teorema 4.6.2, visto de izquierda a derecha, indica que cualquier equivalencia cuyos operandos sean iguales puede ser reducida a la expresión *true*. Finalmente, el Teorema 4.6.3 indica que la equivalencia es reflexiva.

Ejercicios

1. Demuestre el Teorema 4.6.3 (ayuda: use las demostraciones de los teoremas 4.6.1 y 4.6.2).
2. Demuestre que cada uno de los axiomas de DS es una tautología. Es decir, si ϕ es un axioma de DS, entonces $\models \phi$.
3. Demuestre que el Teorema 4.6.1 es una tautología, es decir, $\models \mathit{true}$.
4. Demuestre que el Teorema 4.6.2 es una tautología, es decir, $\models ((\phi \equiv \phi) \equiv \mathit{true})$.
5. Demuestre que el Teorema 4.6.3 es una tautología, es decir, $\models (\phi \equiv \phi)$.
6. Encuentre proposiciones concretas ϕ, ψ, τ para las cuales $\models (\phi[p := \psi] \equiv \phi[p := \tau])$ pero no $\models (\psi \equiv \tau)$.

4.2. Propiedades estructurales de la equivalencia

El sistema DS basa su poder deductivo, en gran medida, en las propiedades de la equivalencia que pueden ser usadas para aplicar las reglas de inferencia Ecuanimidad y Leibniz. La equivalencia lógica cuenta con otras propiedades que pueden ser útiles como mecanismo deductivo en DS. Esta sección introduce nuevas reglas de inferencia para DS, todas ellas relacionadas con propiedades deductivas de la equivalencia, las cuales se pueden "derivar" de los axiomas y las reglas de inferencia de DS. En este sentido, dado que las nuevas reglas de inferencia pueden ser derivadas, estas no agregan nuevo poder deductivo a DS; es decir, estas nuevas reglas de inferencia pueden verse como "macros" que pueden ahorrar esfuerzo en la tarea de demostración de teoremas de DS. En consecuencia, el conjunto de teoremas de DS no cambia al agregar las reglas de inferencia presentadas en esta sección.

Inicialmente se establecen algunas propiedades que permiten usar las reglas de inferencia de DS con más libertad.

Metateorema 4.7

Sean ϕ, ψ, τ proposiciones. Las siguientes afirmaciones sobre DS son ciertas:

1. Si $\vdash_{\mathsf{DS}} \psi$ y $\vdash_{\mathsf{DS}} (\phi \equiv \psi)$, entonces $\vdash_{\mathsf{DS}} \phi$.
2. Si $\vdash_{\mathsf{DS}} (\tau \equiv \psi)$, entonces $\vdash_{\mathsf{DS}} (\phi[p := \psi] \equiv \phi[p := \tau])$.

Demostración. A continuación se presenta una demostración de (1); establecer la propiedad (2) se propone como ejercicio para el lector. Suponga que $\psi_0, \ldots, \psi_n$ es una demostración de ψ y que $\gamma_0, \ldots, \gamma_m$ es una demostración de $(\phi \equiv \psi)$. Note

que $\psi_n = \psi$ y $\gamma_m = (\phi \equiv \psi)$. El objetivo es encontrar una demostración de ϕ en DS. La idea es combinar las demostraciones de ψ y $(\phi \equiv \psi)$ de la siguiente manera:

$0.$	ψ_0	$(\cdots)$
$\vdots$	$\vdots$	$\vdots$
$n.$	ψ	$(\cdots)$
$n+1.$	γ_0	$(\cdots)$
$\vdots$	$\vdots$	$\vdots$
$n+m.$	$(\phi \equiv \psi)$	$(\cdots)$
$n+m+1.$	$((\phi \equiv \psi) \equiv (\psi \equiv \phi))$	$(Ax2)$
$n+m+2.$	$(\psi \equiv \phi)$	(Ecuanimidad $n+m$ y $n+m+1$)
$n+m+3.$	ϕ	(Ecuanimidad n y $n+m+2$).

Se concluye que si $\vdash_{\mathsf{DS}} \psi$ y $\vdash_{\mathsf{DS}} (\phi \equiv \psi)$, entonces $\vdash_{\mathsf{DS}} \phi$. □

Nota 4.8

La expresión "corrección de una regla de inferencia" en el contexto de un sistema formal quiere decir que si las premisas de una regla de inferencia son teoremas, entonces su conclusión también debe ser teorema.

Nota 4.9

El Metateorema 4.7 justifica la *corrección* de las siguientes reglas de inferencia:

$$\frac{\psi \qquad (\phi \equiv \psi)}{\phi}\ \textsc{Ecuanimidad}^* \qquad \frac{(\tau \equiv \psi)}{(\phi[p := \psi] \equiv \phi[p := \tau])}\ \textsc{Leibniz}^*$$

A continuación se presentan las propiedades de *transitividad* e *identidad* de la equivalencia.

Metateorema 4.10

Sean ϕ, ψ, τ proposiciones. La siguientes afirmaciones sobre DS son ciertas:

1. Si $\vdash_{\mathsf{DS}} (\phi \equiv \psi)$ y $\vdash_{\mathsf{DS}} (\psi \equiv \tau)$, entonces $\vdash_{\mathsf{DS}} (\phi \equiv \tau)$.
2. $\vdash_{\mathsf{DS}} \phi$ sii $\vdash_{\mathsf{DS}} (\phi \equiv \mathit{true})$.

Demostración. Las demostraciones se proponen como ejercicio para el lector. □

Nota 4.11

El Metateorema 4.10 justifica la corrección de las siguientes reglas de inferencia:

$$\frac{(\phi \equiv \psi) \qquad (\psi \equiv \tau)}{(\phi \equiv \tau)}\ \textsc{Transitividad}$$

$$\frac{(\phi \equiv \mathit{true})}{\phi}\ \textsc{Identidad} \qquad\qquad \frac{\phi}{(\phi \equiv \mathit{true})}\ \textsc{Identidad}$$

Finalmente, el Metateorema 4.12 establece algunas propiedades *estructurales* de la equivalencia.

Metateorema 4.12

Sean ϕ, ψ, τ proposiciones. Las siguientes afirmaciones sobre DS son ciertas:

1. $\vdash_{\mathsf{DS}} (\phi \equiv (\psi \equiv \tau))$ si y solo si $\vdash_{\mathsf{DS}} ((\phi \equiv \psi) \equiv \tau)$.
2. $\vdash_{\mathsf{DS}} (\phi \equiv \psi)$ si y solo si $\vdash_{\mathsf{DS}} (\psi \equiv \phi)$.

Demostración. A continuación se presenta una demostración de (1); establecer la propiedad (2) se propone como ejercicio para el lector. La propiedad (1) en palabras:

> En DS, hay una demostración de $(\phi \equiv (\psi \equiv \tau))$ si y solo si hay una demostración de $((\phi \equiv \psi) \equiv \tau)$.

La demostración se obtiene por doble implicación.

- Suponga que hay una demostración $\phi_0, \ldots, \phi_n$ con $\phi_n = (\phi \equiv (\psi \equiv \tau))$. El objetivo es exhibir una demostración de $((\phi \equiv \psi) \equiv \tau)$:

0.	ϕ_0	$(\cdots)$
	$\cdots$	$\cdots$
n.	$(\phi \equiv (\psi \equiv \tau))$	$(\cdots)$
$n+1$.	$((\phi \equiv (\psi \equiv \tau)) \equiv ((\phi \equiv \psi) \equiv \tau))$	(Ax1)
$n+2$.	$((\phi \equiv \psi) \equiv \tau)$	(Ecuanimidad n y $n+1$).

- Suponga que hay una demostración $\psi_0, \ldots, \psi_n$ con $\psi_n = ((\phi \equiv \psi) \equiv \tau)$. El objetivo es exhibir una demostración de $(\phi \equiv (\psi \equiv \tau))$:

0.	ψ_0	$(\cdots)$
	$\cdots$	$\cdots$
n.	$((\phi \equiv \psi) \equiv \tau)$	$(\cdots)$
$n+1$.	$((\phi \equiv (\psi \equiv \tau)) \equiv ((\phi \equiv \psi) \equiv \tau))$	(Ax1)
$n+2$.	$(\phi \equiv (\psi \equiv \tau))$	(Ecuanimidad* n y $n+1$).

□

Nota 4.13

El Metateorema 4.12 justifica la corrección de las siguientes reglas de inferencia:

$$\frac{(\phi \equiv (\psi \equiv \tau))}{((\phi \equiv \psi) \equiv \tau)}\ \textsc{Asociatividad} \qquad \frac{((\phi \equiv \psi) \equiv \tau)}{(\phi \equiv (\psi \equiv \tau))}\ \textsc{Asociatividad}$$

$$\frac{(\phi \equiv \psi)}{(\psi \equiv \phi)}\ \textsc{Conmutatividad}$$

Las reglas de inferencia introducidas en esta sección ahora son parte de DS. Recuerde que estas reglas no le otorgan más poder deductivo a DS, simplemente facilitan la demostración de teoremas en el sistema formal.

Ejercicios

1. Demuestre el Metateorema 4.7.2.
2. Demuestre:
   a) El Metateorema 4.10.1.
   b) El Metateorema 4.10.2.
3. Demuestre el Metateorema 4.12.2.
4. Utilice las reglas de inferencia introducidas en esta sección para simplificar las demostraciones presentadas en la Sección 4.1.
5. Sean ϕ y ψ proposiciones. Demuestre o refute:
   a) Si $\vdash_{\mathsf{DS}} \phi$ y $\vdash_{\mathsf{DS}} \psi$, entonces $\vdash_{\mathsf{DS}} (\phi \equiv \psi)$.
   b) Si $\vdash_{\mathsf{DS}} (\phi \equiv \psi)$, entonces $\vdash_{\mathsf{DS}} \phi$ y $\vdash_{\mathsf{DS}} \psi$.
6. Sean ϕ y ψ proposiciones. Demuestre:

a) Si ϕ es un axioma, entonces $\vdash_{\mathsf{DS}} \phi[p := \psi]$.
b) Si $\vdash_{\mathsf{DS}} \phi$, entonces $\vdash_{\mathsf{DS}} \phi[p := \psi]$.

4.3. La negación y la discrepancia

La negación y la discrepancia son conectivos lógicos del sistema formal. En DS, los dos conectivos están relacionados estructuralmente por medio de la equivalencia. Esta sección presenta axiomas *definicionales* para la negación y la discrepancia, es decir, define estos símbolos con base en otros conectivos del sistema formal. Al final de esta sección se encuentra una aplicación del uso de la discrepancia en criptografía.

Definición 4.14

Sean ϕ y ψ proposiciones DS. Los siguientes axiomas de DS definen la negación y la discrepancia:

$(Ax9)$**:** $((\neg\phi) \equiv (\phi \equiv \mathit{false}))$

$(Ax10)$**:** $((\phi \not\equiv \psi) \equiv ((\neg\phi) \equiv \psi))$

La negación es por naturaleza el operador de complemento de la lógica. La discrepancia es el operador opuesto de la equivalencia. En DS se modela el complemento lógico de una proposición haciendo esta equivalente a *false* $(Ax9)$. La discrepancia, al ser el inverso de la equivalencia, se puede definir al negar uno de sus operandos $(Ax10)$. Como se verá más adelante en esta sección por medio de un teorema, no importa cuál de los dos operandos se niegue en la definición de $\not\equiv$ pues las dos proposiciones resultantes son equivalentes.

A continuación se presentan algunas propiedades de la negación.

Teorema 4.15

Para cualesquiera proposiciones ϕ y ψ de DS:

1. $\vdash_{\mathsf{DS}} (\mathit{false} \equiv (\neg \mathit{true}))$
2. $\vdash_{\mathsf{DS}} ((\neg \mathit{false}) \equiv \mathit{true})$
3. $\vdash_{\mathsf{DS}} (\neg \mathit{false})$
4. $\vdash_{\mathsf{DS}} ((\neg(\phi \equiv \psi)) \equiv ((\neg\phi) \equiv \psi))$
5. $\vdash_{\mathsf{DS}} (((\neg\phi) \equiv \psi) \equiv (\phi \equiv (\neg\psi)))$
6. $\vdash_{\mathsf{DS}} ((\neg(\neg\phi)) \equiv \phi)$

7. $\vdash_{\mathsf{DS}} ((\phi \equiv (\neg\phi)) \equiv \mathit{false})$

Las constantes *true* y *false* son opuestos el uno del otro (teoremas 4.15.1 y 4.15.2), y entonces, naturalmente, la negación de *false* es un teorema de DS (Teorema 4.15.3). La negación distribuye sobre la equivalencia (Teorema 4.15.4) y alterna entre los operandos de una equivalencia (Teorema 4.15.5). Este último teorema justifica la afirmación hecha previamente acerca de la definición de $\not\equiv$: desde el punto de vista deductivo resulta indiferente cuál de los operandos de la equivalencia sea negado. El Teorema 4.15.6 establece la propiedad de la doble negación: la negación se cancela con ella misma. Finalmente, el Teorema 4.15.7 caracteriza la contradicción lógica en términos de la equivalencia y la negación. A continuación se presentan demostraciones de algunos de estos teoremas.

Demostración. Para (1):

1.	$((\neg \mathit{true}) \equiv (\mathit{true} \equiv \mathit{false}))$	$(Ax9)$
2.	$(((\neg \mathit{true}) \equiv \mathit{true}) \equiv \mathit{false})$	(Asociatividad 1)
3.	$(\mathit{false} \equiv ((\neg \mathit{true}) \equiv \mathit{true}))$	(Conmutatividad 2)
4.	$(((\neg \mathit{true}) \equiv \mathit{true}) \equiv (\neg \mathit{true}))$	$(Ax3)$
5.	$(\mathit{false} \equiv (\neg \mathit{true}))$	(Transitividad 3,4).

Para (2):

1.	$((\neg \mathit{false}) \equiv (\mathit{false} \equiv \mathit{false}))$	$(Ax9)$
2.	$((\mathit{false} \equiv \mathit{false}) \equiv \mathit{true})$	(Teorema 4.6.2 con $\phi = \mathit{false}$)
3.	$((\neg \mathit{false}) \equiv \mathit{true})$	(Transitividad 1,2).

Para (4):

1.	$((\neg(\phi \equiv \psi)) \equiv ((\phi \equiv \psi) \equiv \mathit{false}))$	$(Ax9)$
2.	$((\phi \equiv \psi) \equiv (\psi \equiv \phi))$	$(Ax2)$
3.	$(((\phi \equiv \psi) \equiv \mathit{false}) \equiv ((\psi \equiv \phi) \equiv \mathit{false}))$	(Leibniz 2)
4.	$(((\psi \equiv \phi) \equiv \mathit{false}) \equiv (\psi \equiv (\phi \equiv \mathit{false})))$	$(Ax1)$
5.	$((\neg\phi) \equiv (\phi \equiv \mathit{false}))$	$(Ax9)$
6.	$((\psi \equiv (\phi \equiv \mathit{false})) \equiv (\psi \equiv (\neg\phi)))$	(Leibniz* 5)
7.	$((\psi \equiv (\neg\phi)) \equiv ((\neg\phi) \equiv \psi))$	$(Ax2)$
8.	$((\neg(\phi \equiv \psi)) \equiv ((\neg\phi) \equiv \psi))$	(Transitividad 1,3,4,6,7).

Finalmente, para (6):

1.	$((\neg(\neg\phi)) \equiv ((\neg\phi) \equiv \mathit{false}))$	$(Ax9)$
2.	$(((\neg\phi) \equiv \mathit{false}) \equiv (\phi \equiv (\neg\mathit{false})))$	(Teorema 4.15.5)
3.	$((\neg\mathit{false}) \equiv \mathit{true})$	(Teorema 4.15.2)
4.	$((\phi \equiv (\neg\mathit{false})) \equiv (\phi \equiv \mathit{true}))$	(Leibniz 3)
5.	$((\neg(\neg\phi)) \equiv (\phi \equiv \mathit{true}))$	(Transitividad 1,2,4)
6.	$(((\neg(\neg\phi)) \equiv \phi) \equiv \mathit{true})$	(Asociatividad 5)
7.	$((\neg(\neg\phi)) \equiv \phi)$	(Identidad 6).

Demostraciones para los demás teoremas se proponen como ejercicios para el lector. □

Las demostraciones presentadas anteriormente sirven para ilustrar el primer uso de algunas reglas de inferencia de DS y para introducir algunas convenciones:

- En la demostración del Teorema 4.15.1 se usan las reglas estructurales de asociatividad y conmutatividad de la equivalencia presentadas en la Sección 4.2.
- La demostración presentada del Teorema 4.15.2 no es precisamente una demostración en el sentido estricto de la definición de DS. En particular, el renglón 2 viola la definición de demostración porque la proposición que allí aparece no es una axioma ni tampoco se obtiene usando una regla de inferencia con premisas que previamente aparecen en dicha demostración. Sin embargo, usar teoremas en los renglones de una demostración es una libertad que se permite para simplicar el trabajo deductivo. Estas simplificaciones son aceptadas bajo el siguiente acuerdo: dicho teorema debe ser anterior y estar demostrado; además se debe indicar cómo reemplazar las proposiciones del teorema usado en la proposición que aparece en la demostración actual. El efecto formal de esta convención es el de 'copiar y pegar' una demostración del teorema usado en la demostración actual con las proposiciones reemplazadas adecuadamente. Por ejemplo, en el renglón 2 de la demostración del Teorema 4.15.2, se usa el Teorema 4.6.2 (i.e, $((\phi \equiv \phi) \equiv \mathit{true})$) en donde ϕ se reemplaza con *false*. Recuerde que toda instancia de un axioma es un axioma. También, toda instancia de un teorema es a su vez un teorema (Ejercicio 1, Sección 4.1).
- La regla Leibniz se utiliza en el renglón 3 de la demostración del Teorema 4.15.4. Note que esta deducción corresponde a la siguiente inferencia:

$$\frac{((\phi \equiv \psi) \equiv (\psi \equiv \phi))}{((p \equiv \mathit{false})[p := (\phi \equiv \psi)] \equiv (p \equiv \mathit{false})[p := (\psi \equiv \phi)])}\ \textsc{Leibniz}$$

- Finalmente, observe que la regla TRANSITIVIDAD es usada en varias de las demostraciones. En particular, se usa en el renglón 8 de la demostración del Teorema 4.15.4 con la explicación 'Transitividad 1,3,4,6,7'. Esta es una convenición para usar la transitividad de la equivalencia en cadena o cascada (i.e., transitivamente). Por ejemplo, la secuencia '1,3,4,6,7' en dicha leyenda indica que se aplica TRANSITIVIDAD con premisas en los renglones 1 y 3, la proposición resultante se usa como premisa junto con la proposición en el renglón 4 para otra vez aplicar la regla, y así sucesivamente.

La discrepancia es un operador muy conocido y usado en informática, pero bajo otro nombre: '*xor*'. Su interpretación es la de 'o exclusivo'. En el lenguaje cotidiano, la discrepancia elimina la ambiguedad de la disyunción cuando se desea enunciar una situación en la cual exactamente uno de sus operandos es cierto (pero no ambos). En informática, más precisamente en criptografía, la discrepancia es utilizada para obtener un simple pero efectivo método de ciframiento y desciframiento conocido como *ciframiento 'xor'* (en inglés, '*xor cypher*'). Este método está basado en el hecho de que la discrepancia puede interpretarse como la suma de *bits* modulo 2 en donde *true* corresponde a 1 y *false* corresponde a 0: $0 + 0 = 0 = 1 + 1$ y $0+1 = 1 = 1+0$. El uso de la discrepancia en criptografía se ilustra con un ejemplo al final de esta sección.

A continuación se presentan algunas propiedades de la discrepancia.

Teorema 4.16

Para cualesquiera proposiciones ϕ, ψ y τ de DS:

1. $\vdash_{\mathsf{DS}} ((\phi \not\equiv (\psi \not\equiv \tau)) \equiv ((\phi \not\equiv \psi) \not\equiv \tau))$
2. $\vdash_{\mathsf{DS}} ((\phi \not\equiv \psi) \equiv (\psi \not\equiv \phi))$
3. $\vdash_{\mathsf{DS}} ((\phi \not\equiv \mathit{false}) \equiv \phi)$
4. $\vdash_{\mathsf{DS}} ((\phi \not\equiv \phi) \equiv \mathit{false})$
5. $\vdash_{\mathsf{DS}} (((\phi \not\equiv \psi) \not\equiv \psi) \equiv \phi)$

La discrepancia es asociativa, conmutativa y tiene elemento identidad *false* (teoremas 4.16.1-3). La discrepancia es irreflexiva (Teorema 4.16.4) y acepta una ley de cancelación (Teorema 4.16.5). A continuación se presentan demostraciones de algunos de estos teoremas.

Demostración. Para (4):

1. $((\phi \not\equiv \phi) \equiv ((\neg\phi) \equiv \phi))$ $(Ax10)$
2. $(((\neg\phi) \equiv \phi) \equiv (\phi \equiv (\neg\phi)))$ $(Ax2)$
3. $((\phi \equiv (\neg\phi)) \equiv \mathit{false})$ (Teorema 4.15.7)
4. $((\phi \not\equiv \phi) \equiv \mathit{false})$ (Transitividad 1,2,3).

Demostraciones para los demás teoremas se proponen como ejercicios para el lector. □

4.3.1. Ciframiento de texto con *xor*. Como se explicó al inicio de esta sección, en informática hay (y se usa) un método de ciframiento basado en la discrepancia. En particular, este método está basado en las propiedades de la discrepancia enunciadas en los teoremas 4.16.1-5. La idea es que dada una cadena texto t y una llave de ciframiento k, ambas representadas como secuencias de bits, t puede ser cifrada con la llave k por medio de la operación $(t \not\equiv k)$ interpretada bit a bit, obteniendo así un texto secreto. Suponga que este texto secreto se denota con t_k. Para recuperar el texto original t del texto secreto t_k, es decir, para descrifrar t_k, basta con aplicar la operación $(t_k \not\equiv k)$. ¿Por qué? Observe que, de acuerdo con el Teorema 4.16.5, la operación de desciframeinto garantiza la obtención del texto original:

$$(t_k \not\equiv k) = ((t \not\equiv k) \not\equiv k) = t.$$

A continuación se presenta un ejemplo que ilustra paso a paso el proceso de ciframiento y desciframiento de texto basado en la discrepancia.

Ejemplo 4.1

Este ejemplo está basado en material disponible públicamente en Wikipedia. La cadena de texto `Wiki`, representada en código ASCII de 8 bits como

`01010111 01101001 01101011 01101001,`

puede ser cifrada con repeticiones de la llave `11110011` de la siguiente forma:

```
    01010111  01101001  01101011  01101001
≢   11110011  11110011  11110011  11110011
-------------------------------------------
    10100100  10011010  10011000  10011010.
```

Para revertir el proceso se repite la misma operación bit a bit usando como operandos el texto cifrado y la llave de ciframiento originalmente usada:

```
    10100100  10011010  10011000  10011010
≢   11110011  11110011  11110011  11110011
-------------------------------------------
    01010111  01101001  01101011  01101001.
```

Fíjese que, al usar ciframiento basado en la discrepancia, la seguridad se preserva mientras la llave no sea comprometida. Por ejemplo, si se usa una llave de ciframiento buena y esta se pierde, es prácticamente imposible recuperar el texto original a partir del texto cifrado (vea los ejercicios 13 y 14 al final de esta sección).

Nota 4.17

El ciframiento basado en *xor* es muy usado gracias a su simplicidad y facilidad de implementación. La siguiente función `xor_cypher` en el lenguaje de programación Python 3, cifra una cadena de texto con una llave (también representada como cadena) usando *xor* con el método descrito anteriormente:

```
1 def xor_cypher(text, key):
2     """xor de las cadenas text y key; se supone que
3        len(text)<=len(key)"""
4     return ''.join(chr(ord(a)^ord(b)) for a,b in zip(text,key))
```

Ejercicios

1. Demuestre que el axioma $(Ax9)$ es una tautología.
2. Demuestre que el axioma $(Ax10)$ es una tautología.
3. Ilustre con una inferencia el uso de la regla Leibniz* en el renglón 6 de la demostración del Teorema 4.15.4.
4. Demuestre el Teorema 4.15.3.
5. Demuestre el Teorema 4.15.5.
6. Demuestre el Teorema 4.15.7.
7. Demuestre el Teorema 4.16.1 (solo si tiene mucho tiempo disponible).
8. Demuestre el Teorema 4.16.2.
9. Demuestre el Teorema 4.16.3.
10. Demuestre el Teorema 4.16.5.
11. Investigue acerca de los siguientes operadores de Python 3, explique su uso e ilústrelo con ejemplos:
    - `^`
    - `zip`
    - `ord`
    - `chr`
    - `join`

12. Implemente la función `xor_cypher` y calcule el resultado de las siguientes invocaciones:
    a) `xor_cypher("h","&")`
    b) `xor_cypher(xor_cyper("h","&"),"&")`
    c) `xor_cypher("hola","mundo")`
    d) `xor_cypher(xor_cypher("hola","mundo"),"Mundo")`
    e) `xor_cypher("hola","----")`
    f) `xor_cypher(xor_cypher("hola","----"),"--")`
13. Considere la siguiente situación: uno de sus amigos del curso de lógica cifró un archivo de texto con la función `xor_cypher` pero ha perdido la llave de ciframiento. El amigo tiene a disposición el archivo cifrado y recuerda que la llave de ciframiento es una cadena formada a partir de una única letra minúscula. Sin embargo, él no recuerda la longitud de la llave de ciframiento. ¿Es posible ayudar a su amigo a recuperar el archivo original? Explique su respuesta.
14. Considere el mismo ejercicio del numeral anterior, pero con una llave conformada a partir de dos letras minúsculas distintas.

4.4. La disyunción

La disyunción $\vee$ es uno de los operadores básicos del sistema formal DS, junto con las constantes *true* y *false*, y la equivalencia $\equiv$. Su interpretación es la del 'o inclusivo', es decir, es verdadera cuando al menos uno de sus operandos es verdadero.

A continuación se recuerdan los axiomas de DS que están relacionados directamente con la disyunción.

Definición 4.18

Sean ϕ, ψ, τ proposiciones de DS. El *conjunto de axiomas* de DS incluye los siguientes axiomas relacionados directamente con la disyunción:

$(Ax4)$**:** $((\phi \vee (\psi \vee \tau)) \equiv ((\phi \vee \psi) \vee \tau))$

$(Ax5)$**:** $((\phi \vee \psi) \equiv (\psi \vee \phi))$

$(Ax6)$**:** $((\phi \vee \mathit{false}) \equiv \phi)$

$(Ax7)$**:** $((\phi \vee \phi) \equiv \phi)$

$(Ax8)$**:** $((\phi \vee (\psi \equiv \tau)) \equiv ((\phi \vee \psi) \equiv (\phi \vee \tau)))$

La disyunción es asociativa $(Ax4)$, conmutativa $(Ax5)$ y tiene elemento identidad *false* $(Ax6)$. Además, la disyunción es idempotente $(Ax7)$ y distribuye sobre la equivalencia $(Ax8)$.

A continuación se presentan algunas propiedades de la disyunción.

Teorema 4.19

Para cualesquiera proposiciones ϕ y ψ de DS:

1. $\vdash_{\mathsf{DS}} (\phi \vee (\neg\phi))$
2. $\vdash_{\mathsf{DS}} ((\phi \vee \mathit{true}) \equiv \mathit{true})$
3. $\vdash_{\mathsf{DS}} (\phi \vee \mathit{true})$
4. $\vdash_{\mathsf{DS}} ((\phi \vee \psi) \equiv ((\phi \vee (\neg\psi)) \equiv \phi))$

La disyunción satisface la ley del 'tercero excluído', es decir, bajo cualquier valuación una proposición o su negación son verdaderas (Teorema 4.19.1). La disyunción tiene elemento anulador *true* (teoremas 4.19.2-3). El Teorema 4.19.4 es una propiedad de caracterización extraña que es útil a la hora de simplificar algunas demostraciones que involucran a la disyunción. A continuación se presentan demostraciones de algunos de estos teoremas.

Demostración. Para (1):

1.	$((\neg\phi) \equiv (\phi \equiv \mathit{false}))$	$(Ax9)$
2.	$((\phi \vee (\neg\phi)) \equiv (\phi \vee (\phi \equiv \mathit{false})))$	(Leibniz 1)
3.	$((\phi \vee (\phi \equiv \mathit{false})) \equiv ((\phi \vee \phi) \equiv (\phi \vee \mathit{false})))$	$(Ax8)$
4.	$((\phi \vee \phi) \equiv \phi)$	$(Ax7)$
5.	$(((\phi \vee \phi) \equiv (\phi \vee \mathit{false})) \equiv (\phi \equiv (\phi \vee \mathit{false})))$	(Leibniz 4)
6.	$((\phi \vee \mathit{false}) \equiv \phi)$	$(Ax6)$
7.	$((\phi \equiv (\phi \vee \mathit{false})) \equiv (\phi \equiv \phi))$	(Leibniz 6)
8.	$((\phi \equiv \phi) \equiv \mathit{true})$	(Teorema 4.6.2)
9.	$((\phi \vee (\neg\phi)) \equiv \mathit{true})$	(Transitividad 2,3,5,7,8)
10.	$(\phi \vee (\neg\phi))$	(Identidad).

Demostraciones para los demás teoremas se proponen como ejercicios para el lector. □

Históricamente, la disyunción ha desempeñando un papel protagónico en la lógica y en su estudio, pues este conectivo lógico sirve para explicar la diferencia entre dos tipos de lógicas: la *lógica clásica* y la *lógica intuicionista* (o *constructivista*). Desde el punto de vista filosófico, la lógica clásica puede verse como una versión Platónica de la lógica en donde la verdad de una proposición *siempre* existe

y un observador únicamente puede encontrar una demostración *directa* o *indirecta* de ella. La lógica intuicionista es diferente en este sentido dado que la verdad de una proposición únicamente es aceptada cuando se construye una demostración *directa* de ella; en el contexto de la lógica intuicionista, la verdad no existe sin una demostración directa. Una ejemplo de una demostración directa es cualquier demostración de un teorema de DS que se haya hecho hasta ahora. Un ejemplo de una demostración indirecta de una proposición ϕ es una demostración por contradicción: si se supone que $(\neg\phi)$ es cierto, entonces se llega a un absurdo. Y es así como surge la siguiente pregunta: ¿para demostrar una propiedad ϕ basta con (i) exhibir una demostración para ϕ o (ii) justificar la imposibilidad de la ausencia de una demostración para ϕ? Los intuicionistas solo aceptan como razonable la situación (i), mientras que los demás aceptan tanto (i) como (ii). El sistema DS es un sistema lógico *clásico*. En este sentido, en DS se puede demostrar como teorema una proposición ϕ ya bien sea dando una demostración directa o una indirecta de ella. Como sistema lógico clásico que es, DS está caracterizado por el teorema del 'tercero excluído' (Teorema 4.19.1).

A continuación se presenta una demostración *no constructiva* de un hecho muy conocido en matemáticas: $\sqrt{2}$ es irracional. Un número real x es irracional sii x no puede expresarse como una fracción $\frac{a}{b}$ con $a, b \in \mathbb{Z}$ y $b \neq 0$.

Ejemplo 4.2

Observación: $\sqrt{2}$ es irracional.

Argumentación: Suponga hacia una contradicción que $\sqrt{2}$ es racional, es decir, que hay $a, b \in \mathbb{Z}$ con $b \neq 0$ tales que $x = \frac{a}{b}$. Como $\sqrt{2} > 0$, se puede suponer que $a > 0$ y $b > 0$. Además, siempre se puede suponer que la fracción $\frac{a}{b}$ es irreducible (i.e., a y b no tiene factores positivos en común aparte de 1). Observe entonces lo siguiente:

$$\sqrt{2} = \frac{a}{b} \quad \text{sii} \quad 2 = \frac{a^2}{b^2} \quad \text{sii} \quad 2b^2 = a^2.$$

Esto indica que a es par (Ejercicio 7), es decir, $a = 2c$ para algún $c \in \mathbb{N}$ (Ejercicio 6). Entonces se tiene:

$$2b^2 = a^2 \quad \text{sii} \quad 2b^2 = 4c^2 \quad \text{sii} \quad b^2 = 2c^2.$$

Esto a su vez indica que b es par, es decir, $b = 2d$ para algún $d \in \mathbb{N}$. Consecuentemente, $\frac{a}{b} = \frac{2c}{2d} = \frac{c}{d}$ y por tanto a y b tienen a 2 como factor común. Esto contradice la suposición sobre la irreducibilidad de la fracción $\frac{a}{b}$ y así la suposición inicial de que $\sqrt{2}$ no es irracional es falso. Por tanto, $\sqrt{2}$ es irracional.

Note que el argumento en el Ejemplo 4.2 está basado en el hecho de que $\sqrt{2}$ es o no irracional (tercero excluído). El argumento justifica que $\sqrt{2}$ no es racional (i.e., no es no irracional). Entonces, necesariamente, $\sqrt{2}$ es irracional. Sin embargo, en ningún momento se 'construye' una demostración directa de la irracionalidad de $\sqrt{2}$ sino que se presenta una demostración indirecta.

Ejemplo 4.3

Los pasos deductivos en la demostración del Ejemplo 4.2 pueden expresarse en lógica proposicional. Considere la siguiente simbolización:

p**:** $\sqrt{2}$ es irracional.

La demostración del Ejemplo 4.2 se justifica en DS de la siguiente forma:

1.	$(p \vee (\neg p))$	(Teorema 4.19.1)
2.	$(\neg(\neg p))$	($\sqrt{2}$ no es racional –Ejemplo 4.2)
3.	$((\neg(\neg p)) \equiv ((\neg p) \equiv \mathit{false}))$	($Ax9$)
4.	$((\neg p) \equiv \mathit{false})$	(Ecuanimidad 2,3)
5.	$((p \vee (\neg p)) \equiv (p \vee \mathit{false}))$	(Leibniz 4)
6.	$(p \vee \mathit{false})$	(Ecuanimidad 1,5)
7.	$((p \vee \mathit{false}) \equiv p)$	($Ax6$)
8.	p	(Ecuanimidad 6,7).

Desde el punto de vista computacional, cuando la disyunción es interpretada como un operador binario sobre *bits* (es decir, sobre los valores 0 y 1 en donde 0 corresponde a *false* y 1 a *true*), la disyunción resultan en el operador de *máximo* binario: el bit 1 es el máximo entre los bits 0 y 1. Traduciendo esta interpretación al sistema formal DS, se puede decir que la disyunción establece un orden estricto entre las constantes lógicas del sistema formal: *true* es el máximo entre *false* y *true*. En realidad, la disyunción no determina únicamente que *true* es el máximo entre las dos constantes del sistema formal DS; también determina que *true* es la proposición *máxima* entre *todas* las proposiciones del sistema formal (Teorema 4.19.2). Como se aprenderá en secciones próximas, también hay un operador que caracteriza una proposición mínima entre todas las proposiciones (Ejercicio 4) y un operador que determina un 'orden parcial' entre las proposiciones de DS (Ejercicio 5).

Ejercicios

1. Demuestre el Teorema 4.19.2.
2. Demuestre el Teorema 4.19.3.
3. Demuestre el Teorema 4.19.4.
4. Apelando a la interpretación formal de los conectivos lógicos, es decir, basándose en las funciones $H_{true}, H_{false}, \ldots$ de la Sección 4.1, identifique cuál de los operadores lógicos de DS determina que *false* es la proposición mínima entre todas las proposiciones de DS. Justifique su respuesta.
5. Apelando a la interpretación formal de los conectivos lógicos, es decir, basándose en las funciones $H_{true}, H_{false}, \ldots$ de la Sección 4.1, identifique cuál de los conectivos lógicos de DS es un orden parcial. Un conectivo lógico binario $\otimes$ es un *orden parcial* sii $\otimes$ es *reflexivo* (i.e., $\mathbf{v}(\phi \otimes \phi) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$ de ϕ), *transitivo* (i.e., si $\mathbf{v}(\phi \otimes \psi) = \mathtt{T}$ y $\mathbf{v}(\psi \otimes \tau) = \mathtt{T}$, entonces $\mathbf{v}(\phi \otimes \tau) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$ de ϕ, ψ, τ) y *antisimétrico* (i.e., si $\mathbf{v}(\phi \otimes \psi) = \mathtt{T}$ y $\mathbf{v}(\psi \otimes \phi) = \mathtt{T}$, entonces $\mathbf{v}(\phi \equiv \psi) = \mathtt{T}$ para cualquier valuación $\mathbf{v}$ de ϕ, ψ).
6. Demuestre que si $a \in \mathbb{N}$ y a es par, entonces hay un $c \in \mathbb{N}$ tal que $a = 2c$.
7. Demuestre que si $a \in \mathbb{N}$ y a^2 es par, entonces a es par.
8. Demuestre que $\sqrt{3}$ es irracional.
9. Investigue cuáles son los operadores de disyunción lógica y disyunción entre *bits* de Python. De ejemplos del uso de cada uno de ellos.
10. Considere la siguiente regla de inferencia:

$$\frac{(\phi \vee \psi) \qquad (\neg\phi)}{\psi}\ \textsc{Silogismo disyuntivo}$$

    Explique brevemente el significado de la regla Silogismo disyuntivo, de un ejemplo de su uso y demuestre que es correcta.

11. Considere la siguiente regla de inferencia:

$$\frac{(\phi \vee \psi) \qquad ((\neg\phi) \vee \tau)}{(\psi \vee \tau)}\ \textsc{Corte}$$

    Explique brevemente el significado de la regla Corte, de un ejemplo de su uso y demuestre que es correcta.

12. Considere la siguiente regla de inferencia:

$$\frac{\phi}{(\phi \vee \psi)}\ \textsc{Debilitamiento}$$

Explique brevemente el significado de la regla Debilitamiento, de un ejemplo de su uso y demuestre que es correcta.

13. Considere la siguiente regla de inferencia:

$$\frac{(\phi \vee \psi)}{\phi}\ \text{Debilitamiento?}$$

Explique con un contraejemplo por qué la regla Debilitamiento? es incorrecta.

4.5. *Intermezzo:* derivaciones

La potencia deductiva de DS puede ser llevada a otro nivel usando el concepto de derivación, una técnica de alto nivel para 'calcular' demostraciones. La idea principal es que a partir de una derivación se puede obtener una demostración. Esta situación es similar a la que sucede con algunos lenguajes de programación. Por ejemplo, un programa en el lenguaje de programación Java es compilado en un lenguaje de bajo nivel llamado Java Bytecode. El programa resultante en Java Bytecode es entonces ejecutado por la máquina virtual de Java (JVM). Sin embargo, un usuario común de Java no debe conocer el lenguaje Java Bytecode para hacer aplicaciones Java. Además, programar directamente en Java Bytecode puede ser una experiencia algo tediosa. En este sentido, se puede pensar que una demostración en DS es una secuencia de fórmulas de bajo nivel (mucho detalle), mientras que una derivación es una secuencia de fórmulas de alto nivel que puede ser 'compilada' para obtener una demostración.

Definición 4.20

Una *derivación* en DS es una secuencia finita de proposiciones $\phi_0, \phi_1, \ldots, \phi_n$ de DS tales que $\vdash_{\mathsf{DS}} (\phi_{k-1} \equiv \phi_k)$ para $0 < k \leq n$.

Note que una derivación es una secuencia finita (y no vacía) de proposiciones en DS. En particular, se acepta como derivación una secuencia con una única proposición. A pesar de que las demostraciones y las derivaciones tienen la misma estructura sintáctica (i.e., ambas son secuencias de proposiciones), la Definición 4.20 no asocia relación alguna entre una demostración y una derivación. Para el propósito del estudio de la lógica proposicional, la relación que existe entre estos dos objetos se establece en el Metateorema 4.21.

Metateorema 4.21

Sea $\phi_0, \phi_1, \ldots, \phi_n$ una derivación en DS. Entonces se tiene $\vdash_{\mathsf{DS}} (\phi_0 \equiv \phi_n)$.

Demostración. Se supone que $\phi_0, \phi_1, \ldots, \phi_n$ es una derivación en DS y se procede por inducción sobre $n \in \mathbb{N}$.

Caso base: Si $n = 0$, entonces se debe demostrar que $\vdash_{\mathsf{DS}} (\phi_0 \equiv \phi_0)$. Esto es trivial porque la equivalencia es reflexiva (Teorema 4.6.3).

Caso inductivo: Se supone que la propiedad es cierta para $n \geq 0$ y se demuestra para $n + 1$. Considere la derivación $\phi_0, \phi_1, \ldots, \phi_n, \phi_{n+1}$. Por la hipótesis inductiva se tiene $\vdash_{\mathsf{DS}} (\phi_0 \equiv \phi_n)$. Por la definición de derivación se tiene $\vdash_{\mathsf{DS}} (\phi_n \equiv \phi_{n+1})$. Finalmente, por la regla TRANSITIVIDAD se tiene $\vdash_{\mathsf{DS}} (\phi_0 \equiv \phi_{n+1})$.

Por el principio de inducción matemática se concluye que si $\phi_0, \phi_1, \ldots, \phi_n$ es una derivación en DS, entonces $\vdash_{\mathsf{DS}} (\phi_0 \equiv \phi_n)$. □

En una derivación se preserva la equivalencia entre cualquier par de proposiciones consecutivas y por ello, dado que la equivalencia es transitiva, se establece la equivalencia entre la primera y la última proposición de una derivación. A diferencia de una demostración, no todas las proposiciones en una derivación son teoremas. Esta flexibilidad resulta en más libertad para calcular, pero requiere cuidado a la hora de construir e interpretar una derivación. Por ejemplo, como se establece en el Ejercicio 4.5.3, cualquier demostración es necesariamente una derivación. Intuitivamente, esto es cierto dado que todas las proposiciones en una demostración son axiomas o teoremas, y entre ellos son equivalentes. Sin embargo, no toda derivación es una demostración debido a que dos proposiciones pueden ser equivalentes aún sin que cada una de ellas sea un axioma o teorema (ver Ejercicio 4.5.4).

Nota 4.22

Una derivación $\phi_0, \phi_1, \ldots, \phi_n$ en DS se puede diagramar esquemáticamente de la siguiente forma:

$$
\begin{array}{ll}
 & \phi_0 \\
\equiv & \langle\ \text{explicación}_0\text{: por qué ``}\vdash_{\mathsf{DS}} (\phi_0 \equiv \phi_1)\text{''}\ \rangle \\
 & \phi_1 \\
\vdots & \langle\ \ldots\ \rangle \\
 & \phi_{n-1} \\
\equiv & \langle\ \text{explicación}_{n-1}\text{: por qué ``}\vdash_{\mathsf{DS}} (\phi_{n-1} \equiv \phi_n)\text{''}\ \rangle \\
 & \phi_n
\end{array}
$$

en donde explicación$_i$ es un texto explicándo por qué $\vdash_{\mathsf{DS}} (\phi_i \equiv \phi_{i+1})$. Note que con la representación gráfica de una derivación se distingue claramente entre una demostración y una derivación.

El Ejemplo 4.4 presenta una derivación y se sugiere al lector compararla con la demostración inicialmente presentada del Teorema 4.15.1.

Ejemplo 4.4

La siguiente derivación es una demostración alternativa del Teorema 4.15.1.

$$
\begin{array}{ll}
 & (\neg \mathit{true}) \\
\equiv & \langle\ \mathit{Ax}9\ \rangle \\
 & (\mathit{true} \equiv \mathit{false}) \\
\equiv & \langle\ \text{conmutatividad de } \equiv\ \rangle \\
 & (\mathit{false} \equiv \mathit{true}) \\
\equiv & \langle\ \mathit{Ax}3\ \rangle \\
 & \mathit{false}
\end{array}
$$

Por el Metateorema 4.21 se obtiene $\vdash_{\mathsf{DS}} ((\neg \mathit{true}) \equiv \mathit{false})$. Por la regla Conmutatividad se concluye $\vdash_{\mathsf{DS}} (\mathit{false} \equiv (\neg \mathit{true}))$.

Ejercicios

1. Investigue el significado y el orígen de la palabra *intermezzo*. Ilustre el uso de esta palabra con un par de ejemplos.

2. Considere una proposición ϕ y la siguiente argumentación:
   > La secuencia unitaria ϕ es una derivación. Entonces, por el Metateorema 4.21 se tiene $\vdash_{\mathsf{DS}} \phi$.

   ¿Por qué esta argumentación es incorrecta? Justifique su respuesta.
3. Demuestre que si una secuencia $\phi_0, \ldots, \phi_n$ de proposiciones en DS es una demostración, entonces es una derivación.
4. Formule un ejemplo en donde una secuencia de proposiciones que tengan las misma tabla de verdad y en las cuales haya al menos un renglón que resulte en `F`. Intuitivamente, justifique por qué esta derivación no es una demostración.
5. Considere una secuencia $\phi_0, \ldots, \phi_n$ de proposiciones en DS. Demuestre que las siguientes dos afirmaciones son equivalentes:
   - $\phi_0, \ldots, \phi_n$ es una derivación
   - $\phi_n, \ldots, \phi_0$ es una derivación
6. Considere una secuencia $\phi_0, \ldots, \phi_n$ de proposiciones en DS y las siguientes afirmaciones:
   - $\phi_0, \ldots, \phi_n$ es una derivación
   - $\vdash_{\mathsf{DS}} (\phi_0 \equiv \cdots \equiv \phi_n)$

   Justifique por qué estas dos afirmaciones no son equivalentes.
7. Siguiendo la demostración del Metateorema 4.21, obtenga la demostración en DS correspondiente a la derivación en el Ejemplo 4.4. En otras palabras, compile dicha derivación en una demostración de DS.

4.6. La conjunción

La conjunción $\wedge$ es un conectivo lógico que se define en términos de otros conectivos lógicos del sistema formal DS. Su interpretación es la del 'y', es decir, una conjunción es únicamente verdadera cuando sus dos operandos son verdaderos.

A continuación se presenta el axioma definicional de la conjunción en DS.

Definición 4.23

Sean ϕ y ψ proposiciones de DS. El siguiente axioma de DS define la conjunción:

$(Ax11)$**:** $((\phi \wedge \psi) \equiv (\phi \equiv (\psi \equiv (\phi \vee \psi))))$

En la Definición 4.23, el axioma $(Ax11)$ define la conjunción en términos de la equivalencia y la disyunción. El axioma $(Ax11)$ es conocido también como la 'regla dorada' y se originó en el estudio del álgebra abstracta. En particular, este axioma es la piedra angular que permitió demostrar a comienzos del siglo XX que el álgebra

Booleana y los anillos Booleanos, dos teorías que hasta ese momento se estudiaban por separado al ser consideradas completamente diferentes, eran lo mismo. Por eso su nombre.

Note que la conjunción se define únicamente por medio del axioma ($Ax11$) y en ningún momento se establecen como dadas las propiedades estructurales del operador, como lo son la asociatividad, conmutatividad, identidad, etc.. A continuación se enuncian algunas propiedades estructurales de la conjunción.

Teorema 4.24

Para cualesquiera proposiciones ϕ, ψ, τ de DS:

1. $\vdash_{\mathsf{DS}} ((\phi \wedge (\psi \wedge \tau)) \equiv ((\phi \wedge \psi) \wedge \tau))$
2. $\vdash_{\mathsf{DS}} ((\phi \wedge \psi) \equiv (\psi \wedge \phi))$
3. $\vdash_{\mathsf{DS}} ((\phi \wedge \mathit{true}) \equiv \phi)$
4. $\vdash_{\mathsf{DS}} ((\phi \wedge \mathit{false}) \equiv \mathit{false})$
5. $\vdash_{\mathsf{DS}} ((\phi \wedge \phi) \equiv \phi)$

La conjunción es asociativa (Teorema 4.24.1), conmutativa (Teorema 4.24.2) y tiene elemento identidad *true* (Teorema 4.24.3). Además, la conjunción tiene elemento anulador *false* (Teorema 4.24.4) y es idempotente (Teorema 4.24.5). A continuación se presentan demostraciones de algunos de estos teoremas.

Demostración. Para (2):

$$
\begin{array}{ll}
 & (\phi \wedge \psi) \\
\equiv & \langle\ Ax11\ \rangle \\
 & (\phi \equiv (\psi \equiv (\phi \vee \psi))) \\
\equiv & \langle\ \text{conmutatividad de } \vee\ \rangle \\
 & (\phi \equiv (\psi \equiv (\psi \vee \phi))) \\
\equiv & \langle\ \text{asociatividad de } \equiv\ \rangle \\
 & ((\phi \equiv \psi) \equiv (\psi \vee \phi)) \\
\equiv & \langle\ \text{conmutatividad de } \equiv\ \rangle \\
 & ((\psi \equiv \phi) \equiv (\psi \vee \phi)) \\
\equiv & \langle\ \text{asociatividad de } \equiv\ \rangle \\
 & (\psi \equiv (\phi \equiv (\psi \vee \phi))) \\
\equiv & \langle\ \text{regla dorada } (Ax11)\ \rangle \\
 & (\psi \wedge \phi)
\end{array}
$$

Demostraciones para los demás teoremas se proponen como ejercicios para el lector. □

Fíjese que en la derivación del Teorema 4.24.2 no se hace referencia alguna a las reglas de inferencia de DS en ninguna de las explicaciones. Esta será una práctica más común a medida que se avance en este texto con el obejtivo de aprovechar el nivel de abstracción que brindan las derivaciones y también simplificar su escritura. Sin embargo, se alerta al lector para que siempre tenga en cuenta qué regla de inferencia se usa en cada paso de una derivación.

A continuación se enuncian algunas propiedades que relacionan la conjunción con otros conectivos lógicos.

Teorema 4.25

Para cualesquiera proposiciones ϕ, ψ, τ de DS:

1. $\vdash_{\mathsf{DS}} ((\phi \wedge (\neg\phi)) \equiv \mathit{false})$
2. $\vdash_{\mathsf{DS}} ((\neg(\phi \wedge \psi)) \equiv ((\neg\phi) \vee (\neg\psi)))$
3. $\vdash_{\mathsf{DS}} ((\neg(\phi \vee \psi)) \equiv ((\neg\phi) \wedge (\neg\psi)))$
4. $\vdash_{\mathsf{DS}} ((\phi \wedge (\psi \equiv \tau)) \equiv (((\phi \wedge \psi) \equiv (\phi \wedge \tau)) \equiv \phi))$
5. $\vdash_{\mathsf{DS}} ((\phi \wedge (\psi \not\equiv \tau)) \equiv ((\phi \wedge \psi) \not\equiv (\phi \wedge \tau)))$
6. $\vdash_{\mathsf{DS}} ((\phi \wedge (\psi \vee \tau)) \equiv ((\phi \wedge \psi) \vee (\phi \wedge \tau)))$
7. $\vdash_{\mathsf{DS}} ((\phi \vee (\psi \wedge \tau)) \equiv ((\phi \vee \psi) \wedge (\phi \vee \tau)))$

El Teorema 4.25.1 es una forma de caracterizar la contradicción en función de la conjunción. Los teoremas 4.25.2-3 son conocidos como las 'leyes de DeMorgan' y establecen cómo la negación distribuye sobre la conjunción y la disyunción, respectivamente. La conjunción pseudo-distribuye sobre la equivalencia (Teorema 4.25.4) y distribuye sobre la discrepancia (Teorema 4.25.5). La conjunción y la disyunción distribuyen mútuamente una sobre la otra (teoremas 4.25.6-7). Demostraciones de estos teoremas se proponen como ejercicios para el lector.

Desde el punto de vista computacional, cuando la conjunción es interpretada como un operador binario sobre *bits* (es decir, sobre los valores 0 y 1 en donde 0 corresponde a *false* y 1 a *true*), la conjunción no es otra cosa más que el operador de *mínimo* binario: el bit 0 es el mínimo entre los bits 0 y 1. Traduciendo esta interpretación al sistema formal DS, se puede decir que la conjunción completa el orden estricto para las constantes Booleanas introducido al final de la Sección 4.4: *true* es el máximo entre *false* y *true*, y *false* es el mínimo entre *false* y *true*. En el mismo sentido en que la disyunción establece a *true* como el máximo entre todas

las proposiciones de DS, la conjunción establece a *false* como el *mínimo* entre todas las proposiciones de DS (Teorema 4.24.4).

Ejercicios

1. Justifique que el axioma $(Ax11)$ es una tautología.
2. Demuestre el Teorema 4.24.1.
3. Explique qué regla de inferencia se usa, y con qué premisas, en cada uno de los pasos de la demostración del Teorema 4.24.2.
4. Demuestre el Teorema 4.24.3.
5. Demuestre el Teorema 4.24.4.
6. Demuestre el Teorema 4.24.5.
7. Demuestre la siguiente versión de la 'regla dorada' para la disyunción y la discrepancia:

   $$\vdash_{\mathsf{DS}} ((\phi \vee \psi) \equiv (\phi \not\equiv (\psi \not\equiv (\phi \wedge \psi)))).$$

8. Demuestre el Teorema 4.25.1.
9. Demuestre el Teorema 4.25.2.
10. Demuestre el Teorema 4.25.3.
11. Demuestre el Teorema 4.25.4.
12. Demuestre el Teorema 4.25.5.
13. Demuestre el Teorema 4.25.6.
14. Demuestre el Teorema 4.25.7.
15. Investigue cuáles son los operadores de conjunción lógica y conjunción entre *bits* de Python. De ejemplos del uso de cada uno de ellos.
16. Considere la siguiente regla de inferencia:

    $$\frac{(\phi \wedge \psi)}{\phi}\ \textsc{Debilitamiento}$$

    Explique brevemente el significado de la regla Debilitamiento, de un ejemplo de su uso y demuestre que es correcta.
17. Considere la siguiente regla de inferencia:

    $$\frac{\phi \qquad \psi}{(\phi \wedge \psi)}\ \textsc{Unión}$$

    Explique brevemente el significado de la regla Unión, de un ejemplo de su uso y demuestre que es correcta.

4.7. La implicación y la consecuencia

La implicación lógica suele ser el conectivo lógico preferido en los sistemas de cálculo tradicionales de la lógica. Esto se debe a que la implicación puede combinarse fácilmente con la mayoría de los demás conectivos lógicos. Por ejemplo, como se verá en el desarrollo de esta sección, la implicación distribuye sobre la equivalencia, la disyunción, la conjunción y sobre ella misma. Dado que la consecuencia lógica se define fácilmente con base en la implicación, y cualquier propiedad que tenga la impliacación automáticamente es heredada por la consecuencia, el objetivo en esta sección es estudiar principalmente las propiedades de la implicación. Esta sección contiene una cantidad significativa de teoremas y ejercicios y, como tal, brinda una oportunidad al lector para practicar cómo *calcular* teoremas en DS.

A continuación se presentan los axiomas definicionales de la implicación y la consecuencia en DS.

Definición 4.26

Sean ϕ y ψ proposiciones de DS. Los siguientes axiomas de DS definen la implicación y la consecuencia:

$(Ax12)$**:** $((\phi \to \psi) \equiv ((\phi \vee \psi) \equiv \psi))$

$(Ax13)$**:** $((\phi \leftarrow \psi) \equiv (\psi \to \phi))$

En la Definición 4.26, el axioma $(Ax12)$ define la implicación en términos de la disyunción y la equivalencia lógica. El axioma $(Ax13)$ define la consecuencia directamente en términos de la implicación; una definición como esta comúnmnente se denomina 'azúcar sintáctico' dado que representa una sencilla traducción de símbolos. Dada la similitud entre la implicación y la consecuencia en DS, los teoremas que se enuncien para la implicación en el resto de esta sección, se obtienen fácilmente para la consecuencia por virtud del axioma $(Ax13)$.

Nota 4.27

Interpretando en términos de *bits* el axioma $(Ax12)$ que define la implicación, la implicación binaria establece que el máximo entre el *bit* antecedente y el *bit* consecuente es el *bit* consecuente. Esto es compatible con la semántica de la implicación: es falsa únicamnete cuando el antecedente es verdadero y el consecuenta falso, es decir, cuando el 'máximo' entre el antecedente y el consecuente *no* es el consecuente.

La primera cuestión a observar es que la implicación puede ser definida de varias formas. En este sentido, los primeros teoremas acerca de la implicación son propiedades que permiten 'reescribir' la implicación de manera alternativa y complementaria a aquella indicada por el Axioma ($Ax12$).

Teorema 4.28

Para cualesquiera proposiciones ϕ y ψ de DS:

1. $\vdash_{\mathsf{DS}} ((\phi \to \psi) \equiv ((\neg\phi) \vee \psi))$.
2. $\vdash_{\mathsf{DS}} ((\phi \to \psi) \equiv ((\phi \wedge \psi) \equiv \phi))$.

Demostración. Para (1):

$$
\begin{array}{ll}
 & ((\neg\phi) \vee \psi) \\
\equiv & \langle \text{ definición de } \neg \rangle \\
 & ((\phi \equiv \mathit{false}) \vee \psi) \\
\equiv & \langle \text{ conmutatividad de } \vee \rangle \\
 & (\psi \vee (\phi \equiv \mathit{false})) \\
\equiv & \langle \text{ distribución de } \vee \text{ sobre } \equiv \rangle \\
 & ((\psi \vee \phi) \equiv (\psi \vee \mathit{false})) \\
\equiv & \langle \text{ identidad de } \vee \rangle \\
 & ((\psi \vee \phi) \equiv \psi) \\
\equiv & \langle \text{ conmutatividad de } \vee \rangle \\
 & ((\phi \vee \psi) \equiv \psi) \\
\equiv & \langle \text{ definición de } \to \rangle \\
 & (\phi \to \psi)
\end{array}
$$

La demostración para (2) se propone como ejercicio para el lector. □

Los siguientes teoremas relacionan la implicación y las constantes Booleanas.

Teorema 4.29

Para cualquier proposición ϕ de DS:

1. $\vdash_{\mathsf{DS}} (\phi \to \mathit{true})$
2. $\vdash_{\mathsf{DS}} (\mathit{false} \to \phi)$

3. $\vdash_{\mathsf{DS}} ((\mathit{true} \to \phi) \equiv \phi)$
4. $\vdash_{\mathsf{DS}} ((\phi \to \mathit{false}) \equiv (\neg\phi))$

La implicación tiene dos anuladores: *true* en el consecuente (Teorema 4.29.1) y *false* en el antecedente (Teorema 4.29.2). La implicación tiene a *true* en el antecedente como único elemento identidad (Teorema 4.29.3). La negación de una proposición puede caracterizarse con la implicación; esta es una forma adicional de caracterizar la contradicción (Teorema 4.29.4). Demostraciones de estos teoremas se proponen como ejercicios para el lector.

Los siguientes teoremas indican cómo la implicación distribuye sobre algunos conectivos lógicos de DS.

Teorema 4.30

Para cualesquiera proposiciones ϕ, ψ, τ de DS:

1. $\vdash_{\mathsf{DS}} ((\phi \to (\psi \equiv \tau)) \equiv ((\phi \to \psi) \equiv (\phi \to \tau)))$
2. $\vdash_{\mathsf{DS}} ((\phi \to (\psi \vee \tau)) \equiv ((\phi \to \psi) \vee (\phi \to \tau)))$
3. $\vdash_{\mathsf{DS}} ((\phi \to (\psi \wedge \tau)) \equiv ((\phi \to \psi) \wedge (\phi \to \tau)))$
4. $\vdash_{\mathsf{DS}} ((\phi \to (\psi \to \tau)) \equiv ((\phi \to \psi) \to (\phi \to \tau)))$
5. $\vdash_{\mathsf{DS}} ((\phi \to (\psi \leftarrow \tau)) \equiv ((\phi \to \psi) \leftarrow (\phi \to \tau)))$

La implicación distribuye sobre la equivalencia, la disyunción, la conjunción, la implicación y la consecuencia (Teoremas4.30.1-5).

Dado que el sistema DS está basado en la equivalencia, demostraciones y derivaciones que involucran la implicación pueden ser torpes. A continuación se presentan algunas propiedades de la implicación que son útiles para simplificar cálculos con proposiciones que mencionan la implicación.

Teorema 4.31

Para cualesquiera proposiciones ϕ, ψ, τ de DS:

1. $\vdash_{\mathsf{DS}} (((\neg\phi) \to (\neg\psi)) \equiv (\psi \to \phi))$
2. $\vdash_{\mathsf{DS}} ((\neg(\phi \to \psi)) \equiv (\phi \wedge (\neg\psi)))$

3. $\vdash_{\mathsf{DS}} ((\phi \equiv \psi) \equiv ((\phi \to \psi) \wedge (\psi \to \phi)))$
4. $\vdash_{\mathsf{DS}} ((\phi \equiv \psi) \to (\phi \to \psi))$
5. $\vdash_{\mathsf{DS}} ((\phi \to (\psi \to \tau)) \equiv ((\phi \wedge \psi) \to \tau))$
6. $\vdash_{\mathsf{DS}} (\phi \vee (\phi \to \psi))$
7. $\vdash_{\mathsf{DS}} ((\phi \vee (\psi \to \phi)) \equiv (\psi \to \phi))$
8. $\vdash_{\mathsf{DS}} ((\phi \wedge (\phi \to \psi)) \equiv (\phi \wedge \psi))$
9. $\vdash_{\mathsf{DS}} ((\phi \wedge (\psi \to \phi)) \equiv \phi)$

El Teorema 4.31.1 es conocido como la propiedad 'contrapositiva' y el Teorema 4.31.2 indica cómo se niega una implicación. El Teorema 4.31.3 establece una caracterización de la equivalencia en función de la implicación y la conjunción: la 'doble implicación' o 'implicación mútua'. El Teorema 4.31.4 establece que la implicación no es más fuerte que la equivalencia y el Teorema 4.31.5 una propiedad de acoplamiento de la implicación. Los teoremas 4.31.6-9 establencen propiedades de absorción de la implicación.

Nota 4.32

Es importante que el lector no confunda el concepto de contrapositiva de un implicación (Teorema 4.31.1) con el concepto de *converso* de una implicación. Dada una implicación $(\phi \to \psi)$, su converso es la implicación $(\psi \to \phi)$.

A continuación se presenta la demostración del Teorema 4.31.1.

Demostración. Considere la siguiente derivación:

$$
\begin{array}{ll}
 & ((\neg\phi) \to (\neg\psi)) \\
\equiv & \langle \text{ definición alternativa de } \to \text{ (Teorema 4.28-1) } \rangle \\
 & ((\neg(\neg\phi)) \vee (\neg\psi)) \\
\equiv & \langle \text{ doble negación (Teorema 4.15.6) } \rangle \\
 & (\phi \vee (\neg\psi)) \\
\equiv & \langle \text{ conmutatividad de } \vee \rangle \\
 & ((\neg\psi) \vee \phi) \\
\equiv & \langle \text{ definición alternativa de } \to \text{ (Teorema 4.28-1) } \rangle \\
 & (\psi \to \phi)
\end{array}
$$

Demostraciones de los demás teoremas se proponen como ejercicios para el lector. □

De acuerdo con la explicación dada en la Sección 4.4, en la lógica proposicional hay un conectivo lógico que establece una relación de orden entre las proposiciones (ver Ejercicio 4.4.5). La implicación es dicho conectivo y define un *orden parcial* sobre las proposiciones, es decir, la implicación es reflexiva, transitiva y antisimétrica. Estas propiedades de la implicación se enuncian con los siguientes teoremas.

Teorema 4.33

Para cualesquiera proposiciones ϕ, ψ, τ de DS:

1. $\vdash_{\mathsf{DS}} (\phi \to \phi)$
2. $\vdash_{\mathsf{DS}} (((\phi \to \psi) \wedge (\psi \to \tau)) \to (\phi \to \tau))$
3. $\vdash_{\mathsf{DS}} (((\phi \to \psi) \wedge (\psi \to \phi)) \to (\phi \equiv \psi))$

El Teorema 4.33.1 indica que la implicación es reflexiva, el Teorema 4.33.2 que es transitiva y el Teorema 4.33.3 que es antisimétrica.

Apelando a la implicación como el orden natural entre las proposiciones, es posible pensar en asociar una noción de 'cantidad' de información a una proposición. Por ejemplo, cuando las proposiciones p y $(p \vee q)$ son ciertas, la primera de ellas tiene más información que la segunda: en la primera de ellas necesariamente p es cierta pero en la segunda alguna entre p y q son ciertas, pero no se sabe cuál. Similarmente, cuando las proposiciones $(p \wedge q)$ y p son ciertas, la primera tiene más información que la segunda porque establece que tanto p como q son ciertas. El adjetivo 'parcial' en la expresión 'orden parcial' asociado a la implicación se puede explicar intuitivamente desde la noción de 'cantidad' de información. Por ejemplo, cuando las proposiciones p y q son ciertas, es imposible determinar si la verdad de p se puede establecer a partir de la verdad de q de acuerdo con el orden impuesto por la implicación o viceversa. En este sentido, las proposiciones p y q son incomparables: note que ni $(p \to q)$ o $(q \to p)$ son teoremas de DS.

Nota 4.34

Para cualesquiera proposiciones ϕ y ψ, si $\vdash_{\mathsf{DS}} (\phi \to \psi)$, entonces se dice que ϕ *es tan débil como* ψ (o, equivalentemente, ψ *es tan fuerte como* ϕ).

Teorema 4.35

Para cualesquiera proposiciones ϕ y ψ de DS:

1. $\vdash_{\mathsf{DS}} (\phi \to (\phi \vee \psi))$
2. $\vdash_{\mathsf{DS}} ((\phi \wedge \psi) \to \phi)$
3. $\vdash_{\mathsf{DS}} ((\phi \wedge \psi) \to (\phi \vee \psi))$
4. $\vdash_{\mathsf{DS}} (((\phi \vee \psi) \to (\phi \wedge \psi)) \equiv (\phi \equiv \psi))$
5. $\vdash_{\mathsf{DS}} (((\phi \vee \psi) \to \tau) \equiv ((\phi \to \tau) \wedge (\psi \to \tau)))$

Los teoremas 4.35.1-4 son caracterizaciones típicas de fortalecimientos y debilitamientos. El Teorema 4.35.5 establece una propiedad de análisis de casos pues indica que si en el antecedente de una implicación hay una disyunción, entonces dicha implicación es equivalente a la conjunción de dos implicaciones más sencillas: una por cada operando de la disyunción y con el mismo consecuente de la implicación inicial.

Finalmente, se presentan algunos teoremas acerca de la transitividad de la equivalencia y la implicación. En particular, se establece que la equivalencia es transitiva (hecho que en este punto se conoce y se usa gracias a la regla de inferencia Transitividad, pero que no ha sido demostrado como teorema de DS). También se identifica cómo la transitividad de la implicación conmuta con la transitividad de la equivalencia y viceversa.

Teorema 4.36

Para cualesquiera proposiciones ϕ, ψ, τ de DS:

1. $\vdash_{\mathsf{DS}} (((\phi \equiv \psi) \wedge (\psi \equiv \tau)) \to (\phi \equiv \tau))$
2. $\vdash_{\mathsf{DS}} (((\phi \equiv \psi) \wedge (\psi \to \tau)) \to (\phi \to \tau))$
3. $\vdash_{\mathsf{DS}} (((\phi \to \psi) \wedge (\psi \equiv \tau)) \to (\phi \to \tau))$

Demostraciones de estos teoremas se proponen como ejercicios para el lector.

Ejercicios

1. Demuestre que el axioma $(Ax12)$ es una tautología.
2. Demuestre que el axioma $(Ax13)$ es una tautología.
3. Demuestre el Teorema 4.28.2.

4. Demuestre el Teorema 4.29.1.
5. Demuestre el Teorema 4.29.2.
6. Demuestre el Teorema 4.29.3.
7. Demuestre el Teorema 4.29.4.
8. Demuestre el Teorema 4.30.1.
9. Demuestre el Teorema 4.30.2.
10. Demuestre el Teorema 4.30.3.
11. Demuestre el Teorema 4.30.4.
12. Demuestre el Teorema 4.30.5.
13. Demuestre o refute: la implicación distribuye sobre la discrepancia.
14. Demuestre el Teorema 4.31.2.
15. Demuestre el Teorema 4.31.3.
16. Demuestre el Teorema 4.31.4
17. Demuestre el Teorema 4.31.5.
18. Demuestre el Teorema 4.31.6.
19. Demuestre el Teorema 4.31.7.
20. Demuestre el Teorema 4.31.8.
21. Demuestre el Teorema 4.31.9.
22. Demuestre el Teorema 4.33.1.
23. Demuestre el Teorema 4.33.2.
24. Demuestre el Teorema 4.33.3.
25. Demuestre o refute: la discrepancia es reflexiva.
26. Demuestre o refute: la discrepancia es transitiva.
27. Demuestre o refute: la discrepancia es antisimétrica.
28. Demuestre o refute: la consecuencia es reflexiva.
29. Demuestre o refute: la consecuencia es transitiva.
30. Demuestre o refute: la consecuencia es antisimétrica.
31. Demuestre el Teorema 4.35.1.
32. Demuestre el Teorema 4.35.2.
33. Demuestre el Teorema 4.35.3.
34. Demuestre el Teorema 4.35.4.
35. Demuestre el Teorema 4.35.5.
36. Demuestre el Teorema 4.36.1.
37. Demuestre el Teorema 4.36.2.

38. Demuestre el Teorema 4.36.3.

39. Sean ϕ y ψ proposiciones de DS. Demuestre o refute:
    a) $\vdash_{\mathsf{DS}} ((\phi \vee \psi) \to (\phi \wedge \psi))$
    b) $\vdash_{\mathsf{DS}} ((\phi \equiv \psi) \to (\phi \to \psi))$
    c) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to (\phi \equiv \psi))$
    d) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \vee (\psi \to \phi))$

40. Sean ϕ, ψ, τ proposiciones de DS. Demuestre o refute las siguientes monotonías:
    a) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to ((\phi \equiv \tau) \to (\psi \equiv \tau)))$
    b) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to ((\phi \not\equiv \tau) \to (\psi \not\equiv \tau)))$
    c) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to ((\phi \vee \tau) \to (\psi \vee \tau)))$
    d) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to ((\phi \wedge \tau) \to (\psi \wedge \tau)))$
    e) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to ((\phi \to \tau) \to (\psi \to \tau)))$
    f) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to ((\phi \leftarrow \tau) \to (\psi \leftarrow \tau)))$

41. Demuestre la propiedad de 'Modus Ponens': $\vdash_{\mathsf{DS}} ((\phi \wedge (\phi \to \psi)) \to \psi)$.

42. La propiedad de 'Modus Ponens', más que como teorema, se usa como regla de inferencia en los sistemas formales tradicionales de la lógica proposicional:

$$\frac{\phi \qquad (\phi \to \psi)}{\psi}\ \textsc{Modus Ponens}$$

    Demuestre que Modus Ponens es correcta.

43. Considere la siguiente regla de inferencia:

$$\frac{(\phi \to \psi) \qquad (\neg\psi)}{(\neg\phi)}\ \textsc{Modus Tolens}$$

    a) Explique brevemente el significado de la regla Modus Tolens, de un ejemplo de su uso y demuestre que es correcta.
    b) Encuentre y explique la relación entre la regla Modus Tolens y la regla Modus Ponens (Ejercicio 42).

44. Considere la siguiente regla de inferencia:

$$\frac{(\phi \to \psi) \qquad (\psi \to \tau)}{(\phi \to \tau)}\ \textsc{Transitividad}$$

    Esta regla también es conocida como 'silogismo hipotético'.
    a) Explique brevemente el significado de la regla Transitividad, de un ejemplo de su uso y demuestre que es correcta.
    b) Encuentre y explique la relación entre la regla Transitividad y la regla Corte (Ejercicio 4.4.11).

45. El sistema formal PM de Whitehead y Rusell usa únicamente el conjunto $\{\vee, \to\}$ de conectivos lógicos. Para ϕ, ψ, τ proposiciones de PM, los axiomas de PM están dados por el siguiente esquema axiomático:

$(PM1)$**:** $((\phi \vee \phi) \to \phi)$
$(PM2)$**:** $(\phi \to (\phi \vee \psi))$
$(PM3)$**:** $((\phi \to \psi) \to ((\tau \vee \phi) \to (\tau \vee \psi)))$
$(PM4)$**:** $((\phi \vee \psi) \to (\psi \vee \phi))$

El sistema PM tiene como única regla de inferencia Modus Ponens.

a) Demuestre que los axiomas de PM son tautologías.
b) Demuestre que los axiomas de PM son teoremas de DS.
c) Sea ϕ una proposición de PM. Demuestre que si $\vdash_{\mathsf{PM}} \phi$, entonces $\vdash_{\mathsf{DS}} \phi$.
d) (Difícil) Sea ϕ una proposición de PM. Demuestre que si $\vdash_{\mathsf{DS}} \phi$, entonces $\vdash_{\mathsf{PM}} \phi$.

46. El nombre PM (Ejercicio 45) es una abreviación de *Principia Mathematica*, nombre de un texto de tres volúmenes acerca de los fundamentos de las matemáticas escrito por Alfred North Whitehead y Bertrand Russell a principios del siglo XX. El sistema PM se propone por primera vez en *Principia Mathematica* y luego es usado por Hilbert y Ackerman en *Principles of Mathematical Logic*. Una particularidad de *Principia Mathematica* es que su contenido está agrupado en párrafos enumerados. Investigue y transcriba al castellano el párrafo (1) del Volúmen 1 acerca de proposiciones elementales.

47. El sistema formal L, propuesto por Gottlob Frege, usa los conectivos $\{\neg, \to\}$. Para ϕ, ψ, τ proposiciones de L, los axiomas de L están dados por el siguiente esquema axiomático:

$L1$**:** $(\phi \to (\psi \to \phi))$
$L2$**:** $((\phi \to (\psi \to \tau)) \to ((\phi \to \psi) \to (\phi \to \tau)))$
$L3$**:** $(((\neg\phi) \to (\neg\psi)) \to (\psi \to \phi))$

El sistema L tiene como única regla de inferencia Modus Ponens. Lleve a cabo los numerales (a)–(d) del Ejercicio 45 para L en lugar de PM.

48. El sistema formal R, propuesto por Barkley Rosser, usa los conectivos $\{\neg, \wedge, \to\}$. Para ϕ, ψ, τ proposiciones de R, los axiomas de R están dados por el siguiente esquema axiomático:

$R1$**:** $(\phi \to (\phi \wedge \phi))$
$R2$**:** $((\phi \wedge \psi) \to \phi)$
$R3$**:** $((\phi \to \psi) \to ((\neg(\psi \wedge \tau)) \to (\neg(\phi \wedge \tau))))$

El sistema R tiene como única regla de inferencia Modus Ponens. Lleve a cabo los numerales (a)–(d) del Ejercicio 45 para R en lugar de PM.

49. El sistema formal LUK, propuesto por Jan Łukasiewicz (pronunciado "vu-ca-che-vich"), usa los conectivos $\{\neg, \to\}$. Para ϕ, ψ, τ proposiciones de LUK, los axiomas de LUK están dados por el siguiente esquema axiomático:

$LUK1$**:** $((\phi \to \psi) \to ((\tau \vee \phi) \to (\tau \vee \psi)))$
$LUK2$**:** $(((\neg\phi) \to \phi) \to \phi)$
$LUK3$**:** $(\phi \to ((\neg\phi) \to \psi))$

El sistema LUK tiene como única regla de inferencia MODUS PONENS. Lleve a cabo los numerales (a)–(d) del Ejercicio 45 para LUK en lugar de PM.

50. El sistema formal KA, propuesto por Stig Kanger, usa los conectivos $\{\neg, \rightarrow\}$. Para ϕ, ψ, τ proposiciones de KA, los axiomas de KA están dados por el siguiente esquema axiomático:

    $KA1$**:** $(\phi \rightarrow (\psi \rightarrow \phi))$

    $KA2$**:** $((\phi \rightarrow \psi) \rightarrow ((\tau \vee \phi) \rightarrow (\tau \vee \psi)))$

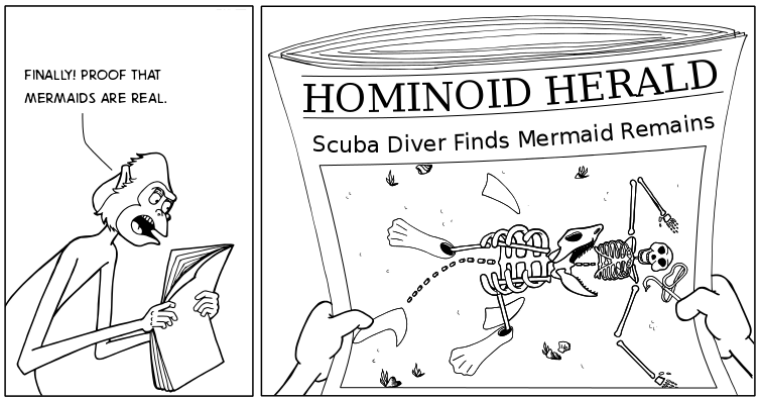
    $KA3$**:** $((\phi \rightarrow (\neg\psi)) \rightarrow (\psi \rightarrow (\neg\phi)))$

    $KA4$**:** $((\neg(\neg\phi)) \rightarrow ((\neg\phi) \rightarrow \psi))$

    El sistema KA tiene como única regla de inferencia MODUS PONENS y es un sistema proposicional de la lógica intuicionista. Lleve a cabo los numerales (a)–(c) del Ejercicio 45 para KA en lugar de PM.

Capítulo 5

Técnicas de razonamiento y demostración



Fallacious Logic Comic: Confirmation Bias (http://fallaciouslogic.net/11)

Este capítulo presenta técnicas de razonamiento y demostración que son útiles para analizar proposiciones de DS y, en general, propiedades de otras teorías (especialmente cuando sus aparatos deductivos están basados en DS). Por ejemplo, además de ilustrar demostraciones de algunos teoremas proposicionales, este capítulo aborda extensivamente teoremas sencillos de la teoría de conjuntos y de la teoría de números como ejemplo del uso y del alcance de las técnicas presentadas.

La Sección 5.1 propone una convención sobre los conectivos lógicos de DS que permite escribir proposiciones con menos paréntesis y sin alterar su semántica. Las técnicas de razonamiento y demostración se presentan en dos etapas. Inicialmente, la Sección 5.2 presenta un recuento de técnicas básicas para demostrar teoremas de DS. Estas técnicas incluyen la reducción a *true* y el tránsito de la equivalencia, las cuales han sido usadas en el Capítulo 4 para derivar algunos teoremas de

DS. El resto de las técnicas se presenta en la Sección 5.5 que incluye, entre otras, las técnicas de suposición del antecedente, doble implicación y contradicción. Estas dos secciones están separadas por secciones que introducen nuevos conceptos y establecen propiedades importantes del sistema DS, y que son necesarias para el desarrollo del resto del capítulo. La Sección 5.3 presenta las *derivaciones relajadas*, un concepto que extiende el de derivación, y que facilita encontrar demostraciones de implicaciones y consecuencias lógicas por tránsito desde el antecedente al consecuente (y viceversa). La Sección 5.4 es especial porque presenta dos resultados importantes acerca del sistema DS. El primero de estos resultados es el Metateorema de la Deducción 5.9, una herramienta útil para demostrar implicaciones lógicas. El segundo resultado (se presenta sin demostración) es el Metateorema de Solvencia y Completitud 5.11; sin necesidad de exagerar, este metateorema puede ser el más importante de DS. Su importancia radica en establecer la coincidencia entre la relación semántica '$\models$' y la relación deductiva '$\vdash_{\mathsf{DS}}$'. En otras palabras, el Metateorema 5.11 establece que una proposición es tautología si y solo si esta es teorema de DS. En la práctica la coincidencia semántica y deductiva es útil para analizar proposiciones pues permite cambiar entre estos dos "mundos", como más convenga. Adicionalmente, la Sección 5.3 extiende el concepto de demostración de un sistema formal para aceptar demostraciones bajo un conjunto de suposiciones, en adición a los axiomas del sistema formal.

5.1. Eliminación de paréntesis

El lector posiblemente ha notado que algunas fórmulas de DS tienen una cantidad abrumadora de paréntesis; sin ellos, muchas expresiones son *ambigüas* en el sentido que una expresión puede tener más de un significado. Esta corta sección establece algunas convenciones para simplificar la escritura de proposiciones al permitir obviar paréntesis en muchos casos.

Los paréntesis son escenciales cuando una máquina es programada para reconocer una fórmula o, en general, una expresión como un programa en un lenguaje de programación. En el caso del lenguaje de programación LISP (cuyo nombre proviene del inglés "*LISt Processing*"), famoso por el uso extensivo de paréntesis en sus expresiones, algunos creen que el nombre del lenguaje realmente abrevia "*lots of unnecessary stupid parenthesis*". En contraste, si se tiene suficiente cuidado, para un humano no es necesario el uso extensivo de paréntesis, aún para leer código LISP. En el caso de una proposición, por ejemplo, algunos paréntesis pueden ser eliminados cuando dicha omisión no genera ambigüedad, es decir, cuando exista una única forma de interpretar la proposición resultante de eliminar los paréntesis.

Una forma de evitar algunos paréntesis es por medio de convenciones sintácticas sobre los conectivos lógicos. Al asignar una *precedencia* a cada uno de los conectivos lógicos de DS es posibe evitar la escritura de algunos paréntesis en las proposiciones.

Nota 5.1

Se adoptan las siguientes convenciones sobre la precedencia de los conectivos lógicos de DS:

- Las constantes *true* y *false* tienen más precedencia que $\neg$.
- El conectivo $\neg$ tiene más precedencia que $\vee$ y $\wedge$.
- Los conectivos $\vee$ y $\wedge$ tienen más precedencia que $\rightarrow$ y $\leftarrow$.
- Los conectivos $\rightarrow$ y $\leftarrow$ tienen más precedencia que $\equiv$ y $\not\equiv$.

Note que la precedencia de los conectivos lógicos es transitiva. Por ejemplo, de esta convención, se puede establecer que la negación tiene más precedencia que la implicación y que la discrepancia. Es importante observar que algunos conectivos no tienen precedencia sobre otros; por ejemplo, la disyunción y la conjunción tienen la misma precedencia. En estos casos de conectivos con la misma precedencia, la precedencia no ayuda a omitir paréntesis en 'combinaciones' de expresiones que los usen porque sin los paréntesis puede surgir ambigüedad.

Ejemplo 5.1

Dado que la disyunción tiene más precedencia que la implicación, la proposición $((p \vee q) \rightarrow r)$ se escribe de forma simplificada como $p \vee q \rightarrow r$. De manera similar, la expresión $((p \vee q) \wedge r)$ se escribe como $(p \vee q) \wedge r$. Sin embargo, esta combinación de conjunción y disyunción no se puede escribir como $p \vee q \wedge r$ porque esta última expresión es ambigüa: $(p \vee q) \wedge r$ y $p \vee (q \wedge r)$ tienen interpretaciones distintas.

Otro mecanismo que sirve el propósito de eliminar paréntesis es por medio de las propiedades algebraicas de los conectivos lógicos.

Ejemplo 5.2

Al saber que la equivalencia es asociativa, los paréntesis en

$$(p \equiv (q \equiv r))$$

son innecesarios y, entonces, esta proposición se escribe como

$$p \equiv q \equiv r.$$

Note que por una razón similar, al saber que la equivalencia es conmutativa además de ser asociativa, esta última expresión representa a su vez las proposiciones $((p \equiv q) \equiv r)$ o $(r \equiv p) \equiv q$, lo cual sugiere potencialmente cierto tipo de ambigüedad. Sin embargo, este no es el caso gracias a las propiedades algebraicas de la equivalencia: estas proposiciones son equivalentes y entonces no existe riesgo alguno de confundir o cambiar la semántica de la proposición original.

Una voz de precaución para el lector: no es prudente eliminar paréntesis de una proposición si la situación es tal que dicha omisión potencie la introducción de alguna ambigüedad.

Ejercicios

1. Elimine tantos paréntesis como sea posible de la siguientes proposiciones sin introducir ambigüedad:
   a) $((\phi \vee (\psi \vee \tau)) \equiv ((\phi \vee \psi) \vee \tau))$
   b) $((\phi \vee \psi) \equiv (\psi \vee \phi))$
   c) $((\phi \vee \mathit{false}) \equiv \phi)$
   d) $((\phi \vee \phi) \equiv \phi)$
   e) $((\phi \vee (\psi \equiv \tau)) \equiv ((\phi \vee \psi) \equiv (\phi \vee \tau)))$
   f) $((\neg\phi) \equiv (\phi \equiv \mathit{false}))$
   g) $(\mathit{false} \equiv (\neg \mathit{true}))$
   h) $((\neg \mathit{false}) \equiv \mathit{true})$
   i) $(\neg \mathit{false})$
   j) $((\neg(\phi \equiv \psi)) \equiv ((\neg\phi) \equiv \psi))$
   k) $(((\neg\phi) \equiv \psi) \equiv (\phi \equiv (\neg\psi)))$
   l) $((\neg(\neg\phi)) \equiv \phi)$
   m) $((\phi \equiv (\neg\phi)) \equiv \mathit{false})$
   n) $((\phi \vee \psi) \equiv ((\phi \vee (\neg\psi)) \equiv \phi))$
   ñ) $((\phi \wedge \psi) \equiv (\phi \equiv (\psi \equiv (\phi \vee \psi))))$
   o) $((\phi \wedge \phi) \equiv \phi)$
   p) $((\phi \wedge (\neg\phi)) \equiv \mathit{false})$
   q) $((\neg(\phi \wedge \psi)) \equiv ((\neg\phi) \vee (\neg\psi)))$
   r) $((\phi \wedge (\psi \vee \tau)) \equiv ((\phi \wedge \psi) \vee (\phi \wedge \tau)))$
   s) $((\phi \vee (\psi \wedge \tau)) \equiv ((\phi \vee \psi) \wedge (\phi \vee \tau)))$
   t) $((\phi \rightarrow \psi) \equiv ((\phi \vee \psi) \equiv \psi))$
   u) $((\phi \leftarrow \psi) \equiv (\psi \rightarrow \phi))$

v) $((\phi \to \mathit{false}) \equiv (\neg\phi))$
w) $((\phi \to (\psi \vee \tau)) \equiv ((\phi \to \psi) \vee (\phi \to \tau)))$
x) $((\phi \to (\psi \wedge \tau)) \equiv ((\phi \to \psi) \wedge (\phi \to \tau)))$
y) $((\phi \vee (\psi \to \phi)) \equiv (\psi \to \phi))$
z) $(((\phi \vee \psi) \to (\phi \wedge \psi)) \equiv (\phi \equiv \psi))$

2. Determine si alguna de las siguientes expresiones es ambigüa. De serlo, liste todas las posibles formas de parentizarla de manera tal que cada una de ellas resulte en una proposición con significado distinto:
   a) $p \vee q \wedge r$
   b) $p \wedge q \vee r$
   c) $p \to q \to r$
   d) $p \to q \leftarrow r$
   e) $p \leftarrow q \to r$
3. Justifique por qué la expresión $p \equiv q \not\equiv r$ no es ambigüa.
4. Demuestre que la equivalencia y la discrepancia son mutuamente asociativas.
5. Considere las siguientes expresiones en el lenguaje de DS; algunas de ellas no son fórmulas del sistema formal. Para cada una de ellas encuentre una parentización de tal manera que la proposición resultante sea una tautología:
   a) $\mathit{true} \vee p \wedge q$
   b) $p \equiv p \vee q$
   c) $p \to q \equiv r \equiv p \wedge q \equiv p \wedge r$
   d) $p \equiv q \not\equiv r \leftarrow \mathit{false} \wedge p$
   e) $\neg p \wedge p \equiv p \to r$

5.2. Técnicas básicas

Esta sección presenta un recuento de técnicas básicas para razonar acerca de equivalencias lógicas. Algunas de estas técnicas han sido previamente usadas en el Capítulo 4. Las técnicas discutidas en esta sección son de uso general y pueden ser útiles para demostrar *cualquier* teorema de DS.

5.2.1. Reducción a *true*. Considere la situación en la cual se quiere demostrar que una proposición ϕ es teorema. La técnica de *reducción a true* consiste en construir una derivación de la forma

$$\phi, \ldots, \mathit{true}.$$

Esta derivación establece $\vdash_{\mathsf{DS}} \phi \equiv \mathit{true}$ por el Metateorema 4.21. Dado que todo teorema de DS es equivalente a *true* (Metateorema 4.10.2), se concluye $\vdash_{\mathsf{DS}} \phi$.

Ejemplo 5.3

Considere el Teorema 4.33.1 de la reflexividad de la implicación; el objetivo es demostrar $\vdash_{\mathsf{DS}} \phi \to \phi$. Se propone la siguiente derivación:

$$\begin{aligned} & \phi \to \phi \\ \equiv \quad & \langle \text{ definición alterna de la implicación } \rangle \\ & \neg\phi \vee \phi \\ \equiv \quad & \langle \text{ tercero excluído } \rangle \\ & \mathit{true}. \end{aligned}$$

Dado que $\phi \to \phi$ es equivalente a *true*, se establece $\vdash_{\mathsf{DS}} \phi \to \phi$.

La técnica de reducción a *true* tiene ventajas y desventajas. Una ventaja es la claridad en su método: reducir ϕ a *true* por medio de simplificaciones que provienen de axiomas o teoremas previamente demostrados. Otra ventaja es que la información de la proposición a ser demostrada está disponible en cualquier paso de derivación, lo cual potencia las posibilidades de simplificación. En contraste, una desventaja de la técnica de reducción a *true* es que la derivación resultante puede ser verbosa dado que en cada paso de la derivación se puede repetir una y otra vez una misma subproposición. En este sentido, es preferible contar con una estrategia de demostración, aún cuando esto demande más tiempo antes de iniciar la demostración.

5.2.2. Tránsito. Dado que la equivalencia es un conectivo lógico importante en DS, en muchas ocasiones el objetivo de una demostración es una equivalencia lógica. En realidad, cualquier demostración de una proposición ϕ en DS puede plantearse como una demostración de $\phi \equiv \mathit{true}$ porque *true* es la identidad de la equivalencia.

Para demostrar que una equivalencia de la forma $\phi \equiv \psi$ es teorema de DS, tal y como su nombre lo indica, la técnica de *tránsito* tiene como objetivo construír una derivación de la forma

$$\phi, \ldots, \psi.$$

El Metateorema 4.21 establece entonces que $\vdash_{\mathsf{DS}} \phi \equiv \psi$. Cuando una de las proposiciones ϕ o ψ es la constante *true*, entonces las técnicas de reducción a *true* y tránsito coinciden. Ejemplos de derivaciones que usan la técnica de tránsito incluyen las demostraciones del Teorema 4.15.1 y del Teorema 4.24.2, entre otros.

La ventaja de la técnica de tránsito sobre la técnica de reducción a *true* es que las derivaciones resultantes pueden ser más sucintas. La razón es que si uno de los

dos operandos de una equivalencia exhibe una estructura más compleja que su contraparte, iniciar el tránsito desde esta proposición puede permitir simplificaciones significativas en algunos pasos de derivación.

5.2.3. Uso de lemas. En algunas ocasiones puede ser de ayuda separar una derivación larga o complicada en varias demostraciones.

Nota 5.2

Un *lema* representa un teorema auxiliar que sirve para demostrar otro teorema.

La palabra 'lema' tiene sus raíces en el griego y se entiende como 'teorema auxiliar'; una traducción más literal sería 'lo que se asume'. El proceso de identificar lemas en una demostración puede resaltar propiedades que de otra forma pueden resultar desapercibidas. La diferencia tácita entre un lema y un teorema es subjetiva pues depende del gusto de quien escribe o quien lee.

Nota 5.3

En este texto, un lema es un sustantivo que identifica una propiedad *auxiliar* que ayuda en la demostración de un teorema de interés.

La ventaja potencial de los lemas es que pueden reducir sustancialmente el esfuerzo de prueba porque pueden ser reutilizados una y otra vez. Esta situación es similarmente ventajosa a aquella en programación cuando se definen procedimientos que encapsulan cierto tipo de cálculos específicos y que pueden ser llamadas una y otra vez.

Ejemplo 5.4

Una propiedad que es laboriosa de establecer, es la asociatividad de la conjunción (Teorema 4.24.1). Se sugiere al lector tratar de construír una derivación por tránsito de esta propiedad antes de continuar con el ejemplo. Alternativamente, es posible encontrar: (i) una proposición τ que es equivalente al lado izquierdo de la equivalencia (Lema 1) y (ii) que el lado derecho de la equivalencia es equivalente a τ (Lema 2). Entonces, por la transitividad de la equivalencia, se puede establecer que el lado izquierdo y el lado derecho de la equivalencia son equivalentes. Se sugiere al lector obtener una demostración del Teorema 4.24.1 usando estas observaciones.

Ejercicios

1. Demuestre que si $\phi_0, \ldots, \phi_n, \mathit{true}$ es una derivación por reducción a *true*, entonces $\phi_0, \ldots, \phi_n$ es una demostración de DS.
2. Demuestre el Teorema 4.24.1 por reducción a *true*.
3. Demuestre el Teorema 4.24.1 por tránsito.
4. Demuestre el Teorema 4.24.1 usando lemas.

5.3. Derivaciones relajadas

Hasta ahora, la técnica de reducción a *true* (Sección 5.2.1) es la única alternativa para demostrar en DS que una implicación lógica es teorema. Esta sección presenta el concepto de *derivación relajada*, una extensión del concepto de derivación. La utilidad de una derivación relajada es que permite demostraciones de implicaciones y consecuencias lógicas por tránsito desde el antecedente al consecuente (y viceversa).

Es importante entender por qué una derivación no es útil, en general, para derivar por tránsito una implicación lógica. Por definición, en una derivación cualquier par de proposiciones consecutivas son equivalentes; de esta forma, apelando a la transitividad de la equivalencia, la primera y última proposiciones en una derivación son equivalentes. Por tanto, cuando dos proposiciones no son equivalentes, es imposible que haya una derivación por tránsito de una a la otra. Por ejemplo, es imposible que exista una derivación por tránsito de $p \wedge q$ a $p \vee q$, porque estas no son equivalentes a pesar de que estar relacionadas por la implicación lógica.

En una derivación relajada se 'relajan' los requisitos deductivos de una derivación para permitir pasos deductivos en donde es suficiente preservar la implicación lógica, una práctica muy común en la práctica de las matemáticas. De esta forma, y apelando a la transitividad de la implicación, con una derivación relajada se puede demostrar por tránsito que $p \wedge q$ implica $p \vee q$ (o, alternativamente, que $p \vee q$ es consecuencia lógica de $p \wedge q$).

Definición 5.4

Una secuencia de proposiciones $\phi_0, \ldots, \phi_n$ de DS es:

1. Una *derivación (relajada) de debilitamiento* si y solo si $\vdash_{\mathsf{DS}} \phi_{k-1} \to \phi_k$ para cualquier $0 < k \leq n$.

2. Una *derivación (relajada) de fortalecimiento* si y solo si $\vdash_{\mathsf{DS}} \phi_{k-1} \leftarrow \phi_k$ para cualquier $0 < k \leq n$.

Hay dos tipos de derivaciones relajadas: de debilitamiento (en donde predomina la implicación lógica) y de fortalecimiento (en donde predomina la consecuencia lógica). Dado que la implicación no es conmutativa, el orden de las proposiciones en una derivación relajada es importante.

Metateorema 5.5

Sean $\phi_0, \ldots, \phi_n$ proposiciones de DS.

1. Si $\phi_0, \ldots, \phi_n$ es una derivación de debilitamiento, entonces $\vdash_{\mathsf{DS}} \phi_0 \rightarrow \phi_n$.
2. Si $\phi_0, \ldots, \phi_n$ es una derivación de fortalecimiento, entonces $\vdash_{\mathsf{DS}} \phi_0 \leftarrow \phi_n$.

Demostración. A continuación se presenta una demostración para (1); una demostración para (2) se obtiene de forma similar y se propone como ejercicio para el lector. Sea $\phi_0, \ldots, \phi_n$ una derivación de debilitamiento. Se procede por inducción sobre $n \in \mathbb{N}$.

Caso base: si $n = 0$, entonces el objetivo es demostrar $\vdash_{\mathsf{DS}} \phi_0 \rightarrow \phi_0$ lo cual es cierto porque la implicación es reflexiva (Teorema 4.33.1).

Caso inductivo: se tiene que $\phi_0, \ldots, \phi_n, \phi_{n+1}$ es una derivación de debilitamiento; el objetivo es demostrar que $\vdash_{\mathsf{DS}} \phi_0 \rightarrow \phi_{n+1}$ usando como hipótesis inductiva $\vdash_{\mathsf{DS}} \phi_0 \rightarrow \phi_n$. Por la Definición 5.4.1 se tiene también $\vdash_{\mathsf{DS}} \phi_n \rightarrow \phi_{n+1}$. Como la implicación es transitiva (Teorema 4.33.2), se concluye $\vdash_{\mathsf{DS}} \phi_0 \rightarrow \phi_{n+1}$.

□

Es importante notar que cualquier derivación es simultáneamente una derivación de debilitamiento y una de fortalecimiento. La observación clave es que si $\phi \equiv \psi$ es teorema, también lo son $\phi \rightarrow \psi$ y $\phi \leftarrow \psi$ por el Teorema 4.31.4. Como consecuencia práctica, en una derivación de debilitamiento (respectivamente, de fortalecimiento) se pueden mezclar pasos de equivalencia y de implicación (respectivamente, de consecuencia) conjuntamente. Otra forma de justificar esta observación es con base en los teoremas 4.36.2-3.

En este punto es clave hacer una advertencia acerca de cómo usar correctamente derivaciones relajadas: una derivación relajada no puede resultar de mezclar

derivaciones de debilitamiento y de fortalecimiento. Note que, de acuerdo con la Definición 5.4, en una derivación relajada se pueden combinar pasos de equivalencia y de implicación ó de equivalencia y consecuencia, pero no pasos de implicación y consecuencia. Semánticamente, si se permitiera esta mezcla, se perdería completamente la relación de verdad entre la primera y última proposición en una derivación relajada (ver Ejercicio 5.3.3).

Nota 5.6

Un paso de derivación de debilitamiento de ϕ a ψ se puede diagramar esquemáticamente de la siguiente forma:

$$\begin{array}{ll} & \phi \\ \rightarrow & \langle \text{ explicación: por qué "} \vdash_{\mathsf{DS}} \phi \rightarrow \psi \text{" } \rangle \\ & \psi. \end{array}$$

De manera similar, un paso de derivación de fortalecimiento de ϕ a ψ se puede diagramar esquemáticamente de la siguiente forma:

$$\begin{array}{ll} & \phi \\ \leftarrow & \langle \text{ explicación: por qué "} \vdash_{\mathsf{DS}} \phi \leftarrow \psi \text{" } \rangle \\ & \psi. \end{array}$$

A continuación se presentan ejemplos del uso de derivaciones relajadas.

Ejemplo 5.5

Considere la siguiente derivación de $\vdash_{\mathsf{DS}} p \wedge q \rightarrow p \vee q$ (Teorema 4.35.3):

$$\begin{array}{ll} & p \wedge q \\ \rightarrow & \langle \text{ debilitamiento: Teorema 4.35.2 } \rangle \\ & p \\ \rightarrow & \langle \text{ debilitamiento: Teorema 4.35.1 } \rangle \\ & p \vee q. \end{array}$$

Por el Metateorema 5.5.1 se concluye $\vdash_{\mathsf{DS}} p \wedge q \rightarrow p \vee q$.

Ejemplo 5.6

Considere la siguiente derivación del axioma ($KA4$) del sistema formal KA (Ejercicio 4.7.50):

$$\begin{aligned} & \neg\phi \to \psi \\ \equiv \quad & \langle \text{ definición alternativa de } \to \rangle \\ & \neg\neg\phi \vee \psi \\ \leftarrow \quad & \langle \text{ fortalecimiento: Teorema 4.35.1} \rangle \\ & \neg\neg\phi. \end{aligned}$$

Por el Metateorema 5.5.2 se concluye $\vdash_{\mathsf{DS}} \neg\neg\phi \to (\neg\phi \to \psi)$.

Ejercicios

1. Demuestre el Metateorema 5.5.2.
2. Considere dos proposiciones ϕ y ψ de DS. ¿Es posible que exista una derivación de debilitamiento y otra de fortalecimiento de ϕ a ψ? Justifique su respuesta.
3. Proponga una secuencia de proposiciones en las cuales haya pasos de debilitamiento y fortalecimiento, de tal forma que dicha secuencia no sea una derivación relajada. Justifique su respuesta.
4. Usando como guía la derivación de debilitamiento en el Ejemplo 5.5, proponga una derivación de fortalecimiento para demostrar $\vdash_{\mathsf{DS}} p \wedge q \to p \vee q$.
5. Derive por tránsito de debilitamiento/fortalecimiento los siguientes teoremas:
   a) $\vdash_{\mathsf{DS}} \phi \to (\psi \to \phi)$.
   b) $\vdash_{\mathsf{DS}} ((\phi \to \psi) \to \phi) \to \phi$.
   c) $\vdash_{\mathsf{DS}} (\phi \to \psi) \to ((\psi \to \tau) \to (\phi \to \tau))$.
6. Derive por tránsito de debilitamiento/fortalecimiento los siguientes teoremas:
   a) $\vdash_{\mathsf{DS}} (\phi \to \psi) \to (\phi \vee \tau \to \psi \vee \tau)$.
   b) $\vdash_{\mathsf{DS}} (\phi \to \psi) \to (\phi \wedge \tau \to \psi \wedge \tau)$.
7. Demuestre las siguientes proposiciones:
   a) Una derivación es una derivación de debilitamiento.
   b) Una derivación es una derivación de fortalecimiento.
8. Refute las siguientes proposiciones:
   a) Una derivación de debilitamiento es una derivación.
   b) Una derivación de fortalecimiento es una derivación.

9. Sean $\phi_0, \ldots, \phi_n$ proposiciones de DS. Demuestre: $\phi_0, \ldots, \phi_n$ es una derivación de debilitamiento sii $\phi_n, \ldots, \phi_0$ es una derivación de fortalecimiento.
10. Considere el siguiente esquema:

$$\begin{array}{ll} & \phi \\ \rightarrow & \langle \ldots \rangle \\ & \psi \\ \leftarrow & \langle \ldots \rangle \\ & \tau \end{array}$$

Formule ejemplos de proposiciones concretas para ϕ, ψ, τ de tal forma que las siguientes afirmaciones sean ciertas y adapte el esquema de tal manera que no se usen conjuntamente implicaciones y consecuencias:

a) $\vdash_{\mathsf{DS}} \phi \rightarrow \tau$
b) $\vdash_{\mathsf{DS}} \phi \equiv \tau$
c) $\vdash_{\mathsf{DS}} \phi \leftarrow \tau$

5.4. Deducción con suposiciones y el Metateorema de la Deducción

Esta sección presenta dos resultados importantes del sistema DS, cada uno de ellos en la forma de un metateorema. El primero de ellos es el Metateorema de la Deducción el cual facilita la demostración de implicaciones lógicas en DS por medio del mecanísmo de suposición del antecedente. El segundo, que se presenta superficialmente, es el Metateorema de Coherencia y Completitud el cual establece que el concepto semántico de tautología y el concepto deductivo de teorema coinciden en DS. Estas dos propiedades de DS se presentan para el caso general de demostraciones con suposiciones, una extensión del concepto de demostración en un sistema formal.

5.4.1. Demostración con suposiciones. A continuación se presenta el concepto de demostración son suposiciones.

Definición 5.7

Sea Γ un conjunto de proposiciones de DS. Una *demostración con suposiciones en* Γ es una secuencia de proposiciones $\phi_0, \ldots, \phi_n$ de DS tal que para $0 \leq k \leq n$ una de las siguientes condiciones es cierta:

1. ϕ_k es un axioma de DS,

2. $k > 0$ y ϕ_k es la conclusión de una regla de inferencia de DS cuyas premisas aparecen en la secuencia $\phi_0, \ldots, \phi_{k-1}$, o
3. $\phi_k \in \Gamma$.

Si $\phi_0, \ldots, \phi_n$ es una demostración con suposiciones en Γ, entonces se dice que ϕ_n es un *teorema de* Γ *en* DS, lo cual se escribe como

$$\Gamma \vdash_{\mathsf{DS}} \phi_n.$$

La Definición 5.7 generaliza la definición de demostración en un sistema formal (Definición 0.10) al permitir usar información de un conjunto de suposiciones en un paso de inferencia (condición (3)). Cuando el conjunto de suposiciones Γ es vacío (i.e., $\Gamma = \{\}$), esta condición es inútil y entonces una demostración con suposiciones coincide con una demostración en un sistema formal. En consecuencia, se prefiere escribir $\vdash_{\mathsf{DS}} \phi$ cuando $\{\} \vdash_{\mathsf{DS}} \phi$.

Las definiciones de derivación, derivación de debilitamiento y derivación de fortalecimiento también se extendienden para usar un conjunto de suposiciones; la formulación de dichas extensiones se propone como ejercicios para el lector.

Nota 5.8

Intuitivamente, una demostración con suposiciones es la contraparte deductiva del concepto semántico de consecuencia tautológica. Esta observación se hace explícita al final de esta sección por medio del Metateorema de Coherencia y Completitud.

5.4.2. Metateorema de la Deducción. El Metateorema de la Deducción formaliza y justifica una técnica de demostración muy común en la práctica de la informática y de las matemáticas conocida como la técnica de demostración por suposición del antecedente. El Metateorema de la Deducción establece que para demostar una implicación lógica basta con construír una demostración del consecuente a partir de la información del antecedente.

Metateorema 5.9

Sean ϕ, ψ proposiciones de DS y Γ un conjunto de proposiciones de DS:

$$\text{si} \quad \Gamma \cup \{\psi\} \vdash_{\mathsf{DS}} \phi, \quad \text{entonces} \quad \Gamma \vdash_{\mathsf{DS}} \psi \to \phi.$$

El Metateorema de la Deducción establece que si una proposición ϕ se puede demostrar a partir de un conjunto de suposiciones $\Gamma \cup \{\psi\}$, entonces la implicación $\psi \to \phi$ se puede demostrar a partir de Γ. En el caso especial cuando $\Gamma = \{\}$, el metateorema de la deducción establece que si $\{\psi\} \vdash_{\mathsf{DS}} \phi$, entonces $\vdash_{\mathsf{DS}} \psi \to \phi$. Note que $\{\psi\} \vdash_{\mathsf{DS}} \phi$ es equivalente a decir que se demuestra ϕ suponiendo ψ como "axioma adicional".

Demostración. Sea $\phi_0, \ldots, \phi_n$ una demostración de ϕ con suposiciones en $\Gamma \cup \{\psi\}$ (note que $\phi_n = \phi$). El objetivo es demostrar que $\psi \to \phi$ es teorema de Γ en DS. La demostración procede por inducción sobre $n \in \mathbb{N}$.

Caso base: si $n = 0$, entonces ϕ es un axioma de DS o $\phi \in \Gamma \cup \{\psi\}$. En cualquiera de los dos casos se tiene $\Gamma \vdash_{\mathsf{DS}} \psi \to \phi$ (ejercicios 5.4.3.2 y 5.4.3.3).

Caso inductivo: si $\phi_0, \ldots, \phi_n, \phi_{n+1}$ es una demostración de ϕ con suposiciones en $\Gamma \cup \{\psi\}$ (note que $\phi_{n+1} = \phi$), suponga que la propiedad vale para todo $m \leq n$ (i.e., $\Gamma \vdash_{\mathsf{DS}} \psi \to \phi_m$ para $0 \leq m \leq n$). Si ϕ es un axioma de DS o $\phi \in \Gamma \cup \{\psi\}$, se tiene $\Gamma \vdash_{\mathsf{DS}} \psi \to \phi$ al igual que en el caso base. De lo contrario, ϕ se obtiene como conclusión de una regla de inferencia de DS:

Ecuanimidad: ϕ se obtiene con premisas $\phi_i = \gamma$ y $\phi_j = \gamma \equiv \phi$, para algún $0 \leq i \neq j \leq n$. Por la hipotesis inductiva, se tiene $\Gamma \vdash_{\mathsf{DS}} \psi \to \gamma$ y $\Gamma \vdash_{\mathsf{DS}} \psi \to (\gamma \equiv \phi)$. Considere la siguiente derivación con suposiciones Γ:

$$
\begin{array}{ll}
 & \psi \to (\gamma \equiv \phi) \\
\equiv & \langle \text{ distribución de } \to \text{ sobre } \equiv \rangle \\
 & \psi \to \gamma \equiv \psi \to \phi \\
\equiv & \langle \text{ hipótesis inductiva: } \Gamma \vdash_{\mathsf{DS}} \psi \to \gamma \text{; identidad de la equivalencia} \rangle \\
 & \psi \to \phi.
\end{array}
$$

Entonces, $\Gamma \vdash_{\mathsf{DS}} \psi \to \phi$.

Leibniz: ϕ se obtiene con premisa $\phi_k = \tau_0 \equiv \tau_1$, para algún $0 \leq k \leq n$. Entonces ϕ es de la forma $\gamma[p := \tau_0] \equiv \gamma[p := \tau_1]$. Por la hipótesis inductiva se tiene $\Gamma \vdash_{\mathsf{DS}} \psi \to (\tau_0 \equiv \tau_1)$. El objetivo es demostrar

$$\Gamma \vdash_{\mathsf{DS}} \psi \to (\gamma[p := \tau_0] \equiv \gamma[p := \tau_1]).$$

Esta demostración se puede obtener por inducción estructural sobre γ y se propone como ejercicio para el lector (Ejercicio 5.4.3.4).

En cualquiera de los casos se tiene que si $\Gamma \cup \{\psi\} \vdash_{\mathsf{DS}} \phi$, entonces $\Gamma \vdash_{\mathsf{DS}} \psi \to \phi$.

□

A pesar de parecer un resultado "obvio" y de ser usado por matemáticos durante siglos, el Metateorema de la Deducción fue demostrado (en su versión general) por

Herbrand y Tarski a principios del siglo XX. A continuación se presentan algunos ejemplos del uso del metateorema de la deducción.

Ejemplo 5.7

Considere el axioma ($L1$) del sistema formal L de Frege para la lógica proposicional (Ejercicio 4.7.47): $\phi \to (\psi \to \phi)$. Por el Metateorema de la Deducción basta con demostrar

$$\{\phi\} \vdash_{\mathsf{DS}} \psi \to \phi$$

o, *in extremis*, demostrar

$$\{\phi, \psi\} \vdash_{\mathsf{DS}} \phi.$$

Considere la siguiente derivación con suposiciones en $\{\phi, \psi\}$:

$$\begin{array}{ll} & \phi \\ \equiv & \langle \text{ suposición } \rangle \\ & \mathit{true}. \end{array}$$

Por el Metateorema de la Deducción se concluye $\vdash_{\mathsf{DS}} \phi \to (\psi \to \phi)$.

Ejemplo 5.8

Considere el Teorema 4.36.2:

$$\vdash_{\mathsf{DS}} (\phi \equiv \psi) \wedge (\psi \to \tau) \to (\phi \to \tau).$$

Por el Teorema 4.31.5, basta con demostrar:

$$\vdash_{\mathsf{DS}} (\phi \equiv \psi) \to ((\psi \to \tau) \to (\phi \to \tau)).$$

Considere la siguiente derivación con suposiciones en $\{\phi \equiv \psi, \psi \to \tau\}$:

$$\begin{array}{ll} & \phi \\ \equiv & \langle \text{ suposición: } \phi \equiv \psi \rangle \\ & \psi \\ \to & \langle \text{ debilitamiento: } \psi \to \tau \rangle \\ & \tau. \end{array}$$

Se tiene $\{\phi \equiv \psi, \psi \to \tau\} \vdash_{\mathsf{DS}} \phi \to \tau$, y por el Metateorema de la Deducción se concluye

$$\vdash_{\mathsf{DS}} (\phi \equiv \psi) \to ((\psi \to \tau) \to (\phi \to \tau)).$$

El converso del Metateorema de la Deducción también es cierto y se formula a continuación.

Metateorema 5.10

Sean ϕ, ψ proposiciones de DS y Γ un conjunto de proposiciones de DS:

$$\text{si} \quad \Gamma \vdash_{\mathsf{DS}} \psi \to \phi, \quad \text{entonces} \quad \Gamma \cup \{\psi\} \vdash_{\mathsf{DS}} \phi.$$

Demostración. Esta demostración es fácil de obtener y se propone como ejercicio para el lector. □

5.4.3. Metateorema de Coherencia y Completitud. Finalmente, se presenta (sin demostración, la cual está fuera del alcance de este capítulo) el Metateorema de Coherencia y Completitud. Este resultado es importante porque establece una coherencia entre la semántica de la lógica proposicional y el aparato deductivo de DS.

Metateorema 5.11

Sea Γ un conjunto de proposiciones de DS y ϕ una proposición de DS.

Coherencia: Si $\Gamma \vdash_{\mathsf{DS}} \phi$, entonces $\Gamma \models \phi$.

Completitud: Si $\Gamma \models \phi$, entonces $\Gamma \vdash_{\mathsf{DS}} \phi$.

En particular, si $\Gamma = \{\}$, entonces:

$$\vdash_{\mathsf{DS}} \phi \quad \text{sii} \quad \models \phi.$$

El Metateorema de Coherencia y Completitud establece que analizar una proposición con métodos semánticos o con métodos deductivos en DS es igualmente potente. Coherencia significa que todo teorema de Γ es una consecuencia tautológica de Γ; completitud siginifica que para cualquier consecuencia tautológica de Γ hay una demostración en DS con suposiciones en Γ. En particular, cuando el conjunto de suposiciones Γ es vacío, la coherencia indica que todo teorema es una tautología y la completitud indica que toda tautología es teorema.

Ejercicios

1. Basándose en la Definición 5.7, proponga definiciones para:
   a) Derivación con suposiciones.

b) Derivación de debilitamiento con suposiciones.
c) Derivación de fortalecimiento con suposiciones.

2. Sean Γ un conjunto de proposiciones de DS y ϕ una proposición. Demuestre que si ϕ es teorema de DS, entonces ϕ es teorema de Γ en DS. Es decir,

   $$\text{si } \vdash_{\mathsf{DS}} \phi, \quad \text{entonces} \quad \Gamma \vdash_{\mathsf{DS}} \phi.$$

3. Sean Γ un conjunto de proposiciones, y ϕ, ψ proposiciones de DS. Demuestre que si $\phi \in \Gamma \cup \{\psi\}$, entonces $\Gamma \vdash_{\mathsf{DS}} \psi \to \phi$.

4. Sea Γ un conjunto de proposiciones de DS, y $\phi, \psi, \gamma, \tau_0, \tau_1$ proposiciones de DS. El objetivo es demostrar que si $\Gamma \vdash_{\mathsf{DS}} \psi \to (\tau_0 \equiv \tau_1)$, entonces:

   $$\Gamma \vdash_{\mathsf{DS}} \psi \to (\gamma[p := \tau_0] \equiv \gamma[p := \tau_1])$$

   Una demostración se puede obtener por inducción sobre la estructura de γ; lleve a cabo los siguientes pasos:
   - Demuestre que la propiedad es cierta cuando γ es una constante.
   - Demuestre que la propiedad es cierta cuando γ es una variable proposicional (considere dos casos, cuando es p y cuando no es p).
   - Demuestre que la propiedad es cierta cuando γ es de la forma $\gamma_0 \equiv \gamma_1$ suponiendo que la propiedad es cierta para γ_0 y γ_1.
   - Demuestre que la propiedad es cierta cuando γ es de la forma $\gamma_0 \vee \gamma_1$ suponiendo que la propiedad es cierta para γ_0 y γ_1.

   Dado que en DS los conectivos lógicos $\mathit{true}, \mathit{false}, \equiv, \vee$ bastan para axiomatizar cualquier otro conectivo lógico, los casos anteriores son suficientes para establecer la veracidad de la afirmación inicial.

5. Sean Γ y Δ conjuntos de proposiciones de DS, y ϕ, ψ proposiciones de DS. Demuestre:
   a) Si $\Gamma \subseteq \Delta$ y $\Gamma \vdash_{\mathsf{DS}} \phi$, entonces $\Delta \vdash_{\mathsf{DS}} \phi$.
   b) Si $\Gamma \cup \{\psi\} \vdash_{\mathsf{DS}} \phi$ y $\Gamma \vdash_{\mathsf{DS}} \psi$, entonces $\Gamma \vdash_{\mathsf{DS}} \phi$.

6. Sea Γ un conjunto de proposiciones de DS y ϕ una proposición de DS. Demuestre que si $\Gamma \vdash_{\mathsf{DS}} \phi$ y cada una de las proposiciones en Γ es teorema de DS, entonces $\vdash_{\mathsf{DS}} \phi$.

7. Demuestre el Metateorema 5.10.

8. Sean Γ un conjunto de proposiciones y ϕ, ψ proposiciones de DS. Demuestre:
   a) Si $\Gamma \vdash_{\mathsf{DS}} \phi$, entonces $\Gamma \vdash_{\mathsf{DS}} \phi \vee \psi$.
   b) Si $\Gamma \vdash_{\mathsf{DS}} \phi \wedge \psi$, entonces $\Gamma \vdash_{\mathsf{DS}} \phi$.
   c) Si $\Gamma \vdash_{\mathsf{DS}} \phi$ y $\Gamma \vdash_{\mathsf{DS}} \psi$, entonces $\Gamma \vdash_{\mathsf{DS}} \phi \wedge \psi$.

9. Sean $\phi_0, \ldots, \phi_n$ tautologías. Demuestre que si $\{\phi_0, \ldots, \phi_n\} \models \phi$, entonces ϕ es teorema de DS.

10. Demuestre: $\Gamma \vdash_{\mathsf{DS}} \phi$ si y solo si $\Gamma \cup \{\neg\phi\}$ es insatisfacible.

11. Sea Γ un conjunto de proposiciones satisfacible. Demuestre o refute:
    a) Si $\Gamma \vdash_{\mathsf{DS}} \phi \vee \psi$, entonces $\Gamma \cup \{\neg\phi\}$ o $\Gamma \cup \{\neg\psi\}$ es satisfacible.
    b) Si $\Gamma \vdash_{\mathsf{DS}} \neg(\phi \vee \psi)$, entonces $\Gamma \cup \{\neg\phi, \neg\psi\}$ es satisfacible.

5.5. Técnicas complementarias

Esta sección presenta técnicas complementarias para razonar sobre proposiciones. A diferencia de las técnicas inicialmente presentadas en la Sección 5.2, las técnicas en esta sección son más generales y no están reservadas únicamente para razonar sobre la equivalencia lógica. De hecho, la mayoría de las técnicas presentadas en esta sección son especialmente útiles para razonar sobre implicaciones lógicas o, en su defecto, están basadas en la implicación lógica. Como ejemplos, esta sección hace uso extensivo de teorías como la artimética y los conjuntos, apelando a definiciones básicas e intuitivas.

Es importante resaltar que las técnicas presentadas en esta sección son de caracter general en un sentido complementario: pueden ser usadas en cualquier otro sistema formal de la lógica proposicional y no son exclusivas de DS. En particular, el material de esta sección sirve para sustentar algunas técnicas de demostración que comúnmente se encuentran en textos y que hasta ahora pueden ser inaccesibles para el lector.

5.5.1. Suposición del antecedente. Una práctica común para demostrar una implicación lógica es usar la técnica de suposición del antecedente. Esta técnica consiste en demostrar ϕ bajo la suposición de ψ, para concluír que $\psi \to \phi$ es cierto. La técnica de demostración por suposición del antecedente también se conoce como *demostración constructiva*: se trata de construir una demostración a partir de una lista de axiomas y suposiciones.

Metateorema 5.12

Sean Γ una colección de proposiciones y ϕ, ψ proposiciones:

$$\Gamma \vdash_{\mathsf{DS}} \psi \to \phi \qquad \text{sii} \qquad \Gamma \cup \{\psi\} \vdash_{\mathsf{DS}} \phi.$$

La demostración por suposición del antecedente se justifica desde el Metateorema de la Deducción y su converso (metateoremas 5.9 y 5.10). Para demostrar que $\psi \to \phi$ es un teorema de Γ, la técnica de suposición del antecedente se puede usar de la siguiente forma:

1. suponer que ψ es un teorema de Γ, extendiendo el conjunto de suposiciones de Γ a $\Gamma \cup \{\psi\}$, y

2. demostrar que ϕ es teorema de $\Gamma \cup \{\psi\}$.

A continuación se presenta un ejemplo sobre una propiedad de paridad de los números enteros $\mathbb{Z}$ que usa la técnica de suposición del antecedente. Para ello, para cualquier número entero $a \in \mathbb{Z}$, se introduce la proposición $impar(a)$ de la siguiente manera:

$$impar(a) : \text{“hay un } c_a \in \mathbb{Z} \text{ tal que } a = 2c_a + 1 \text{ (i.e., } a \text{ es } impar\text{).”}$$

Ejemplo 5.9

Si a y b son números enteros consecutivos, entonces $a + b$ es impar.

Demostración. A continuación se especifica el objetivo de la demostración en el lenguaje del sistema DS :

$$\vdash_{\mathsf{DS}} (a = b + 1) \to impar(a + b)$$

Esta especificación establece un ordeden entre a y b, lo cual no importa porque la suma de números enteros es conmutativa. En favor de la brevedad, se abusa la notación dado que los símbolos $+, =, impar$ no hacen parte de DS. Al suponer el antecedente, basta con demostrar (paso 1):

$$\{a = b + 1\} \vdash_{\mathsf{DS}} impar(a + b).$$

Considere la siguiente derivación en DS (paso 2):

$$\begin{aligned}
& impar(a + b) \\
\equiv \quad & \langle \text{ suposición: } a = b + 1 \rangle \\
& impar((b + 1) + b) \\
\equiv \quad & \langle \text{ aritmética} \rangle \\
& impar(2b + 1) \\
\equiv \quad & \langle \text{ definición de } impar \text{ con testigo } c_{a+b} = b \rangle \\
& true.
\end{aligned}$$

Por el Meateorema 5.12 se concluye $\vdash_{\mathsf{DS}} (a = b + 1) \to impar(a + b)$.

5.5.2. Doble implicación. En matemáticas e informática es común encontrar propiedades que se especifican como una equivalencia lógica. Una práctica efectiva para demostrar una equivalencia lógica es hacerlo usando la técnica de doble implicación. Esta técnica consiste en demostrar indirectamente una proposición de la forma $\phi \equiv \psi$ por medio de las demostraciones de $\phi \to \psi$ y $\psi \to \phi$. En principio se duplican los esfuerzos de demostración al cambiar un objetivo por dos. Sin embargo, cada uno de los dos nuevos objetivos es una implicación y, entonces,

pueden ser abordado usando, por ejemplo, la técnica de suposición del antecedente (Sección 5.5.1), y esto puede resultar conveniente.

Metateorema 5.13

Sean Γ un conjunto de proposiciones y ϕ, ψ proposiciones:

$$\Gamma \vdash_{\mathsf{DS}} \phi \equiv \psi \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}} \phi \to \psi \quad \text{y} \quad \Gamma \vdash_{\mathsf{DS}} \psi \to \phi.$$

La técnica de suposición del antecedente (Metateorema 5.13) basa su corrección en el Teorema 4.31.3. Para demostrar que $\phi \equiv \psi$ es un teorema de Γ con la técnica de doble implicación hay exactamente dos pasos:

1. demostrar que $\phi \to \psi$ es teorema de Γ y
2. demostrar que $\psi \to \phi$ es teorema de Γ.

A continuación se presenta un ejemplo de la teoría de conjuntos para ilustrar la estructura de una demostración por doble implicación. Para ello se supone que el lector está familiarizado con los símbolos $\in, =, \subseteq, \cup, \cap$ de la teoría de conjuntos. La expresión '$x \in A$' denota la *pertenencia* del elemento x al conjunto A. La expresión '$A = B$' denota la relación de *igualdad* de conjuntos y se axiomatiza con la proposición $(x \in A) \equiv (x \in B)$ para cualquier elemento x. La expresión '$A \subseteq B$' denota la relación de *inclusión* de conjuntos y se axiomatiza con la proposición $(x \in A) \to (x \in B)$ para cualquier elemento x. La expresión '$A \cup B$' denota la operación de *unión* de conjuntos y la proposición '$x \in (A \cup B)$' se axiomatiza como $(x \in A) \vee (x \in B)$ para cualquier elemento x. La expresión '$A \cap B$' denota la operación de *intersección* de conjuntos y la proposición '$x \in (A \cap B)$' se axiomatiza como $(x \in A) \wedge (x \in B)$ para cualquier elemento x.

Ejemplo 5.10

Sean A y B conjuntos: $A \cup B = B$ sii $A \subseteq B$.

Demostración. A continuación se presenta una especificación de la propiedad que se quiere demostrar:

$$\vdash_{\mathsf{DS}} (A \cup B = B) \equiv (A \subseteq B).$$

Usando la técnica de doble implicación, basta con demostrar:

1. $\vdash_{\mathsf{DS}} (A \cup B = B) \to (A \subseteq B)$.
2. $\vdash_{\mathsf{DS}} (A \subseteq B) \to (A \cup B = B)$.

A continuación se presenta una demostración de (1); una demostración de (2) se propone como ejercicio para el lector. Usando la técnica de suposición del antecedente, basta con demostrar para cualquier x:

$$\{A \cup B = B, x \in A\} \vdash_{\mathsf{DS}} x \in B.$$

Considere la siguiente derivación:

$$\begin{array}{ll} & x \in B \\ \equiv & \langle \text{ suposición: } B = A \cup B \rangle \\ & x \in (A \cup B) \\ \equiv & \langle \text{ pertenencia a la unión de dos conjuntos} \rangle \\ & (x \in A) \vee (x \in B) \\ \equiv & \langle \text{ suposición } x \in A \text{; anulador de la disyunción} \rangle \\ & \mathit{true}. \end{array}$$

5.5.3. Contradicción. Demostrar por contradicción que una proposición es cierta, está basado en el hecho de que dicha proposición es verdadera o falsa (Teorema 4.19.1, tercero excluído), pero no puede ser verdadera y falsa a la vez. Se encuentra una contradicción cuando se establece que una proposición es verdadera y falsa a la vez, indicando que alguna de las suposiciones es incorrecta. La técnica de demostración por contradicción puede ser usada para demostrar cualquier tipo de proposiciones y es especialmente útil para demostrar implicaciones lógicas.

Metateorema 5.14

Sean Γ un conjunto de proposiciones y ϕ, ψ proposiciones:

1. $\Gamma \vdash_{\mathsf{DS}} \phi$ sii $\Gamma \vdash_{\mathsf{DS}} \neg\phi \to \mathit{false}$.
2. $\Gamma \vdash_{\mathsf{DS}} \psi \to \phi$ sii $\Gamma \vdash_{\mathsf{DS}} \psi \wedge \neg\phi \to \mathit{false}$.

La técnica de demostración por contradicción (Metateorema 5.14) se usó previamente en el Ejemplo 4.2 para demostrar que $\sqrt{2}$ es irracional. Dicho ejemplo se desarrolla usando el Metateorema 5.14.1 al negar la proposición que se quiere demostrar (i.e., negando que $\sqrt{2}$ es irracional, es decir, suponiendo que $\sqrt{2}$ es racional) y estableciendo una contradicción ($\sqrt{2}$ no se puede expresar como una fracción).

Nota 5.15

Observe que en DS una contradicción puede representarse como $\phi \equiv \neg\phi$ o $\phi \wedge \neg\phi$, para cualquier proposición ϕ.

En general, para aplicar el Metateorema 5.14.1 queriendo demostrar que proposición ϕ es un teorema de Γ, se usan los siguientes pasos:

1. suponer que $\neg\phi$ es cierto (i.e., extender Γ a $\Gamma \cup \{\neg\phi\}$) y
2. demostrar que *false* (i.e., una contradicción) es teorema de $\Gamma \cup \{\neg\phi\}$.

De forma similar, para usar el Metateorema 5.14.2 queriendo demostrar que una implicación $\psi \to \phi$ es un teorema de Γ, se usa el siguiente método:

1. suponer que ψ y $\neg\phi$ son ciertos (i.e., extender Γ a $\Gamma \cup \{\psi, \neg\phi\}$) y
2. demostrar que *false* (i.e., una contradicción) es teorema de $\Gamma \cup \{\psi, \neg\phi\}$.

Ejemplo 5.11

Si a y b son enteros impares, entonces $a + b$ es par.

Demostración. Una especificación de la proposición puede ser la siguiente:

$$\vdash_{\mathsf{DS}} \mathit{impar}(a) \wedge \mathit{impar}(b) \to \mathit{par}(a + b),$$

en donde $\mathit{par}(x) \equiv \neg\mathit{impar}(x)$ para cualquier número x. Aplicando el Metateorema 5.14.2, basta con suponer $\{\mathit{impar}(a), \mathit{impar}(b), \mathit{impar}(a + b)\}$ y llegar a una contradicción:

$$\begin{aligned}
& \mathit{true} \\
\equiv\ & \langle \text{ suposición: } \mathit{impar}(a + b) \rangle \\
& \mathit{impar}(a + b) \\
\equiv\ & \langle \text{ suposición: } \mathit{impar}(a) \text{ e } \mathit{impar}(b) \rangle \\
& \mathit{impar}((2c_a + 1) + (2c_b + 1)) \\
\equiv\ & \langle \text{ aritmética} \rangle \\
& \mathit{impar}(2(c_a + c_b + 1)) \\
\equiv\ & \langle \text{ definición de } \mathit{impar}\text{: } 2(c_a + c_b + 1) \text{ no es impar} \rangle \\
& \mathit{false}.
\end{aligned}$$

5.5.4. Contrapositiva. De acuerdo con el Teorema 4.31.1, es cierto que una implicación $\psi \to \phi$ es equivalente a $\neg\phi \to \neg\psi$. Recuerde que la proposición

$\neg\phi \to \neg\psi$ se denomina la *contrapositiva* de $\psi \to \phi$. La técnica de demostración por contrapositiva consiste en demostrar indirectamente que una implicación es teorema al demostrar que su contrapositiva es teorema.

Metateorema 5.16

Sean Γ un conjunto de proposiciones y ϕ, ψ proposiciones:

$$\Gamma \vdash_{\mathsf{DS}} \psi \to \phi \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}} \neg\phi \to \neg\psi.$$

La técnica de demostración por contrapositiva (Metateorema 5.16) a veces no es considerada como una técnica en si misma, sino más bien como un mecanismo que permite usar otras técnicas. Note que la contrapositiva de una implicación lógica es también una implicación lógica y por tanto otras técnicas pueden ser usadas conjuntamente como, por ejemplo, suposición del antecedente o contradicción.

Aunque aparentemente es fácil, se debe tener cuidado especial al transformar una implicación en su contrapositiva. Por ejemplo, la contrapositiva de la proposición "si x es impar, entonces $5x$ es impar" es la proposición "si $5x$ no es impar, entonces x no es impar" o, mejor, "si $5x$ es par, entonces x es par". Determinar la contrapositiva de una proposición puede ser difícil cuando dicha proposición exhibe una estructura con varios conectivos lógicos. En general, existen variantes a la demostración por contrapositiva; a continuación se presentan algunas de ellas.

Metateorema 5.17

Sean Γ un conjunto de proposiciones y $\phi, \phi_0, \phi_1, \psi, \psi_0, \psi_1$ proposiciones:

1. $\Gamma \vdash_{\mathsf{DS}} \phi \to \psi_0 \vee \psi_1$ sii $\Gamma \vdash_{\mathsf{DS}} \phi \wedge \neg\psi_1 \to \psi_0$.
2. $\Gamma \vdash_{\mathsf{DS}} \phi_0 \wedge \phi_1 \to \psi$ sii $\Gamma \vdash_{\mathsf{DS}} \phi_0 \to \psi \vee \neg\phi_1$.
3. $\Gamma \vdash_{\mathsf{DS}} \phi_0 \wedge \phi_1 \to \psi_0 \vee \psi_1$ sii $\Gamma \vdash_{\mathsf{DS}} \phi_0 \wedge \neg\psi_1 \to \psi_0 \vee \neg\phi_1$.

Por ejemplo, una contrapositiva de la proposición "si x es impar y y es par, entonces xy es par" es la proposición "si xy es impar, entonces x es par o y es impar" (¿por qué?). En este caso, se podría suponer que xy es impar (técnica de suposición del antecedente) y demostrar que al menos uno de x e y es impar.

Ejemplo 5.12

Si $a + b$ es par, entonces a y b no son consecutivos.

Demostración. El objetivo es demostrar:

$$\vdash_{\mathsf{DS}} par(a+b) \to (a \neq b+1).$$

Al igual que en Ejemplo 5.9, y sin que esto sea importante para la demostración, se fija un orden entre a y b. Note que por el Metateorema 5.16 basta con demostrar:

$$\vdash_{\mathsf{DS}} (a = b+1) \to impar(a+b),$$

lo cual es la conclusión del Ejemplo 5.9.

En general, es difícil establecer unívocamente si es más fácil demostrar una implicación o una de sus contrapositivas. En principio, vale la pena primero tratar de encontrar una demostración de dicha implicación en su forma original. Si se falla, entonces la demostración de 'una' de sus contrapositivas puede ser una buena alternativa.

5.5.5. Análisis de casos. El siguiente texto se toma prestado de Wikipedia y es traducido al castellano por el autor:

> *"El análisis por casos es uno de los métodos más generales y aplicables en el pensamiento analítico. Este depende únicamente de la división de un problema, decisión o situación en una cantidad suficiente de subproblemas (o casos). El análisis de cada uno de los casos puede ser suficiente para resolver el problema original."*

El principio de análisis por casos se usa en una observación famosa de Sherlock Holmes, explicando que cuando se ha eliminado lo imposible, lo que reulte debe ser verdadero, por muy inverosímil que parezca. Las raíces lógicas de la observación de Holmes están basadas en el principio del tercero excluído (¿por qué?).

En este texto se distinguen dos principios de análisis por casos:

- Análisis de casos sobre una variable proposicional.
- Análisis de casos exhaustivos.

El análisis de casos sobre una variable proposicional para una proposición ϕ que se quiere demostrar como teorema, consiste en demostrar que las proposiciones $\phi[p := true]$ y $\phi[p := false]$ son teoremas, para alguna variable proposicional p. Semánticamente esto es completamente intuitivo porque si $\phi[p := true]$ y $\phi[p := false]$ son teoremas, entonces son tautologías. En consecuencia ϕ es una tautología y, por ende, un teorema.

Metateorema 5.18

Sean Γ una colección de proposiciones y ϕ una proposición:

$$\Gamma \vdash_{\mathsf{DS}} \phi \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}} \phi[p := \mathit{true}] \quad \text{y} \quad \Gamma \vdash_{\mathsf{DS}} \phi[p := \mathit{false}]\,.$$

El Metateorema 5.18 está basado en una propiedad de la lógica proposicional conocida como la *Regla de Shannon*.

Teorema 5.19

Sean p una variable proposicional y ϕ una proposición:

$$\vdash_{\mathsf{DS}} \phi \equiv (p \to \phi[p := \mathit{true}]) \wedge (\neg p \to \phi[p := \mathit{false}]).$$

La demostración del Teorema 5.19 se propone como ejercicio para el lector.

En general, para aplicar el Metateorema 5.18 queriendo demostrar que proposición ϕ es un teorema de Γ, se usa el siguiente método:

1. demostrar que $\phi[p := \mathit{true}]$ es teorema de Γ y
2. demostrar que $\phi[p := \mathit{false}]$ es teorema de Γ.

Ejemplo 5.13

Considere el Teorema 4.24.1 de asociatividad de la conjunción:

$$\vdash_{\mathsf{DS}} (p \wedge (q \wedge r)) \equiv ((p \wedge q) \wedge r).$$

El Metateorema 5.18 establece que basta con demostrar, por ejemplo:

$$\vdash_{\mathsf{DS}} (\mathit{true} \wedge (q \wedge r)) \equiv ((\mathit{true} \wedge q) \wedge r) \quad \text{y}$$
$$\vdash_{\mathsf{DS}} (\mathit{false} \wedge (q \wedge r)) \equiv ((\mathit{false} \wedge q) \wedge r).$$

Estas dos demostraciones se proponen como ejercicio para el lector.

El segundo principio de análisis de casos está basado en la siguiente observación:

> *"Si ψ_0 implica ϕ, ψ_1 implica ϕ y $\psi_0 \vee \psi_1$ es cierto, entonces ϕ debe ser cierto."*

El detalle más importante en este tipo de anásis por casos para demostar que una proposición ϕ es teorema, es que los casos ψ_0 y ψ_1 sean colectivamente *exhaustivos*,

i.e., $\psi_0 \vee \psi_1$ debe ser teorema. Si este es el "caso", basta con demostrar que en cada uno de los casos ψ_0 y ψ_1 la proposición ϕ es cierta.

En una demostración exahustiva por análisis de casos no es necesario restringirse a dos casos únicamente. En general, puede haber más de dos casos. Antes de presentar formalmente el análisis de casos exhaustivo, se introduce notación auxiliar.

Definición 5.20

Sean $\phi_0, \ldots, \phi_n$ proposiciones:

- La expresión $\bigvee_{i=0}^{n} \phi_i$, llamada *disyunción generalizada* (o *disyuntoria*), abrevia la proposición

  $$\phi_0 \vee \cdots \vee \phi_n.$$

- La expresión $\bigwedge_{i=0}^{n} \phi_i$, llamada *conjunción generalizada* (o *conjuntoria*), abrevia la proposición

  $$\phi_0 \wedge \cdots \wedge \phi_n.$$

Una diyunción generalizada es una secuencia *finita* de proposiciones operada bajo la disyunción y una conjunción generalizada es una secuencia *finita* de proposiciones operada bajo la conjunción. Note que el orden en cada operación generalizada es inmaterial porque la disyunción y la conjunción son conectivos lógicos asociativos y conmutativos.

Metateorema 5.21

Sean Γ una colección de proposiciones y $\phi, \psi_0, \ldots, \psi_n$ proposiciones. Si

1. $\Gamma \vdash_{\mathsf{DS}} \bigvee_{i=0}^{n} \psi_i$ y
2. $\Gamma \vdash_{\mathsf{DS}} \psi_i \to \phi$, para cada $0 \leq i \leq n$,

entonces $\Gamma \vdash_{\mathsf{DS}} \phi$.

En el Metateorema 5.21 la variable n indica la cantidad de casos a ser considerados: $n = 0$ indica un caso, $n = 1$ indica dos casos, etc. Se llama la atención del lector sobre el hecho de que este metateorema se formula como una implicación, a diferencia de los demás metateoremas vistos en esta sección que han sido formulados como una equivalencia lógica. Finalmente, note que cuando $n = 0$, este metateorema corresponde a una versión de la regla de inferencia Modus Ponens (Ejercicio 4.7.42) con suposiciones.

Ejemplo 5.14

Si $a \in \mathbb{Z}$, entonces $a^2 \neq 5$.

Demostración. Considere los siguientes tres casos:

$$\psi_0 : a < -2, \qquad \psi_1 : -2 \leq a \leq 2, \qquad \psi_2 : 2 < a.$$

Los casos ψ_0, ψ_1, ψ_2 son exhaustivos sobre a (¿por qué?). Por el Metateorema 5.21 basta con demostrar que las implicaciones $\psi_i \to \phi$, para $i = 0, 1, 2$, son teoremas. Estas tres demostraciones se proponen como ejercicios para el lector.

El análisis de casos exhaustivo puede adaptarse para demostrar una implicación cuando su estructura interna lo permite. A continuación se formulan dos principios de análisis por casos cuando el objetivo es demostrar una implicación y esta exhíbe una estructura interna especial.

Metateorema 5.22

Sean Γ una colección de proposiciones y $\phi, \psi_0, \ldots, \psi_n$ proposiciones:

1. $\Gamma \vdash_{\mathsf{DS}} (\bigvee_{i=0}^{n} \psi_i) \to \phi$ sii $\Gamma \vdash_{\mathsf{DS}} \psi_i \to \phi$, para cada $0 \leq i \leq n$.
2. $\Gamma \vdash_{\mathsf{DS}} \phi \to (\bigwedge_{i=0}^{n} \psi_i)$ sii $\Gamma \vdash_{\mathsf{DS}} \phi \to \psi_i$, para cada $0 \leq i \leq n$.

El Metateorema 5.22 indica cómo se pueden demostrar implicaciones cuando el antecedente es una disyunción de proposiciones (5.22.1) y cuando el consecuente es una conjunción de proposiciones (5.22.2).

A continuación se presenta un ejemplo sobre una propiedad de divisibilidad de los números enteros $\mathbb{Z}$. Para ello, para cualquier par de números enteros $a, b \in \mathbb{Z}$, se introduce la proposición '$a \cdot| \, b$' de la siguiente manera:

$$a \cdot| \, b : \text{“hay un } c_{ab} \in \mathbb{Z} \text{ tal que } b = ac_{ab} \text{ (i.e., } b \text{ es } \textit{múltiplo} \text{ de } a).\text{”}$$

Ejemplo 5.15

Si $a \in \mathbb{Z}$ no es múltiplo de 3, entonces $a^2 - 1$ es múltiplo de 3.

Demostración. El objetivo es demostrar:

$$\vdash_{\mathsf{DS}} \neg(3 \cdot| \, a) \to 3 \cdot| \, (a^2 - 1).$$

Al hablar de múltiplos de 3, note que cualquier a se puede escribir de alguna de las siguientes formas para alguna constante $c \in \mathbb{Z}$:

$$a = 3c, \qquad a = 3c + 1 \qquad \text{o} \qquad a = 3c + 2.$$

Dado que a no es múltiplo de 3, se descarta el caso $a = 3c$. Entonces, el objetivo de la demostración se simplifica a demostrar para algún $c \in \mathbb{Z}$:

$$\vdash_{\mathsf{DS}} (a = 3c + 1) \vee (a = 3c + 2) \to 3 \cdot | \, (a^2 - 1).$$

Usando el Metateorema 5.22.1, basta con demostrar:

1. $\vdash_{\mathsf{DS}} (a = 3c + 1) \to 3 \cdot | \, (a^2 - 1)$.
2. $\vdash_{\mathsf{DS}} (a = 3c + 2) \to 3 \cdot | \, (a^2 - 1)$.

Estas dos demostraciones se proponen como ejercicio para el lector.

En la práctica es preferible evitar demostraciones por análisis de casos cuando la cantidad de casos es significativa. El análisis de casos, en estas situaciones, es poco elegante. Otro motivo, para nada estético, es que la cantidad de posibles errores en una demostración aumenta con la cantidad de casos considerados. En general, cuando un teorema se demuestra con muchos casos, existe la percepción de que dicho teorema es una coincidencia y no una propiedad fundamental. Sin embargo, hay algunos teoremas importantes para los cuales únicamente se conocen demostraciones por casos exhaustivos; esto no los hace menos importantes o útiles. El teorema de los 4 colores y la conjetura de Kepler hacen parte de este exclusivo grupo de teoremas (para más información, ver ejercicios al final de esta sección).

Ejercicios

1. Demuestre el Metateorema 5.12.
2. Explique, paso a paso, por qué en la demostración de la propiedad (1) en el Ejemplo 5.10 basta con demostrar para cualquier x:

   $$\{A \cup B = B, x \in A\} \vdash_{\mathsf{DS}} x \in B.$$

3. Complete el Ejemplo 5.10 con la demostración de la propiedad (2).
4. Sean A, B y C conjuntos. Demuestre:
   a) $A \cap B \subseteq A$.
   b) $A \subseteq A \cup B$.
   c) $(C \subseteq A) \wedge (C \subseteq B) \equiv C \subseteq (A \cap B)$.

5. Sean A y B conjuntos. En cada uno de los siguientes casos establezca si la condición dada es suficiente para establecer la igualdad de A y B (i.e, $A = B$):
   a) $A \subseteq B$ y $B \subseteq A$.
   b) $A \cup B = B$ y $A \cap B = B$.
   c) $A \cup B = B$ y $A \cap B = A$.
   Justifique su respuesta: en el caso afirmativo presente una demostración; en el caso negativo presente un contraejemplo.
6. Demuestre el Metateorema 5.13.
7. Demuestre el converso de la propiedad enunciada en el Ejemplo 5.15: si $a^2 - 1$ es múltiplo de 3, entonces a no es múltiplo de 3.
8. Para $a \in \mathbb{Z}$ demuestre: $\neg(3 \cdot | \, a) \equiv 3 \cdot | \, (a^2 - 1)$.
9. Para $a \in \mathbb{Z}$ demuestre: $\neg(5 \cdot | \, a) \equiv 5 \cdot | \, (a^4 - 1)$.
10. Sean Γ una colección de proposiciones y $\phi_0, \ldots, \phi_n$ proposiciones. Demuestre que la siguiente técnica de demostración para una conjuntoria es correcta:
$$\Gamma \vdash_{\mathsf{DS}} \bigwedge_{i=0}^{n} \phi_i \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}} \phi_i, \quad \text{para cada } 0 \leq i \leq n.$$
11. Sean Γ una colección de proposiciones y $\phi_0, \ldots, \phi_n$ proposiciones. Demuestre o refute:
$$\Gamma \vdash_{\mathsf{DS}} \bigvee_{i=0}^{n} \phi_i \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}} \phi_i, \quad \text{para } \textit{algún } 0 \leq i \leq n.$$
12. Demuestre el Metateorema 5.14.1.
13. Demuestre el Metateorema 5.14.2.
14. Demuestre que $\sqrt[3]{2}$ es irracional.
15. Demuestre que el inverso aditivo de un número irracional es irracional.
16. Demuestre que no hay un número entero mayor o igual que todos los números enteros.
17. Una ecuación *Diofantina* es una ecuación para la cual se buscan soluciones en $\mathbb{Z}$. Por ejemplo, una tripla Pitagórica (x, y, z) es una solución a la ecuación Diofantina $x^2 + y^2 = z^2$.
   a) Demuestre que no hay soluciones en los enteros *positivos* para la ecuación Diofantina $x^2 - y^2 = 1$.
   b) Demuestre que no hay soluciones en los enteros *positivos* para la ecuación Diofantina $x^2 - y^2 = 10$.
18. Considere el problema de escribir una colección de números de tal manera que se use cada uno de los 10 digitos exactamente una vez. Por ejemplo, la colección $\{19, 28, 30, 7, 6, 5, 4\}$ usa cada uno de los 10 dígitos exactamente una vez. ¿Existe una colección de números que satisfaga esta condición y cuya suma sea 100? Justifique su respuesta.

19. Sean Γ una colección de proposiciones y ϕ, ψ proposiciones. Demuestre que el siguiente principio de demostración por contradicción es correcto:

$$\Gamma \vdash_{\mathsf{DS}} \psi \to \phi \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}} \psi \wedge \neg\phi \to \neg\psi.$$

20. Demuestre el Metateorema 5.16.
21. Demuestre el Metateorema 5.17.1.
22. Demuestre el Metateorema 5.17.2.
23. Demuestre el Metateorema 5.17.3.
24. Basándose en el Metateorema 5.17, proponga tres proposiciones equivalentes a "si x es impar y y es par, entonces xy es par".
25. Demuestre que si $a, b \in \mathbb{Z}$ son tales que $par(a+b)$, entonces a y b tienen la misma paridad. Dos números tienen la misma paridad cuando ambos son impares o ambos son pares.
26. Demuestre que si $a, b \in \mathbb{Z}$ son tales que $par(ab)$, entonces al menos uno de a y b es par.
27. Demuestre que si $a, b \in \mathbb{Z}$ son tales que $impar(ab)$, entonces a y b es son impares.
28. Demuestre que si a es un número entero *positivo* de la forma $a = 3c + 2$, para algún $c \in \mathbb{Z}$, entonces a no es un cuadrado perfecto. Ayuda: note que todo número entero a puede escribirse como $a = 3c$, $a = 3c + 1$ o $a = 3c + 2$ para algún $c \in \mathbb{Z}$.
29. Demuestre que si a es un número entero de la forma $a = 4c + 2$ o $a = 4c + 3$ para algún $c \in \mathbb{Z}$, entonces a no es un cuadrado perfecto. Ayuda: note que todo número entero a puede escribirse como $a = 4c$, $a = 4c + 1$, $a = 4c + 2$ o $a = 4c + 3$, para algún $c \in \mathbb{Z}$.
30. Demuestre que si $a, b \in \mathbb{R}$ son tales que ab es irracional, entonces al menos uno de a y b es irracional.
31. Demuestre el Teorema 5.19.
32. Demuestre el Metateorema 5.18.
33. Complete las demostraciones en el Ejemplo 5.13.
34. Demuestre el Metateorema 5.21 por inducción sobre $n \in \mathbb{N}$.
35. Complete las demostraciones en el Ejemplo 5.14.
36. Demuestre que si $a \in \mathbb{Z}$, entonces alguno de a^3, $a^3 + 1$ o $a^3 - 1$ es múltiplo de 9.
37. Demuestre que si $a \in \mathbb{N}$, entonces $a^7 - a$ es múltiplo de 7. Ayuda: use análsis de casos sobre a con respecto a múltiplos de 7.
38. Demuestre el Metateorema 5.22.1.
39. Demuestre el Metateorema 5.22.2.
40. Complete el Ejemplo 5.15 con las demostraciones de las condiciones (1) y (2).

41. Investigue y explique brevemente en qué consiste el "teorema de los 4 colores" (en inglés, *four color theorem*). En particular, describa cómo fue útil un computador para demostrarlo y explique qué técnica de demostración (abordada en esta sección) fue extensivamente usada en dicha demostración mecánica.
42. Investigue sobre la "conjetura de Kepler" (en inglés, *Kepler conjecture*). Explique en qué consiste la conjetura y el estado actual de su demostración. ¿Qué técnica fue usada extensivamente en su demostración? Explique brevemente.
43. En internet circulan listas de "técnicas" de demostración como, por ejemplo, demostración por intimidacíon, por omisión o por ofuscación. A la luz de la lógica este tipo de razonamientos carecen de cualquier justificación. Sin embargo, puede ser interesante conocer algunos de estos argumentos para evitarlos a toda costa, al menos en un curso de lógica. Investigue y explique en qué consisten las siguientes "técnicas" de demostración:
    a) Demostración por intimidacíon.
    b) Demostración por omisión.
    c) Demostración por ofuscación.
    d) Demostración por intuición.
    e) Demostración por acumulación de evidencia.
    f) Demostración por referencia a una fuente inaccesible.
    g) Demostración por autoridad eminente.
    h) Demostración por creencia religiosa.
    i) Demostración por asombro.
    j) Demostración por repetición.
    k) Demostración por reducción al problema equivocado.
    l) Demostración por importancia.
    m) Demostración por eliminación del contraejemplo.
    n) Demostración por cambio de definición.
    ñ) Demostración por notación ilegible.

Parte 2

Lógica de predicados

Capítulo 6

Lenguaje y especificación

La lógica de predicados, o lógica de primer orden, es conocida coloquialmente como la *lógica de cuantificadores*. De forma más precisa, la lógica de predicados se obtiene de la lógica proposicional al agregar los cuantificadores lógicos *para todo*, denotado con $\forall$ y llamado cuantificador *universal*, y *existe*, denotado con $\exists$ y llamado cuantificador *existencial*. Para explicar por qué estos nuevos conectivos lógicos son necesarios, considere la siguiente argumentación.

Ejemplo 6.1

Todos los informáticos son intelectualmente destacados. Turing es un informático. Entonces, Turing es intelectualmente destacado.

Desde el punto de vista de la lógica proposicional, la argumentación en el Ejemplo 6.1 tiene la forma

$$p, q, r,$$

la cual, de acuerdo con los métodos estudiados en los capítulos 2 y 4, no es válida. Sin embargo, intuitivamente este argumento es válido: de lo contrario habría un informático intelectualmente no destacado, contradiciendo la primera proposición en la argumentación. Esto puede indicar que la lógica proposicional no es lo suficientemente expresiva para simbolizar, especificar y verificar la validez de algunas argumentaciones.

En este capítulo se estudiarán los lenguajes de primer orden. Estos lenguajes tienen la característica de ser más generales que el lenguaje de la lógica proposicional y, gracias a su poder expresivo, permitirán especificar y verificar argumentaciones

como la del Ejemplo 6.1. Anticipando el desarrollo de este capítulo, la argumentación del Ejemplo 6.1 en lógica de predicados tendría la siguiente forma:

$$(\forall x\,(I(x) \to D(x))), I(t), D(t),$$

en donde x es una variable sobre elementos, t denota el elemento "Turing", $I(x)$ un predicado denotando que "x es informático" y $D(x)$ un predicado denotando que "x es intelectualmente destacado".

6.1. Lenguajes de primer orden

La discusión que sirve como introducción a este capítulo ilustra la necesidad de contar con un lenguaje lógico para especificar propiedades en diferentes ámbitos, con un nivel de detalle mayor que en la lógica proposicional. Esta sección presenta la noción de lenguaje de primer orden, como herramienta sintáctica para desarrollar la lógica de predicados.

El lenguaje de la lógica de predicados consta de símbolos de función, símbolos de predicado, variables, una función de aridad, paréntesis, comas y conectivos lógicos.

Definición 6.1

Los *símbolos del lenguaje de la lógica de predicados* son:

- Una colección $\mathcal{F}$ de *símbolos de función.*
- Una colección $\mathcal{P}$ de *símbolos de predicado.*
- Una colección infinita $\mathcal{X}$ de *variables*

  $$x_0, x_1, x_2, \ldots$$

- Una función $ar : \mathcal{F} \cup \mathcal{P} \to \mathbb{N}$ de *aridad.*
- Paréntesis izquierdo '(' y paréntesis derecho ')', y la coma ','.
- Una colección de *conectivos lógicos*

  $$\mathit{true}, \mathit{false}, \equiv, \not\equiv, \vee, \wedge, \to, \leftarrow, \forall, \exists.$$

Los símbolos de función y de predicado son parámetros del lenguaje, es decir, hay diferentes lenguajes dependiendo de qué símbolos se usen como parámetros. Los conectivos lógicos de la lógica de predicados extienden aquellos de la lógica proposicional con los cuantificadores universal y existencial. Cada símbolo de función y de predicado tiene una *aridad* asociada indicando la cantidad de argumentos que espera. En particular, una constante c es un símbolo tal que $ar(c) = 0$. Aunque la notación para identificar las variables de un lenguaje de primer orden aparece fija, se entiende que dicho conjunto también es paramétrico en la definición de un lenguaje de primer orden.

Nota 6.2

Un lenguaje de primer orden $\mathcal{L}$ con símbolos de función $\mathcal{F}$, símbolos de predicado $\mathcal{P}$ y variables $\mathcal{X}$ se denota como $\mathcal{L} = (\mathcal{F}, \mathcal{P}, \mathcal{X})$.

Desde el punto de vista de un sistema formal, las expresiones bien formadas de la lógica de predicados se clasificarán en uno de dos tipos. El primer tipo corresponde a expresiones que permiten identificar los objetos de interés; cada una de estas expresiones se denomina *término*. El segundo tipo de expresiones denota valores de verdad sobre dichos objetos; cada una de estas expresiones se denomina *fórmula*.

Nota 6.3

Un *término* es una expresión que identifica objetos.

Nota 6.4

Una *fórmula* es una expresión que denota valores de verdad sobre los objetos.

Las reglas de construcción de términos y fórmulas se presenta en las secciones 6.2 y 6.3, respectivamente. Sin embargo, es natural ver cómo un símbolo de función constante o una variable puede identificar objetos, un símbolo de función referirse a objetos de manera indirecta y un símbolo de predicado cualificar objetos.

Ejemplo 6.2

Teniendo como referencia el Ejemplo 6.1, si f es un símbolo de función que denota el padre de un individuo y el símbolo de función t identifica a Turing, entonces $f(t)$ identifica al padre de Turing. De manera similar, si I y D son símbolos de predicado como los presentados en la introducción del capítulo, entonces la fórmula $I(x)$ indica que x es informático, $(I(x) \rightarrow D(x))$ que aquel que sea informático es intelectualmente destacado y $D(t)$ que Turing es intelectualmente destacado.

Ejercicios

1. Los grupos son estructuras matemáticas que se definen con base en un conjunto y una operación binaria sobre elementos del conjunto. Investigue acerca de estas estructuras y:
   a) Formule 3 ejemplos de grupos identificando claramente el conjunto y la operación binaria que lo definen.
   b) Defina un lenguaje de primer orden para los grupos.
   c) Enuncie los axiomas de un grupo.
   d) Explique brevemente cómo los grupos han sido útiles para resolver problemas cotidianos.
2. Una lista es un tipo abstracto de datos que representa una colección ordenada. Hay dos tipos de listas: la lista vacía y la lista que resulta de agregar un elemento al frente de una lista. Por ejemplo, si el foco es en las listas de números enteros, los siguientes son ejemplos de listas:

   $$\lambda \qquad 3;\lambda \qquad -1;(4;(3;\lambda))$$

   en donde λ representa la lista vacía. Defina un lenguaje de primer orden para listas, incluyendo un símbolo de función que represente el tamaño de una lista y un símbolo de predicado que represente la igualdad de dos listas.
3. Defina un lenguaje de primer orden que permita definir árboles binarios, incluyendo algunas de las operaciones (de consulta) más habituales sobre ellos.
4. La *lógica Aristotélica* o *lógica silogística* se define sobre un lenguaje de primer orden. Investigue sobre esta lógica y caracterice el lenguaje de primer orden sobre el cual se define.

6.2. Términos

Dado un lenguaje de primer orden $\mathcal{L} = (\mathcal{F}, \mathcal{P}, \mathcal{X})$, los términos de $\mathcal{L}$ se construyen con base en los símbolos de función $\mathcal{F}$ y las variables $\mathcal{X}$.

Definición 6.5

Los *términos* de la lógica de predicados sobre $\mathcal{F}$ son aquellas cadenas obtenidas usando una cantidad finita de veces las siguientes reglas de construcción:

1. Cada variable es un término.
2. Si $c \in \mathcal{F}$ y $ar(c) = 0$, entonces c es un término.

3. Si $t_1, \ldots, t_n$ son términos y $f \in \mathcal{F}$ es tal que $ar(f) = n$ con $n > 0$, entonces $f(t_1, \ldots, t_n)$ es un término.

La expresión $\mathcal{T}_{\mathcal{F}}(\mathcal{X})$ denota la *colección de términos* sobre $\mathcal{F}$ con variables en $\mathcal{X}$.

Usando la notación Backus-Naur, la Definición 6.5 puede escribirse compactamente como:

$$t ::= x \mid c \mid f(t, \ldots, t), \tag{6.1}$$

en donde $x \in \mathcal{X}$ y $c, f \in \mathcal{F}$ son tales que $ar(c) = 0$ y $ar(f) > 0$.

Es importante notar que los bloques de construcción básicos de los términos son las variables y las constantes (i.e., símbolos de función con aridad 0). Términos más complejos se construyen a partir de símbolos de función aplicados sobre términos previamente construídos. Note que la definición de término depende del conjunto $\mathcal{F}$ y, consecuentemente, si este conjunto cambia, entonces también cambia el conjunto de términos de la lógica de predicados.

Ejemplo 6.3

Sea $\mathcal{F} = \{n, f, g\}$ con $ar(n) = 0$, $ar(f) = 1$ y $ar(g) = 2$. Entonces n, $g(f(n), n)$ y $f(g(f(n), n))$ son términos. Sin embargo, las expresiones $n(f)$, $g(f(n))$ y $g(f(n), n, n)$ no lo son. ¿Por qué?

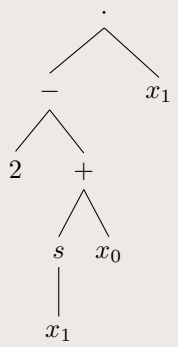
Ejemplo 6.4

Sea $\mathcal{F} = \{0, 1, 2, \ldots, s, +, -, \cdot\}$ con $ar(n) = 0$ si $n \in \mathbb{N}$, $ar(s) = 1$ y $ar(+) = ar(-) = ar(\cdot) = 2$. Entonces $\cdot(-(2, +(s(x_1), x_0)), x_1)$ es un término. Usualmente los símbolos de función binarios se escriben con notación *infija* en favor de la notación *prefija*; consecuentemente, este término se escribe como $(2 - (s(x_1) + x_0)) \cdot x_1$.

Los términos de la lógica de predicados pueden ser representados visualmente por medio de árboles de sintaxis. Un árbol de sintaxis de un término corresponde a una variable, un símbolo de función constante o se construye a partir de un símbolo de función f y una lista de términos $t_1, \ldots, t_n$ (i.e., una término $f(t_1, \ldots, t_n)$). Este último caso, la raíz en un árbol de sintaxis tiene etiqueta f y n subárboles, uno por cada subtérmino t_i.

Ejemplo 6.5

Considere los símbolos de función presentados en el Ejemplo 6.4. El siguiente árbol de sintaxis representa el término $(2 - (s(x_1) + x_0)) \cdot x_1$:



Ejercicios

1. Explique por qué las expresiones $n(f)$, $g(f(n))$ y $g(f(n), n, n)$ del Ejemplo 6.3 no son términos.
2. Dibuje el árbol de sintaxis de los términos n, $g(f(n), n)$ y $f(g(f(n), n))$ del Ejemplo 6.3.
3. Sea $\mathcal{F} = \{a, b, c\}$ con a, b, c constantes (i.e., $ar(a) = ar(b) = ar(c) = 0$). ¿Cuáles son los términos sobre $\mathcal{F}$ libres de variables (i.e., sin apariciones de variable alguna)?
4. Sea $\mathcal{F} = \{a, f\}$ con a constante y f un símbolo unario (i.e., $ar(a) = 0$ y $ar(f) = 1$). ¿Cuáles son los términos sobre $\mathcal{F}$ libres de variables?
5. Sea $\mathcal{F} = \{d, f, g\}$ con $ar(d) = 0$, $ar(f) = 3$ y $ar(g) = 2$, y suponga $x, y, z \in \mathcal{X}$. ¿Cuáles de las siguientes expresiones son términos sobre $\mathcal{F}$? Dibuje el árbol de sintaxis cuando la expresión sea un término; en el caso contrario, explique por qué no es un término.
   a) $g(d, d)$
   b) $f(x, g(y, z), d)$
   c) $g(x, f(y, z), d)$
   d) $g(x, h(y, z), d)$
   e) $f(f(g(d, x), f(g(d, x), t, g(y, d)), g(d, d)), g(f(d, d, x), d), z)$
6. Sea $\mathcal{F} = \{d, f, g\}$ con $ar(d) = 0$, $ar(f) = 3$ y $ar(g) = 2$.

a) La *longitud* de un término sobre $\mathcal{F}$ corresponde a la cantidad de símbolos que se usan en su representación, incluyendo las comas y paréntesis. Por ejemplo, la longitud de $f(x_0, g(x_1, x_2), x_2)$ es 13. Liste todos los términos sobre $\mathcal{F}$ libres de variables cuya longitud sea menor a 10.

b) La *altura* de una término sobre $\mathcal{F}$ se define como 1 más la longitud del camino más largo en el árbol de sintaxis. Por ejemplo, la altura del término $f(x_0, g(x_1, x_2), x_2)$ es 3. Liste todos los términos sobre $\mathcal{F}$ libres de variables cuya altura sea menor a 4.

7. Dibuje el árbol de sintaxis del término $(2 - s(x_0)) + (x_1 \cdot x_0)$, teniendo en cuenta que $-, +, \cdot$ se usan con notación infija.
8. Investigue y explique brevemente en qué consisten las notaciones prefija, infija y posfija. Iluestre con un ejemplo en cada uno de los casos.

6.3. Fórmulas

Dado un lenguaje de primer orden $\mathcal{L} = (\mathcal{F}, \mathcal{P}, \mathcal{X})$, las fórmulas de $\mathcal{L}$ se construyen con base en los símbolos de predicado $\mathcal{P}$ y los términos de $\mathcal{L}$.

Definición 6.6

Las *fórmulas* de la lógica de predicados sobre $(\mathcal{F}, \mathcal{P}, \mathcal{X})$ son aquellas cadenas obtenidas usando una cantidad finita de veces las siguientes reglas de construcción:

1. Las constantes *true* y *false* son fórmulas.
2. Si $P \in \mathcal{P}$ y $ar(P) = 0$, entonces P es una fórmula.
3. Si $t_1, \ldots, t_n$ son términos sobre $\mathcal{F}$ y $Q \in \mathcal{P}$ es tal que $ar(Q) = n$ con $n > 0$, entonces $Q(t_1, \ldots, t_n)$ es una fórmula.
4. Si ϕ es una fórmula, entonces $(\neg\phi)$ es una fórmula.
5. Si ϕ y ψ son fórmulas, y $\otimes \in \{\equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$, entonces $(\phi \otimes \psi)$ es una fórmula.
6. Si ϕ es una fórmula y x es una variable, entonces $(\forall x\, \phi)$ y $(\exists x\, \phi)$ son fórmulas.

La expresión $\mathcal{T}_{(\mathcal{F},\mathcal{P})}(\mathcal{X})$ denota la *colección de fórmulas* sobre $(\mathcal{F}, \mathcal{P})$ con variables en $\mathcal{X}$.

De acuerdo con la Definición 6.6, observe cómo en una fórmula de la lógica de predicados los argumentos de un símbolo de predicado son términos. La Definición 6.6

puede escribirse usando la notación Backus-Naur de la siguiente manera:

$$\begin{aligned}(6.2)\qquad \phi ::= \ & \mathit{true} \mid \mathit{false} \mid P \mid Q(t_1, \ldots, t_n) \mid (\neg\phi) \mid (\phi \equiv \phi) \mid (\phi \not\equiv \phi) \mid (\phi \vee \phi) \\ & \mid (\phi \wedge \phi) \mid (\phi \to \phi) \mid (\phi \leftarrow \phi) \mid (\forall x\, \phi) \mid (\exists x\, \phi),\end{aligned}$$

en donde $P, Q \in \mathcal{P}$ con $ar(P) = 0$, $ar(Q) = n > 0$, t_i son términos sobre $\mathcal{F}$ y $x \in \mathcal{X}$. Recuerde que todo lo que aparece a la derecha del símbolo ::= corresponde a una fórmula que ya ha sido construída siguiendo las mismas reglas de construcción.

A continuación se presenta una tabla con los nombres y las interpretaciones intuitivas que reciben los cuantificadores:

Símbolo	**Nombre**	**Interpretación**
$\forall$	cuantificador universal	todo $\cdots$ es tal que $\cdots$
$\exists$	cuantificador existencial	hay un $\cdots$ tal que $\cdots$

Nota 6.7

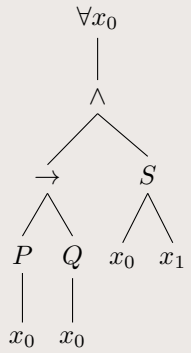
Por conveniencia, en la lógica de predicados se preserva la convención sobre la precedencia de los conectivos lógicos de la lógica proposicional (Nota 5.1). Se establece que los cuantificadores $\forall, \exists$ tienen la misma precedencia que $\neg$. En consecuencia, se adoptan las siguientes convenciones sobre la precedencia de los conectivos lógicos de la lógica de predicados:

- Las constantes *true* y *false* tienen más precedencia que $\neg, \forall, \exists$.
- Los conectivos $\neg, \forall, \exists$ tienen más precedencia que $\vee, \wedge$.
- Los conectivos $\vee, \wedge$ tienen más precedencia que $\to, \leftarrow$.
- Los conectivos $\to, \leftarrow$ tienen más precedencia que $\equiv, \not\equiv$.

Las fórmulas de la lógica de predicados, al igual que los términos, pueden ser representadas por medio de árboles sintácticos. En un árbol de sintaxis, los cuantificadores $\forall$ y $\exists$ tienen exactamente un subárbol, como la negación $\neg$, que corresponde al árbol de la fórmula que cuantifican. Las fórmulas que se construyen a partir de un símbolo de predicado P y una lista de términos $t_1, \ldots, t_n$ (i.e., una fórmula $P(t_1, \ldots, t_n)$), en un árbol de sintaxis tienen raíz con etiqueta P y n subárboles, uno por cada t_i.

Nota 6.8

Considere los símbolos de función presentados en el Ejemplo 6.4. El siguiente árbol de sintaxis representa la fórmula $\forall x_0\,((P(x_0) \to Q(x_0)) \wedge S(x_0, x_1))$:



Las fórmulas sin variables libres tienen un tratamiento especial en algunos casos, como se verá en capítulos posteriores.

Definición 6.9

Una fórmula ϕ se llama *sentencia* si y sólo sí no tiene variables libres.

Ejemplo 6.6

La fórmula

$$\forall x_0\,((P(x_0) \to Q(x_0)) \wedge S(x_0, x_1))$$

de la Nota 6.8 no es una sentencia porque la variable x_1 aparece libre en ella. En cambio, la fórmula

$$\exists x_1(\forall x_0\,((P(x_0) \to Q(x_0)) \wedge S(x_0, x_1)))$$

es una sentencia.

Como se ilustró en la introducción de este capítulo, la lógica de predicados generaliza la lógica proposicional y cuenta con un lenguaje que permite modelar en más detalle situaciones de interés. Considere el problema de simbolizar y especificar la siguiente proposición en un lenguaje de la lógica de predicados:

Todo hijo de mi padre es mi hermano.

A continuación se presentan dos ejemplos de especificación con lenguajes similares, solo que en uno de ellos *padre* se usa como símbolo de predicado y en el otro como símbolo de función.

Ejemplo 6.7

En este ejemplo se usa *padre* como predicado. Los símbolos de función se definen como $\mathcal{F} = \{yo\}$ tal que *yo* es una constante que denota el sujeto "yo" o "mi". El conjunto de símbolos de predicado se define como $\mathcal{P} = \{H, I, P\}$ con los siguientes significados:

$H(x, y)$: "x es hermano de y",

$I(x, y)$: "x es hijo de y",

$P(x, y)$: "x es *el* padre de y".

Con esta simbolización, la especificación de la proposición bajo cuestión es la siguiente:

$$\forall x_0 \, \forall x_1 \, (P(x_0, yo) \wedge I(x_1, x_0) \rightarrow H(x_1, yo)),$$

que literalmente traduce "para todo x_0 y x_1, si x_0 es el padre de *yo* y x_1 es hijo de x_0, entonces x_1 es hermano de *yo*".

Ejemplo 6.8

En este ejemplo se usa *padre* como función. Los símbolos yo, H, I se definen como en el Ejemplo 6.7. En adición, el símbolo de función p es tal que $p(x)$ representa el padre de x. Note que dada una persona, su padre está unívocamente definido y por ende la expresión $p(x)$ está bien definida. Con esta simbolización, la especificación de la proposición bajo cuestión es la siguiente:

$$\forall x_0 \, (I(x_0, p(yo)) \rightarrow H(x_0, yo)),$$

que literalmente traduce "para todo x_0, si x_0 es hijo del padre de *yo*, entonces x_0 es hermano de *yo*".

Note que, en principio, la fórmula obtenida en el Ejemplo 6.8 es más sencilla que aquella del Ejemplo 6.7 porque involucra únicamente un cuantificador.

Finalmente, se adoptan algunas convenciones para escribir cuantificaciones. La motivación para esta convención será evidente a medida que se use la lógica de predicados como lenguaje de especificación.

Nota 6.10

Para ϕ y ψ fórmulas, y $x \in \mathcal{X}$:

- La expresión $(\forall x \mid \psi : \phi)$ es azúcar sintáctico para la fórmula $\forall x\,(\psi \to \phi)$; en particular, $(\forall x \mid \mathit{true} : \phi)$ se puede abreviar como $(\forall x \mid: \phi)$.
- La expresión $(\exists x \mid \psi : \phi)$ es azúcar sintáctico para la fórmula $\exists x\,(\psi \wedge \phi)$; en particular, $(\exists x \mid \mathit{true} : \phi)$ se puede abreviar como $(\exists x \mid: \phi)$.

En las fórmulas $(\forall x \mid \psi : \phi)$ y $(\exists x \mid \psi : \phi)$, la fórmula ψ recibe el nombre de *rango* de la cuantificación y ϕ el nombre de *término* de la cuantificación. Usando esta notación, la fórmula del Ejemplo 6.8 se escribe como

$$(\forall x_0 \mid I(x_0, p(yo)) : H(x_0, yo)).$$

En este caso $I(x_0, p(yo))$ es el rango y $H(x_0, yo)$ el término de la cuantificación.

Ejercicios

1. Suponga que $\mathcal{F} = \{\}$ y $\mathcal{P} = \{p_0, p_1, \ldots\}$ es una colección infinita de símbolos de función tales que $ar(p_n) = 0$ para $n \in \mathbb{N}$.
   a) Liste cinco fórmulas sobre $(\mathcal{F}, \mathcal{P})$ con variables en $\mathcal{X}$.
   b) ¿Cómo puede relacionar la colección de proposiciones $\mathcal{T}(\mathcal{V})$ del sistema DS con las fórmulas $\mathcal{T}_{(\mathcal{F},\mathcal{P})}(\mathcal{X})$ de la lógica de predicados sobre $(\mathcal{F}, \mathcal{P})$ con variables en $\mathcal{X} = \{\}$? Explique su respuesta.
2. Complete el Ejemplo 6.7 con la definición de la función ar para los símbolos de función y de predicado.
3. Complete el Ejemplo 6.8 con la definición de los conjuntos $\mathcal{F}$ y $\mathcal{P}$, y la definición de la función ar para los símbolos en estos conjuntos.
4. Justifique detalladamente por qué la expresión obtenida en el Ejemplo 6.8 es una fórmula.
5. Escriba la fórmula del Ejemplo 6.7 siguiendo la convención notacional de la Nota 6.10.
6. Sean m un símbolo de función constante, f un símbolo de función con un argumento y S, B símbolos de predicado con dos argumentos. Suponga que x, y, z son variables en $\mathcal{X}$. ¿Cuáles de las siguientes expresiones son fórmulas? Dibuje el árbol de sintaxis cuando la expresión sea una fórmula; en el caso contrario, explique por qué no es una fórmula.
   a) $S(m, x)$
   b) $B(m, f(m))$

c) $f(m)$
d) $B(B(m,x),y)$
e) $S(B(m),z)$
f) $B(x,y) \to \exists z\, S(z,y)$
g) $B(x,y) \to (\exists z \mid: S(z,y))$
h) $S(x,y) \to S(y,f(f(x)))$
i) $B(x) \not\equiv B(B(x))$

7. Sean c, d símbolos de función constantes, f un símbolo de función con un argumento y h un símbolo de función con dos argumentos. Además, sean P, Q símbolos de predicado con tres argumentos. Suponga que x, y, z son variables en $\mathcal{X}$. ¿Cuáles de las siguientes expresiones son fórmulas? Dibuje el árbol de sintaxis cuando la expresión sea una fórmula; en el caso contrario, explique por qué no es una fórmula.
   a) $\forall x\, P(f(d), h(g(c,x),d,y))$
   b) $\forall x\, P(f(d), h(P(x,y),d,y))$
   c) $\forall x\, Q(g(h(x,f(d),x),g(x,x)),h(x,x,x),c)$
   d) $\forall z\, (Q(z,z,z) \to P(z))$
   e) $(\forall z \mid Q(z,z,z) : P(z))$
   f) $\forall x\, \forall y\, (g(x,y) \to P(x,y,z))$
   g) $Q(c,d,c)$

8. Use los símbolos de predicado

   $A(x,y)$: "x admira a y",
   $B(x,y)$: "x asistió a y",
   $P(x)$: "x es profesor",
   $E(x)$: "x es estudiante",
   $C(x)$: "x es una clase",

   y el símbolo de función constante

   m : "María"

   para especificar las siguientes frases en el lenguaje de la lógica de predicados:
   a) María admira a todos los profesores (ayuda: la respuesta no es $(\forall x \mid: A(m, P(x)))$).
   b) Algún profesor admira a María.
   c) María se auto-admira.
   d) No todos los estudiantes asisten a todas las clases.
   e) Ninguna clase tuvo como asistentes a todos los estudiantes.
   f) Ninguna clase tuvo como asistentes a estudiante alguno.

9. Use los símbolos de predicado

   $G(x, y)$: "x le gana y",

   $F(x)$: "x es un equipo de fútbol",

   $A(x, y)$: "x es arquero de y",

   $P(x, y)$: "x pierde con y",

   y los símbolos de función constantes

   b : "El Bosque",

   t : "TikiTiki"

   para especificar las siguientes frases en el lenguaje de la lógica de predicados:
   a) Todo equipo de fútbol tiene un arquero.
   b) Si El Bosque le gana a TikiTiki, entonces El Bosque no pierde contra todos los equipos.
   c) TikiTiki le ganó a un equipo, el cual le ganó a El Bosque.

10. Suponga que $P(x, y)$ denota "x es padre de y" y $M(x, y)$ denota "x es madre de y". De forma similar, suponga que $E(x, y)$, $A(x, y)$ y $H(x, y)$ denotan "x es esposo/hermana/hermano de y", respectivamente. Finalmente, suponga que puede usar símbolos de función consante para identificar individuos como 'Juan' y 'Juana'. Sin embargo, no es permitido usar símbolos de predicado distintos a los introducidos inicialmente para especificar en el lenguaje de la lógica de predicados las siguientes proposiones:
    a) Todos tienen un madre.
    b) Todos tienen una madre y un padre.
    c) Quien sea que tiene una madre tiene un padre.
    d) Juan es abuelo.
    e) Ana y Jaime son primos.
    f) Algunas madres son tías.
    g) Ningún tío es padre.
    h) La abuela de nadie es padre de alguien.
    i) Juan y Juana son marido y mujer.
    j) Carlos es el cuñado de Mónica.

11. Suponga que $\mathcal{F} = \{\}$ y $\mathcal{P} = \{P\}$ es tal que $ar(P) = 2$. Además suponga que $P(x, y)$ simboliza "x e y son iguales" (en este caso, la pareja $(\mathcal{F}, \mathcal{P})$ se denomina el lenguaje de la igualdad). Especifique las siguientes frases en el lenguaje de la igualdad:
    a) Hay al menos dos elementos.
    b) Hay a lo sumo dos elementos.
    c) Hay exactamente tres elementos.
    d) Para cualquier par de elementos, hay otro elemento distinto a ellos.

12. Suponga que $\mathcal{F} = \{\}$ y $\mathcal{P} = \{R\}$ es tal que $ar(R) = 1$. Especifique las siguientes frases en el lenguaje de la lógica de predicados:
    a) Exactamente un elemento tiene la propiedad R.
    b) Todos, excepto dos elementos, tienen la propiedad R.
13. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados las siguientes proposiones:
    a) Cada quien ama a alguien.
    b) Alguien ama a alguien.
    c) Cada quien ama a todo el mundo.
    d) Nadie ama a todo el mundo.
    e) Alguien no ama a nadie.
14. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados las siguientes proposiones:
    a) Todos los humanos son egoístas.
    b) Ningún humano es egoísta.
    c) Algunos humanos son egoístas.
    d) Algunos humanos no son egoístas.
15. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados las siguientes proposiones:
    a) Todas las cosas rojas están en la caja.
    b) Únicamente cosas rojas están en la caja.
    c) Ningún animal es gato y perro.
    d) Todos los premios fueron ganados por un niño.
    e) Un niño ganó todos los premios.
16. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados las siguientes proposiones:
    a) Usted puede engañar a algunos algunas veces.
    b) Usted puede engañar a todos algunas veces.
    c) Usted no puede engañarlos a todos algunas veces.
    d) Usted no puede engañar a alguien todas las veces.
17. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados las siguientes proposiones:
    a) Mensajes enviados de un proceso a otro son recibidos en el orden en que fueron enviados.
    b) Mensajes enviados a todos los procesos por un proceso son recibidos por todos los procesos en el orden en que fueron enviados.
    c) Todos los mensajes son recibidos en el mismo orden por todos los procesos.
18. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados las siguientes proposiones:
    a) Un estudiante recibe una nota por cada curso en que esté registrado.

b) El registro de un curso requiere haber aprobado todos sus prerrequisitos.
c) Ningún estudiante con una nota menor a 3 en un curso requerido se gradúa con honores.

19. Identifique los conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados la siguiente proposición:

    Suponiendo que cada tarea t toma $tarea(t)$ segundos, el tiempo de inicio $inicio(t)$ de la tarea t es el tiempo más temprano en el cual todas las tareas prerrequisito en la colección $prer(t)$ han sido completadas.

20. Las siguientes frases del inglés han sido tomadas de "RFC3157 Internet Task Force Document *Securely Available Credentials – Requirements*". Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados cada una de ellas:
    a) An attacker can persuade a server that a successful login has occurred, even if it hasn't.
    b) An attacker can overwrite someone else's credentials on the server.
    c) All users enter passwords instead of names.
    d) Credential transfer both to and from a device MUST be supported.
    e) Credentials MUST NOT be forced by the protocol to be present in clear text at any device other than the end user's.

21. Investigue y explique brevemente en qué consisten las notaciones prefija, infija y posfija. Iluestre con un ejemplo en cada uno de los casos.

22. Investigue la definición de un grupo en un libro de álgebra abstracta.
    a) Describa el lenguaje $(\mathcal{F}, \mathcal{P})$ para grupos.
    b) Proponga fórmulas sobre $(\mathcal{F}, \mathcal{P})$ que correspondan a los axiomas de los grupos.

23. Investigue la definición de un espacio vectorial en un libro de álgebra lineal.
    a) Describa el lenguaje $(\mathcal{F}, \mathcal{P})$ para espacios vectoriales.
    b) Proponga fórmulas sobre $(\mathcal{F}, \mathcal{P})$ que correspondan a los axiomas de los espacios vectoriales.

    Suponga que las variables "varían" sobre vectores y escalares. Para distinguir entre ellas, incluya en el lenguaje los símbolos de predicado V, S con aridad $ar(V) = ar(S) = 0$ de forma tal que $V(x)$ denote "x es un vector" y $S(x)$ denote "x es un escalar".

24. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados la siguiente argumentación:

    Hay un hombre que desprecia la humanidad. Entonces hay un hombre despreciado por toda humanidad.

    Intuitivamente, ¿es el argumento correcto?

25. Defina unos conjuntos de símbolos de función y de predicado adecuados, y especifique en el lenguaje de la lógica de predicados la siguiente argumentación:

Las hienas son peligrosas. Ningún gato es peligroso. Entonces, los gatos no son hienas.

Intuitivamente, ¿es el argumento correcto?

26. Proponga un lenguaje de primer orden $\mathcal{L}$, con al menos un símbolo de función y uno de predicado, de tal forma que cualquier fórmula sea una sentencia.

6.4. Variables libres y acotadas

Esta sección inicia la discusión de un tema delicado de la lógica de predicados: cómo sustituír en una fórmula un término por una variable. El problema radica en que dados una fórmula ϕ, una variable x y un término t sobre $(\mathcal{X}, \mathcal{F})$, al sustituír x por t en ϕ, se desea que la fórmula resultante "diga" sobre t lo mismo que ϕ dice sobre x. Dado que x puede aparecer dentro de subfórmulas cuantificadas de ϕ, alguna variable de t, si hay alguna, puede ser "capturada" por dichos cuantificadores. En general, este fenómeno potencialmente cambia el significado de una fórmula, lo cual es problemático y por ende se debe evitar. Esta sección introduce nociones que permiten clasificar las apariciones de las variables en una fórmula. En particular, estas nociones serán útiles en la Sección 6.5 para responder clara y correctamente la pregunta: ¿cómo sustituír adecuadamente una variable por un término en un fórmula?

En un árbol de sintaxis de una fórmula de la lógica de predicados, las variables aparecen como etiquetas en dos tipos de nodos. Unas variables pueden aparecer en nodos internos del árbol de sintaxis indicando el *alcance* de un cuantificador en una subfórmula. Otras variables aparecen en las hojas del árbol de sintaxis, algunas de ellas bajo el alcance de un cuantificador. Una variable x que aparece en una hoja del árbol de sintaxis que no tiene como ancestro un nodo con etiqueta '$\forall x$' o '$\exists x$' recibe el nombre de variable *libre* porque representa objetos que aún están por concretar. El el caso contrario, estas variables que aparecen en las hojas del árbol reciben el nombre de variables *acotadas* porque su valor está sujeto al cuantificador que la precede.

Considere el árbol de sintaxis en la Figura 2 correspondiente a la fórmula

$$(\forall x_0\, P(x_0, f(x_1))) \equiv Q(x_2) \wedge \exists x_2\, P(x_2, x_0),$$

en donde el símbolo de función f tiene aridad 1 y los símbolos de predicado P, Q tienen aridad 2 y 1, respectivamente. En este árbol de sintaxis hay un nodo interno con etiqueta '$\forall x_0$' y otro con etiqueta '$\exists x_2$'. Dos de las hojas del árbol están etiquetadas con 'x_0' y 'x_2', respectivamente, mientras que la hoja restante está etiquetada con 'x_1'. Si se recorre el árbol hacia arriba desde la primera hoja que corresponde a x_0, de izquierda a derecha, entonces se encuentra el nodo etiquetado $\forall x_0$. Como ambas etiquetas se refieren a la misma variable, en este caso x_0, se tiene que la

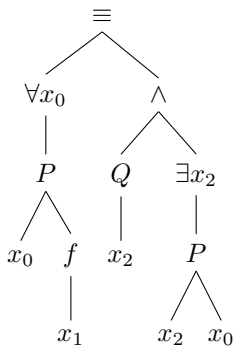


Figura 2. Árbol de sintaxis de $(\forall x_0\, P(x_0, f(x_1))) \equiv Q(x_2) \wedge \exists x_2\, P(x_2, x_0)$.

aparición de x_0 desde donde se inició el recorrido está *acotada* en la fórmula asociada al subárbol con raíz $\forall x_0$. En otras palabras, esa aparición de x_0 está *acotada* en la fórmula $\forall x_0\, P(x_0, f(x_1))$. En contraste, si se inicia un recorrido similar desde la otra hoja asociada a x_0, se observa que no hay ningún nodo interno que indique una cuantificación sobre x_0. Entonces, se tiene que la segunda aparición de x_0 es *libre*. Una situación similar se tiene para la variable x_2; en este caso, su primera aparición en el árbol, de izquierda a derecha, es libre mientras que la segunda es acotada (¿por qué?). En el caso de la hoja correspondiente a la variable x_1, si se inicia un recorrido hacia arriba se encuentra un único nodo interno etiquetado con una cuantificación, es decir, '$\forall x_0$'. Como x_1 no tiene nada que ver con x_0 (i.e., $x_1 \neq x_0$), entonces la única aparición de la variable x_1 es libre en esta fórmula.

A continuación se definen los conceptos de aparición *libre* y *acotada* de una variable en una fórmula de la lógica de predicados.

Definición 6.11

Sea ϕ una fórmula en $\mathcal{T}_{(\mathcal{F},\mathcal{P})}(\mathcal{X})$ y x una variable en $\mathcal{X}$.

- Una aparición de x es *libre en* ϕ sii el recorrido en el árbol de sintaxis de ϕ desde la hoja asociada a esa aparición de x no tiene etiquetas '$\forall x$' o '$\exists x$'.
- Una aparición de x es *acotada en* ϕ sii dicha aparición no es libre en ϕ.

De acuerdo con la Definición 6.11, y como se ha visto a lo largo de esta sección, es posible que una variable sea libre y acotada en una misma fórmula dado que una variable puede aparecer más de una vez en una fórmula.

Ejemplo 6.9

Considere la fórmula representada por el árbol de sintaxis en la Figura 2. La variable x_0 ocurre libre y acotada en esta fórmula. A su vez, la variable x_1 únicamente ocurre libre, mientras que x_2, al igual que x_0, ocurre libre y acotada en la fórmula.

Nota 6.12

Considere una variable x y una fórmula ϕ. Intuitivamente, una aparición acotada de x por un cuantificador *universal* en ϕ representa *cualquier* valor posible para esa aparición de x (e.g., todo número, cualquier conjunto, etc.). Una aparición acotada de x por un cuantificador *existencial* en ϕ representa *algún* valor posible para esa aparición de x. Una aparición libre de x en ϕ representa un valor externo (o desconocido) que debe ser suministrado (por ejemplo, por el contenido de una dirección de memoria en un computador).

A continuación se hace precisa la noción de alcance de un cuantificador.

Definición 6.13

Sean ϕ una fórmula en $\mathcal{T}_{(\mathcal{F},\mathcal{P})}(\mathcal{X})$ y x una variable en $\mathcal{X}$. El *alcance* del cuantificador $\forall x$ (resp., $\exists x$) en la fórmula $\forall x\, \phi$ (resp., $\exists x\, \phi$) es la fórmula ϕ sin sus subfórmulas de la forma $\forall x\, \psi$ y $\exists x\, \psi$.

En términos de árboles de sintaxis, el alcance de un cuantificador sobre una variable x es un subárbol de ϕ en el cual se ignoran aquellos subárboles cuya etiqueta corresponde a un cuantificador sobre x. Note que de acuerdo con la Definición 6.13, una aparición de una variable x es acotada sii dicha aparición está al alcance de algún $\forall x$ o $\exists x$; de lo contrario, esa aparición de x es libre en la fórmula.

Ejemplo 6.10

Considere la fórmula en la Figura 2:

$$((\forall x_0\, P(x_0, f(x_1))) \equiv Q(x_2) \wedge \exists x_2\, P(x_2, x_0)).$$

El alcance de $\forall x_0$ es la fórmula $P(x_0, f(x_1))$. El alcance de $\exists x_2$ es la fórmula $P(x_2, x_0)$.

Ejemplo 6.11

Considere la fórmula

$$\forall x_0\,(P(x_0, x_1) \to \exists x_0\, Q(x_0)).$$

En este caso, el alcance de $\forall x_0$ es $P(x_0, x_1)$ y el alcance de $\exists x_0$ es $Q(x_0)$. ¿Por qué?

Es importante notar, en general, que una hoja de un árbol de sintaxis está o no en el alcance de un cuantificador. Consecuentemente, una aparición *individual* de una variable es libre o acotada, pero no puede ser ambas a la vez.

Ejercicios

1. Considere el árbol de sintaxis en la Figura 2. Explique por qué una aparición de x_2 es libre y la otra es acotada.
2. Considere la fórmula en el Ejemplo 6.11. Dibuje el árbol de sintaxis, y explique por qué el alcance de $\forall x_0$ es $P(x_0, x_1)$ y el alcance de $\exists x_0$ es $Q(x_0)$.
3. Sean m un símbolo de función constante (i.e., $ar(m) = 0$), f un símbolo de función con un argumento y S, B símbolos de predicado con dos argumentos. Suponga que x, y, z son variables en $\mathcal{X}$. Para cada una de las siguientes fórmulas, indique (i) cuáles apariciones de x, y, z son libres y (ii) cuáles son acotadas.
   a) $S(m, x)$
   b) $B(m, f(m))$
   c) $B(x, y) \to \exists z\, S(z, y)$
   d) $\exists y\, B(x, y) \to \exists z\, S(z, y)$
   e) $S(x, y) \to S(y, f(f(x)))$
4. Sean c, d símbolos de función constantes, f un símbolo de función con un argumento y h un símbolo de función con dos argumentos. Además, sean P, Q símbolos de predicado con tres argumentos. Suponga que x, y, z son variables en $\mathcal{X}$. Para cada una de las siguientes fórmulas, indique (i) cuáles apariciones de x, y, z son libres, (ii) cuáles son acotadas y (iii) el alcance de cada uno de los cuantificadores.
   a) $P(c, c, d) \vee \forall x\, P(f(d), h(h(c, x), d), y)$
   b) $\exists y\, (P(x, y, x) \to \exists z\, Q(z, y, f(z)))$
   c) $\exists y\, P(x, y, x) \not\equiv \forall y\, Q(z, y, f(z))$
   d) $\exists y\, (P(x, y, x) \not\equiv \forall y\, Q(z, y, f(z)))$
   e) $\forall x\, \exists y\, P(x, y, x) \to \exists z\, Q(z, y, f(x))$
   f) $\forall z\, \exists y\, P(x, y, x) \to \exists z\, Q(z, y, f(x))$
   g) $\forall x\, (\exists y\, P(x, y, x) \wedge \exists z\, Q(z, y, f(x)))$

h) $\forall z\,(\exists y\,P(x,y,x) \wedge \exists z\,Q(z,y,f(x)))$

5. Sea ϕ la fórmula

$$\exists x(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z))),$$

en donde x, y, z son variables y P, Q símbolos de predicado con dos argumentos.
   a) Dibuje el árbol de sintaxis de ϕ.
   b) Identifique las apariciones de variables libres y acotadas en ϕ.
   c) ¿Hay alguna variable en ϕ que tenga apariciones libres y acotadas? Explique brevemente.
   d) ¿Cuál es el alcance de $\exists x$ en ϕ?
   e) ¿Cuál es el alcance de $\forall y$ en ϕ?
   f) Cambie la parentización en ϕ de tal manera que el alcance de $\exists x$ en la fórmula resultante sea $P(y,z)$; dibuje el árbol de sintaxis correspondiente.

6. Proponga una definición inductiva sobre la estructura de las fórmulas para la expresión $quant(x,\phi)$ que es cierta si y solo si la variable x aparece cuantificada en la fórmula ϕ.

6.5. Sustitución de términos

Las variables son comodines en una fórmula, luego es natural contar con mecanismos para reemplazarlas por información más concreta. Esta sección presenta la *sustitución textual de términos* como mecanismo para reemplazar variables por términos fórmulas de la lógica de predicados. Adicionalmente, esta sección presenta la noción de cuándo un término t puede reemplazar una variable x en una fórmula ϕ de manera tal que la fórmula resultante "diga" sobre t "lo mismo" que ϕ "dice" sobre x. Así se finaliza la discusión iniciada en la Sección 6.4 acerca cómo sustituír adecuadamente una variable por un término en una fórmula.

De acuerdo con las definiciones de términos y fórmulas (definiciones 6.5 y 6.6) únicamente es correcto sustituír una variable por un término. De lo contrario, se tendrían términos cuyos subtérminos pueden ser fórmulas y esto no tiene sentido.

A continuación se presenta el concepto de sustitución de términos.

Definición 6.14

Una *sustitución de términos* es una función $F : \mathcal{X} \to \mathcal{T}_{\mathcal{F}}(\mathcal{X})$ distinta a la identidad en una cantidad finita de elementos del dominio.

Una sustitución de términos F es una función que asocia un término $F(x)$ a cualquier variable $x \in \mathcal{X}$. Recuerde que $\mathcal{T}_{\mathcal{F}}(\mathcal{X})$ denota la colección de términos sobre

$\mathcal{F}$ (Definición 6.5). Al igual que una sustitución en la lógica proposicional (Definición 3.1), cualquier sustitución de términos F es tal que $F(x) \neq x$ para una cantidad finita de variables x. Entonces, una sustitución también puede ser escrita como un conjunto finito de la forma

$$\{y_0 \mapsto u_0, y_1 \mapsto u_1, \ldots, y_n \mapsto u_n\}$$

indicando que el término u_i está asociado a la variable y_i $(0 \leq i \leq n)$ y cualquier otra variable está asociada a sí misma cuando esta no aparece en la lista de variables $y_0, y_1, \ldots, y_n$.

La definición de cómo una sustitución se aplica a una fórmula se presenta en dos partes. Inicialmente, se define cómo una sustitución se aplica a un término y posteriormente se define cómo se aplica a una fórmula.

Definición 6.15

Sea t un término y $F = \{y_0 \mapsto u_0, y_1 \mapsto u_1, \ldots, y_n \mapsto u_n\}$ una sustitución de términos. La *sustitución textual* de F en t, denotada como $\overline{F}(t)$, se define inductivamente para todo subtérmino de t de la siguiente manera:

1. $\overline{F}(x) = F(x)$, si $x \in \{y_0, \ldots, y_n\}$,
2. $\overline{F}(x) = x$, si $x \notin \{y_0, \ldots, y_n\}$,
3. $\overline{F}(c) = c$, si $c \in \mathcal{F}$ y $ar(c) = 0$ y
4. $\overline{F}(f(t_1, \ldots, t_k)) = f(\overline{F}(t_1), \ldots, \overline{F}(t_k))$ si $f \in \mathcal{F}$ es tal que $ar(f) = k > 0$ y $t_1, \ldots, t_k$ son términos.

De acuerdo con la Definición 6.15, aplicar una sustitución de términos F a un término t resulta en un término $\overline{F}(t)$ en el cual algunas variables de t pueden haber sido reemplazadas. El caso (1) de esta definición indica explícitamente cómo una variable x en t es reemplazada por $F(x)$. En los casos (2) y (3) note que si t es una variable que no es parte de las variables "afectadas" por F o es una constante, entonces $F(t) = t$. El caso (4) corresponde a la definición inductiva que depende de la estructura interna de los subtérminos de t.

Ejemplo 6.12

Considere $F = \{x_0 \mapsto f(x_2), x_1 \mapsto h(x_0, x_3), x_8 \mapsto x_7\}$ una sustitución de términos y t el término $h(x_0, g(x_1, x_0, x_2))$. Entonces $\overline{F}(t)$ es el término:

$$h(f(x_2), g(h(x_0, x_3), f(x_2), x_2)).$$

Por motivos técnicos, es necesario introducir el concepto de restricción de una sustitución de términos antes de definir cómo una sustitución se aplica a una fórmula. La idea es usar las restricciones de una variable en una sustitución para controlar cuáles de las apariciones de una variable se reemplazan.

Definición 6.16

Sea x una variable y F una sustitución de términos. La *restricción de x en F*, denotada $F_{\lhd x}$, es la sustitución de términos definida para cualquier variable y de la

siguiente manera:

$$F_{\lhd x}(y) = \begin{cases} x & \text{, si } x = y \\ F(y) & \text{, si } x \neq y. \end{cases}$$

Como tal, la restricción $F_{\lhd x}$ de una variable x en una sustitución de términos F es una sustitución igual a F pero que es la identidad en la variable restringida x.

Ejemplo 6.13

Sea $F = \{x_0 \mapsto f(x_2), x_1 \mapsto h(x_0, x_3), x_8 \mapsto x_7\}$. Entonces:

- $F_{\lhd x_0}$ es $\{x_1 \mapsto h(x_0, x_3), x_8 \mapsto x_7\}$.
- $F_{\lhd x_1}$ es $\{x_0 \mapsto f(x_2), x_8 \mapsto x_7\}$.
- $F_{\lhd x_2}$ y $F_{\lhd x_7}$ son F.
- $F_{\lhd x_8}$ es $\{x_0 \mapsto f(x_2), x_1 \mapsto h(x_0, x_3)\}$.

Teniendo como referencia el árbol de sintaxis de una fórmula sobre la cual se quiere reemplazar la variable x por un término t, instrumentalmente el objetivo es reemplazar 'los' nodos hoja asociados a x por el árbol de sintaxis de t. Como se plantea al inicio de la Sección 6.4, este reemplazo debe hacerse con cuidado; por eso el artículo 'los' en la frase anterior debe entenderze con un grano de sal. La idea entonces es reemplazar aquellas hojas asociadas a apariciones libres de x por el árbol de t e ignorar los reemplazos para aquellas apariciones acotadas de x. A continuación se define cómo una sustitución de términos se aplica a una fórmula.

Definición 6.17

Sea ϕ una fórmula y $F = \{y_0 \mapsto u_0, y_1 \mapsto u_1, \ldots, y_n \mapsto u_n\}$ una sustitución de términos. La *sustitución textual* de F en ϕ, denotada como $\overline{F}(\phi)$, se define inductivamente para toda subfórmula de ϕ de la siguiente manera:

1. $\overline{F}(\mathit{true}) = \mathit{true}$ y $\overline{F}(\mathit{false}) = \mathit{false}$,
2. $\overline{F}(P) = P$, si $P \in \mathcal{P}$ y $ar(P) = 0$,
3. $\overline{F}(Q(t_1, \ldots, t_k)) = Q(\overline{F}(t_1), \ldots, \overline{F}(t_k))$, si $Q \in \mathcal{P}$ es tal que $ar(Q) = k > 0$ y $t_1, \ldots, t_k$ son términos sobre $\mathcal{F}$,

4. $\overline{F}(\neg\psi) = \neg\overline{F}(\psi)$,
5. $\overline{F}(\psi \otimes \tau) = \overline{F}(\psi) \otimes \overline{F}(\tau)$, si $\otimes \in \{\equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$ y
6. $\overline{F}(\forall x\, \psi) = \forall x\, \overline{F_{\lhd x}}(\psi)$ y $\overline{F}(\exists x\, \psi) = \exists x\, \overline{F_{\lhd x}}(\psi)$.

La Definición 6.17 indica cómo aplicar una sustitución F a una fórmula ϕ resulta en una fórmula $\overline{F}(\phi)$. La sustitución no tiene efecto alguno sobre las constantes *true* y *false*, ni tampoco sobre predicados sin argumentos (casos (1)-(2)). Una sustitución aplicada a una fórmula que corresponde a un predicado con al menos un argumento, resulta en la sustitución aplicada a cada uno de sus subtérminos (caso (3)). Note que en este caso la sustitución se aplica sobre términos como lo establece la Definición 6.15. La definición inductiva para combinaciones Booleanas de fórmulas es considera en los casos (4) y (5). La situación con las cuantificaciones es más interesante (caso (6)). Una sustitución *no* afecta la variable cuantificada en $\forall x$ o $\exists x$. Además, ninguna aparición de x en ψ, libre o acotada, se sustituye cuando se aplica una sustitución a una fórmula $\forall x\, \psi$ o $\exists x\, \psi$. En el caso específico de las cuantificaciones, las restricciones son el instrumento que permite controlar qué variables son reemplazadas.

A continuación se presentan ejemplos ilustrando la aplicación de sustituciones a una fórmula.

Ejemplo 6.14

Considere $F = \{x_0 \mapsto f(x_2), x_1 \mapsto h(x_0, x_3), x_8 \mapsto x_7\}$ una sustitución de términos y ϕ la fórmula $B(g(x_0, x_1), f(x_1)) \wedge \mathit{true}$. Entonces $\overline{F}(\phi)$ es la fórmula:

$$B(g(f(x_2), h(x_0, x_3)), f(h(x_0, x_3))) \wedge \mathit{true}.$$

En el Ejemplo 6.14 todas las variables de la fórmula ϕ aparecen libres. Entonces, la sustitución F se aplica reemplazando todas las apariciones de las variables que afecta. En este caso, las variables sustituídas en ϕ son x_0 y x_1.

Ejemplo 6.15

Considere $F = \{x_0 \mapsto f(x_2), x_1 \mapsto h(x_0, x_3), x_8 \mapsto x_7\}$ una sustitución de términos y ϕ la fórmula en la Figura 2:

$$((\forall x_0\, P(x_0, f(x_1))) \equiv Q(x_2) \wedge \exists x_2\, P(x_2, x_0)).$$

Entonces $\overline{F}(\phi)$ es la fórmula:

$$((\forall x_0\, P(x_0, f(h(x_0, x_3)))) \equiv Q(x_2) \wedge \exists x_2\, P(x_2, f(x_2))).$$

En el Ejemplo 6.15 no todas las variables de la fórmula ϕ aparecen libres. La primera aparición de x_0 (de izquierda a derecha) es acotada y por lo tanto no es afectada por F. Caso contrario ocurre con la segunda aparición de x_0 que es libre y, consecuentemente, es sustituída por F con el término $f(x_2)$. La única aparición de x_1 es libre y es sustituída por F con el término $h(x_0, x_3)$.

Nota 6.18

Para simplificar la escritura de la extensión $\overline{F}$ de una sustitución de términos F, se adopta la convención de referirse a dicha extensión como una sustitución de términos y denotarla como F. Finalmente, si ϕ es una fórmula y F es una sustitución de términos $\{y_0 \mapsto u_0, \ldots, y_n \mapsto u_n\}$, entonces $F(\phi)$ puede escribirse como

$$\phi[y_0, \ldots, y_n := u_0, \ldots, u_n]\,.$$

Desafortunadamente, aún bajo los cuidados de la Definición 6.17, las sustituciones textuales pueden dar lugar a efectos inesperados. En una sustitución textual $\phi[x := t]$, puede suceder que una variable y aparezca en t mientras una aparición libre de x en ϕ está en el alcance de una cuantificación $\forall y$ o $\exists y$. En este caso, la variable y que en ϕ denota un valor externo, da lugar en $\phi[x := t]$ a una variable que está cuantificada universal o existencialmente. En cualquier caso, se puede cambiar el sentido de la fórmula inicial y este es en realidad el problema operativo de las sustituciones en la lógica de predicados.

Definición 6.19

Sean t un término, x una variable y ϕ una fórmula. Se dice que t *es libre para* x *en* ϕ sii ninguna aparición libre de x en ϕ está bajo el alcance de un $\forall y$ o $\exists y$, en donde y es una variable de t.

La Definición 6.19 puede ser fácil de enteder si se piensa en términos de árboles de sintaxis. Considere los árboles de sintaxis de ϕ y de t. Sin ninguna restricción, se puede obtener la fórmula $\phi[x := t]$, en la cual todas las apariciones libres de x han sido sustituídas por el árbol de sintaxis de t. De acuerdo con la Definición 6.19, lo que la expresión 't es libre para x en ϕ' siginifica es que las variables en el árbol de sintaxis de t no estarán acotadas si t sustituye las apariciones libres de x en ϕ, i.e., en $\phi[x := t]$.

Ejemplo 6.16

Sea ϕ la fórmula $B(g(x_0, x_1), f(x_1)) \wedge \mathit{true}$. Considere el término t correspondiente a $f(g(x_1, x_3))$.

- t es libre para x_1 en ϕ.
- t es libre para x_2 en ϕ.

En general, cualquier término u es libre para cada una de las variables x_0, x_1 en ϕ porque esta fórmula no tiene cuantificadores.

Ejemplo 6.17

Sea ϕ la fórmula en la Figura 2:

$$((\forall x_0\, P(x_0, f(x_1))) \equiv Q(x_2) \wedge \exists x_2\, P(x_2, x_0)).$$

- $f(x_2)$ *no* es libre para x_0 en ϕ porque hay una aparición libre de x_0 en ϕ que está bajo el alcance de un cuantificador $\exists x_2$ y x_2 es una variable en $f(x_2)$.
- x_1 es libre para x_1 en ϕ.
- $f(x_2)$ es libre para x_1 en ϕ.
- $h(x_0, x_3)$ *no* es libre para x_1 en ϕ porque hay una aparición libre de x_1 en ϕ que está bajo el alcance de un cuantificador $\forall x_0$, y x_0 es una variable en $h(x_0, x_3)$.

El Ejemplo 6.17 ilustra cómo la sustitución textual en el Ejemplo 6.15 puede tener efectos de borde porque al menos uno de los términos sustituídos fue capturado por un cuantificador. En este sentido, el Ejemplo 6.15 también es un ejemplo de una sustitución que ¡nunca! se debe hacer.

Ejercicios

1. Considere la sustitución textual F y el término t en el Ejemplo 6.12. Escriba en detalle el cálculo de $\overline{F}(t)$.
2. Considere la sustitución textual F y la fórmula ϕ en el Ejemplo 6.14.
   a) Dibuje el árbol de sintaxis de ϕ.
   b) Asocie a cada una de las etiquetas del árbol de sintaxis de ϕ la sustitución correspondiente.
   c) Dibuje el árbol de sintaxis de $\overline{F}(\phi)$.
3. Considere la sustitución textual F y la fórmula ϕ en el Ejemplo 6.15.
   a) Dibuje el árbol de sintaxis de ϕ.
   b) Asocie a cada uno de los nodos del árbol de sintaxis de ϕ la sustitución correspondiente.
   c) Dibuje el árbol de sintaxis de $\overline{F}(\phi)$.
4. Determine la fórmula $\phi[x_2 := f(x_1, x_2)]$ en donde ϕ es cada una de las siguientes fórmulas:
   a) $\forall x_2\,(P(x_1, x_2) \to P(x_2, c))$
   b) $\forall x_2\,P(x_1, x_2) \to P(x_2, c)$
   c) $Q(x_3) \to \neg\forall x_1\,\forall x_2\,R(x_1, x_2, c)$
   d) $\forall x_1\,Q(x_1) \to \forall x_2\,P(x_1, x_2)$
   e) $\forall x_2\,(P(f(x_1, x_2), x_1) \equiv \forall x_1\,S(x_3, g(x_1, x_2)))$
5. Sea t el término $f(x_1, x_2)$. Explique si t es libre para x_2 en cada una de las siguientes fórmulas:
   a) $\forall x_2\,(P(x_1, x_2) \to P(x_2, c))$
   b) $\forall x_2\,P(x_1, x_2) \to P(x_2, c)$
   c) $Q(x_3) \to \neg\forall x_1\,\forall x_2\,R(x_1, x_2, c)$
   d) $\forall x_1\,Q(x_1) \to \forall x_2\,P(x_1, x_2)$
   e) $\forall x_2\,(P(f(x_1, x_2), x_1) \equiv \forall x_1\,S(x_3, g(x_1, x_2)))$
6. En cada uno de los siguientes casos, sea ϕ la fórmula dada. Sea t el término $f(x_1, x_3)$. Calcule $\phi[x_1 := t]$ y luego explique si t es libre para x_1 en ϕ.
   a) $\forall x_2\,(P(x_2, f(x_1, x_2)) \vee Q(x_1))$
   b) $\forall x_2\,P(x_2, f(x_1, x_2)) \vee Q(x_1)$
   c) $\forall x_1\,\forall x_3\,(Q(x_3) \not\equiv Q(x_1))$
   d) $\forall x_1\,\forall x_3\,Q(x_3) \not\equiv Q(x_1)$
   e) $\forall x_2\,R(x_1, g(x_1), x_2) \to \forall x_3\,Q(f(x_1, x_3))$
7. Repita el Ejercicio 6 para cada uno de los siguientes términos t:
   a) x_2
   b) x_3
   c) $f(x_4, x_1)$

d) $h(x_1, x_2, x_3)$

8. Considere la sustitución textual F y la fórmula ϕ en el Ejemplo 6.15. Proponga un renombramiento de las variables en ϕ, sin cambiar el sentido de la fórmula, de manera tal que en la sustitución textual $F(\phi)$ no haya captura de variables.
9. Sean x, y variables, t, u términos y ϕ una fórmula. Formule un contraejemplo para la siguiente igualdad:

   $$(\phi[x := t])[y := u] = (\phi[y := u])[x := t]\,.$$

10. Sean x una variable, t un término y ϕ una fórmula. Demuestre o refute: si x no ocurre libre en ϕ, entonces t es libre para x en ϕ.
11. Sean x una variable y ϕ una fórmula. Demuestre que x es libre para x en ϕ.
12. Sea ϕ una fórmula en la cual x aparece libre y sea y una variable que no aparece libre en ϕ. Demuestre que si y es libre para x en ϕ, entonces x es libre para y en $\phi[x := y]$.

6.6. Un lenguaje para arreglos

En esta sección se presenta $\mathcal{L}_A$, un lenguaje de primer orden para arreglos de números enteros. Un arreglo es una secuencia finita de valores seleccionables por un índice. En informática, los arreglos son estructuras de datos importantes, interesantes en sí mismas, y bloques fundamentales de construcción en otras estructuras de datos como montones, diccionarios y grafos.

El lenguaje $\mathcal{L}_A$ se define por medio de un conjunto $\mathcal{F}_A$ de símbolos de función y un conjunto $\mathcal{P}_A$ de símbolos de predicado, lo cual se escribe como $\mathcal{L}_A = (\mathcal{F}_A, \mathcal{P}_A)$. Considere las siguientes afirmaciones para motivar la definición del lenguaje:

1. El arreglo a no es vacío.
2. El arreglo a está ordenado ascendentemente.
3. Los arreglos a y b son iguales.

De forma más precisa, estas frases pueden escribirse alternativamente como:

1. El arreglo a es tal que $len(a) > 0$.
2. El arreglo a es tal que si i y j son dos de sus índices y $i < j$, entonces $a[i] \leq a[j]$.
3. Los arreglos a y b son tales que $len(a) = len(b)$ y si i es uno de sus índices, entonces $a[i] = b[i]$.

Al leer las frases (1)-(3) es evidente que $\mathcal{L}_A$ debe incluír algunos símbolos que no corresponden a variables o conectivos lógicos. En la frase (1) se usan los símbolos *len* para denotar la longitud de un arreglo, $>$ para denotar una relación entre números y 0 para denotar el número cero. En la frase (2) aparecen las variables i y j que se

refieren a índices de un arreglo, y las relaciones $<, \leq$ para comparar dos números. En esta misma frase, la expresión $a[i]$ denota el valor en a asociado al índice i. En la frase (3) se usan dos nombres de arreglo diferentes, la relación $=$ para comparar dos números y una variable para denotar índices.

El lenguaje $\mathcal{L}_A$ considera tres tipos de objetos: los índices, los valores y los arreglos. Como se verá en los siguientes párrafos, estas distinciones se pueden incluír en $\mathcal{L}_A$ usando la noción de *tipo*.

Nota 6.20

Un *tipo* es una conveción sintáctica asociada a los símbolos de un lenguaje de primer orden, de función y predicado, para clasificar términos.

Los tipos de $\mathcal{L}_A$ son I para índices, V para valores y A para nombres de arreglos.

Definición 6.21

Los símbolos de función $\mathcal{F}_A$ son los siguientes:

- Una colección infinita de símbolos constantes $0, 1, 2, \ldots$ de tipo I.
- Una colección infinita de símbolos constantes $\ldots, -2, -1, 0, 1, 2, \ldots$ de tipo V.
- Una colección infinita de símbolos constantes $a_0, a_1, a_2, \ldots$ de tipo A.
- Un símbolo unario *len* de tipo V con argumento de tipo A.
- Un símbolo binario *read* de tipo V con primer argumento de tipo A y segundo argumento de tipo I.
- Los símbolos binarios $+, -, \cdot$ de tipo V con argumentos de tipo V.

Las constantes $0, 1, 2, \ldots$ representan índices para seleccionar valores almacenados en un arreglo, las constantes $\ldots, -2, -1, 0, 1, 2, \ldots$ representan valores almacenables en un arreglo y las constantes $a_0, a_1, a_2, \ldots$ identifican arreglos. Los símbolos *len* y *read* denotan, respectivamente, la cantidad de elementos de un arreglo y el valor almacenado por un arreglo en un índice dado. La distinción entre, por ejemplo, 0 como símbolo de tipo I y 0 como símbolo de tipo V se hace explícita a propósito: a pesar de las similitud caligráfica, la constante 0 es una como índice y otra como valor.

Nota 6.22

Si se abstrae el concepto de índice en $\mathcal{L}_A$, se pueden modelar arreglos más generales. Por ejemplo, se pueden considerar arreglos cuyos índices inicien desde 1 y, en un caso más general, arreglos cuyos índices sean caracteres o cadenas.

Definición 6.23

La expresión $\mathcal{X}_I$ denota una colección infinita de variables de tipo I y $\mathcal{X}_V$ una colección infinita de variables de tipo V.

La Definición 6.23 establece que $\mathcal{X}$, la colección de variables de un lenguaje de primer orden, en $\mathcal{L}_A$ se descompone en $\mathcal{X}_I$ y $\mathcal{X}_V$ (i.e., $\mathcal{X} = \mathcal{X}_I \cup \mathcal{X}_V$). Note que $\mathcal{L}_A$ no considera variables de tipo A y entonces en este lenguaje no es posible cuantificar sobre arreglos. Para facilitar el desarrollo de la sección se supondrá que $\mathcal{X}_I$ y $\mathcal{X}_V$ no tienen elementos en común (i.e., $\mathcal{X}_I \cap \mathcal{X}_V = \{\}$).

Sabiendo qué denota $\mathcal{F}_A$, la Definición 6.5 automáticamente indica cuáles son los términos del lengueja $\mathcal{L}_A$.

Ejemplo 6.18

Sean a, b un símbolos constantes de tipo A, i, j símbolos constantes de tipo I y n, m símbolos constantes de tipo V. Los siguientes son términos de $\mathcal{L}_A$:

$$i \quad j \quad a \quad len(a) \quad len(b) \quad read(a, i) \quad read(b, i) \quad n + m \quad n \cdot m$$

Los términos i, j tienen tipo I, el término a tiene tipo A, los términos $len(a), len(b)$ tienen tipo V, mientras que $read(a, i), read(b, i), n + m, n \cdot m$ tienen tipo V.

A continuación se definien los símbolos de predicado de $\mathcal{L}_A$.

Definición 6.24

Los símbolos de predicado $\mathcal{P}_A$ son los siguientes:

- Un símbolo binario $=$ cuyos dos argumentos son de tipo I.
- Símbolos binarios $=, \leq, <, \geq, >$ cuyos dos argumentos son de tipo V.

Las fórmulas de $\mathcal{L}_A$ modelan propiedades sobre arreglos y números. En este lenguaje se pueden expresar comparaciones entre números, índices y valores en arreglos. Intencionalmente no se incluye un símbolo de predicado para denotar igualdad entre arreglos por el motivo que dicho predicado puede ser definido con base en los predicados de igualdad de índices y valores (ver Ejemplo 6.21).

Al igual que en la Definición 6.21, la Definición 6.24 introduce dos símbolos con la misma notación. Este es el caso del símbolo de predicado $=$ que se usa para denotar igualdad entre índices y para denotar igualdad entre valores. Esta situación es particular en $\mathcal{L}_A$ porque la interpretación deseada de un índice es un número natural y la de un valor es un número entero, y las relaciones de igualdad en $\mathbb{N}$ y $\mathbb{Z}$ coinciden. Esta situación no es posible con un lenguaje general de arreglos. Por ejemeplo, si los valores almacenados en un arreglo son caracteres o cadenas, entonces las relaciones de igualdad entre índices y valores son distintas e incompatibles.

Nota 6.25

Se usarán algunas convenciones para simplificar la escritura de fórmulas. Las expresiones $\forall x{:}T$ y $\exists x{:}T$ denotan cuantificaciones sobre una variable x de tipo T (i.e., $x \in \mathcal{X}_T$). Si a es una constante de tipo A e i es un término de tipo I, entonces la expresión $a[i]$ denota el término $\mathit{read}(a, i)$.

A continuación se presentan ejemplos de cómo especificar en $\mathcal{L}_A$ las frases (1)-(3) en la introducción de la sección. En estas especificaciones se opta por usar arreglos cuyo primer índice es 0. Esto quiere decir que si un arreglo almacena n elementos, entonces el último índice de dicho arreglo es $n - 1$.

Ejemplo 6.19

Para la frase "el arreglo a no es vacío" se propone la fórmula

$$\mathit{len}(a) > 0.$$

En el Ejemplo 6.19, el nombre a es un parámetro de la fórmula y corresponde a una de las constantes $a_0, a_1, \ldots$ de tipo A (Definición 6.21).

Ejemplo 6.20

Para la frase "el arreglo a está ordenado ascendentemente" se propone la fórmula:

$$\forall i{:}I\,(0 \leq i < \mathit{len}(a) \rightarrow \forall j{:}I\,(0 \leq j < \mathit{len}(a) \rightarrow (i < j \rightarrow a[i] \leq a[j]))).$$

Alternativamente, usando la notación introducida en la Nota 6.10:

$$(\forall i{:}I \mid 0 \leq i < \mathit{len}(a) : (\forall j{:}I \mid 0 \leq j < \mathit{len}(a) : i < j \rightarrow a[i] \leq a[j])).$$

En el Ejemplo 6.20, el nombre a es un parámetro de la fórmula. Las fórmulas $0 \leq i < \mathit{len}(a)$ y $0 \leq j < \mathit{len}(a)$ denotan, respectivamente, que i y j son índices del arreglo a, es decir, cantidades que indexan algún valor almacenado en a.

Ejemplo 6.21

Para la frase "los arreglos a y b son iguales" se propone la fórmula:

$$(\mathit{len}(a) = \mathit{len}(b)) \wedge (\forall i{:}I \mid (\mathit{len}(a) = \mathit{len}(b)) \wedge 0 \leq i < \mathit{len}(a) : a[i] = b[i]).$$

En el Ejemplo 6.21, los nombres a y b son parámetros de la fórmula. Para que dos arreglos sean iguales es necesario que tengan la misma longitud, los mismos elementos y en el mismo orden.

Se puede ir un poco más allá de lo hecho en los ejemplos 6.19-6.21 En particular, se pueden abstraer algunos conceptos y definir nuevos símbolos de predicado, permitiendo modularidad en las fórmulas y distintos niveles de granularidad en una especificación.

Considere los siguientes símbolos: *empty* un símbolo de predicado unario con argumento de tipo A, *asc* un símbolo de predicado ternario con primer argumento de tipo A y demás argumentos de tipo I, y $=$ un símbolo de predicado binario con dos argumentos de tipo A. El significado intuitivo de estos predicados es el siguiente:

$\mathit{empty}(a)$: "el arreglo a es vacío",
$\mathit{asc}(a, x, y)$: "el subarreglo $a[x], a[x+1], \ldots, x[y-1]$ es ascendente",
$a = b$: "los arreglos a y b son iguales".

Las definiciones de estos símbolos de predicado en el lenguaje $\mathcal{L}_A$ son las siguientes:

$$\begin{aligned} empty(a) &\equiv (len(a) = 0) \\ asc(a, x, y) &\equiv 0 \leq x \leq len(a) \wedge 0 \leq y \leq len(a) \wedge \\ &\qquad (\forall i{:}I \mid x \leq i < y : (\forall j{:}I \mid x \leq j < y : i \leq j \rightarrow a[i] \leq a[j])) \\ a = b &\equiv (len(a) = len(b)) \wedge \\ &\qquad (\forall i{:}I \mid (len(a) = len(b)) \wedge 0 \leq i < len(a) : a[i] = b[i]). \end{aligned}$$

El predicado *empty* indica que un arreglo es vacío, lo cual se establece únicamente cuando su longitud es 0. El predicado *asc* generaliza el concepto de "ser ascendente" para un arreglo permitiendo que también se consideren segmentos de arreglos (i.e., subarreglos). El predicado $=$ de igualdad entre arreglos se define punto a punto y para ello es necesario que las longitudes coincidan.

Ejemplo 6.22

Usando los símbolos *empty*, *asc* y $=$, las frases (1)-(3) que motivan esta sección pueden ser especificadas por las siguientes fórmulas:

(1) : $\neg empty(a)$

(2) : $asc(a, 0, len(a))$

(3) : $a = b$.

Un ejercicio interesante y relacionado con el lenguaje $\mathcal{L}_A$, consiste en interpretar en castellano sus fórmulas.

Ejemplo 6.23

Sea a una constante de tipo A. Considere la siguiente fórmula de $\mathcal{L}_A$:

$$(\forall i{:}I \mid 0 \leq i < len(a) : (\exists j{:}I \mid 0 \leq j < len(a) : a[i] + a[j] = 0)).$$

¿Cómo se interpreta esta fórmula? En castellano, esta fórmula podría ser traducida así:

> si se toma cualquier elemento v en a ($v = a[i]$), hay un elemento u en a ($u = a[j]$) tal que u es el inverso aditivo de v.

Ejemplo 6.24

Sea a una constante de tipo A. Considere la siguiente fórmula de $\mathcal{L}_A$:

$$(\exists i{:}I \mid 0 \leq i < len(a) : a[i] = i).$$

En castellano, esta fórmula indica que el arreglo a tiene al menos un *punto fijo*.

Se concluye esta sección mostrando cómo traducir algunas fórmulas de $\mathcal{L}_A$ a código Python. En general, la práctica de mecanizar fórmulas en un lenguaje de programación es muy atractiva y útil porque entonces dichas fórmulas pueden ser evaluadas automáticamente sobre elementos concretos. En particular, al contar con fórmulas de $\mathcal{L}_A$ mecanizadas en Python, se pueden *verificar* automáticamente propiedades sobre arreglos de números enteros por medio de la ejecución de programas.

Ejemplo 6.25

Considere la siguiente función `empty` en Python3:

```
1 def empty(a):
2   assert type(a)==list
3   return len(a)==0
```

En el Ejemplo 6.25, se define el símbolo *empty* con la función `empty`. Esta función retorna de forma afirmativa únicamente cuando el arreglo dado tiene longitud 0 y de forma negativa de lo contrario; esta es la definición lógica del símbolo *empty* en $\mathcal{L}_A$. La instrucción `assert` se usa para validar la *precondición* de la función `empty`: el argumento de la función debe ser de tipo arreglo (i.e., `type(a)` debe ser `list`). La instrucción `assert` permite evaluar *aserciones*: si la condición dada se cumple, entonces el código de la función sigue su ejecución; de lo contrario, la ejecución de la función termina abruptamente con un error de violación de la aserción. A pesar de ser muy sencillo, el Ejemplo 6.25 sirve el propósito de resaltar por qué es importante identificar qué símbolos son parámetros en una fórmula y qué debe ser cierto sobre ellos. En la fórmula $empty(a)$, el nombre de arreglo a es un parámetro y, consecuentemente, la función Python `empty` tiene un argumento que corresponde al nombre del arreglo sobre el cual se hace la verificación. Note que el símbolo de función *len* de $\mathcal{L}_A$ coincide con el nombre de la función `len` de Python3, pero son símbolos en dos "mundos" distintos.

Ejemplo 6.26

Considere la siguiente función `asc` en Python3:

```
def asc(a,x,y):
  assert type(a)==list
  r = 0<=x<=len(a) and 0<=y<=len(a)
  i = x
  while r and i<y:
    j = x
    while r and j<y:
      if i<j:
        r = a[i]<=a[j]
      j += 1
    i += 1
  return r
```

La función `asc(a,x,y)` calcula en la variable `r` si el subarreglo $a[x], ..., a[y-1]$ está ordenado ascendentemente o no. Para ello es necesario que x e y sean cantidades correctas como cotas para índices de a. Si este es el caso y la función retorna negativamente, entonces hay un par de índices i y j, cumpliendo $x \leq i < y$, $x \leq j < y$ y $i < j$, tales que los valores $a[i]$ y $a[j]$ están desordenados. De lo contrario, la función retorna afirmativamente porque el subarreglo de interés no tiene desorden. La variable `r` se usa como *centinela* de los ciclos: si `r` corresponde a *false* en algún momento de la ejecución, entonces no es necesario seguir explorando el arreglo dado.

Para alguien familiarizado con programación y el uso de arreglos, la función `asc` en el Ejemplo 6.26 posiblemente no sea la primera opción cuando se desee verificar que un arreglo está ordenado ascendentemente. El principal motivo es que la función `asc` puede realizar una cantidad significativa de comparaciones innecesarias. A pesar de este defecto, la función `asc` sirve el propósito de evidenciar lo fácil que puede ser obtener una versión ejecutable de una fórmula de $\mathcal{L}_A$.

Nota 6.26

Hay un defecto fundamental con la función `asc`: mecaniza el predicado *asc* con la posibilidad de ejecutar demasiadas instrucciones para ello. Específicamente, el defecto radica en que la cantidad de iteraciones de los ciclos en `asc` crece *cuadráticamente* en función de la longitud del segmento a verificar. En la práctica, se puede formular una mecanización de *asc* de forma tal que la cantidad de iteraciones crezca *linealmente* en función de la longitud del segmento a verificar.

Finalmente, se presenta un ejemplo correspondiente a la verificación del predicado de igualdad de arreglos.

Ejemplo 6.27

Considere la siguiente función `equal` en Python3:

```
1 def equal(a,b):
2   assert type(a)==type(b)==list
3   r,i = len(a)==len(b),0
4   while r and i<len(a):
5     r,i = a[i]==b[i],i+1
6   return r
```

Ejercicios

1. Dibuje el árbol de sintaxis de los términos en el Ejemplo 6.18.
2. Sean x una variable en $\mathcal{X}_V$, i una variable en $\mathcal{X}_I$ y a un nombre de arreglo. Determine cuáles de las siguientes expresiones son términos de $\mathcal{L}_A$. En el caso de que la expresión sea un término, dibuje el árbol de sintaxis; en el caso contrario explique por qué no es un término.
   a) $len(a, 0)$
   b) $read(a, 0)$
   c) $a[0]$
   d) $read(x, i)$
   e) $x[i]$
3. Dibuje el árbol de sintaxis de la fórmula en el Ejemplo 6.19.
4. Dibuje el árbol de sintaxis de la fórmula en el Ejemplo 6.20.
5. Dibuje el árbol de sintaxis de la fórmula en el Ejemplo 6.21.
6. Dibuje el árbol de sintaxis de las fórmulas en el Ejemplo 6.22.
7. Dibuje el árbol de sintaxis de la fórmula en el Ejemplo 6.23.
8. Dibuje el árbol de sintaxis de la fórmula en el Ejemplo 6.24.
9. Sean x una variable en $\mathcal{X}_V$ y a un nombre de arreglo. Determine cuáles de las siguientes expresiones son fórmulas de $\mathcal{L}_A$. En el caso de que la expresión sea una fórmula, dibuje el árbol de sintaxis; en el caso contrario explique por qué no es una fórmula.
   a) $\forall i{:}I\,(a[i] = x)$
   b) $\forall i{:}I\,(0 \leq i < len(a) \rightarrow a[i] = x)$
   c) $\forall b{:}A\,(b[i] = x)$

d) $len(a) > 1 \wedge \exists j{:}I\, \neg(a[0] \cdot a[j] = a[0])$
e) $(\forall i{:}I \mid 0 < i < len(a) : a[i-1] \leq a[i])$

10. Exhiba tres arreglos concretos para a que satisfagan la fórmula del Ejemplo 6.20; exhiba tres que no la satisfagan.

11. Especifique en $\mathcal{L}_A$ las siguientes afirmaciones:
    a) El arreglo a es decreciente.
    b) Los arreglos a y b son distintos.
    c) El arreglo a no tiene puntos fijos.
    d) El arreglo a no tiene elementos repetidos.
    e) El arreglo a es la identidad.

12. Sea a un arreglo. Traduzca al castellano cada una de las siguientes fórmulas, y en cada caso proponga un ejemplo de un arreglo que la satisfaga y otro que no:
    a) $\exists x{:}V\, (len(a) = 2 \cdot x + 1)$
    b) $(\forall i{:}I \mid 0 \leq i < len(a) : a[i] = a[0])$
    c) $a[1] = 5 \wedge (\forall i{:}I \mid 0 \leq i < len(a) : a[i] = a[0])$
    d) $(\exists i{:}I \mid 0 \leq i < len(a) : a[i] = -a[i])$

13. Proponga una mecanización para la fórmula en el Ejemplo 6.23 en Python.

14. Proponga una mecanización para la fórmula en el Ejemplo 6.24 en Python.

15. Considere un arreglo a y la fórmula ϕ correspondiente a

$$(\forall i{:}I \mid 0 < i < len(a) : a[i-1] \leq a[i]).$$

    a) Traduzca ϕ al castellano.
    b) Identifique los parámetros de ϕ.
    c) Proponga una mecanización para ϕ en Python.

16. Una secuencia finita de valores es llamada *palíndrome* si su lectura hacia adelante y hacia atrás es la misma.
    a) ¿Es un arreglo vacío palíndrome?
    b) Presente ejemplos de tres arreglos que sean palíndromes y de tres que no lo sean.
    c) Defina en $\mathcal{L}_A$ un símbolo de predicado unario *pal* con argumento de tipo A que corresponda a la siguiente definición intuitiva: $pal(a)$ indica que el arreglo a es palíndrome.
    d) Proponga una mecanización para *pal* en Python.

17. Una secuencia finita de valores es llamada *alíndrome* si resulta de *concatenar* dos secuencias palíndromes no vacías.
    a) ¿Es un arreglo vacío alíndrome?
    b) Presente ejemplos de tres arreglos que sean alíndromes y de tres que no lo sean.

c) Defina en $\mathcal{L}_A$ un símbolo de predicado unario *al* con argumento de tipo A que corresponda a la siguiente definición intuitiva: $al(a)$ indica que el arreglo a es alíndrome.

d) Proponga una mecanización para *al* en Python.

18. Una secuencia finita de valores es llamada *bitónica* si resulta de *concatenar* una secuencia creciente (posiblemente vacía) con una secuencia decreciente (posiblemente vacía).

a) ¿Es un arreglo vacío bitónico?

b) Presente ejemplos de tres arreglos que sean bitónicos y de tres que no lo sean.

c) Defina en $\mathcal{L}_A$ un símbolo de predicado unario *bit* con argumento de tipo A que corresponda a la siguiente definición intuitiva: $bit(a)$ indica que el arreglo a es bitónico.

d) Proponga una mecanización para *bit* en Python.

19. Investigue acerca de los siguientes conceptos, explique su uso e ilústrelo con ejemplos:

- aserción
- precondición
- poscondición
- invariante

Capítulo 7

El sistema de Dijkstra y Scholten para predicados

El sistema de Dijkstra y Scholten para lógica de predicados es una extensión del sistema DS de la lógica proposicional y está paremetrizado por un lenguaje de primer orden $\mathcal{L}$. Por ello, este sistema se denomina DS($\mathcal{L}$). Al igual que DS, el sistema DS($\mathcal{L}$) está fundamentado en la equivalencia lógica y en el cambio de 'iguales por iguales'. Dado que en la lógica de predicados hay dos universos, el de los términos y el de las fórmulas, el cambio de iguales por iguales en DS($\mathcal{L}$) ocurre en los dos niveles. Es correcto decir que el enfoque 'calculativo' de la lógica de Dijkstra y Scholten es famoso gracias a su sistema de primer orden.

7.1. El sistema formal DS($\mathcal{L}$)

Inicialmente se definen los símbolos de DS($\mathcal{L}$).

Definición 7.1

Los *símbolos de* DS($\mathcal{L}$) son:

- Una colección infinita $\mathcal{X}$ de *variables*

  $x_0, x_1, x_2, \ldots$

- Una colección $\mathcal{F}$ de *símbolos de función.*

- Una colección $\mathcal{P}$ de *símbolos de predicado* que incluye una colección infinita de símbolos constantes

  $$p_0, p_1, p_2, \ldots$$

- Una función $ar : \mathcal{F} \cup \mathcal{P} \to \mathbb{N}$ de *aridad*.
- Paréntesis izquierdo '(' y paréntesis derecho ')', y la coma ','.
- Una colección de *conectivos lógicos*

  $$true, false, \neg, \equiv, \not\equiv, \vee, \wedge, \to, \leftarrow, \forall, \exists.$$

De acuerdo con la Definición 7.1, $\mathcal{L}$ es un lenguaje de primer orden (Definición 6.1). La única novedad está relacionada con los símbolos de predicado $\mathcal{P}$. En particular, $\mathsf{DS}(\mathcal{L})$ supone una cantidad infinita de símbolos de predicado de aridad 0 (i.e., símbolos de predicado constantes). Esta es una decisión técnica la cual se justificará en los siguientes párrafos.

Al igual que DS, el sistema formal $\mathsf{DS}(\mathcal{L})$ basa su poder deductivo en las propiedades de la equivalencia lógica y en el cambio de 'iguales por iguales'. Como en lógica de predicados hay dos universos, el de los términos y el de las fórmulas, en $\mathsf{DS}(\mathcal{L})$ son naturales los conceptos de cambio de iguales por iguales en dos niveles: a nivel de términos y a nivel de fórmulas. En el nivel más básico, la sustitución de términos facilita el cambio de iguales por iguales. Sin embargo, no hay un concepto similar a nivel de fórmulas y por ello es necesario definir qué significa hacer sustituciones de fórmulas en $\mathsf{DS}(\mathcal{L})$.

Definición 7.2

Una *sustitución de fórmulas* es una función

$$F : \{p_0, p_1, \ldots\} \to \mathcal{T}_{(\mathcal{F},\mathcal{P})}(\mathcal{X})$$

distinta a la identidad en una cantidad finita de elementos del dominio.

Una sustitución de fórmulas F es una función que asocia una fórmula $F(x)$ a cualquier símbolo constante de predicado $p \in \mathcal{P}$. Recuerde que $\mathcal{T}_{(\mathcal{F},\mathcal{P})}(\mathcal{X})$ denota la colección de fórmulas sobre el lenguaje $(\mathcal{F}, \mathcal{P}, \mathcal{X})$ (Definición 6.6). Al igual que una sustitución de la lógica proposicional y de términos, cualquier sustitución de fórmulas F es tal que $F(p) \neq p$ para una cantidad finita de símbolos p de aridad 0. Entonces, una sustitución de fórmulas también puede ser escrita como un conjunto finito de la forma $\{q_0 \mapsto \phi_0, q_1 \mapsto \phi_1, \ldots, q_n \mapsto \phi_n\}$ indicando que la fórmula ϕ_i está

asociada al símbolo constante q_i $(0 \leq i \leq n)$ y cualquier otro símbolo constante está asociado a sí mismo cuando este no aparece en la lista de símbolos $q_0, q_1, \ldots, q_n$.

Una sustitución de fórmulas únicamente se aplica a una fórmula. Note que no tiene sentido aplicar una sustitución de fórmulas a un término. Primero, porque un término no menciona símbolos de predicado. Segundo, porque en general un término no puede tener una fórmula como subtérmino.

Definición 7.3

Sea ϕ una fórmula y $F = \{q_0 \mapsto \phi_0, q_1 \mapsto \phi_1, \ldots, q_n \mapsto \phi_n\}$ una sustitución de fórmulas. La *sustitución textual* de F en ϕ, denotada como $\overline{F}(\phi)$, se define inductivamente para toda subfórmula de ϕ de la siguiente manera:

1. $\overline{F}(\mathit{true}) = \mathit{true}$ y $\overline{F}(\mathit{false}) = \mathit{false}$,
2. $\overline{F}(p) = F(p)$, si $p \in \{q_0, \ldots, q_n\}$,
3. $\overline{F}(p) = p$, si $\mathit{ar}(p) = 0$ y $p \notin \{q_0, \ldots, q_n\}$,
4. $\overline{F}(Q(t_1, \ldots, t_k)) = Q(t_1, \ldots, t_k)$, si $Q \in \mathcal{P}$ es tal que $\mathit{ar}(Q) = k > 0$ y $t_1, \ldots, t_k$ son términos sobre $\mathcal{F}$,
5. $\overline{F}(\neg\psi) = \neg\overline{F}(\psi)$,
6. $\overline{F}(\psi \otimes \tau) = \overline{F}(\psi) \otimes \overline{F}(\tau)$, si $\otimes \in \{\equiv, \not\equiv, \vee, \wedge, \rightarrow, \leftarrow\}$ y
7. $\overline{F}(\forall x\, \psi) = \forall x\, \overline{F}(\psi)$ y $\overline{F}(\exists x\, \psi) = \exists x\, \overline{F}(\psi)$.

Intuitivamente, la Definición 7.3 indica que la aplicación de una sustitución de fórmulas reemplaza *todas* las apariciones de los símbolos de predicado constantes asociados en dicha sustitución. Aplicar una sustitución de fórmulas F a una fórmula ϕ resulta en una fórmula $\overline{F}(\phi)$ en la cual algunos símbolos de predicado constantes pueden haber sido reemplazados. En el caso (1) se indica que una sustitución de fórmulas no afecta las constantes *true* y *false*. El caso (2) indica explícitamente cómo un símbolo de predicado constante p es reemplazado por la fórmula $F(p)$. En los casos (3) y (4) se indica cómo un símbolo de predicado constante que no aparece en el dominio de la sustitución o cuya aridad es al menos 1, nunca es afectado por una sustitución de fórmulas. Los casos (5) y (6) presentan la definición inductiva para combinaciones Booleanas de fórmulas. Finalmente, el caso (7) indica cómo se aplica una sustitución de fórmulas a una fórmula cuantificada. A diferencia de la aplicación de una sustitución de términos a una fórmula, en el caso de la sustitución de fórmulas no es necesario restringir variables: un símbolo de predicado nunca es objeto de una cuantifcación.

Nota 7.4

Para simplificar la escritura de la extensión $\overline{F}$ de una sustitución de fórmulas F, se adopta la convención de referirse a dicha extensión como una sustitución de fórmulas y denotarla como F. Finalmente, si ϕ es una fórmula y F es una sustitución de fórmulas $\{q_0 \mapsto \phi_0, \ldots, q_n \mapsto \phi_n\}$, entonces $F(\phi)$ puede escribirse como

$$\phi[q_0, \ldots, q_n := \phi_0, \ldots, \phi_n]\,.$$

A continuación se define el conjunto de axiomas de $\mathsf{DS}(\mathcal{L})$.

Definición 7.5

Sean x una variable, t un término y ϕ, ψ, τ fórmulas. El *conjunto de axiomas* de $\mathsf{DS}(\mathcal{L})$ está dado por el siguiente esquema axiomático:

$(Ax\cdot)$**:** Cualquier axioma de DS.

$(Bx1)$**:** $(\forall x\, \phi) \equiv \phi$, si x no aparece libre en ϕ.

$(Bx2)$**:** $\phi \vee (\forall x\, \psi) \equiv \forall x\, (\phi \vee \psi)$, si x no aparece libre en ϕ.

$(Bx3)$**:** $(\forall x\, \phi) \wedge (\forall x\, \psi) \equiv \forall x\, (\phi \wedge \psi)$.

$(Bx4)$**:** $(\forall x\, \phi) \to \phi[x := t]\,,$ si t es libre para x en ϕ.

Cualquier axioma de DS es un axioma de $\mathsf{DS}(\mathcal{L})$ y para su identificación se conservan los nombres de DS. La cuantificación de una variable no tiene efecto sobre una fórmula en la cual dicha variable no aparece libre $(Bx1)$. La disyunción distribuye sobre la cuantificación universal siempre y cuando la fórmula siendo distribuída no sea capturada por el cuantificador $(Bx2)$. La cuantificación universal y la conjunción conmutan $(Bx3)$. Una fórmula cuantificada universalmente puede ser 'particularizada' por cualquier término, siempre y cuando dicho término sea libre para la variable cuantificada en dicha fórmula $(Bx4)$. Finalmente, note que algunos paréntesis en la Definición 7.5 se incluyen únicamente por claridad en la lectura y pueden ser eliminados de acuerdo con las convenciones de precedencia en la Nota 6.7.

A continuación se define el conjunto de reglas de inferencia de $\mathsf{DS}(\mathcal{L})$.

Definición 7.6

Sean x una variable, p un símbolo de predicado con aridad 0 y ϕ, ψ, τ fórmulas. Las reglas de inferencia de DS($\mathcal{L}$) son:

$$\frac{\psi \qquad \psi \equiv \phi}{\phi}\ \textsc{Ecuanimidad}$$

$$\frac{\psi \equiv \tau}{\phi[p := \psi] \equiv \phi[p := \tau]}\ \textsc{Leibniz} \qquad\qquad \frac{\phi}{\forall x\, \phi}\ \textsc{Generalización}$$

El sistema DS($\mathcal{L}$) cuenta con tres reglas de inferencia. Las reglas Ecuanimidad y Leibniz son similares a las reglas de inferencia de DS. La tercera regla de inferencia es Generalización e indica que si una fórmula es teorema, entonces también lo es cualquiera de sus versiones cuantificadas universalmente (i.e., para cualquier variable en $\mathcal{X}$).

Nota 7.7

Note que los conceptos de derivación (Definición 4.20) y derivación relajada (Definición 5.4) definidos para DS pueden ser definidos de manera similar para DS. Algo similar sucede con las reglas de inferencia derivadas en el Capítulo 4 para DS, las cuales deben ser correctas para DS($\mathcal{L}$), al igual que con las técnicas de razonamiento y demostración del Capítulo 5. Sin embargo, se debe tener cuidado con el Metateorema de la Deducción para DS($\mathcal{L}$) que es distinto a su contraparte en DS (ver Sección 7.4).

Se concluye esta sección con un ejemplo de una demostración en DS($\mathcal{L}$).

Ejemplo 7.1

Considere la argumentación del Ejemplo 6.1:

> Todos los informáticos son intelectualmente destacados. Turing es un informático. Entonces, Turing es intelectualmente destacado.

Además, considere la siguiente simbolización para cualquier término t:

$I(t)$: "t es informático".

$D(t)$: "t es intelectualmente destacado".

Usando $\mathsf{DS}(\mathcal{L})$ se tiene el siguiente razonamiento:

1.	$\forall x\,(I(x) \to D(x))$	(suposición 1)
2.	$I(\mathit{Turing})$	(suposición 2)
3.	$\forall x\,(I(x) \to D(x)) \to (I(\mathit{Turing}) \to D(\mathit{Turing}))$	(Bx4)
4.	$I(\mathit{Turing}) \to D(\mathit{Turing})$	(Modus Ponens 1,3)
5.	$D(\mathit{Turing})$	(Modus Ponens 2,4).

Ejercicios

1. Sean x, y variables, f un símbolo de función con $ar(f) = 2$ y H un símbolo de predicado con $ar(H) = 1$. Sea F la sustitución de fórmulas

   $$\{p_0 \mapsto p_1, p_2 \mapsto \mathit{true}, p_3 \mapsto H(x), p_4 \mapsto p_4\}.$$

   Determine la sustitución textual de F para cada una de las siguientes fórmulas:
   a) p_0
   b) $H(y) \equiv \forall x\, H(x) \wedge \mathit{false}$
   c) $\exists x \forall y\,(H(f(x, y)) \vee p_3)$

2. Sean x, y, f, H como en el Ejercicio 1. Para cada una de las siguientes fórmulas ϕ, determine si $F(p_i)$ es libre para p_i en ϕ $(i = 0, 1, 2, 3, 4)$:
   a) p_0
   b) $H(y) \equiv \forall x\, H(x) \wedge \mathit{false}$
   c) $\exists x \forall y\,(H(f(x, y)) \vee p_3)$

3. Defina cada uno de los siguientes conceptos para $\mathsf{DS}(\mathcal{L})$:
   a) Demostración.
   b) Derivación.
   c) Derivación de debilitamiento.
   d) Derivación de fortalecimiento.

4. Sea una ϕ una proposición. Demuestre que si $\vdash_{\mathsf{DS}} \phi$, entonces $\vdash_{\mathsf{DS}(\mathcal{L})} \phi$. Note que los símbolos de predicado $p_0, p_1, \ldots$ en $\mathcal{P}$ corresponden a las variables proposicionales de DS. Entonces, cualquier proposición es una fórmula de $\mathsf{DS}(\mathcal{L})$.

5. Demuestre que la regla de inferencia Modus Ponens es correcta en $\mathsf{DS}(\mathcal{L})$:

   $$\frac{\phi \qquad \phi \to \psi}{\psi}\ \text{Modus Ponens}$$

en donde ϕ y ψ son fórmulas de $\mathsf{DS}(\mathcal{L})$.

6. Simbolice cada una de las siguientes argumentaciones y para cada una de ellas demuestre que es correcta o formule un contraejemplo.
   a) Todos los hombres son mortales. Sócrates es hombre. Entonces Sócrates es mortal.
   b) No todos los estudiantes asisten a todas las clases. Entonces, todas las clases están vacías.
   c) La relación binaria R es reflexiva y transitiva. Entonces R es antisimétrica.
7. Demuestre que el Metateorema 5.13 de demostración por doble implicación es cierto para $\mathsf{DS}(\mathcal{L})$. En particular, demuestre para cualquier par de fórmulas ϕ y ψ, y Γ un conjunto de fórmulas:

$$\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi \equiv \psi \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi \to \psi \quad \text{y} \quad \Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \phi.$$

8. Demuestre el Metateorema 5.14 de demostración por contradicción para $\mathsf{DS}(\mathcal{L})$. En particular, para ϕ y ψ fórmulas, y Γ un conjunto de fórmulas:
   a) $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi$ sii $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \neg\phi \to \mathit{false}$.
   b) $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \phi$ sii $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \psi \wedge \neg\phi \to \mathit{false}$.
9. Demuestre el Matateorema 5.16 de demostración por contrapositiva para $\mathsf{DS}(\mathcal{L})$.
10. Demuestre el Matateorema 5.17 de demostración por contrapositiva para $\mathsf{DS}(\mathcal{L})$.
11. Demuestre el Metateorema 5.18 para $\mathsf{DS}(\mathcal{L})$, en donde $p \in \mathcal{P}$ es tal que $ar(p) = 0$.
12. Demuestre el Metateorema 5.21 para $\mathsf{DS}(\mathcal{L})$.
13. Demuestre el Metateorema 5.22 para $\mathsf{DS}(\mathcal{L})$.

7.2. La cuantificación universal

El cuantificador universal $\forall$ en la lógica de predicados es un conectivo lógico que expresa la veracidad de una fórmula para *cada uno* de los términos del lenguaje (i.e., para cada uno de los elementos del universo del discurso).

Inicialmente se presentan algunos teoremas básicos.

Teorema 7.8

Para cualquier variable x y fórmula ϕ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \mathit{true} \equiv \mathit{true}$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \mathit{false} \equiv \mathit{false}$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x \forall x\, \phi \equiv \forall x\, \phi$

Los teoremas 7.8.1 y 7.8.2 establecen que la cuantificación universal no tiene efecto alguno sobre las constantes Booleanas y el Teorema 7.8.3 establece que la cuantificación universal es idempotente.

Demostración. A continuación se presenta una demostración de (1); demostraciones de (2) y (3) se proponen como ejercicio para el lector.

$$
\begin{array}{ll}
& \forall x\ \mathit{true} \\
\equiv & \langle\ \text{axioma } (Bx1)\text{: } x \text{ no aparece libre en } \mathit{true}\ \rangle \\
& \mathit{true}.
\end{array}
$$

□

De acuerdo con la Nota 6.10, la expresión $(\forall x \mid \psi : \phi)$ es azúcar sintáctico para la fórmula $\forall x\,(\psi \to \phi)$, en donde la fórmula ψ es denominada el rango de la cuantificación y ϕ el término de la cuantificación. A continuación se presentan algunos teoremas de *trueque* útiles para manipular rangos y términos en una cuantificación universal.

Teorema 7.9

Para cualesquiera variable x y fórmulas ϕ, ψ, τ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \psi : \phi) \equiv (\forall x \mid: \neg\psi \vee \phi)$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \psi \wedge \tau : \phi) \equiv (\forall x \mid: \psi \wedge \tau \to \phi)$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \psi \wedge \tau : \phi) \equiv (\forall x \mid \psi : \tau \to \phi)$

Los teoremas de trueque indican cómo mover fórmulas entre el rango y el término de una cuantificación. El Teorema 7.9.1 establece que una fórmula en el rango puede ser negada y pasada al término bajo una disyunción y viceversa. Los teoremas 7.9.2 y teoremas 7.9.3 indican cómo hacer trueque de fórmulas que corresponden a una conjunción.

Demostración. Se presenta una demostración de (1); demostraciones de (2) y (3) se proponen como ejercicio para el lector.

$$
\begin{array}{ll}
 & (\forall x \mid \psi : \phi) \\
\equiv & \langle \text{ azúcar sintáctico } \rangle \\
 & \forall x\, (\psi \to \phi) \\
\equiv & \langle \text{ definición alternativa de } \to \rangle \\
 & \forall x\, (\neg\psi \vee \phi) \\
\equiv & \langle \text{ identidad de } \to \rangle \\
 & \forall x\, (\mathit{true} \to \neg\psi \vee \phi) \\
\equiv & \langle \text{ azúcar sintáctico } \rangle \\
 & (\forall x \mid \mathit{true} : \neg\psi \vee \phi) \\
\equiv & \langle \text{ azúcar sintáctico } \rangle \\
 & (\forall x \mid : \neg\psi \vee \phi).
\end{array}
$$

□

Algunas monotonías de la lógica de predicados son especialmente útiles para hacer cálculos. El Teorema 7.10 presenta algunas de ellas.

Teorema 7.10

Para cualesquiera variable x y fórmulas ϕ, ψ, τ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, (\psi \to \phi) \to (\forall x\, \psi \to \forall x\, \phi)$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, (\psi \equiv \phi) \to (\forall x\, \psi \equiv \forall x\, \phi)$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \tau : \psi \to \phi) \to ((\forall x \mid \tau : \psi) \to (\forall x \mid \tau : \phi))$
4. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \tau : \psi \equiv \phi) \to ((\forall x \mid \tau : \psi) \equiv (\forall x \mid \tau : \phi))$

Los teoremas 7.10.1 y 7.10.3 son versiones de la monotonía de la implicación bajo la cuantificación universal, mientras que los teoremas 7.10.2 y 7.10.4 son versiones de la monotonía de la equivalencia bajo la cuantificación universal. Recuerde que la expresión $\forall x\, (\psi \equiv \phi) \to (\forall x\, \psi \equiv \forall x\, \phi)$ corresponde a la fórmula $\forall x\, (\psi \equiv \phi) \to ((\forall x\, \psi) \equiv (\forall x\, \phi))$. Las demostraciones de estos teoremas se proponen como ejercicios para el lector.

A continuación se presentan algunos teoremas de distribución de conectivos lógicos sobre la cuantificación universal.

Teorema 7.11

Sean x una variable y ϕ, ψ fórmulas. Si x no aparece libre en ψ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} \psi \wedge \forall x\, \phi \equiv \forall x\, (\psi \wedge \phi)$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \forall x\, \phi \equiv \forall x\, (\psi \to \phi)$

Los teoremas 7.11.1-2 indican, respectivamente, cómo la conjunción y la implicación distribuyen sobre la cuantificación universal cuando la fórmula distribuída no es afectada por dicho cuantificador.

Demostración. Se presenta una demostración de (1); una demostración de (2) se propone como ejercicio para el lector.

$$
\begin{array}{ll}
 & \psi \wedge \forall x\, \phi \\
\equiv & \langle \text{ axioma } (Bx1)\text{: } x \text{ no aparece libre en } \psi \rangle \\
 & \forall x\, \psi \wedge \forall x\, \phi \\
\equiv & \langle \text{ axioma } (Bx3) \rangle \\
 & \forall x\, (\psi \wedge \phi).
\end{array}
$$

□

Finalmente, se presentan algunos teoremas para la manipulación de rangos y el renombramiento de variables.

Teorema 7.12

Para cualesquiera variables x e y, y fórmulas ϕ, ψ, τ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \mathit{false} : \phi)$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \psi \vee \tau : \phi) \equiv (\forall x \mid \psi : \phi) \wedge (\forall x \mid \tau : \phi)$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \phi \equiv \forall y\, (\phi[x := y])$, si y no aparece libre en ϕ.
4. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \forall y\, \phi \equiv \forall y\, \forall x\, \phi$.

El Teorema 7.12.1 se llama *regla del rango vacío* pues indica cómo operar una cuantificación universal cuando ningún elemento del dominio del discurso satisface su rango. El Teorema 7.12.2 se llama *regla de ruptura de rango* pues indica cómo operar una cuantificación universal cuando su rango corresponde a una disyunción. El Teorema 7.12.3 permite el renombramiento de variables y el Teorema 7.12.4 permite el intercambio de variables cuantificadas universalmente.

Ejercicios

1. Demuestre el Teorema 7.8.2.
2. Demuestre el Teorema 7.8.3.
3. Demuestre el Teorema 7.9.2.
4. Demuestre el Teorema 7.9.3.
5. Sean ϕ, ψ, τ fórmulas; demuestre los siguientes teoremas de trueque:
   a) $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid \psi : \phi) \equiv (\forall x \mid: \psi \vee \phi \equiv \phi)$
   b) $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid: \phi \wedge \psi \to \tau) \equiv (\forall x \mid \phi : (\psi \to \tau))$
6. Demuestre el Teorema 7.10.1.
7. Demuestre el Teorema 7.10.2.
8. Demuestre el Teorema 7.10.3.
9. Demuestre el Teorema 7.10.4.
10. Demuestre o refute: $\vdash_{\mathsf{DS}(\mathcal{L})} \psi \leftarrow \forall x\, \phi \equiv \forall x\, (\psi \leftarrow \phi)$.
11. Demuestre el Teorema 7.11.1.
12. Demuestre el Teorema 7.11.2.
13. Demuestre el Teorema 7.12.1.
14. Demuestre el Teorema 7.12.2.
15. Demuestre el Teorema 7.12.3.
16. Demuestre el Teorema 7.12.4.

7.3. La cuantificación existencial

El cuantificador existencial $\exists$ en la lógica de predicados es un conectivo lógico que expresa la veracidad de una fórmula para *alguno* de los términos del lenguaje (i.e., para al menos uno de los elementos del universo del discurso).

La cuantificación existencial se define con base en la cuantificación universal.

Definición 7.13

Sea ϕ una fórmula y x una variable. El siguiente axioma de $\mathsf{DS}(\mathcal{L})$ define la cuantificación existencial:

$(Bx5)$**:** $\exists x\, \phi \equiv \neg \forall x\, \neg \phi$.

Intuitivamente, la Definición 7.13 indica que al menos un elemento satisface una fórmula siempre y cuando no todos los elementos satisfagan su negación.

Inicialmente se presenta un teorema que generaliza la definición del cuantificador existencial a fórmulas existenciales escritas con azúcar sintáctico.

Teorema 7.14

Para cualquier variable x y fórmulas ϕ, ψ:

$$\vdash_{\mathsf{DS}(\mathcal{L})} (\exists x \mid \psi : \phi) \equiv \neg(\forall x \mid \psi : \neg\phi).$$

La propiedad presentada en el Teorema 7.14 está dada por una fórmula con azúcar sintáctico para complementar el axioma que define la cuantificación existencial. Note que el rango no se niega al cambiar el cuantificador, solo el término (además de la fórmula como tal).

Demostración.

$$\begin{array}{ll} & (\exists x \mid \psi : \phi) \\ \equiv & \langle \text{ azúcar sintáctico } \rangle \\ & \exists x\, (\psi \wedge \phi) \\ \equiv & \langle \text{ axioma } (Bx5) \rangle \\ & \neg(\forall x\, \neg(\psi \wedge \phi)) \\ \equiv & \langle \text{ DeMorgan: Teorema 4.25.2 } \rangle \\ & \neg(\forall x\, (\neg\psi \vee \neg\phi)) \\ \equiv & \langle \text{ definición alternativa de la implicación: Teorema 4.28.1 } \rangle \\ & \neg(\forall x\, (\psi \to \neg\phi)) \\ \equiv & \langle \text{ azúcar sintáctico } \rangle \\ & \neg(\forall x \mid \psi : \neg\phi). \end{array}$$

□

El comportamiento del cuantificador existencial es similar al del universal en presencia de constantes y de sentencias. Al igual que sucede con la cuantificación universal, un cuantificador existencial no tiene efecto sobre las constantes Booleanas (teoremas 7.15.1 y 7.15.2) y es idempotente (Teorema 7.15.3).

Teorema 7.15

Para cualquier variable x y fórmula ϕ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, \mathit{true} \equiv \mathit{true}$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, \mathit{false} \equiv \mathit{false}$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x \exists x\, \phi \equiv \exists x\, \phi$

A continuación se presenta un teorema de la cuantificación existencial como pareja del Axioma $(Bx4)$ de la cuantificación universal.

Teorema 7.16

Sean x una variable, t un término y ϕ una fórmula. Si t es libre para x en ϕ:

$$\vdash_{\mathsf{DS}(\mathcal{L})} \phi[x := t] \to \exists x\, \phi.$$

El Teorema 7.16 indica que $\exists x\, \phi$ es cierto siempre y cuando haya un término t que reemplace a x y que haga $\phi[x := t]$ cierta. Este teorema es importante en la práctica porque establece una condición suficiente para que una fórmula cuantificada existencialmente sea cierta: basta con encontrar un *testigo* que permita concluir que la fórmula es cierta.

La cuantificación existencial conmuta con la disyunción, es decir, se pueden juntar dos cuantificaciones existenciales cuando están conectadas por una disyunción. El Teorema 7.17 formaliza esta propiedad.

Teorema 7.17

Sean x una variable y ϕ, ψ fórmulas:

$$\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, \phi \vee \exists x\, \psi \equiv \exists x\, (\phi \vee \psi).$$

Note que la distribución de la cuantificación existencial sobre la disyunción no está supeditada a que las fórmulas sean cerradas o no mencionen a dicha variable. Básicamente, gracias a este teorema, una disyunción de cuantificaciones existenciales se puede simplificar a una fórmula con un solo cuantificador existencial.

A continuación se presentan algunos teoremas de distribución de conectivos lógicos sobre la cuantificación existencial.

Teorema 7.18

Sean x una variable y ϕ, ψ fórmulas. Si x no aparece libre en ψ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} \psi \vee \exists x\, \phi \equiv \exists x\, (\psi \vee \phi)$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} \psi \wedge \exists x\, \phi \equiv \exists x\, (\psi \wedge \phi)$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \exists x\, \phi \equiv \exists x\, (\psi \to \phi)$

Para la cuantificación existencial también hay teoremas de manipulación de rangos y cambio de variables. Cuando el rango es vacío, una cuantificación existencial es vacua. El rango de una cuantificación existencial puede comutar con el término cuando hay conjunciones. Cuando en el rango hay una disyunción, se puede dividir la cuantificación. Estas dos propiedades son fácilmente verificables deshaciendo el azúcar sintáctico de las fórmulas. Se puede hacer renombramiento de varibles únicamente cuando las nuevas variables sean frescas. Finalmente, los cuantificadores existenciales pueden alternar sin alterar el valor de verdad de una fórmula.

Teorema 7.19

Para cualesquiera variables x e y, término t y fórmulas ϕ, ψ, τ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} (\exists x \mid \mathit{false} : \phi) \equiv \mathit{false}$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} (\exists x \mid \tau \wedge \phi : \psi) \equiv (\exists x \mid \tau : \phi \wedge \psi)$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} (\exists x \mid \psi \vee \tau : \phi) \equiv (\exists x \mid \psi : \phi) \vee (\exists x \mid \tau : \phi)$
4. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, \phi \equiv \exists y\, (\phi[x := y])$, si y no aparece libre en ϕ.
5. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, \exists y\, \phi \equiv \exists y\, \exists x\, \phi$.

El Teorema 7.20 presenta propiedades que relacionan la cuantificación universal y la cuantificación existencial por medio de la implicación lógica.

Teorema 7.20

Sean x una variable y ϕ, ψ fórmulas. Si x no aparece libre en ϕ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \psi \to \phi \equiv \exists x\, (\psi \to \phi)$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, \psi \to \phi \equiv \forall x\, (\psi \to \phi)$
3. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, (\phi \to \psi) \equiv \forall x\, \phi \to \exists x\, \psi$

Estos teoremas pueden parecer contraintuitivos dados los cambios que hay, entre antecedente y consecuente, de los cuantificadores. Una sugerencia para no equivocarse al manipular cuantificadores en la presencia de la implicación: pensar en la implicación en términos de la disyunción, y usar el hecho de que la disyunción distribuye, cuando no hay captura de variables, sobre fórmulas cuantificadas universal o existencialmente.

Para terminar la sección, se presentan teoremas de debilitamiento/fortalecimiento y monotonía.

Teorema 7.21

Para cualesquiera variable x y fórmulas ϕ, ψ:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} \exists x\, \phi \to \exists x\, (\phi \vee \psi)$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, (\phi \to \psi) \to (\exists x\, \phi \to \exists x\, \psi)$

El Teorema 7.21.1 es una versión de debilitamiento/fortalecimiento del cuantificador existencial. El Teorema 7.21.2 es una monotonía de la cuantificación universal para implicaciones. Las demostraciones de estos teoremas, y los demás que no han sido demostrados en el desarrollo de la sección, se proponen como ejercicio para el lector.

Ejercicios

1. Simbolice cada una de las siguientes argumentaciones y para cada una de ellas demuestre que es correcta o formule un contraejemplo.
   a) Si hay quien pague impuestos, entonces todos los políticos pagan impuestos. Si hay algún filántropo, entonces todo aquel que pague impuestos es filántropo. Consecuentemente, si hay algún filántropo que pague impuestos, entonces todos los políticos son filántropos.
   b) Si hay quien pague impuestos, entonces todos los políticos pagan impuestos. Si hay algún filántropo, entonces todo aquel que pague impuestos es filántropo. Consecuentemente, si hay algún filántropo que pague impuestos, entonces todos los filántropos son políticos.
2. Demuestre el Teorema 7.15.1.
3. Demuestre el Teorema 7.15.2.
4. Demuestre el Teorema 7.15.3.
5. Demuestre el Teorema 7.16.
6. Demuestre el Teorema 7.17.

7. Demuestre el Teorema 7.18.1.
8. Demuestre el Teorema 7.18.2.
9. Demuestre el Teorema 7.18.3.
10. Demuestre el Teorema 7.19.1.
11. Demuestre el Teorema 7.19.2.
12. Demuestre el Teorema 7.19.3.
13. Demuestre el Teorema 7.19.4.
14. Demuestre el Teorema 7.19.5.
15. Demuestre el Teorema 7.20.1.
16. Demuestre el Teorema 7.20.2.
17. Demuestre el Teorema 7.20.3.
18. Demuestre el Teorema 7.21.1.
19. Demuestre el Teorema 7.21.2.
20. Suponga que x no aparece libre en ϕ. Demuestre o refute:

$$\vdash_{\mathsf{DS}(\mathcal{L})} \phi \leftarrow \exists x\, \psi \equiv \exists x\, (\phi \leftarrow \psi).$$

7.4. Algunos metateoremas

Esta sección presenta algunos metateoremas útiles para obtener demostraciones en $\mathsf{DS}(\mathcal{L})$. Estos metateoremas son ciertos no solamente para $\mathsf{DS}(\mathcal{L})$ sino también para cualquier sistema razonable de primer orden. En el desarrollo de la sección se usan los conceptos de demostración y derivación con suposiciones, cuyas definiciones corresponden a extensiones de aquellas de la lógica proposicional y se proponen como ejercicio para el lector.

El primer metateorema recibe el nombre de *Metateorema de Generalización*, al igual que una de las reglas de inferencia de $\mathsf{DS}(\mathcal{L})$.

Metateorema 7.22

Sean x una variable, ϕ una fórmula y Γ una colección de fórmulas:

$$\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \phi \qquad \text{sii} \qquad \Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi.$$

El Metateorema de Generalización (7.22) indica que para demostrar que una fórmula cuantificada universalmente es teorema, basta con ignorar el cuantificador universal. Es importante advertir que este metateorema indica que una fórmula y su versión cuantificada universalmente son *equidemostrables*, mas no equivalentes.

Confundir estos dos conceptos es un error común en quienes inician el estudio de la lógica.

Demostración. Se procede por doble implicación; basta con demostrar:

1. Si $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \phi$, entonces $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi$.
2. Si $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi$, entonces $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \phi$.

Para (1) se presenta la siguiente demostración con suposiciones en Γ:

1.	$\forall x\, \phi$	(suposición)
2.	$\forall x\, \phi \to \phi[x := x]$	($Bx4$; x es libre para x en ϕ)
3.	$\phi[x := x]$	(Modus Ponens 1,2).

Note que $\phi[x := x]$ y ϕ son la misma fórmula. En consecuencia, si $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \phi$, entonces $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi$.

Para (2) se propone la siguiente demostración con suposiciones en Γ:

1.	ϕ	(suposición)
2.	$\forall x\, \phi$	(Generalización 1).

En consecuencia, si $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \phi$, entonces $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, \phi$. □

Al igual que DS, el sistema $\mathsf{DS}(\mathcal{L})$ cuenta una versión del Metateorema de la Deducción en la cual hay diferencias sutiles. Estas diferencias se introducen para evitar cometer errores a causa de las variables libres.

Metateorema 7.23

Sean ϕ, ψ fórmulas y Γ un conjunto de fórmulas. Si

1. $\Gamma \cup \{\psi\} \vdash_{\mathsf{DS}(\mathcal{L})} \phi$ y
2. la demostración en (1) no usa la regla Generalización sobre una variable libre de ψ,

entonces $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \phi$.

En realidad, la única restricción para usar el Metateorema de la Deducción (7.23) consiste no aplicar la regla Generalización sobre alguna de las variables libres de la fórmula que se está utilizando como suposición. Si este es el caso, entonces este metateorema coincide con su versión proposicional. La demostración del Metateorema de la Deducción se propone como ejercicio para el lector.

A continuación se presenta un ejemplo que muestra la importancia de la condición (2) en el Metateorema de la Deducción (7.23).

Ejemplo 7.2

Sea E un predicado unario cuya interpretación en $\mathbb{Z}$ es la siguiente:

$$E(x) : \text{“}x = 0\text{”}.$$

Fíjese que por la regla Generalización, se tiene que $\{E(x)\} \vdash_{\mathsf{DS}(\mathcal{L})} \forall x\, E(x)$. Sin embargo, la fórmula $(E(x) \to \forall x\, E(x))$ no puede ser cierta porque no todos los números enteros son iguales a 0.

Para cierto tipo de fórmulas se puede formular una versión del Metateorema 7.23 más sencilla. En particular, para aquellas fórmulas que no tienen variables libres el Metateorema de la Deducción se puede usar como en lógica proposicional cuando las suposiciones son sentencias (i.e., no tienen variables libres).

Metateorema 7.24

Sean ϕ, ψ fórmulas y Γ un conjunto de fórmulas. Si

1. $\Gamma \cup \{\psi\} \vdash_{\mathsf{DS}(\mathcal{L})} \phi$ y
2. ψ es una sentencia

entonces $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \phi$.

Demostración. Es suficiente con demostrar las condiciones (1) y (2) del Metateorema 7.23. Note que la condición (1) del Metateorema 7.23 coincide con la suposición (1) (del metateorema que está siendo demostrado). Por la suposición (2) se tiene que ψ es una sentencia; consecuentemente ψ no tiene variables libres. De esta forma la condición (2) del Metateorema 7.23 se cumple trivialmente. Entonces, $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \phi$. □

El converso del Metateorema de la Deducción es cierto sin las condiciones sobre el uso de la regla Generalización para las variables libres de la suposición. A continuación se formula este metateorema.

Metateorema 7.25

Sean ϕ, ψ fórmulas y Γ un conjunto de fórmulas:

$$\text{si} \quad \Gamma \vdash_{\mathsf{DS}(\mathcal{L})} \psi \to \phi \quad \text{entonces} \quad \Gamma \cup \{\psi\} \vdash_{\mathsf{DS}(\mathcal{L})} \phi.$$

El Metateorema 7.25 tiene la misma formulación que su contraparte en DS; su demostración se propone como ejercicio para el lector.

En lo que resta de esta sección, se presentan ejemplos que ilustran el uso de los metateoremas introducidos hasta ahora. Para ello, se apela al concepto de *sucesión de números reales*, un objeto matemático estudiado en los cursos introductorios de cálculo diferencial.

Nota 7.26

Una *sucesión de números reales* (o *sucesión*) es una función $f : \mathbb{N} \to \mathbb{R}$.

Se distinguen tres tipos para modelar sucesiones en DS($\mathcal{L}$): uno para números naturales, uno para números reales y uno para sucesiones. El tipo de números naturales se denota con la letra N y el de los números reales con la letra R.

El interés principal es el estudio de algunas propiedades del límite de una sucesión, cuando este existe. En particular, el interés es establecer propiedades del predicado binario *limit* definido para cualquer sucesión f y número real x de la siguiente manera:

$$\mathit{limit}(f, x) : \text{“el límite de } f \text{ es } x\text{.”}$$

Claramente esta no es una definición formal de lo que significa que un valor x sea el límite de una sucesión f. De acuerdo con Wikipedia, una definición (adaptada) es la siguiente:

> una sucesión f tiene límite x cuando n tiende a infinito, si para todo valor $\epsilon > 0$ por pequeño que sea, existe un valor m a partir del cual si $n > m$ se tiene que la distancia de x a $f(n)$ es menor que ε.

Esta definición, aún informal, se especifica formalmente en el siguiente ejemplo con ayuda de DS($\mathcal{L}$).

Ejemplo 7.3

A continuación se presenta una definición de *limit*:

$$\mathit{limit}(f, x) \equiv (\forall \epsilon{:}R \mid \epsilon > 0 : (\exists m{:}R \mid m \geq 0 : (\forall n{:}N \mid n > m : \mathit{abs}(f(n) - x) < \epsilon))).$$

Note que en la definición de *limit* (Ejemplo 7.3) se hace explícito el tipo de las variables que se emplean. Además, es claro el alcance de cada uno de los cuantificadores. La expresión *abs* es un símbolo de función unario de tipo R cuyo argumento es de tipo R que denota el valor absoluto. Los símbolos $<, >, \geq, \leq$ son los predicados de comparación usuales para números.

En el siguiente ejemplo se usan propiedades básicas de los números naturales y reales.

Ejemplo 7.4

Sea f la función definida por $f(n) = 1$ para todo $n \in \mathbb{N}$. A continuación se demuestra que 1 es el límite de f. El objetivo es demostrar:

$$\vdash_{\mathsf{DS}(\mathcal{L})} (\forall \epsilon{:}R \mid \epsilon > 0 : (\exists m{:}R \mid m \geq 0 : (\forall n{:}N \mid n > m : abs(f(n) - 1) < \epsilon))).$$

Por el Metateorema 7.22, basta con demostrar:

$$\{\epsilon > 0\} \vdash_{\mathsf{DS}(\mathcal{L})} (\exists m{:}R \mid m \geq 0 : (\forall n{:}N \mid n > m : abs(f(n) - 1) < \epsilon)).$$

Considere la siguiente derivación:

$$\begin{array}{ll} & (\exists m{:}R \mid m \geq 0 : (\forall n{:}N \mid n > m : abs(f(n) - 1) < \epsilon)) \\ \equiv & \langle \text{ azúcar sintáctico } \rangle \\ & \exists m{:}R\, (m \geq 0 \wedge (\forall n{:}N \mid n > m : abs(f(n) - 1) < \epsilon)) \\ \leftarrow & \langle \text{ instanciación con testigo } m = 0 \text{ (Teorema 7.16)} \rangle \\ & 0 \geq 0 \wedge (\forall n{:}N \mid n > 0 : abs(f(n) - 1) < \epsilon) \\ \equiv & \langle \text{ reflexividad de } \geq \text{; identidad de la conjunción } \rangle \\ & (\forall n{:}N \mid n > 0 : abs(f(n) - 1) < \epsilon). \end{array}$$

Por el Metateorema 7.22, basta con demostrar:

$$\{\epsilon > 0, n > 0\} \vdash_{\mathsf{DS}(\mathcal{L})} abs(f(n) - 1) < \epsilon.$$

$$\begin{array}{ll} & abs(f(n) - 1) < \epsilon \\ \equiv & \langle \text{ definición de } f \rangle \\ & abs(1 - 1) < \epsilon \\ \equiv & \langle \text{ aritmética; definición de } abs \rangle \\ & 0 < \epsilon \\ \equiv & \langle \text{ suposición } \rangle \\ & true. \end{array}$$

Entonces $\vdash_{\mathsf{DS}(\mathcal{L})} limit(f, 1)$.

Ejercicios

1. Sea Γ una colección de fórmulas. Defina cada uno de los siguientes conceptos para $\mathsf{DS}(\mathcal{L})$:

a) Demostración con suposiciones en Γ.
b) Derivación con suposiciones en Γ.
c) Derivación de debilitamiento con suposiciones en Γ.
d) Derivación de fortalecimiento con suposiciones en Γ.

2. Sea x una variable y ϕ una fórmula. Refute: $\forall x\, \phi \equiv \phi$.
3. (Difícil) Demuestre el Metateorema de la Deducción (7.23).
4. Demuestre el Metateorema 7.25.
5. Repita la demostración en el Ejemplo 7.4 con testigo $m = 10$.
6. Sea f la sucesión definida por $f(n) = \frac{1}{n}$ para $n > 0$. Demuestre que f tiene límite 0.
7. Sea f la sucesión definida por $f(n) = \frac{1}{n+1}$ para $n \in \mathbb{N}$. Demuestre que f tiene límite 0.
8. Sea f la sucesión definida por $f(n) = \frac{1}{n^2}$ para $n > 0$. Demuestre que f tiene límite 0.
9. Sea f la sucesión definida por $f(n) = n$ para $n \in \mathbb{N}$. Demuestre que f no tiene límite.
10. Sean f, g sucesiones y a, b números reales. Demuestre que si $\mathit{limit}(f, a)$ y $\mathit{limit}(g, b)$, entonces $\mathit{limit}(f + g, a + b)$, en donde $(f + g)(n) = f(n) + g(n)$.
11. Demuestre que si una sucesión tiene límite, entonces este es único.

7.5. La igualdad

La igualdad es un predicado indispensable para muchas aplicaciones de la lógica de predicados. Esta sección presenta una extensión de $\mathsf{DS}(\mathcal{L})$ en la cual se incluye la igualdad como un predicado binario que permite comparar términos, el cual se denota como '='.

La relación de igualdad se incorpora en $\mathsf{DS}(\mathcal{L})$ por medio de dos axiomas definicionales.

Definición 7.27

Sea x una variable, t un término y ϕ una fórmula. Los siguientes axiomas de $\mathsf{DS}(\mathcal{L})$ definen la igualdad entre términos:

$(Bx6)$**:** $x = x$

$(Bx7)$**:** $(x = t) \to (\phi \equiv \phi[x := t])$, si t es libre para x en ϕ.

La igualdad es reflexiva para variables ($Bx6$) y permite la sustitución de una variable libre por un término en una fórmula siempre y cuando las variables de dicho término no sean capturadas en el proceso ($Bx7$).

En algunas ocasiones, cuando el rango de una cuantificación corresponde a una igualdad, es posible simplificar dicha fórmula.

Teorema 7.28

Sea x una varible, t un término y ϕ una fórmula. Si t es libre para x en ϕ y x no aparece en t:

1. $\vdash_{\mathsf{DS}(\mathcal{L})} (\forall x \mid x = t : \phi) \equiv \phi[x := t]$
2. $\vdash_{\mathsf{DS}(\mathcal{L})} (\exists x \mid x = t : \phi) \equiv \phi[x := t]$

Cada uno de los teoremas en 7.28 se llama *regla de un punto* porque indican cómo operar una cuantificación, universal o existencial, cuando exactamente un elemento del dominio del discurso satisface su rango.

Demostración. A continuación se presenta una demostración de (1); una demostración de (2) se propone como ejercicio para el lector. Se procede por doble implicación.

$$
\begin{array}{ll}
 & (\forall x \mid x = t : \phi) \\
\equiv & \langle \text{ azúcar sintáctico } \rangle \\
 & \forall x\, ((x = t) \to \phi) \\
\to & \langle\ (Bx4)\text{: } t \text{ es libre para } x \text{ en } \phi\ \rangle \\
 & ((x = t) \to \phi)[x := t] \\
\equiv & \langle \text{ definición de sustitución textual; } x \text{ no aparece en } t \rangle \\
 & (t = t) \to \phi[x := t] \\
\equiv & \langle\ (Bx6)\text{; identidad de } \to \rangle \\
 & \phi[x := t]\,.
\end{array}
$$

En el otro sentido suponga $\vdash_{\mathsf{DS}(\mathcal{L})} \phi[x := t]$, y note que por el axioma ($Bx7$) las fórmulas $((x = t) \to \phi)$ y $((x = t) \to \phi[x := t])$ son equivalentes. Entonces se tiene que $\vdash_{\mathsf{DS}(\mathcal{L})} (x = t) \to \phi$. Por la regla Generalización se obtiene que la fórmula $\forall x\, ((x = t) \to \phi)$ es teorema. Finalmente, esta fórmula puede ser escrita como $(\forall x \mid x = t : \phi)$. Entonces:

$$\vdash_{\mathsf{DS}(\mathcal{L})} \phi[x := t] \to (\forall x \mid x = t : \phi).$$

□

Para ilustrar el uso de la igualdad, se retoma el concepto de *divisibilidad* en $\mathbb{Z}$. Recuerde el predicado de divisibilidad '$\cdot|$' entre números enteros introducido en la Sección 5.5:

$$a \cdot| \, b : \text{“hay un } c_{ab} \in \mathbb{Z} \text{ tal que } b = ac_{ab}\text{.”}$$

Para especificar divisibilidad en $\mathcal{L}$, se supone un único tipo (i.e., números enteros) y por lo tanto no es necesario asociarle un nombre para identificarlo.

Ejemplo 7.5

Se presenta una definición del predicado de divisibilidad para cualquier par de números enteros a y b:

$$a \cdot| \, b \equiv \exists x \, (ax = b)$$

La relación de divisibilidad tiene muchas propiedades. Entre ellas que es reflexiva y transitiva. A continuación se presenta como ejemplo la demostración de que la relación de divisibilidad es reflexiva y se propone para el lector la demostración de la transitividad.

Ejemplo 7.6

Se presenta una demostración de la reflexividad de la relación de divisibilidad. Es decir, el objetivo es demostrar:

$$\vdash_{\mathsf{DS}(\mathcal{L})} \forall a \, (a \cdot| \, a).$$

Por el Metateorema 7.22 basta con demostrar:

$$\vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| \, a.$$

Considere la siguiente derivación:

$$
\begin{array}{ll}
 & a \cdot| \, a \\
\equiv & \langle \text{ definición } \rangle \\
 & \exists x \, (ax = a) \\
\leftarrow & \langle \text{ instanciación con testigo } x = 1 \text{ (Teorema 7.16) } \rangle \\
 & a1 = a \\
\equiv & \langle \text{ arimtética } \rangle \\
 & a = a \\
\equiv & \langle \, (Bx6) \, \rangle \\
 & \mathit{true}.
\end{array}
$$

Otra propiedad de la relación de divisibilidad es la siguiente: si a es divisor de b, entonces a también es divisor de cualquier múltiplo de b.

Ejemplo 7.7

Considere el siguiente objetivo:

$$\vdash_{\mathsf{DS}(\mathcal{L})} \forall a \, \forall b \, \forall c \, (a \cdot| \, b \rightarrow a \cdot| \, bc).$$

Basta con demostrar:

$$\vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| \, b \rightarrow a \cdot| \, bc.$$

Considere la siguiente derivación:

$$
\begin{array}{ll}
 & a \cdot| \, b \rightarrow a \cdot| \, bc \\
\equiv & \langle \text{ definición } \rangle \\
 & \exists x \, (ax = b) \rightarrow a \cdot| \, bc \\
\equiv & \langle \text{ Teorema 7.20.2 } \rangle \\
 & \forall x \, ((ax = b) \rightarrow a \cdot| \, bc).
\end{array}
$$

Por los metateoremas 7.22 y 7.23, basta con demostrar:

$$\{ax = b\} \vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| \, bc.$$

Considere la siguiente derivación:

$$
\begin{array}{ll}
& a \cdot| \, bc \\
\equiv & \langle \text{ definición } \rangle \\
& \exists y \, (ay = bc) \\
\equiv & \langle \text{ suposición; } ax \text{ es libre para } b \rangle \\
& \exists y \, (ay = axc) \\
\leftarrow & \langle \text{ instanciación con testigo } y = xc \rangle \\
& axc = axc \\
\equiv & \langle \, (Bx6) \, \rangle \\
& true.
\end{array}
$$

Note que deliberadamente, en la última derivación del Ejemplo 7.7, se escoge una variable y distanta a x en el primer paso de la derivación. Esto se debe a que esta variable es distinta a la variable x en la suposición.

Ejercicios

1. Demuestre que $=$ es reflexivo, i.e., $\vdash_{\mathsf{DS}(\mathcal{L})} t = t$ para cualquier término t.
2. Demuestre que $=$ es simétrico, i.e, $\vdash_{\mathsf{DS}(\mathcal{L})} t = u \to u = t$ para cualesquiera términos t y u.
3. Demuestre que $=$ es transitivo.
4. Sean x una variable, t, u términos y ϕ una fórmula. Demuestre que si t y u son libres para x en ϕ, entonces:

   $$\vdash_{\mathsf{DS}(\mathcal{L})} (t = u) \to (\phi[x := t] \equiv \phi[x := u]).$$

   Ayuda: proceda por inducción sobre la complejidad de ϕ.
5. Demuestre el Teorema 7.28.2.
6. Demuestre que la relación de divisibilidad es reflexiva.
7. Demuestre que $a \in \mathbb{Z}$ es divisor de cualquiera de sus múltiplos, i.e., $\vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| \, ab$ para cualquier $b \in \mathbb{Z}$.
8. Sean a y b números enteros. Demuestre que si $a \cdot| \, b$ y $b \cdot| \, a$, entonces $a = b$ o $a = -b$.
9. Sean a, b, c números enteros. Demuestre:

a) $\vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| b \wedge a \cdot| c \to a \cdot| b + c$.
b) $\vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| b \wedge a \cdot| c \to a \cdot| bc$.
c) $\vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| b \wedge a \cdot| c \to a^2 \cdot| bc$.

10. Sean a, b, c, d números enteros. Demuestre o refute:

$$\vdash_{\mathsf{DS}(\mathcal{L})} a \cdot| c \wedge b \cdot| d \to ab \cdot| cd.$$

Bibliografía

[1] J. Bohórquez. *Lógica y matemáticas discretas en la informática: el estilo calculatorio*. Escuela Colombiana de Ingeniería, 2012.

[2] E. Dijkstra and C. Scholten. *Predicate calculus and program semantics*. Texts and monographs in computer science. Springer-Verlag, 1990.

[3] D. Gries and F. B. Schneider. *A logical approach to discrete math*. Texts and Monographs in Computer Science. Springer, 1993.

[4] A. Hamilton. *Logic for Mathematicians*. Cambridge University Press, 1988.

[5] D. Hilbert and P. Bernays. *Grundlagen der mathematik*. Springer, 1st edition, 1968–1970.

[6] R. Hodel. *An introduction to mathematical logic*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 2013.

[7] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2004.

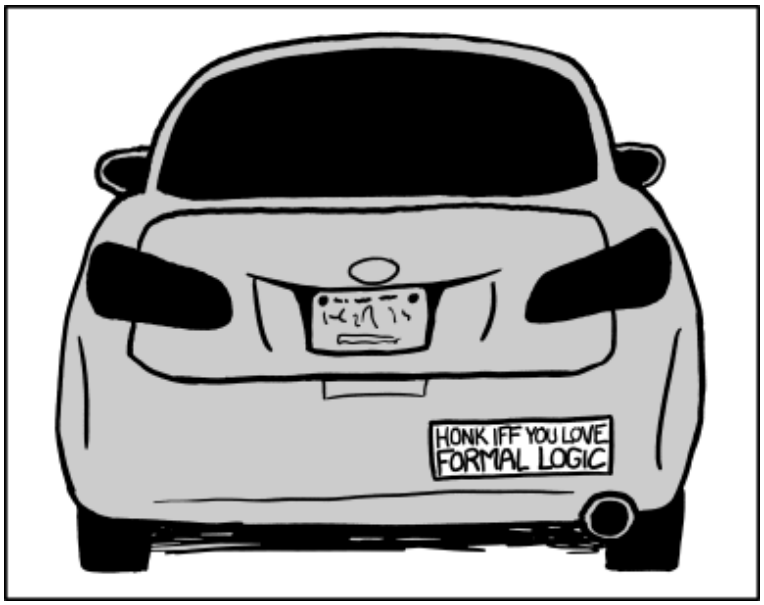
[8] C. Rocha and J. Meseguer. A rewriting decision procedure for Dijkstra-Scholten's syllogistic logic with complements. *Revista Colombiana de Computación*, 8(2), 2007.

[9] C. Rocha and J. Meseguer. Theorem proving modulo based on Boolean equational procedures. In R. Berghammer, B. Möller, and G. Struth, editors, *RelMiCS*, volume 4988 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2008.

[10] G. Tourlakis. *Mathematical logic*. John Wiley & Sons, 1st edition, 2008.

[11] A. N. Whitehead and B. Russell. *Principia mathematica*. Cambridge University Press, Cambridge, 2nd edition, 1910–1913.

Índice alfabético



XKCD: Formal Logic (https://xkcd.com/1033/)