Resumen Lógica e Inteligencia Artificial, y Verificación de Programas

Lógica de Enunciados o Proposicional

Definiciones Generales

Lógica: Disciplina que estudia las formas de *razonamiento*. Trata acerca de los medios a través de los cuales puede propagarse y articularse el *conocimiento*.

Razonamiento: Encadenamiento de enunciados formado por premisas y conclusiones. Se apoya en verdades supuestas, para obtener otras como resultado de la actividad de razonamiento.

- Las *premisas* del razonamiento son el conocimiento que tenemos de base y asumimos verdadero.
- La *conclusión* es el conocimiento nuevo generado a partir de las premisas y un mecanismo de razonamiento.

Lenguaje de la Lógica

La **lógica proposicional**, también conocida como **lógica de enunciados**, es un sistema formal cuyos elementos representan *proposiciones o enunciados*.

- No tiene mucha utilidad por si misma para la representación del conocimiento.

A nosotros nos interesa examinar los mecanismos de razonamiento con precisión **matemática**. Esta precisión requiere que *el lenguaje que usemos no dé lugar a confusiones*, lo cual conseguimos mediante un *lenguaje simbólico* donde cada símbolo tenga un significado bien definido.

Dada una frase en lenguaje natural, **en primer lugar**, podemos observar si se trata de una frase simple o de una frase compuesta.

En segundo lugar, vamos a suponer que **todas las frases simples** (enunciados o proposiciones) pueden ser verdaderas o falsas. Y distinguiremos entre **enunciados simples** (atómicos) o **enunciados compuestos**.

Denotamos los enunciados con letras mayúsculas **A, B, C, ...**

Para construir enunciados compuestos introducimos símbolos para las **conectivas:**

- $\neg A$ - Negación.
- $A \wedge B$ - Conjunción.
- $A \vee B$ - Disyunción.
- $A \rightarrow B$ - Si A entonces B.
- $A \leftrightarrow B$ - A si y sólo si B.

Lenguaje simbólico de la lógica

Para estudiar los **principios del razonamiento**, necesitamos:

- Capturar y formalizar las estructuras del lenguaje natural en un lenguaje simbólico,
- Para luego formalizar los mecanismos de razonamiento que se aplican sobre dichas estructuras lingüísticas.

El lenguaje tiene SINTAXIS y SEMÁNTICA

Elementos del lenguaje simbólico

El lenguaje simbólico consta de un conjunto de:

- Símbolos primitivos (el **alfabeto** o vocabulario) y
- Un conjunto de reglas de formación (la **gramática**) que nos dice cómo construir fórmulas bien formadas a partir de los símbolos primitivos.

Alfabeto

Conjunto de símbolos que pertenecen al lenguaje del sistema. **Consiste en:**

- Una cantidad finita pero arbitrariamente grande de *variables proposicionales* (o variables de enunciado). Las $p_1, p_2, \ldots, p_n$.
- Un conjunto de *operadores lógicos o conectivas*.
- Dos *signos de puntuación*: el paréntesis izquierdo y el paréntesis derecho.

Gramática

Conjunto de reglas que definen recursivamente las cadenas de caracteres que pertenecen al lenguaje. A las cadenas de caracteres construidas según estas reglas se las llama **fórmulas bien formadas (fbf)**, y también se las conoce como **formas enunciativas.**

Las reglas del Lenguaje son:

- Las *variables de enunciado* del alfabeto de L son formas enunciativas. Es decir, $p, q, \ldots$

- Si A y B son formas enunciativas de L, entonces también lo son $\neg A, A \wedge B, A \vee B, A \rightarrow B, A \leftrightarrow B$.
- Sólo las expresiones que pueden ser generadas mediante las cláusulas anteriores en un número finito de pasos son formas enunciativas de L.

Semántica

Como todo enunciado simple es verdadero o falso, una variable de enunciado tomará uno u otro valor de verdad: **V (verdadero) o F (falso)**.

La verdad o falsedad de un enunciado compuesto depende de la verdad o falsedad de los enunciados simples que lo constituyen, y de la forma en que están conectados.

Razonamiento y Deducción

El razonamiento puede ser:

- *Correcto o válido:* si la manera en que está construido garantiza la conservación de la verdad. Si las premisas son V, entonces la conclusión es necesariamente V.
  - **Deducción = razonamiento correcto**.
- *Incorrecto o inválido:* su construcción es defectuosa (no hay garantía acerca del valor de verdad de la conclusión), es decir, cuando:
  - A partir de premisas **verdaderas** permite arribar a una conclusión **falsa**,
  - O porque tiene la **estructura de un razonamiento incorrecto** (aunque la conclusión sea verdadera).

Argumentaciones

Una **forma argumentativa** es una sucesión finita de formas enunciativas, de las cuales la última se considera como la *conclusión* de las anteriores, conocidas como premisas. La notación es: $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{A}$

Una **forma argumentativa válida** es aquella a la que es posible asignar valores de verdad a las variables de enunciado que aparecen en ella, de tal manera que alguna $\mathcal{A}_1, \ldots, \mathcal{A}_n$ tome el valor F y $\mathcal{A}$ tome el valor F. Es decir, bajo cualquier asignación de valores de verdad a las variables de enunciado, si las premisas toman el valor V, la conclusión también debe tomar el valor V.

- Para referirnos a formas argumentativas válidas utilizamos la siguiente *notación*: $\Gamma \models A$.

Una **forma argumentativa inválida** es aquella a la que es posible asignar valores de verdad a las variables de enunciado que aparecen en ella, de tal manera que $\mathcal{A}_1, \ldots, \mathcal{A}_n$ tomen el valor V y $\mathcal{A}$ tome el valor F.

Definición 1.28

La forma argumentativa

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n; \therefore \mathscr{A}$$

es *inválida* si es posible asignar valores de verdad a las variables de enunciado que aparecen en ella, de tal manera que $\mathscr{A}_1, \ldots, \mathscr{A}_n$ tomen el valor V y $\mathscr{A}$ tome el valor F. En otro caso, la forma argumentativa es válida.

Página 32

Ejercicios relacionados

Para cada una de las siguientes argumentaciones, escribir una forma argumentativa que se corresponda con ella y determinar si es válida o inválida.

1. Si la función f no es continua, entonces la función g no es diferenciable. g es diferenciable. Por lo tanto, f no es continua.

Variables de enunciado

- p: "La función f es continua"
- q: "La función g es diferenciable"

Premisas

- $\mathcal{A}_1 \ : \ (\neg p) \rightarrow (\neg q)$
- $\mathcal{A}_2 \ : \ q$

La función f no es continua?

Tomamos como $\mathcal{A}$ a $(\neg p)$. Analizamos: $\mathcal{A}_1, \ \mathcal{A}_2 \ \therefore \ \mathcal{A}$.

$(\neg p) \rightarrow (\neg q)$	q	$(\neg p)$
$(\neg p)$ - F $(\neg q)$ - F $(\neg p) \rightarrow (\neg q)$ - V	q - V	F

La forma argumentativa es inválida ya que es posible asignar valores de verdad a las variables de enunciado que aparecen en ella, de tal manera que $\mathcal{A}_1, \ \mathcal{A}_2$ tomen el valor V y $\mathcal{A}$ tome el valor F. Por lo tanto, no se presentan pruebas suficientes para garantizar que la función f no es continua.

2. Sí Juan ha instalado la calefacción central, entonces ha vendido su coche o ha pedido dinero prestado al banco. Juan no ha vendido su coche ni ha pedido dinero prestado al banco. Por lo tanto, Juan no ha instalado la calefacción central.

Variables de enunciado

- p: "Juan ha instalado la calefacción central"
- q: "Juan ha vendido su coche"
- r: "Juan ha pedido dinero prestado al banco"

Premisas

- $\mathcal{A}_1 \ : \ p \rightarrow (q \vee r)$
- $\mathcal{A}_2 \ : \ (\neg q) \wedge (\neg r)$

Juan no ha instalado la calefacción central?

Tomamos como $\mathcal{A}$ a $(\neg p)$. Analizamos: $\mathcal{A}_1, \ \mathcal{A}_2 \ \therefore \ \mathcal{A}$.

$p \rightarrow (q \vee r)$	$(\neg q) \wedge (\neg r)$	$(\neg p)$
p - V q - F r - V $(q \vee r)$ - V $p \rightarrow (q \vee r)$ - V	$(\neg q)$ - V $(\neg r)$ - F $(\neg q) \wedge (\neg r)$ - F	F

La forma argumentativa es válida ya que es posible asignar valores de verdad a las variables de enunciado que aparecen en ella, de tal manera que alguna $\mathcal{A}_1, \ \mathcal{A}_2$ tome el valor F y $\mathcal{A}$ tome el valor F. Por lo tanto, se presentan pruebas suficientes para garantizar que Juan no ha instalado la calefacción central.

Interpretación y Satisfacción

Una interpretación es una función que relaciona los elementos de los dominios sintáctico y semántico de la lógica considerada.

En el caso particular de la lógica proposicional, una interpretación I consiste en una *función de valuación* v que asigna a cada *variable de enunciado* el valor de verdad V o F.

Se define una regla semántica para cada una de las reglas semánticas de la gramática en pos de extender el dominio de la función de valuación

- $\models_v p$ si y sólo si $v(p) = V$.

- $\models_v (\neg A)$ si y sólo si no es el caso que $\models_v A$.
- $\models_v (A \vee B)$ si y sólo si o bien $\models_v A$ o bien $\models_v B$ o ambos.
- $\models_v (A \wedge B)$ si y sólo si $\models_v A$ y $\models_v B$.
- $\models_v (A \to B)$ si y sólo si no es el caso que $\models_v A$ y no $\models_v B$.
- $\models_v (A \leftrightarrow B)$ si y sólo si $\models (A \to B)$ y $\models (B \to A)$.

Tautología y Contradicción

Definición 1.5

(*a*) Una forma enunciativa es una *tautología* si toma el valor de verdad *V* bajo cada una de las posibles asignaciones de valores de verdad a las variables de enunciado que aparecen en ella.

(*b*) Una forma enunciativa es una *contradicción* si toma el valor de verdad *F* bajo cada una de las posibles asignaciones de valores de verdad a las variables de enunciado que aparecen en ella.

▷No toda forma enunciativa cae en una u otra de estas categorías. De hecho, ninguna de las consideradas hasta aquí cae en ellas.

El método que se utiliza para comprobar si una forma enunciativa dada es una tautología o una contradicción consiste en construir la tabla de verdad.

Página 17 ambas

Equivalencias Lógicas e Implicaciones Lógicas

Con n variables de enunciado construimos:

- *Tablas de verdad* de tamaño 2^n.
- 2^{2^n} *funciones de verdad distintas de* n *argumentos* (en **criollo**, 2^{2^n} formas posibles de poner los V y los F en la la última columna de una tabla de verdad de 2^n filas).
- *Infinitas formas enunciativas* (muchas estarán asociadas a la misma función de verdad).

Definiciones

Definición 1.7

Si $\mathscr{A}$ y $\mathscr{B}$ son formas enunciativas, diremos que $\mathscr{A}$ *implica lógicamente* a $\mathscr{B}$ si $(\mathscr{A} \rightarrow \mathscr{B})$ es una tautología, y que $\mathscr{A}$ es *lógicamente equivalente* a $\mathscr{B}$ si $(\mathscr{A} \leftrightarrow \mathscr{B})$ es una tautología.

Observación: Sean $\mathscr{A}$ y $\mathscr{B}$ formas enunciativas que contengan las mismas variables de enunciado. Si $\mathscr{A}$ y $\mathscr{B}$ son lógicamente equivalentes, entonces representan una misma función de verdad. Puesto que si $(\mathscr{A} \leftrightarrow \mathscr{B})$ es una tautología, no toma nunca el valor F, de modo que $\mathscr{A}$ y $\mathscr{B}$ han de tomar siempre el mismo valor de verdad. Así pues, las funciones de verdad correspondientes a $\mathscr{A}$ y $\mathscr{B}$ han de ser forzosamente iguales.

Página 18 ambas

Corolario. Toda forma enunciativa, que no es una contradicción, es lógicamente equivalente a una forma enunciativa restringida de la forma:

$$\bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} Q_{ij}$$

Es cierto que se pueden escribir **dos fbfs que tengan diferentes letras de proposición y aún así sean lógicamente equivalentes:**

- Si se toman dos fórmulas fbfs que son tautologías, siempre serán equivalentes sin importar su estructura o las variables que utilicen ya que al ser tautologías su resultado final siempre es Verdadero.

Equivalencias lógicas que podemos usar de chill

Ley de Doble Negación	$\neg (\neg p) \Leftrightarrow p$
Ley Conmutativa de la Conjunción	$p \wedge q \Leftrightarrow q \wedge p$
Ley Conmutativa de la Disyunción	$p \vee q \Leftrightarrow q \vee p$
Ley Asociativa de la Conjunción	$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
Ley Asociativa de la Disyunción	$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
Leyes de De Morgan	$\neg (p \vee q) \Leftrightarrow (\neg p) \wedge (\neg q)$
	$\neg (p \wedge q) \Leftrightarrow (\neg p) \vee (\neg q)$
Leyes de Distribución	$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
	$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
Leyes de Absorción	$p \wedge p \Leftrightarrow p$
	$p \vee p \Leftrightarrow p$

Ejercicios relacionados

1. ¿"$(p \to q)$" es lógicamente equivalente a "$(p \vee \neg q)$" ?

No son lógicamente equivalentes.

p	q	$\neg q$	$(p \to q)$	$(p \vee \neg q)$	$(p \to q) \leftrightarrow (p \vee \neg q)$
V	V	F	V	V	V
V	F	V	F	V	F
F	V	F	V	F	F
F	F	V	V	V	V

3. ¿"$(\neg(p \wedge q))$" es lógicamente equivalente a "$(\neg p \vee \neg q)$" ?

Son lógicamente equivalentes.

p	q	$\neg p$	$\neg q$	$(p \wedge q)$	$(\neg(p \wedge q))$	$(\neg p \vee \neg q)$	$(\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$
V	V	F	F	V	F	F	V
V	F	F	V	F	V	V	V
F	V	V	F	F	V	V	V
F	F	V	V	F	V	V	V

Determinar cuáles de las siguientes fbfs son lógicamente implicadas por la fbf $(A \wedge B)$. Fundamentar.

1. A

Es lógicamente implicada por $(A \wedge B)$.

$v(A)$	$v(B)$	$v((A \wedge B))$	$v((A \wedge B) \to A)$
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	V

Conjuntos adecuados de conectivas

Definición 1.23

Un conjunto *adecuado* de conectivas es un conjunto tal que toda función de verdad puede representarse por medio de una forma enunciativa que contenga solamente conectivas del conjunto.

Los pares $\{\sim, \wedge\}$, $\{\sim, \vee\}$ y $\{\sim, \rightarrow\}$ son conjuntos adecuados de conectivas.

Página 28 ambas

Tener en cuenta que todos los conjuntos adecuados tienen a la negación, si nos plantean alguno sin la negación como adecuado that shit is fake as fuck!

Equivalencias en las conectivas

- $(A \vee B) = \neg(\neg A \wedge \neg B)$
- $(A \wedge B) = \neg(\neg A \vee \neg B)$ o $\neg(A \rightarrow \neg B)$
- $(A \rightarrow B) = (\neg A \vee B)$ o $\neg(A \wedge \neg B)$

Técnicas de Prueba

Demostración por contra-ejemplo

Planteamos un contra-ejemplo específico para demostrar que no se cumple lo que nos plantean.

Ejemplo

¿Es cierto que si una fbf A es satisfactible entonces A es una tautología?

Esta propiedad no vale, lo vamos a demostrar construyendo un contraejemplo. Si tomamos $\mathcal{A} = p \wedge q$ podemos ver:

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

$\mathcal{A}$ es una fbf satisfactible ya que existe al menos usa asignación de valores de verdad que la vuelve verdadera pero no es una tautología.

Demostración por el absurdo

Asumimos lo contrario a lo que nos piden, a partir de eso realizamos la demostración usando el conocimiento que nos brinde el enunciado y el que sepamos. Si llegamos a una contradicción, entonces lo que nos piden se cumple.

Ejemplo

Demostrar utilizando la técnica del absurdo que $(p \land \neg p) \to q$ es una tautología.

Asumimos que $(p \land \neg p) \to q$ no es una tautología.

Demostración:

- Por construcción de tablas de verdad podemos demostrar que la valoración $v((p \land \neg p)) = F$ ya que $(p \land \neg p)$ es una *contradicción*.
- Entonces, por definición de valoración *"La valoración $v((\mathcal{A} \to \mathcal{B}))$ será Verdadera si y solo si no se cumple que la $v(\mathcal{A}) = V$ y la $v(\mathcal{B}) = F$"* y por saber que $v((p \land \neg p)) = F$, no nos importa que valor tome la valoración $v(q)$ ya que siempre la valoración $v((p \land \neg p)) = F$, por lo tanto, $(p \land \neg p) \to q$ es una tautología.

Hemos llegado a una contradicción y, por lo tanto, demostrado que la propiedad se cumple.

Demostración por inducción 💀

1. Tenemos que tomar el caso base para lo que nos piden y demostrar que lo que nos piden se cumple para esa caso base.
2. Planteamos la Hipótesis Inductiva (creemos en el corazón de las cartas como Yugi y asumimos que para un n se cumple lo que nos piden).
3. Usamos la Hipótesis para demostrar el caso $n + 1$.

Ejemplo

Demostrar utilizando la técnica de inducción que cualquier fórmula bien formada A que contenga sólo los conectivos $\{\lor, \land\}$ puede tomar el valor F.

Tomamos como Caso base $n = 0$ donde n representa la cantidad de conectivas de la fórmula, en este caso, $\mathcal{A}$ no tiene conectivas, es decir, solo tiene una variable de enunciado p, por ejemplo.

- $\mathcal{A} = p$
- Existe la valoración $v(p) = F$, por lo tanto, $v(\mathcal{A}) = F$

Se cumple el Caso base

Para nuestra **Hipótesis inductiva** asumimos que para toda fbf $\mathcal{A}$ (donde aparecen solo las conectivas $\vee, \wedge$) de n o menos conectivas se cumple que $v(\mathcal{A}) = F$.

Veamos nuestro **Paso de inducción** donde $\mathcal{A}$ posee $n+1$ conectivas. Debido a las 2 formas de construir formas enunciativas según nuestro conjunto de conectivas vamos a considerar los 2 casos posibles:

- *$\mathcal{A}$ es de la forma $(\mathcal{B} \vee \mathcal{C})$.*
  - Cada fórmula $\mathcal{B}$ y $\mathcal{C}$ tienen n conectivas, por lo tanto, por **Hipótesis inductiva** se cumple que $v(\mathcal{B}) = F$ y que $v(\mathcal{C}) = F$.
  - Por definición de valoración *"La valoración $v((\mathcal{B} \vee \mathcal{C}))$ será Verdadera si y solo si no se cumple que la $v(\mathcal{B}) = F$ y la $v(\mathcal{C}) = F$"* vemos entonces que se cumple $v((\mathcal{B} \vee \mathcal{C})) = F$.
- *$\mathcal{A}$ es de la forma $(\mathcal{B} \wedge \mathcal{C})$*
  - Cada fórmula $\mathcal{B}$ y $\mathcal{C}$ tienen n conectivas, por lo tanto, por **Hipótesis inductiva** se cumple que $v(\mathcal{B}) = F$ y que $v(\mathcal{C}) = F$.
  - Por definición de valoración *"La valoración $v((\mathcal{B} \wedge \mathcal{C}))$ será Verdadera si y solo si no se cumple que la $v(\mathcal{B}) = F$ y la $v(\mathcal{C}) = F$"* vemos entonces que se cumple $v((\mathcal{B} \wedge \mathcal{C})) = F$.

Con este análisis hemos demostrado que $v(\mathcal{A}) = F$, para cualquier fbf (donde aparecen solo las conectivas $\vee, \wedge$)

Cálculo de Enunciados L

Mecanismos formales de razonamiento

Un **mecanismo formal de razonamiento** consiste en una colección de reglas que pueden ser aplicadas sobre cierta información inicial para derivar información adicional, en una forma puramente sintáctica.

Sistema formal L

Sistema deductivo axiomático compuesto por:

- Un conjunto de *axiomas* (en realidad esquemas de axiomas)
- Un conjunto de *reglas de inferencia*.

Axiomas

Fórmulas bien formadas que se toman como punto de partida para las demostraciones. Tienen infinitas "realizaciones" según varían las fórmulas que los componen:

(L1) $(\mathscr{A} \to (\mathscr{B} \to \mathscr{A}))$.

(L2) $((\mathscr{A} \to (\mathscr{B} \to \mathscr{C})) \to ((\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C})))$.

(L3) $(((\sim \mathscr{A}) \to (\sim \mathscr{B})) \to (\mathscr{B} \to \mathscr{A}))$.

Página 38

Reglas de Inferencia

Una **regla de inferencia** es una función que asigna una fórmula (*conclusión*) a un conjunto de fórmulas (*premisas*). Determinan qué fórmulas pueden inferirse a partir de qué fórmulas.

4. Reglas de deducción. En L hay solamente una regla de deducción, la regla *modus ponens* (abreviadamente, MP), que afirma: De $\mathscr{A}$ y $(\mathscr{A} \to \mathscr{B})$ se deduce como consecuencia directa $\mathscr{B}$, siendo $\mathscr{A}$ y $\mathscr{B}$ *fbfs* cualesquiera de L.

Página 38

Esta regla siempre **preserva la verdad**.

Demostración o Derivación en L

Una *demostración en* L es una sucesión finita de *fbfs*. $\mathscr{A}_1, \ldots, \mathscr{A}_n$ tal que para todo i $(1 \leqslant i \leqslant n)$, o $\mathscr{A}_i$ es un axioma de L o $\mathscr{A}_i$ se deduce de dos miembros anteriores de la sucesión, digamos $\mathscr{A}_j$ y $\mathscr{A}_k$ $(j < i,\ k < i)$ como consecuencia directa, aplicando la regla de deducción MP. Una tal demostración diremos que es una *demostración de* $\mathscr{A}_n$ *en* L, y también que $\mathscr{A}_n$ *es un teorema de* L.

Página 39

Notación: $\Gamma \vdash_L A_n$ → "A partir de las premisas de Gamma se deduce A_n"

- Tanto premisas como conclusión son fbf.

Definición de Deducción

Definición 2.5

Sea Γ un conjunto de *fbfs* de L (que pueden o no ser axiomas o teoremas de L). Una sucesión finita $\mathscr{A}_1, \ldots, \mathscr{A}_n$ de *fbfs* de L es una *deducción a partir de* Γ si para todo i $(1 \leqslant i \leqslant n)$ se verifica alguna de las condiciones siguientes:

(a) $\mathscr{A}_i$ es un axioma de L,
(b) $\mathscr{A}_i$ es miembro de Γ,
(c) $\mathscr{A}_i$ se deduce directamente de dos miembros anteriores de la sucesión mediante MP.

Página 40

Ejercicios relacionados

Dar una demostración sintáctica en L de las siguientes deducciones.

3. $\{((A \to B) \to C),\ B\} \vdash_L (A \to C)$

- $((B \to (A \to B))$ - *Instancia de Axioma $L1$*
- B - *Hipótesis*
- $(A \to B)$ - *Modus ponens entre 1 y 2*
- $((A \to B) \to C)$ - *Hipótesis*
- C - *Modus ponens entre 3 y 4*
- $((C \to (A \to C))$ - *Instancia de Axioma $L1$*
- $(A \to C)$ - *Modus ponens entre 5 y 6*

Teorema en L

Cuando una fbf se puede demostrar a partir del conjunto vacío de hipótesis se dice que es Teorema en L.

Teoremas que podemos usar para la práctica:

Proposición 2.11

Dadas *fbfs* cualesquiera $\mathscr{A}$ y $\mathscr{B}$ de L, los dos siguientes son teoremas de L.

(a) $(\sim\mathscr{B} \to (\mathscr{B} \to \mathscr{A}))$
(b) $((\sim\mathscr{A} \to \mathscr{A}) \to \mathscr{A})$.

Página 45

Metateorema de la Deducción

Proposición 2.8 (El Teorema de Deducción)
Si $\Gamma \cup \{\mathscr{A}\} \vdash_L \mathscr{B}$ entonces $\Gamma \vdash_L (\mathscr{A} \to \mathscr{B})$, siendo $\mathscr{A}$ y $\mathscr{B}$ *fbfs* de L y Γ un conjunto de *fbfs* de L (eventualmente vacío).

Página 43

Ejercicio Relacionados

Dar una demostración sintáctica en L de las siguientes deducciones.

3. $\{((A \to B) \to C),\ B\} \vdash_L (A \to C)$

Se puede hacer más sencillo demostrarlo aplicando el Metateorema de la deducción demostrando $\{((A \to B) \to C),\ B\} \cup \{A\} \vdash_L (C)$

- $((B \to (A \to B))$ - *Instancia de Axioma* $L1$
- B - *Hipótesis*
- $(A \to B)$ - *Modus ponens entre 1 y 2*
- $((A \to B) \to C)$ - *Hipótesis*
- C - *Modus ponens entre 3 y 4*

Al lograr esta demostración por el Metateorema de la deducción se cumple la demostración $\{((A \to B) \to C),\ B\} \vdash_L (A \to C)$.

Regla del Silogismo Hipotético

Esta regla nos dice que si tenemos $(A \to B)$ y $(B \to C)$, entonces podemos deducir $(A \to C)$.

Corrección (Sensatez), Completitud, Consistencia y Decidibilidad de L

Corrección

El sistema formal L es correcto (produce conclusiones correctas).

- Se cumple que si A es *Teorema* en L, entonces A es una *tautología*.

La corrección de un sistema deductivo está garantizada si se cuenta con axiomas verdaderos y reglas de inferencia correctas (o sensatas) que preservan la verdad.

Completitud

Recíprocamente, el sistema formal L es completo (es capaz de demostrar todas y cada una de las conclusiones).

- Se cumple que si A es una *tautología*, entonces A es un *Teorema* en L.

Consistencia

El sistema formal L es consistente, no se puede derivar una contradicción dentro de él.

Decidibilidad

El sistema formal L es decidible, podemos construir un algoritmo que decida si una fbf es una tautología construyendo su tabla de verdad y verificando que todos sus valores finales sean Verdaderos, de esta forma, si es una tautología entonces es un teorema en L.

Lógica de Predicados de Primer Orden

La Lógica de Predicados resuelve la problemática de la lógica proposicional mediante el uso de Cuantificadores y, Sujetos y Predicados.

La lógica de predicados limitada a representar relaciones entre objetos se denomina de primer orden (la nuestra), la que permite expresar relaciones entre relaciones se conoce como de segundo orden, y así sucesivamente.

Predicados

Es lo que se afirma de un sujeto en una proposición. Pueden definir propiedades sobre uno o más sujetos, a su vez, establecen relaciones entre ellos. Pueden ser unarios, binarios, etc.

Notación: $P_i^j(x)$

- i es el identificador del predicado.
- j nos dice si el predicado es unario, binario, etc.

Cuantificadores

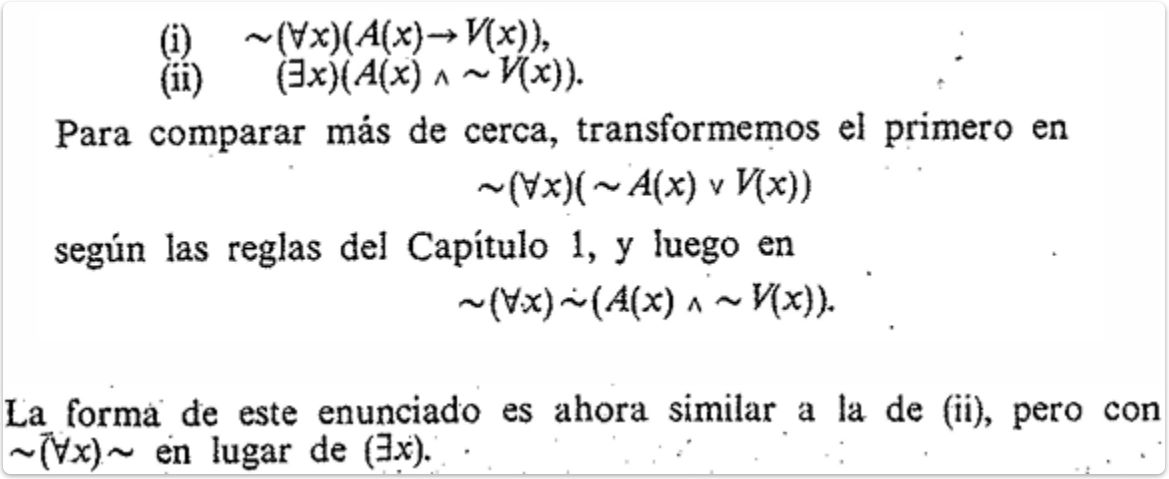
- La frase "Para todo x" se llama cuantificador universal y se simboliza como $\forall x$.
  - Siempre va seguido de una *implicación*.
- La frase "Existe al menos un objeto x tal que" se llama cuantificador existencial y se simboliza como $\exists x$.
  - Siempre va seguido de una *conjunción*.

Los cuantificadores tienen un alcance o scoupe limitado, se pegan directamente a lo que tengan a la derecha básicamente, tener en cuenta el uso de *paréntesis* en las fórmulas para ir determinando el alcance de los cuantificadores.

- Una fbf se considera abierta cuando contiene al menos una variable libre (como Willy la orca), es decir, no la alcanza ningún cuantificador, caso contrario es cerrada.

Relación entre ellos

Para ver la relación podemos pensar en las frases *"No todas las aves vuelan"* y *"Algunas aves no vuelan"*. A partir de la representación de estas frases podemos ver como se relacionan los cuantificadores de esta forma:



(i) $\sim(\forall x)(A(x)\rightarrow V(x))$,
(ii) $(\exists x)(A(x)\wedge\sim V(x))$.

Para comparar más de cerca, transformemos el primero en

$$\sim(\forall x)(\sim A(x)\vee V(x))$$

según las reglas del Capítulo 1, y luego en

$$\sim(\forall x)\sim(A(x)\wedge\sim V(x)).$$

La forma de este enunciado es ahora similar a la de (ii), pero con $\sim(\forall x)\sim$ en lugar de $(\exists x)$.

Páginas 59 y 60

Sintaxis - Lenguaje Simbólico de la Lógica

Para estudiar los principios del razonamiento, la lógica necesita capturar y formalizar las estructuras del lenguaje natural en un lenguaje simbólico, para luego formalizar los mecanismos que se aplican sobre dichas estructuras lingüísticas.

Consta de un Alfabeto y la Gramática.

Alfabeto

Conjunto de símbolos primitivos que pertenecen al lenguaje:

- Un conjunto de símbolos de constantes $C = \{c_1, c_2, \dots\}$.
- Un conjunto de símbolos de variables $X = \{x_1, x_2, \dots\}$.
- Un conjunto de símbolos de funciones $F = \{f_1^1, f_2^1, \dots, f_1^2, f_2^2, \dots\}$.
- Un conjunto de símbolos de predicados $P = \{P_1^1, P_2^1, \dots, P_1^2, P_2^2, \dots\}$.
- Símbolos de conectivas (los mismos de la lógica proposicional).
- Paréntesis de apertura y cierre.
- El cuantificador universal $\forall$ (“para todo”) y el cuantificador existencial $\exists$ (“existe”).

Gramática

Conjunto de reglas que definen recursivamente las cadenas de símbolos que pertenecen al lenguaje. Define dos clases de elementos:

- *Términos:* Expresiones que denotan objetos del dominio.
  - Constantes y variables.
  - Todos los $f_1^n(t_1, \ldots, t_n)$
  - Toda expresión formada a partir de los dos puntos anteriores.
- *Fórmulas bien formadas:* Expresiones que denotan relaciones entre objetos.
  - Fórmulas del tipo $P_1^n(t_1, \ldots, t_n)$.
  - Fórmulas del tipo $(\neg A)$, $(A \vee B)$, $(A \wedge B)$, etc.
  - Fórmulas del tipo $(\forall x)A$, o bien, $(\exists x)A$, etc.
  - Toda expresión formada a partir de los tres puntos anteriores.

Ejercicios Relacionados

Expresar en un lenguaje de predicados de primer orden el conocimiento asociado a las siguientes situaciones:

3. Ningún modelo de IA que se entrena con datos erróneos es preciso.

Definiciones

- $x_1, x_2, \ldots$ son variables.
- $P_3^1(x)$: "x es un modelo de IA".
- $P_1^1(x)$: "x se entrena con datos erróneos".
- $P_2^1(x)$: "x preciso".

Podemos representar con los símbolos definidos:

- $(\forall x_1)((P_3^1(x_1) \wedge (P_1^1(x_1)) \rightarrow (\neg P_2^1(x_1))$.

7. Algunos modelos de inteligencia artificial entrenados por alumnos de FTC lograron superar el umbral de precisión del 90%.

Definiciones

- $x_1, x_2, \ldots$ son variables.
- $P_3^1(x)$: "x es un modelo de Inteligencia Artificial".
- $P_1^1(x)$: "x es entrenado por estudiantes de FTC".
- $P_2^1(x)$: "x logró superar el umbral de precisión del 90%".

Podemos representar con los símbolos definidos:

- $(\exists x_1)(P_3^1(x_1) \land P_1^1(x_1) \land P_2^1(x_1))$.

Semántica - Dominio de Interpretación

Dominio

Está constituido por:

- *Universo (U):* Conjunto no vació de objetos.
- *Conjunto de funciones (F):* Conjunto finito de funciones que toman cierta cantidad de objetos de U y retornan otro que también está dentro de U.
- *Conjunto de Relaciones (P):* Conjunto finito de relaciones entre los objetos de U.

Interpretación

Dados un *lenguaje* y un *dominio*, una **interpretación** I es una función que hace corresponder a los elementos del *lenguaje* con los elementos del *dominio*, satisfaciendo las siguientes condiciones:

- Si c_i es un símbolo de constante, entonces $I(c_i) \in U$.
- Si f_i^n es un símbolo de función de grado n, entonces $I(f_i^n) = U^n \to U$.
- Si P_i^n es un símbolo de predicado de grado n, entonces $I(P_i^n) \subseteq U^n$.

Una interpretación formaliza la noción de que los símbolos representan las abstracciones de la realidad modelada en el dominio correspondiente.

Tener en cuenta que los **símbolos** son sólo eso, **no tienen significado por si mismos**, el **significado** lo obtienen al ser **interpretados** bajo un **Dominio Semántico**, esto nos hace ver que **un mismo símbolo** puede interpretarse de **formas distintas** dependiendo el **Dominio**.

Valoración

Una **valoración** v es una **función** de una **interpretación** que indica cómo se **valoran** los **símbolos de variables**, dado que se define lo siguiente:

- Si c_i es un símbolo de constante, $v(c_i) = I(c_i)$.
- Si f_i^n es un símbolo de función, $v(f_i^n(t_1, \ldots, t_n)) = I(f_i^n)(v(t_1), \ldots, v(t_n))$.

Valoraciones I-equivalentes

En criollo, dos o más valoraciones son i-equivalentes si son **iguales** en todas las variables **excepto** posiblemente en la variable x_i. Ejemplo para entender:

Valoración/Variables	x_1	x_2	x_3
v_1	1	2	2
v_2	3	2	2
v_3	2	2	2

Esas son 1-equivalentes ¿por qué 1? porque coinciden en todas menos en la valoración para x_1, si nos pidieran 2-equivalentes sería x_1 y x_3 iguales pero se difiere en x_2

Satisfacción

$\models_{I,v} (\neg A)$	si y sólo si no es el caso que $\models_{I,v} A$
$\models_{I,v} (A \wedge B)$	si y sólo si $\models_{I,v} A$ y $\models_{I,v} B$
$\models_{I,v} (A \rightarrow B)$	si y sólo si no es el caso que $\models_{I,v} A$ y no $\models_{I,v} B$
$\models_{I,v} (\forall xi)\ A$	si y sólo si para toda valoración w, i-equivalente, se cumple $\models_{I,w} A$
	Es decir que A es verdadera cualquiera sea la valoración de xi.
$\models_{I,v} (\exists x)\ A$	si y sólo si para alguna valoración w, i-equivalente, se cumple $\models_{I,w} A$
	En este caso alcanza con que A sea verdadera en alguna valoración particular de x.

Verdad y Validez

Satisfactible

Una fórmula es Satisfactible cuando existe una Interpretación I y una valoración v, donde $\models_{I,v} A$. (Ver casos de arriba).

Verdadera en alguna Interpretación

Una fórmula es Verdadera en alguna Interpretación si en esa Interpretación I se satisface en todas las valoraciones.

Falsa en alguna Interpretación

Una fórmula es Falsa en alguna Interpretación si en esa Interpretación I no existe valoración que la satisfaga.

Tener en cuenta que existen fórmulas que no son ni Verdaderas ni Falsas.

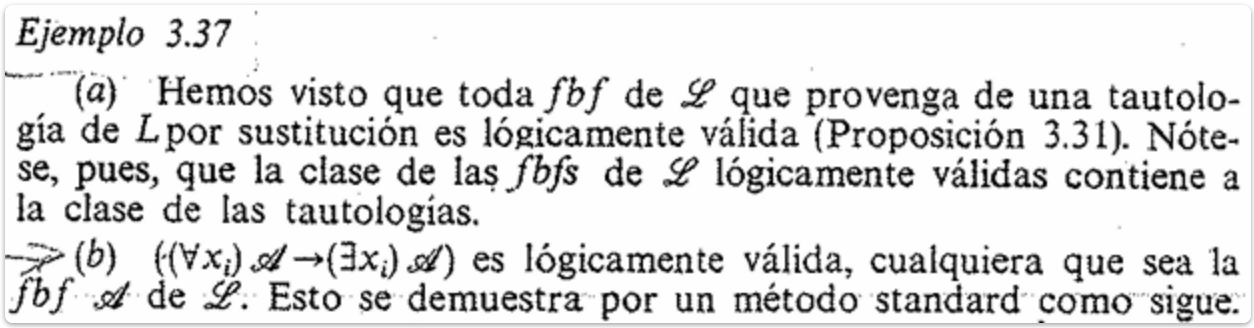
Lógicamente Válida

Una fórmula es Lógicamente Válida si es Verdadera en toda interpretación. Es lo análogo a una tautología de la lógica proposicional.

El conjunto de fórmulas lógicamente válidas incluye al conjunto de tautologías del sistema formal L.

No todas las fórmulas lógicamente válidas tienen estructura de tautologías del sistema formal L, pero toda tautología del sistema formal L es una fórmula lógicamente válida.

Tener en cuenta esto



Ejemplo 3.37

(*a*) Hemos visto que toda *fbf* de $\mathscr{L}$ que provenga de una tautología de L por sustitución es lógicamente válida (Proposición 3.31). Nótese, pues, que la clase de las *fbfs* de $\mathscr{L}$ lógicamente válidas contiene a la clase de las tautologías.

(*b*) $(\forall x_i)\mathscr{A} \rightarrow (\exists x_i)\mathscr{A}$ es lógicamente válida, cualquiera que sea la *fbf* $\mathscr{A}$ de $\mathscr{L}$. Esto se demuestra por un método standard como sigue.

Página 81

Contradicción

Una fórmula es una Contradicción si es Falsa en toda interpretación. Es lo análogo a una contradicción de la lógica proposicional.

El conjunto de fórmulas contradictorias incluye al conjunto de contradicciones del sistema formal L.

Lógicamente Equivalentes (Concepto aplicado solo en la Práctica)

Dos fórmulas son lógicamente equivalentes si tienen el mismo valor de verdad en todas las interpretaciones posibles.

- Para demostrar que *no* son equivalentes, basta encontrar al menos una interpretación donde difieran en valor de verdad.

El Tip que puede servir para descubrir a las que NO son lógicamente equivalentes es ver la conmutación de los Cuantificadores:

- *Cuantificadores del mismo tipo* pueden conmutar entre ellos sin cambiar el significado de la fbf.
- Con *Cuantificadores de distinto tipo* si importa el orden para el significado de la fbf (no es lo mismo $(\forall x)(\exists y)$ que $(\exists y)(\forall x)$ la segunda es más fuerte).
- El *Existe* es *distributivo* en la *disyunción*.
- El *Para todo* es *distributivo* en la *conjunción*.

Cuando las fórmulas son Lógicamente válidas, la justificación que tenemos que dar es la que se vio al principio, tenemos que demostrar que una implica a la otra y viceversa.

Inteligencia Artificial 🤖

Definiciones

La inteligencia se puede definir como la capacidad de un individuo para aprender, razonar, resolver problemas, comprender ideas complejas, adaptarse a nuevas situaciones y utilizar el conocimiento para manipular su entorno.

La inteligencia artificial es un campo de la informática que se centra en la creación de sistemas y programas capaces de realizar tareas que normalmente requieren inteligencia humana. Esto incluye habilidades como el aprendizaje, el razonamiento, la percepción y la toma de decisiones.

IA Débil vs IA Fuerte

IA Débil	IA Fuerte
Sistemas de inteligencia artificial diseñados y entrenados para realizar una tarea específica y limitada. Estos sistemas operan dentro de un conjunto predefinido de reglas y datos, y su "inteligencia" se manifiesta únicamente en la ejecución eficiente de esa tarea particular.	Tipo de inteligencia artificial que posee la capacidad de entender, aprender y aplicar inteligencia a cualquier tarea intelectual que un ser humano pueda realizar. Implica una comprensión genuina, conciencia y la capacidad de razonar, planificar, resolver problemas y aprender de la experiencia en una amplia gama de dominios.

Sistema de Software Complejo vs Sistema de IA

Sistema de Software Complejo	Sistema de IA
Es determinista, se basa en reglas explícitas, busca la optimización y precisión, tiene una entrada y salida definidas, no aprende ni se adapta, entre otras cosas.	Es probabilístico, se basa en el aprendizaje, busca la adaptabilidad y generalización, maneja incertidumbre y ambigüedad, tiene un aprendizaje continuo, entre otras cosas.

IA Simbólica vs IA No-Simbólica

IA Simbólica	IA No-Simbólica
Se basa en la idea de que la inteligencia puede ser modelada mediante la **manipulación de símbolos y reglas lógicas**. Se busca representar el conocimiento humano de forma explícita y estructurada, utilizando lenguajes formales y sistemas de lógica.	Se basa en la idea de que la inteligencia emerge de **patrones y conexiones aprendidas a partir de grandes volúmenes de datos**, sin la necesidad de una representación explícita y simbólica del conocimiento.

Machine Learning

Rama de la Inteligencia Artificial que se enfoca en el desarrollo de algoritmos y modelos que permiten a los sistemas **aprender de los datos sin ser programados explícitamente** para cada tarea. En lugar de seguir instrucciones paso a paso para resolver un problema, los sistemas de ML identifican patrones y relaciones en grandes conjuntos de datos, y utilizan esos patrones para hacer predicciones, clasificaciones o tomar decisiones sobre datos nuevos y no vistos.

Red Neuronal artificial (ANN)

Una **Red neuronal artificial (ANN)** es un modelo computacional inspirado en la estructura y el funcionamiento del cerebro humano. Está diseñada para reconocer patrones, clasificar datos y hacer predicciones, aprendiendo de grandes volúmenes de información.

Funcionamiento Básico:

1. **Entrada:** Las neuronas de la capa de entrada reciben datos (características o atributos).
2. **Propagación:** Estos datos se propagan a través de las capas ocultas. En cada neurona, se realiza una suma ponderada de las entradas (cada entrada multiplicada por el peso de su conexión) y se le añade un "sesgo" (bias).
3. **Función de Activación:** El resultado de esta suma ponderada se pasa a través de una "función de activación" (no lineal) que decide si la neurona debe "activarse" y pasar información a la siguiente capa.
4. **Salida:** Finalmente, las neuronas de la capa de salida producen el resultado de la red (una predicción, una clasificación, etc.).
5. **Aprendizaje (Backpropagation):** Durante el entrenamiento, si la salida de la red es incorrecta, el error se propaga hacia atrás a través de la red (algoritmo de retro-propagación o *backpropagation*), ajustando los pesos de las conexiones para reducir el error en futuras predicciones.

Resumen MUY por arriba del artículo de Alan Turing

El artículo de Alan Turing de 1950, "Computing Machinery and Intelligence", es un texto fundacional en el campo de la inteligencia artificial. En él, Turing propone el "Juego de la Imitación" (conocido popularmente como la Prueba de Turing) como un criterio para determinar si una máquina puede "pensar", y luego aborda una serie de objeciones potenciales a su propuesta.

Verificación de Programas (by the goat Richard 🐐)

Conceptos varios

Diferencia entre Prueba Semántica y Sintáctica de un programa

La prueba semántica usa la semántica de las instrucciones del lenguaje para su prueba, mientras que la prueba sintáctica usa axiomas y reglas de un método lógico-deductivo para su prueba.

Estado de un programa

Un estado σ es una función que asigna a toda variable un valor. Por ejemplo: $\sigma(x) = 1, \sigma(y) = 2$, etc. Un estado σ satisface un predicado p, si p evaluado con σ es verdadero. Se expresa así: $\sigma \models p$. Por ejemplo: si $\sigma(x) = 1$ y $\sigma(y) = 2$, entonces $\sigma \models x < y$.

Especificación de un programa

Un programa se especifica en base a la precondición y post-condición del mismo. ambas son predicados que se deben cumplir justamente antes de empezar el programa y al terminar el programa, podemos usar un par que contenga a estas para especificar al programa. Por ejemplo, el par $(x = X \land y = Y, y = X \land x = Y)$ es la especificación de S_{swap}.

Invariante y Variante de un while

Un invariante es una propiedad lógica que se mantiene verdadera:

- Antes de entrar al bucle
- Durante cada iteración
- Al salir del bucle (si la condición es falsa)

Es una especie de *"regla"* que el programa *nunca rompe*, sin importar cuántas veces se ejecute el bucle.

Una variante es una expresión que decrece en cada iteración del bucle y que sirve para asegurar que el bucle terminará. Nunca se vuelve negativa o llega a un caso base, y representa el número máximo de iteraciones del bucle while.

Métodos axiomáticos sensatos y completos

Un método axiomático es sensato si prueba realmente lo que se quiere probar (no deriva en contradicciones), es decir, si se prueba $\{p\}S\{q\}$, se debe cumplir $\{p\}S\{q\}$. Propiedad obligatoria.

Un método axiomático es completo cuando todo lo que es verdadero dentro de la ejecución de un programa se puede demostrar usando al método, es decir, si se cumple $\{p\}S\{q\}$, se debe poder probar $\{p\}S\{q\}$. Propiedad deseable. La completitud depende de la expresividad del lenguaje de especificación.

¿Lógica de Hoare composicional?

La lógica de Hoare para los programas secuenciales es composicional (en programas concurrentes no) ya que: Dado un programa S, este está compuesto por subprogramas $S_1, \ldots, S_n$ donde para que valga la fórmula $\{p\}S\{q\}$ se depende sólo de que valgan fórmulas $\{p_1\}S_1\{q_1\}, \ldots, \{p_n\}S_n\{q_n\}$, sin importar el contenido de los S_i (noción de caja negra).

Axiomas de la Lógica de Hoare 💀

Axioma de la Asignación (ASI)

Se define como: $\{p(e)\}x := e\{p(x)\}$ → Se lee de atrás para adelante.

Al aplicar la regla ASI, lo que esta en la precondición (izquierda) queda igual que lo que esta en la postcondición (derecha) cambiando solo lo que la instrucción de la asignación hace.

- Ejemplo: $\{0 = 0\}x := 0\{x = 0\}$

Si en la asignación no se tocan las variables de la pre/post condición, queda todo igual.

- Ejemplo: $\{x = 0\}y := 0\{x = 0\}$

Regla de la Secuencia (SEC)

Se define como:

$$\frac{\{p\}S_1\{r\}, \{r\}S_2\{q\}}{\{p\}S_1; S_2\{q\}}$$

Nos permite unir las partes del programa que analizamos por separado en primera instancia, tener en cuenta que la postcondición que es precondición de la otra parte tiene que coincidir exactamente para poder unirse.

Regla del condicional (COND)

Se define como:

$$\frac{\{p \wedge B\}S_1\{q\}, \{p \wedge \neg B\}S_2\{q\}}{\{p\}if\ B\ then\ S_1\ else\ S_2\ fi\{q\}}$$

Es el modo que tenemos de verificar una selección condicional.

Regla de la Repetición (REP)

Se define como:

$$\frac{\{p \wedge B\}S\{p\}, \{p \wedge B \wedge t = Z\}S\{t < Z\}, p \rightarrow t \geq 0}{\{p\}while\ B\ do\ S\ od\{p \wedge \neg B\}}$$

Es el modo que tenemos de verificar un while.

Regla de Consecuencia (CONS)

Se define como:

$$\frac{r \longrightarrow p, \{p\}S\{q\}, q \longrightarrow s}{\{r\}S\{s\}}$$

Nos permite reforzar precondiciones y debilitar postcondiciones. Tenemos que estar muy atentos a la información que nos dan en los ejercicios para saber si hay una precondición más fuerte que implica una que logramos nosotros, o bien si existe una postcondición más débil que quizás nosotros implicamos con una que logramos obtener.

No necesariamente se tienen que modificar las 2 en una aplicación de la regla.

Ejercicios Relacionados

Probar (usando la lógica de Hoare): $\{true\}\mathrm{x} := 0\ ;\ \mathrm{x} := \mathrm{x} + 1\ ;\ \mathrm{x} := \mathrm{x} + 2\{x = 3\}$.

> **Proceso que seguí >**
>
> 1. Empezamos desde la postcondición final.
> 2. Aplicamos ASI hacia atrás para formar las precondiciones necesarias para después usar SEC.
> 3. Encadenamos con SEC (si nos quedan bien las condiciones intermedias y coinciden exactamente)
> 4. Usamos CONS en este caso para generalizar la precondición y probar lo que se pide.

1. $\{x + 2 = 3\}x := x + 2\{x = 3\}$ - ASI
2. $\{(x + 1) + 2 = 3\}x := x + 1\{x + 2 = 3\}$ - ASI

3. $\{(0+1)+2=3\}x := 0\{(x+1)+2=3\}$ - ASI
4. $\{(0+1)+2=3\}x := 0; x := x+1\{x+2=3\}$ - SEC 2 Y 3
5. $\{(0+1)+2=3\}x := 0; x := x+1; x := x+2\{x=3\}$ - SEC 4 Y 1
6. $\{true\}\mathrm{x} := 0\ ;\ \mathrm{x} := \mathrm{x}+1\ ;\ \mathrm{x} := \mathrm{x}+2\{x=3\}$ - CONS ya que: $\{true\} \to \{(0+1)+2=3\}$; es una verdad matemática

Se cumple $\{x=10\}\text{while } \mathrm{x} > 0 \text{ do } \mathrm{x} := \mathrm{x}-1 \text{ od}\{x=0\}$. Se pide probar (usando la lógica de Hoare) que el predicado $p = (x \geq 0)$ es un invariante del while. *Ayuda: hay que probar, por un lado: $x = 10 \longrightarrow p$, y por otro lado: $\{p \wedge x > 0\}\mathrm{x} := \mathrm{x}-1\{p\}$.*

Prueba de $x = 10 \longrightarrow p$, es decir, $x = 10 \longrightarrow (x \geq 0)$:

1. Trivialmente se ve que $x = 10$ es mayor o igual que 0.

Prueba de $\{p \wedge x > 0\}\mathrm{x} := \mathrm{x}-1\{p\}$, es decir, $\{x \geq 0 \wedge x > 0\}\mathrm{x} := \mathrm{x}-1\{x \geq 0\}$:

1. $\{x-1 \geq 0\}\mathrm{x} := \mathrm{x}-1\{x \geq 0\}$ - ASI
2. $\{x \geq 0 \wedge x > 0\}\mathrm{x} := \mathrm{x}-1\{x \geq 0\}$ - CONS ya que $\{x \geq 0 \wedge x > 0\} \to \{x-1 \geq 0\}$ es aritméticamente cierto, sabemos que $x > 0$, entonces se cumple siempre que $x-1 \geq 0$ para cualquier $x > 0$

Con esto, queda demostrado que $p = (x \geq 0)$ es invariante del while.