1. ¿Por qué estudiar lógica computacional?

Porque la lógica en las ciencias de la computación, en cierto sentido es el **cálculo de la computación** el cuál es el fundamento matemático para tratar la información y razonar acerca del comportamiento de programas.

2. ¿Por qué necesitamos la formalización del razonamiento correcto?

La lógica ha sido pieza clave para estructurar el pensamiento y el razonamiento:

- Dar un fundamento a las matemáticas.
- Eliminar errores del razonamiento.
- Encontrar una forma eficiente para llegar a una justificación de una conclusión, dada cierta información en forma de premisas.

3. Argumentos lógicos

3.1. Definición

Es una colección finita de afirmaciones *(proposiciones)* dividida en premisas y conclusión.

3.2. Acerca de las premisas y de la conclusión

Las premisas y la conclusión debe ser susceptibles de recibir un valor de verdad. El argumento lógico puede ser correcto o incorrecto.

3.3. Validez de un argumento

Un argumento es correcto o válido si suponiendo que sus premisas son ciertas, entonces necesariamente la conclusión también lo es.

3.4. Ejemplo

3.4.1. La isla de los caballeros y los bribones

En la isla de los caballeros y bribones sólo hay dos clases de habitantes, los caballeros que siempre dicen la verdad y los bribones que siempre mienten.
Un náufrago llega a la isla y encuentra dos habitantes: A y B. El habitante A afirma: Yo soy un bribón o B es un caballero. El acertijo consiste en averiguar cómo son A y B.

3.4.2. Lógica

$$p := \text{A es bribón}$$
$$q := \text{B es un caballero}$$
$$s := p \vee q$$

s es lo que dijo A.
Supongamos que $\mathcal{I}(p) = \top$
Así $\mathcal{I}(s) = \bot$
Entonces $\mathcal{I}(\neg s) = \top$
Como $\neg s \equiv \neg p \wedge \neg q$

Consecuentemente $\mathcal{I}(\neg p \wedge \neg q) = \top$
Lo que implica que se cumpla

$$\mathcal{I}(\neg p) = \top$$
$$\mathcal{I}(\neg q) = \top$$

Por lo tanto concluimos que A:

$$\mathcal{I}(p) = \bot$$

Por lo tanto hemos llegado a una contradicción, ya que A no puede ser bribón y no serlo al mismo tiempo.
Supongamos que $\mathcal{I}(p) = \bot$
Así $\mathcal{I}(s) = \top$
Como $s \equiv p \vee q$
Así tenemos que $\mathcal{I}(p) = \bot$ y $\mathcal{I}(q) = \top$
En éste caso A no es bribón y B es un caballero, entonces concluimos que:

- A es un caballero.
- B es un caballero.

3.5. Características de un argumento lógico

- Involucran individuos.
- Los individuos que involucran tienen propiedades.
- Una proposición es una oración que puede calificarse como verdadera o falsa y habla de las propiedades de los individuos.
- Se forman medinate proposiciones, clasificadas como premisas y conclusión del argumento.
- Las porposiciones pueden ser compuestas.
- Puede ser correcto o incorrecto.

4. Sistema lógico

Cualquier sistema lógico consta al menos de los siguientes tres componentes:

- Sintaxis.
- Semántica.
- Teoría de la prueba.

4.1. Sintaxis

Lenguaje formal que se utilizará como medio de expresión.

4.2. Semántica

Mecanismo que proporciona significado al lenguaje formal dado por la sintaxis.

4.3. Teoría de la prueba

Se encarga de decidir la correctud de un argumento lógico por medios puramente sintácticos.

4.4. Propiedades

4.4.1. Consistencia

No hay contradicciones

4.4.2. Correctud

Las reglas del sistema no pueden obtener una inferencia falsa a partir de una verdadera.

4.4.3. Completud

Todo lo verdadero es demostrable.

5. Lógica proposicional

5.1. Proposición

Es un enunciado que puede calificarse como verdadero o falso.

5.2. Lenguaje PROP

- Símbolos o variables proposicionales *(número infinito)*: $p_1, \ldots, p_n, \ldots$
- Constantes lógicas: $\top, \bot$
- Operadores lógicos: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- Símbolos auxiliares: ()

El conjunto de expresiones o fórmulas atómicas, denotado como ATOM consta de:

- Variables proposicionales: $p_1, p_2, \ldots, p_n, \ldots$
- Constantes lógicas: $\top, \bot$

5.3. Expresiones en el lenguaje PROP

5.3.1. Definición recursiva

- Si $\varphi \in$ ATOM entonces $\varphi \in$ PROP.
- Si $\varphi \in$ PROP entonces $(\neg\varphi) \in$ PROP.
- Sí $\varphi, \psi \in$ PROP entonces $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi), (\varphi \leftrightarrow \psi) \in$ PROP.
- Son todas.

5.3.2. Backus-Naur

$$
\begin{aligned}
VarP &::= p_1 \mid p_2 \mid \ldots \mid p_n \mid \ldots \\
\varphi, \psi &::= VarP \mid \top \mid \bot \mid (\neg\varphi) \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi) \mid (\varphi \leftrightarrow \psi) \mid
\end{aligned}
$$

5.3.3. ¿Cómo construimos proposiciones a través de la gramática anterior?

Veamos cómo construir $(p \wedge q) \rightarrow ((\neg q) \vee s)$

$$
\begin{aligned}
\varphi &\rightarrow \psi \\
(\varphi_1 \wedge \psi_1) &\rightarrow \psi \\
(\varphi_1 \wedge \psi_1) &\rightarrow (\varphi_2 \vee \psi_2) \\
(\varphi_1 \wedge \psi_1) &\rightarrow ((\neg\varphi_2) \vee \psi_2) \\
(VarP_{\varphi_1} \wedge VarP_{\psi_1}) &\rightarrow ((\neg VarP_{\varphi_2}) \vee VarP_{\psi_2}) \\
(p \wedge VarP_{\psi_1}) &\rightarrow ((\neg VarP_{\varphi_2}) \vee VarP_{\psi_2}) \\
(p \wedge q) &\rightarrow ((\neg VarP_{\varphi_2}) \vee VarP_{\psi_2}) \\
(p \wedge q) &\rightarrow ((\neg q) \vee VarP_{\psi_2}) \\
(p \wedge q) &\rightarrow ((\neg q) \vee s)
\end{aligned}
$$

Veamos cómo construir $((p \vee q) \to r) \leftrightarrow ((\neg r) \to (\neg (p \vee q)))$

$$\begin{aligned}
\varphi &\leftrightarrow \psi \\
(\varphi_1 \to \psi_1) &\leftrightarrow \psi \\
(\varphi_1 \to \psi_1) &\leftrightarrow (\varphi_2 \to \psi_2) \\
((\varphi_3 \vee \psi_3) \to \psi_1) &\leftrightarrow (\varphi_2 \to \psi_2) \\
((\varphi_3 \vee \psi_3) \to \psi_1) &\leftrightarrow ((\neg\varphi_2) \to \psi_2) \\
((\varphi_3 \vee \psi_3) \to \psi_1) &\leftrightarrow ((\neg\varphi_2) \to (\neg\psi_2)) \\
((\varphi_3 \vee \psi_3) \to \psi_1) &\leftrightarrow ((\neg\varphi_2) \to (\neg (\varphi_4 \vee \psi_4))) \\
((VarP_{\varphi_3} \vee VarP_{\psi_3}) \to VarP_{\psi_1}) &\leftrightarrow ((\neg VarP_{\varphi_2}) \to (\neg (VarP_{\varphi_4} \vee VarP_{\psi_4}))) \\
((p \vee VarP_{\psi_3}) \to VarP_{\psi_1}) &\leftrightarrow ((\neg VarP_{\varphi_2}) \to (\neg (VarP_{\varphi_4} \vee VarP_{\psi_4}))) \\
((p \vee q) \to VarP_{\psi_1}) &\leftrightarrow ((\neg VarP_{\varphi_2}) \to (\neg (VarP_{\varphi_4} \vee VarP_{\psi_4}))) \\
((p \vee q) \to r) &\leftrightarrow ((\neg VarP_{\varphi_2}) \to (\neg (VarP_{\varphi_4} \vee VarP_{\psi_4}))) \\
((p \vee q) \to r) &\leftrightarrow ((\neg r) \to (\neg (VarP_{\varphi_4} \vee VarP_{\psi_4}))) \\
((p \vee q) \to r) &\leftrightarrow ((\neg r) \to (\neg (p \vee VarP_{\psi_4}))) \\
((p \vee q) \to r) &\leftrightarrow ((\neg r) \to (\neg (p \vee q)))
\end{aligned}$$

Veamos cómo construir $\neg ((((p \wedge (p \vee (\neg q))) \wedge q) \wedge q))$

$$\begin{aligned}
&\neg (\varphi) \\
&\neg ((\varphi_1 \wedge \psi_1)) \\
&\neg (((\varphi_2 \wedge \psi_2) \wedge \psi_1)) \\
&\neg ((((\varphi_3 \wedge \psi_3) \wedge \psi_2) \wedge \psi_1)) \\
&\neg ((((\varphi_3 \wedge (\varphi_4 \vee \psi_4)) \wedge \psi_2) \wedge \psi_1)) \\
&\neg ((((\varphi_3 \wedge (\varphi_4 \vee (\neg\psi_4))) \wedge \psi_2) \wedge \psi_1)) \\
&\neg ((((VarP_{\varphi_3} \wedge (VarP_{\varphi_4} \vee (\neg VarP_{\psi_4}))) \wedge VarP_{\psi_2}) \wedge VarP_{\psi_1})) \\
&\neg ((((p \wedge (VarP_{\varphi_4} \vee (\neg VarP_{\psi_4}))) \wedge VarP_{\psi_2}) \wedge VarP_{\psi_1})) \\
&\neg ((((p \wedge (p \vee (\neg VarP_{\psi_4}))) \wedge VarP_{\psi_2}) \wedge VarP_{\psi_1})) \\
&\neg ((((p \wedge (p \vee (\neg q))) \wedge VarP_{\psi_2}) \wedge VarP_{\psi_1})) \\
&\neg ((((p \wedge (p \vee (\neg q))) \wedge q) \wedge VarP_{\psi_1})) \\
&\neg ((((p \wedge (p \vee (\neg q))) \wedge q) \wedge q))
\end{aligned}$$

Veamos cómo construir $(\neg s) \to ((\neg t) \wedge \neg (p \vee q))$

$$\begin{aligned}
(\varphi &\to \psi) \\
((\neg\varphi) &\to \psi) \\
((\neg\varphi) &\to (\varphi_1 \wedge \psi_1)) \\
((\neg\varphi) &\to ((\neg\varphi_1) \wedge \psi_1)) \\
((\neg\varphi) &\to ((\neg\varphi_1) \wedge (\neg\psi_1))) \\
((\neg\varphi) &\to ((\neg\varphi_1) \wedge (\neg(\varphi_2 \vee \psi_2)))) \\
((\neg VarP_{\varphi}) &\to ((\neg VarP_{\varphi_1}) \wedge (\neg(VarP_{\varphi_2} \vee VarP_{\psi_2})))) \\
((\neg s) &\to ((\neg VarP_{\varphi_1}) \wedge (\neg(VarP_{\varphi_2} \vee VarP_{\psi_2})))) \\
((\neg s) &\to ((\neg t) \wedge (\neg(VarP_{\varphi_2} \vee VarP_{\psi_2})))) \\
((\neg s) &\to ((\neg t) \wedge (p \vee q))))
\end{aligned}$$

5.3.4. Ejercicios

Realizar las siguientes construcciones:

- $(s \vee r) \wedge (p \to (p \to \neg (q \wedge (s \vee q))))$
- $((\neg (r \to t)) \leftrightarrow ((t \wedge (p \to (q \vee t))) \leftrightarrow s))$
- $((p \to (q \leftrightarrow (s \wedge (t \to q)))) \wedge (q \vee (s \vee (t \leftrightarrow (q \vee (\neg r))))))$

5.4. Precedencia y asociatividad de los operadores lógicos

Precedencia de mayor a menor

- $\neg$
- $\vee, \wedge$
- $\rightarrow$
- $\leftrightarrow$

Operadores asociativos:

- $\neg$
- $\vee, \wedge$
- $\leftrightarrow$

El operador $\rightarrow$ asocia hacia la derecha.
Ejemplificando lo anterior la expresión

$$\varphi_1 \rightarrow \varphi_2 \rightarrow \varphi_3$$

Queda asociada como:

$$\varphi_1 \rightarrow (\varphi_2 \rightarrow \varphi_3)$$

5.5. Eliminación de paréntesis innecesarios

Notemos que:

$$p \wedge q \rightarrow \neg q \vee s$$

es igual a

$$(p \wedge q) \rightarrow ((\neg q) \vee s)$$

Pero ¿Cómo podemos obtener la primera sí nos dan la última?
Simplemente debemos de ir quitando paréntesis guiándonos por la precedencia de operadores

$$\begin{aligned}
&(p \wedge q) \rightarrow ((\neg q) \vee s) && \\
&(p \wedge q) \rightarrow (\neg q \vee s) && \text{Eliminamos el paréntesis del operador } \neg \\
&p \wedge q \rightarrow \neg q \vee s && \text{Eliminamos el paréntesis del operador } \vee \text{ y } \wedge
\end{aligned}$$

Veamos $(p \rightarrow q \wedge r) \leftrightarrow (s \vee (\neg t))$

$$\begin{aligned}
&(p \rightarrow q \wedge r) \leftrightarrow (s \vee (\neg t)) && \\
&(p \rightarrow q \wedge r) \leftrightarrow (s \vee \neg t) && \text{Eliminamos el paréntesis del operador } \neg \\
&(p \rightarrow q \wedge r) \leftrightarrow s \vee \neg t && \text{Eliminamos el paréntesis del operador } \vee \\
&p \rightarrow q \wedge r \leftrightarrow s \vee \neg t && \text{Eliminamos el paréntesis del operador } \rightarrow
\end{aligned}$$

Veamos $((p \vee q) \rightarrow r) \leftrightarrow ((\neg r) \rightarrow (\neg (p \vee q)))$

$$\begin{aligned}
&((p \vee q) \rightarrow r) \leftrightarrow ((\neg r) \rightarrow (\neg (p \vee q))) && \\
&((p \vee q) \rightarrow r) \leftrightarrow (\neg r \rightarrow (\neg (p \vee q))) && \text{Eliminamos el paréntesis del operador } \neg \\
&((p \vee q) \rightarrow r) \leftrightarrow (\neg r \rightarrow \neg (p \vee q)) && \text{Eliminamos el paréntesis del operador } \neg \\
&p \vee q \rightarrow r \leftrightarrow (\neg r \rightarrow \neg (p \vee q)) && \text{Eliminamos el paréntesis del operador } \rightarrow \\
&p \vee q \rightarrow r \leftrightarrow \neg r \rightarrow \neg (p \vee q) && \text{Eliminamos el paréntesis del operador } \rightarrow
\end{aligned}$$

Veamos $\neg (((p \wedge (p \vee (\neg q))) \wedge q) \wedge p)$

$$\begin{aligned}
&\neg (((p \wedge (p \vee (\neg q))) \wedge q) \wedge p) && \\
&\neg (((p \wedge (p \vee \neg q)) \wedge q) \wedge p) && \text{Eliminamos el paréntesis del operador } \neg \\
&\neg ((p \wedge (p \vee \neg q) \wedge q) \wedge p) && \text{Eliminamos el paréntesis del operador } \wedge \\
&\neg (p \wedge (p \vee \neg q) \wedge q \wedge p) && \text{Eliminamos el paréntesis del operador } \wedge
\end{aligned}$$

Veamos $(\neg s) \to ((\neg t) \land \neg (p \lor q))$

$$
\begin{aligned}
&(\neg s) \to ((\neg t) \land \neg (p \lor q)) && \\
&\neg s \to ((\neg t) \land \neg (p \lor q)) && \text{Eliminamos el paréntesis del operador } \neg \\
&\neg s \to (\neg t \land \neg (p \lor q)) && \text{Eliminamos el paréntesis del operador } \neg \\
&\neg s \to \neg t \land \neg (p \lor q) && \text{Eliminamos el paréntesis del operador } \land
\end{aligned}
$$

5.5.1. Ejercicios

Elimina los paréntesis innecesarios de la expresiones:

- $(p \to (q \land (\neg q))) \to ((\neg p) \to p)$
- $(s \lor r) \land (p \to (p \to \neg (q \land (s \lor q))))$

6. Definiciones Recursivas y el Principio de Inducción

6.1. ¿En qué consiste?

En establecer las construcciones para generar elementos de una colección o de una estructura de datos.
La definición del lenguaje `PROP` es un ejemplo de ésta clase de definiciones, podemos identificar las construcciones básicas o atómicas y las construcciones recursivas.
Las definiciones recursivas también sirven para definir propiedades o funciones de una estructura mediante un análisis de casos, es decir de las distintas formas sintácticas que definen a los elementos de dicha estructura.

6.2. Ejemplo

Sea np una función que devuelve el número de paréntesis de una expresión lógica.
Definimos np como sigue:

$$
\begin{aligned}
np \ &: \ \texttt{PROP} \to \mathbb{N} && \\
np\,(\varphi) &= 0 && \text{Si } \varphi \in \texttt{ATOM} \\
np\,((\neg\varphi)) &= 2 + np\,(\varphi) && \text{Si } \varphi \in \texttt{ATOM} \\
np\,((\varphi \star \psi)) &= 2 + np\,(\varphi) + np\,(\psi) && \text{con } \star \in \{\land, \lor, \to, \leftrightarrow\} \\
np\,(\neg\varphi) &= np\,(\varphi) && \\
np\,(\varphi \star \psi) &= np\,(\varphi) + np\,(\psi) && \text{con } \star \in \{\land, \lor, \to, \leftrightarrow\}
\end{aligned}
$$

6.3. Ejemplos de uso

Calculemos $np\,(\neg\,(((p \land (p \lor (\neg q))) \land q) \land p))$

$$
\begin{aligned}
np\,(\neg\,(((p \land (p \lor (\neg q))) \land q) \land p)) &= 2 + np\,(((p \land (p \lor (\neg q))) \land q) \land p) && \text{Aplicamos el segundo caso.} \\
np\,(((p \land (p \lor (\neg q))) \land q) \land p) &= 2 + np\,(((p \land (p \lor (\neg q))) \land q)) + np\,(p) && \text{Aplicamos el quinto caso.} \\
np\,(((p \land (p \lor (\neg q))) \land q)) &= 2 + np\,((p \land (p \lor (\neg q)))) + np\,(q) && \text{Aplicamos el tercer caso.} \\
np\,((p \land (p \lor (\neg q)))) &= 2 + np\,(p) + np\,((p \lor (\neg q))) && \text{Aplicamos el tercer caso.} \\
np\,(p) &= 0 && \text{Aplicamos el primer caso.} \\
np\,((p \lor (\neg q))) &= 2 + np\,(p) + np\,((\neg q)) && \text{Aplicamos el tercer caso.} \\
np\,(p) &= 0 && \text{Aplicamos el primer caso.} \\
np\,((\neg q)) &= 2 + np\,(q) && \text{Aplicamos el segundo caso.} \\
np\,(q) &= 0 && \text{Aplicamos el primer caso.}
\end{aligned}
$$

Por lo tanto

$$\begin{aligned}
np\left((\neg q)\right) &= 2 + np\left(q\right) = 2 + 0 = 2 \\
np\left((p \vee (\neg q))\right) &= 2 + np\left(p\right) + np\left((\neg q)\right) = 2 + 0 + 2 = 4 \\
np\left((p \wedge (p \vee (\neg q)))\right) &= 2 + np\left(p\right) + np\left((p \vee (\neg q))\right) = 2 + 0 + 4 = 6 \\
np\left(((p \wedge (p \vee (\neg q))) \wedge q)\right) &= 2 + np\left((p \wedge (p \vee (\neg q)))\right) + np\left(q\right) = 2 + 6 + 0 = 8 \\
np\left(((p \wedge (p \vee (\neg q))) \wedge q) \wedge p\right) &= 2 + np\left(((p \wedge (p \vee (\neg q))) \wedge q)\right) + np\left(p\right) = 2 + 8 + 0 = 10 \\
np\left(\neg\left(((p \wedge (p \vee (\neg q))) \wedge q) \wedge p\right)\right) &= 2 + np\left(((p \wedge (p \vee (\neg q))) \wedge q) \wedge p\right) = 2 + 10 = 12
\end{aligned}$$

Por lo tanto el número de paréntesis que tiene la fórmula son 12.

6.4. Definición de funciones recursivas

Define las siguientes funciones, indicando dominio y contradominio de la función definida.

- Número de conectivos de una expresión: $con\left(\varphi\right)$ devuelve el número de conectivos de φ.
- Variables de una expresión: $vars\left(\varphi\right)$ devuelve el conjunto de variables que figuran en φ.
- Atómicas en una expresión: $atom\left(\varphi\right)$ devuelve el número de presencias de fórmulas atómicas en φ.

6.4.1. Número de conectivos

Recordemos que los conectivos son:

$$\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$$

Así tenemos que la correspondencia de nuestra función es:

$$con \ : \ \texttt{PROP} \rightarrow \mathbb{N}$$

Definición de la función

$$\begin{aligned}
con\left(\varphi\right) &= 0 && \text{Sí } \varphi \in \texttt{ATOM} \\
con\left(\neg\varphi\right) &= 1 + con\left(\varphi\right) && \\
con\left(\varphi \star \psi\right) &= 1 + con\left(\varphi\right) + cons\left(\psi\right) && \text{con } \star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}
\end{aligned}$$

Hacemos la observación que φ y ψ pueden contener o no paréntesis es decir ambas pueden ser de la forma:

$$(\kappa \star \omega)$$

con $\star \in \{\neg, \wedge, \vee, \rightarrow, \Leftrightarrow\}$ ó

$$(\neg\gamma)$$

Para $\kappa, \omega, \gamma \in \texttt{PROP}$.

6.4.2. Variables de una fórmula

Tenemos que la correspondencia de nuestra función es:

$$vars \ : \ \texttt{PROP} \rightarrow \{\texttt{PROP}\}$$

Definición de la función

$$\begin{aligned}
vars\left(\varphi\right) &= \varnothing && \text{Sí } \varphi \in \{\bot, \top\} \\
vars\left(\varphi\right) &= \{\varphi\} && \text{Sí } \varphi \in \texttt{ATOM} \backslash \{\bot, \top\} \\
vars\left(\neg\varphi\right) &= vars\left(\varphi\right) && \\
vars\left(\varphi \star \psi\right) &= vars\left(\varphi\right) \cup vars\left(\psi\right) && \text{con } \star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}
\end{aligned}$$

Hacemos la observación que φ y ψ pueden contener o no paréntesis es decir ambas pueden ser de la forma:

$$(\kappa \star \omega)$$

con $\star \in \{\neg, \wedge, \vee, \rightarrow, \Leftrightarrow\}$ ó

$$(\neg\gamma)$$

Para $\kappa, \omega, \gamma \in \texttt{PROP}$.

6.4.3. Atómicas de una fórmula

Tenemos que la correspondencia de nuestra función es:

$$atom \ : \ \texttt{PROP} \to \mathbb{N}$$

Definición de la función

$$\begin{aligned} atom\,(\varphi) &= 1 && \text{Sí } \varphi \in \texttt{ATOM} \\ atom\,(\neg\varphi) &= atom\,(\varphi) \\ atom\,(\varphi \star \psi) &= atom\,(\varphi) + atom\,(\psi) && \text{con } \star \in \{\wedge, \vee, \to, \leftrightarrow\} \end{aligned}$$

Hacemos la observación que φ y ψ pueden contener o no paréntesis es decir ambas pueden ser de la forma:

$$(\kappa \star \omega)$$

con $\star \in \{\wedge, \vee, \to, \leftrightarrow\}$ ó

$$(\neg\gamma)$$

Para $\kappa, \omega, \gamma \in \texttt{PROP}$.

6.4.4. Ejercicio

Define una función $depth\,(\varphi)$ que devuelve la altura del árbol de sintaxis abstracta de φ.

6.4.5. Solución

Dominio

$$depth \ : \ \texttt{PROP} \to \mathbb{N}$$

Definición

$$\begin{aligned} depth(\varphi) &= 0 && \text{Si } \varphi \in \texttt{ATOM} \\ depth(\neg\varphi) &= 1 + depth(\varphi) \\ depth(\varphi \star \psi) &= 1 + \max\,(depth(\varphi), depth(\psi)) \end{aligned}$$

6.5. Principio de Inducción Estructural para PROP

Sea $\mathcal{P}$ una propiedad acerca de fórmulas del lenguaje PROP. Para probar que toda fórmula $\varphi \in \texttt{PROP}$ tiene la propiedad $\mathcal{P}$ basta demostrar lo siguiente:

1. **Caso base:** toda variable proposicional tiene la propiedad $\mathcal{P}$.
2. **Hipótesis de Inducción:** suponer que se cumple la propiedad $\mathcal{P}$ para φ y ψ.
3. **Paso Inductivo:** mostrar usando la hipótesis de inducción que:
   a) $(\neg\varphi)$ cumple $\mathcal{P}$.
   b) $(\varphi \star \psi)$ cumple $\mathcal{P}$ para todo $\star \in \{\wedge, \vee, \to, \leftrightarrow\}$.

6.5.1. Ejemplo

Demostrar mediante inducción estructural que se cumplen la siguiente desigualdad

$$atom\,(\varphi) \leq 2 \cdot con\,(\varphi) + 1$$

Demostración. La demostración se hará por medio del **Principio de Inducción Estructural para** PROP.
Caso Base
Sea φ una variable proposicional entonces por definición de la función $atom$ tenemos que:

$$atom\,(\varphi) = 1$$

Mientras por definición de la función $cons$ tenemos que:

$$2 \cdot con\,(\varphi) + 1 = 2 \cdot 0 + 1 = 0 + 1 = 1$$

Entonces se cumple que

$$1 \leq 1$$

Consecuentemente

$$atom\,(\varphi) \leq 2 \cdot con\,(\varphi)$$

Trivialmente

$$2 \cdot con\,(\varphi) < 2 \cdot con\,(\varphi) + 1$$

Por lo tanto

$$atom\,(\varphi) \leq 2 \cdot con\,(\varphi) + 1$$

Hipótesis de Inducción
Sean φ y ψ en PROP entonces φ y ψ cumplen que:

$$atom\,(\varphi) \leq 2 \cdot con\,(\varphi) + 1$$
$$atom\,(\psi) \leq 2 \cdot con\,(\psi) + 1$$

Paso inductivo
Sea $\gamma \in$ PROP entonces demostraremos que

$$atom\,(\gamma) \leq 2 \cdot con\,(\gamma) + 1$$

Caso 1
$\gamma \equiv \neg\varphi$
Por definición de la función $atom$ tenemos que:

$$atom\,(\neg\varphi) = atom\,(\varphi)$$

Por **hipótesis de inducción**

$$atom\,(\varphi) \leq 2 \cdot con\,(\varphi) + 1$$

Notemos que:

$$2 \cdot con\,(\varphi) + 1 = con\,(\varphi) + con\,(\varphi) + 1$$

Es trivial ver que:

$$con\,(\varphi) + con\,(\varphi) + 1 < con\,(\varphi) + con\,(\varphi) + 1 + 1 + 1$$

Por asociatividad tenemos que:

$$con\,(\varphi) + con\,(\varphi) + 1 + 1 + 1 = con\,(\varphi) + 1 + con\,(\varphi) + 1 + 1$$

Que por definición de la función con es:

$$2 \cdot con\,(\varphi) + 1 < con\,(\neg\varphi) + con\,(\neg\varphi) + 1$$

Que es equivalente a

$$2 \cdot con\,(\varphi) + 1 < 2 \cdot con\,(\neg\varphi) + 1$$

Así tenemos que

$$atom\,(\varphi) \leq 2 \cdot con\,(\varphi) + 1 < 2 \cdot con\,(\neg\varphi) + 1$$

Por lo que podemos concluir que

$$atom\,(\neg\varphi) \leq 2 \cdot con\,(\neg\varphi) + 1$$

Caso 2
$\gamma \equiv \varphi \star \psi$
Por definición de la función $atom$ tenemos que:

$$atom\,(\varphi \star \psi) = atom\,(\varphi) + atom\,(\psi)$$

Por **hipótesis de inducción**

$$atom\,(\varphi) \leq 2 \cdot con\,(\varphi) + 1$$
$$atom\,(\psi) \leq 2 \cdot con\,(\psi) + 1$$

Así

$$\begin{aligned}
&atom(\varphi) + atom(\psi) \leq 2 \cdot con(\varphi) + 1 + 2 \cdot con(\psi) + 1 \\
&atom(\varphi) + atom(\psi) \leq con(\varphi) + con(\varphi) + 1 + con(\psi) + con(\psi) + 1 \\
&atom(\varphi) + atom(\psi) \leq con(\varphi) + con(\psi) + 1 + con(\varphi) + con(\psi) + 1 \\
&atom(\varphi) + atom(\psi) \leq con(\varphi \star \psi) + con(\varphi \star \psi) \\
&atom(\varphi) + atom(\psi) \leq 2 \cdot con(\varphi \star \psi)
\end{aligned}$$

Es inmediato ver que:

$$2 \cdot con(\varphi \star \psi) < 2 \cdot con(\varphi \star \psi) + 1$$

Por lo tanto

$$\begin{aligned}
atom(\varphi) + atom(\psi) &\leq 2 \cdot con(\varphi \star \psi) + 1 \\
atom(\varphi \star \psi) &\leq 2 \cdot con(\varphi \star \psi) + 1
\end{aligned}$$

Por lo que podemos concluir que

$$atom(\varphi \star \psi) \leq 2 \cdot con(\varphi \star \psi) + 1$$

Por lo tanto se cumple la propiedad. □

6.5.2. Ejercicios

- Demuestra mediante inducción estructural que se cumpla lo siguiente:

$$con(\varphi) < 2^{depth(\varphi)}$$

- Prueba que cualquier fórmula proposicional ψ que tienen únicamente disyunciones es verdadera si alguna de sus variables proposicionales es verdadera.

6.5.3. Demostración del ejercicio 2

Demostración. La demostración se hará por **Inducción Matemática** sobre el número de variables proposicionales.
Caso base
$n = 2$
Sea ψ una fórmula proposicional con únicamente disyunciones, entonces podemos definir a ψ como sigue:

$$\varphi \vee \gamma$$

con $\varphi, \gamma \in \texttt{PROP}$.
Como $\psi \equiv \top$ entonces, tenemos que se cumplieron alguno de los siguientes tres casos:

- $\varphi \equiv \top$ y $\gamma \equiv \bot$.
- $\varphi \equiv \bot$ y $\gamma \equiv \top$.
- $\varphi \equiv \top$ y $\gamma \equiv \top$.

Lo anterior por la **tabla de verdad** del operador lógico $\vee$, así al menos una variable proposicional es verdadera, por lo tanto se cumple la propiedad.
Hipótesis de Inducción
Sea ψ una fórmula proposicional con n variables proposicionales y la cual consiste de puras disyunciones además ψ es verdadera, entonces alguna de sus variables proposicionales es verdadera.
Paso Inductivo
Sea ψ' una fórmula proposicional con $n + 1$ variables proposicionales tal que $\psi' \equiv \top$, entonces podemos reescribir a ψ' como sigue:

$$\psi \vee \gamma$$

tal que ψ consiste en n variables proposicionales.
Tenemos dos casos
Caso (1)

Sí $\psi \equiv \top$ entonces se cumple la **hipótesis de inducción** y al menos alguna de las variables proposicionales es verdadera.
Caso (2)
Sí no se cumple que $\psi \equiv \top$, entonces por las propiedades del operador $\vee$, γ es verdadera y se cumple que al menos alguna de las variables proposicionales es verdadera.
Caso (3)
$\psi \equiv \top$ y $\gamma \equiv \top$ entonces se sigue el resultado por los casos anteriores.
Por lo tanto se cumple la propiedad. □

7. Sustitución

7.1. Definición

En una fórmula dada φ, una variable proposicional p cambia o se reemplaza por una fórmula ψ. Así generamos una nueva fórmula denotada $\varphi\,[p \;:=\; \psi]$ obtenida al sustituir todas las presencias de p en φ por ψ.

7.2. Definición recursiva

$$
\begin{aligned}
p\,[p \;:=\; \psi] &= \psi \\
q\,[p \;:=\; \psi] &= q \\
\top\,[p \;:=\; \psi] &= \top \\
\bot\,[p \;:=\; \psi] &= \bot \\
(\neg\varphi)\,[p \;:=\; \psi] &= \neg\,(\varphi\,[p \;:=\; \psi]) \\
(\varphi \wedge \gamma)\,[p \;:=\; \psi] &= (\varphi\,[p \;:=\; \psi] \wedge \gamma\,[p \;:=\; \psi]) \\
(\varphi \vee \gamma)\,[p \;:=\; \psi] &= (\varphi\,[p \;:=\; \psi] \vee \gamma\,[p \;:=\; \psi]) \\
(\varphi \to \gamma)\,[p \;:=\; \psi] &= (\varphi\,[p \;:=\; \psi] \to \gamma\,[p \;:=\; \psi]) \\
(\varphi \leftrightarrow \gamma)\,[p \;:=\; \psi] &= (\varphi\,[p \;:=\; \psi] \leftrightarrow \gamma\,[p \;:=\; \psi])
\end{aligned}
$$

7.3. Sustitución simultánea

Sustitución simultánea de n variables proposicionales por n fórmulas, detoda como $\varphi\left[\vec{p} \;:=\; \vec{\psi}\right]$.

7.4. Propiedades

- Si p no figura en φ, es decir $p \notin vars\,(\varphi)$, entonces $\varphi\,[p \;:=\; \psi] = \varphi$.
- Si $p \neq q$ y p no figura en γ entonces

$$\varphi\,[p \;:=\; \psi]\,[q \;:=\; \gamma] = \varphi\,[q \;:=\; \gamma]\,[p \;:=\; \psi\,[q \;:=\; \gamma]]$$

- Si $p \neq q_1, \ldots, q_n$ y p no figura en $\psi_1, \ldots, \psi_n$ entonces

$$\varphi\left[\vec{q}, p \;:=\; \vec{\psi}, \gamma\right] = \varphi\left[\vec{q} \;:=\; \vec{\psi}\right][p \;:=\; \gamma]$$