Resumen - Lógica Computacional

Nicolás Margenat

2Q 2021/1Q 2022

Contents

1 Cardinalidad

1.1 Definiciones Previas

Definición 1. Conjunto Coordinable

A es *coordinable* con B si $\exists f : A \to B$ biyectiva.
Notación: $A \sim B$
Observación: $A\mathcal{R}B$ si $A \sim B$ es una relación de equivalencia.

Definición 2. Sección Inicial

$I_n = \{1, 2, ..., n\}$ con $n \in \mathbb{N}_{\geq 1}$
Ejemplos: $I_1 = \{1\}$, $I_2 = \{1, 2\}$
Observación: $I_n \sim I_m \Leftrightarrow n = m$

Definición 3. Conjuntos finitos e infinitos

A es *finito* si $A = \varnothing$ o $\exists k \in \mathbb{N}_{\geq 1} / A \sim I_k$
Un conjunto es *infinito* si no es finito
Observación: $\mathbb{N}$ es infinito

Definición 4. Cardinales

Card($\mathbb{N}$) = χ_0 ("aleph cero")
Card(I_k) = K (K = k)
Card($\varnothing$) = 0
Card($[0, 1]$) = c ($> \chi_0$)
Card($\mathcal{P}(\mathbb{N})$) $= c$
Card($\mathcal{P}_{finitas}(\mathbb{N})$) $= \chi_0$
Card($\mathcal{P}_{infinitas}(\mathbb{N})$) $= c$

Álgebra de Cardinales importantes

1. $n + \chi_0 = \chi_0$ con $n \in \mathbb{N}$
2. $\chi_0 + \chi_0 = \chi_0$
3. $\sum_{i=1}^{n} \chi_0 = \chi_0$ con $n \in \mathbb{N}_{>0}$
4. $c + \chi_0 = c$
5. $\chi_0 \cdot \chi_0 = \chi_0$
6. $2^{\chi_0} = c$
7. $c \cdot c = c$
8. $\chi_0 \cdot c = c$
9. $c^n = c$ con $n \in \mathbb{N}_{>0}$
10. $c^{\chi_0} = c$
11. $c^c = 2^c$

Definición 5. Comparación de cardinales

A y B conjuntos. $\#A = m, \#B = K$

1. Decimos que $m \leq K$ si $\exists f : A \to B$ inyectiva
2. Decimos que $m \geq K$ si $\exists f : A \to B$ sobreyectiva
3. Decimos que $m = K$ si $\exists f : A \to B$ biyectiva
4. Decimos que $m < K$ si $m \leq K$ y $m \neq K$
5. Decimos que $m > k$ si $m \geq K$ y $m \neq K$

Observación: $\leq$ y $\geq$ son relaciones de orden (R A T).

Definición 6. Conjunto numerable

A es numerable si A es finito o $A \sim \mathbb{N}$

Definición 7. Sucesiones

Una funcion $f : \mathbb{N} \to A$ es una sucesión de elementos de A:

$$\underbrace{f(0)}_{a_0}, \underbrace{f(1)}_{a_1}, \underbrace{f(2)}_{a_2}, \underbrace{f(4)}_{a_4}, ...$$

Notación: (a_n) con $n \in \mathbb{N}$

Definición 8. Álgebra de Cardinales

Sean $a = Card(X)$, $b = Card(Y)$ y $X \cap Y = \varnothing$.
Definimos:

1. $a + b = \#(X \cup Y)$
2. $a * b = \#(X \times Y)$
3. $b^a = \#\{f : X \to Y / \text{ f es funcion}\}$

Algunas **propiedades**:

1. $(x^y)^z = x^{y*z}$
2. Si $y \leq z$, entonces:

$$x^y \leq x^z$$
$$y^x \leq z^x$$
$$x \cdot y \leq x \cdot z$$

3. $x^y \cdot x^z = x^{y+z}$

1.2 Propiedades/Observaciones

1. $m \leq K \Leftrightarrow K \geq m$ con m, K cardinales.
2. Cualquier subconjunto de una sección inicial es finito.
3. $A \subseteq B$ y A es infinito $\Rightarrow$ B es infinito.
4. X infinito $\Rightarrow \exists f : \mathbb{N} \to X$ es inyectiva.
5. A numerable y $A \neq \varnothing \Rightarrow \exists f : \mathbb{N} \to A$ es sobreyectiva.
6. $\exists f : \mathbb{N} \to A$ es sobreyectiva $\Rightarrow$ A es numerable.
7. Si $A \subseteq B \Rightarrow \#A \leq \#B$
8. $(\mathbb{N}^k \times \mathbb{N}) \sim \mathbb{N}$
9. $[0,1] = \{x \in \mathbb{R} / \, 0 \leq x \leq 1\}$ es infinito no numerable.
10. X es infinito no numerable, A numerable $\Rightarrow X \cup A \sim X$
11. X infinito no numerable, A numerable $\Rightarrow X - A \sim X$
12. A, B conjuntos numerables $\Rightarrow A \cup B$ es numerable.
13. $(A_n)_{n \in \mathbb{N}}$ una familia de conjuntos numerables $\Rightarrow \bigcup_{n \in \mathbb{N}} A_n$ es numerable.
14. A conjunto finito no vacío y $S = \bigcup_{m \in \mathbb{N}_{>0}} A^m \Rightarrow \#S = \chi_0$
15. Sea $A \neq \varnothing$ un conjunto. Entonces:
$$A \text{ es infinito } \Leftrightarrow \forall x \in A, \exists f_x : A \to A - \{x\} \text{ es biyectiva}$$
16. $\#\mathcal{P}(X) = 2^{\#X}$

1.3 Teoremas

1.3.1 Teorema de Bernstein

Versión 1 $\#A = m$, $\#B = K$
$$m \leq K \text{ y } m \geq K \Rightarrow m = K$$
Necesito demostrar inyectividad y sobreyectividad.

Versión 2
Utilizando la Propiedad 1.
$\#A = m$, $\#B = K$
$$m \leq K \text{ y } K \leq m \Rightarrow m = K$$
Osea lo puedo demostrar probando inyectividad en ambos casos.

1.3.2 Teorema de Cantor

$$\#X < \#\mathcal{P}(X)$$

1.3.3 Hipótesis del Continuo

DISCLAIMER: Esto en realidad no es un teorema, en el sentido de que no se pudo probar la veracidad del mismo, pero tampoco se pudo probar que es falso. Por ende, toda la comunidad científica/matemática se puso de acuerdo en decir que es verdad en el mientras tanto porque no rompe nada.

Dicho esto, la hipótesis dice lo siguiente:

$$\nexists X \text{ conjunto } / \ \chi_0 < \#X < C$$

2 Lógica Proposicional - Lenguajes

2.1 Definiciones Previas

Definición 1. Alfabeto

A conjunto, $A \neq \varnothing$, se llama *alfabeto.*

Definición 2. Expresión

Una expresión es una sucesión finita de elementos de A o la cadena vacía a la que llamamos λ.

Definición 3. Longitud de una Expresión

Sea $E = e_0, e_1, ..., e_{n-1}$ una expresión de elementos de A, $long(E) = n$ y $long(\lambda) = 0$.

Definición 4. Conjunto A^*

$$A^* = \bigcup_{k \in \mathbb{N}} A_k$$

donde $A_k = \{e \text{ expresión en } A / \, long(e) = k\}$ y A es un alfabeto.
Por ejemplo, $A_0 = \{\lambda\}$, $A_1 = A$, etc.

Definición 5. Lenguaje

A alfabeto, un lenguaje $\Sigma \neq \varnothing$ sobre A es $\Sigma \subseteq A^*$

Definición 6. Igualdad de expresiones

Sean:

$$E = e_0, e_1, ..., e_{n-1}$$
$$F = f_0, f_1, ..., f_j$$

entonces $E = F$ si $j = n - 1$ y $e_i = f_i$ con $0 \leq i \leq n - 1$

Definición 7. Concatenación

$$E = e_0, e_1, ..., e_{n-1} \in A^*$$
$$F = f_0, f_1, ..., f_{k-1} \in A^*$$

Definimos la concatenación como:

$$EF = e_0, ..., e_{n-1}, f_0, ..., f_{k-1}$$

<u>Observaciones</u>:

1. $long(EF) = long(E) + long(F) = n + k$
2. $E\lambda = \lambda E = E$
3. $\lambda\lambda = \lambda$
4. $FE \neq EF$ si $E \neq F$

2.2 Proposiciones

1. A alfabeto, E, F, G, $H \in A^*$, $EF = GH$ y $long(E) \geq long(G)$

$$\Rightarrow \exists H \in A^* / \, E = GH'$$

2. A alfabeto, E, F, G, $H \in A^* / \, EF = GH$ y $long(E) = long(G)$

$$\Rightarrow E = G \text{ y } F = H$$

3 Lógica Proposicional - Semántica

3.1 Definciones Previas

Definición 1. Valuación

Una valuación es una función:

$$v : FORM \to \{0, 1\} \text{ que verifica:}$$

1. $v(\neg\alpha) = 1 - v(\alpha)$
2. $v(\alpha_1 \wedge \alpha_2) = min\{v(\alpha_1), v(\alpha_2)\}$
3. $v(\alpha_1 \vee \alpha_2) = max\{v(\alpha_1), v(\alpha_2)\}$
4. $v(\alpha_1 \to \alpha_2) = max\{1 - v(\alpha_1), v(\alpha_2)\}$

Definición 2. Clasificación Semántica de las fórmulas

Sea $\alpha \in F$. Entonces:

1. α es **TAUTOLOGÍA** si $v(\alpha) = 1\,\forall v$ val.
2. α es **CONTRADICCIÓN** si $v(\alpha) = 0\,\forall v$ val.
3. α es **CONTINGENCIA** si $\exists v$ val $/\, v(\alpha) = 1$ y $\exists w$ val $/\, w(\alpha) = 0$

Definición 3. Equivalencia de fórmulas

Sean α, $\beta \in F$.
Decimos que α *es equivalente a* β si $v(\alpha) = v(\beta) \forall v$ val.
<u>Notación</u>: $\alpha \equiv \beta$
<u>Observación</u>: $\mathcal{R} \in F/\, \alpha\mathcal{R}\beta$ si $\alpha \equiv \beta$. Entonces, $\mathcal{R}$ es de *equivalencia.*

Algunas fórmulas que son equivalentes entre sí son:

1. $\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$
2. $\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$
3. $(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$
4. $(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$

Definición 4. Función booleana

Una función $f : \{0, 1\}^n \to \{0, 1\}$ con $n \in \mathbb{N}_{\geq 1}$ se llama *función booleana.*

Definición 5. Conectivos Adecuados

Sea C un conjunto de conectivos; y sean $F_C = \{$ formulas que tienen conectivos de C $\} \bigcup VAR$.
Entonces:

$$C \textit{ es adecuado } \text{ si } \forall\alpha \in FORM, \exists\beta \in F_C /\, \beta \equiv \alpha$$

3.2 Proposiciones

1. Sea $\alpha \in F$, $(p_1 \to \alpha)$ es tautologia. Entonces,

$$p_1 \notin VAR(\alpha) \Rightarrow \alpha \text{ es tautología}$$

3.3 Teoremas

3.3.1 Teorema 3.A

Dada $f : VAR \to \{0, 1\}$ función. Entonces:

$$\exists!\, v \text{ valuación} : FORM \to \{0, 1\} / \, v \text{ extiende a } f$$

Es decir, $v|_{VAR} = f$

3.3.2 Teorema 3.B

Sea $\alpha \in F$. $Var(\alpha) = \{p_j \in Var/\, p_j \text{ aparece en } \alpha\}$.
v, w son valuaciones tales que $v|_{VAR(\alpha)} = w|_{VAR(\alpha)}$

$$\Rightarrow v(\alpha) = w(\alpha)$$

3.3.3 Teorema 3.C

$$\{f/\, f \text{ es función booleana}\} \to FORM/\equiv$$

es biyectiva.

4 Lógica Proposicional - Parte 1

El alfabeto de la Lógica Proposicional es el que sigue:

$$A = VAR \cup \{(,\,)\} \cup \{\wedge,\, \vee,\, \rightarrow\}$$

donde $VAR = \{p_n /\, n \in \mathbb{N}\}$

4.1 Definiciones Previas

Definición 1. Fórmula

1. $VAR \subseteq FORM = F$
2. $\alpha \in F \;\Rightarrow\; \neg\alpha \in F$
3. $\alpha,\, \beta \in F \;\Rightarrow\; (\alpha \vee \beta),\, (\alpha \wedge \beta),\, (\alpha \rightarrow \beta)$
4. $\alpha \in E^*$ es fórmula si se obtiene aplicando finitas veces 1, 2 y 3.

De esta manera, *las fórmula son el lenguaje de la logica proposicional.*

Definición 2. Cadena de Formación(c.f)

Una sucesion finita $X_1X_2...X_n = \alpha \in F$ de expresiones de A^* es una c.f si:

1. $X_i \in VAR$
2. $\exists j < i /\, X_i = \neg X_j$
3. $\exists j,\, k < i /\, X_i = (X_k \star X_j)$ donde $\star \in \{\wedge,\, \vee,\, \rightarrow\}$

con $1 \leq i,\, j,\, k \leq n$.
Además, cada X_i se llama *eslabón*.

Definición 3. Subcadena

Dada $X_1...X_n = \alpha$ c.f, decimos que $X_{i_1}X_{i_2}...X_{i_k}$ es una subcadena si:

1. Si es c.f
2. $X_{i_k} = X_n = \alpha$
3. $1 \leq i_1 < i_2 < ... < i_k = n$

Observación: Se define $\mathcal{R}$ en el conjunto de c.f de forma tal que $A\mathcal{R}B$ si A es subcadena de B. Entonces, $\mathcal{R}$ es de **orden**.

Definición 4. C.F Minimal

Una c.f es *minimal* si la única subcadena que tiene es ella misma.

Definición 5. Complejidad

Sea $E \in A^*$. Se define la complejidad de E como la cantidad de conectivos que aparecen en E.
Notación: $C(E)$

Definición 6. Complejidad binaria

Sea $E \in A^*$. Se define la complejidad binaria de E como la cantidad de conectivos binarios que aparecen en E.
Notación: $Cb(E)$

Definición 7. Peso

Sea $E \in A^*$. Se define el peso de E como la cantidad de paréntesis que abren menos la cantidad de paréntesis que cierran.
Notación: $p(E)$

Definición 8. Subfórmula

Sea $\alpha \in F$.
Si $C(\alpha) = 0 \Rightarrow \alpha = p_j \in VAR$

$$S(\alpha) = \{\alpha = p_j\}$$

Si $C(\alpha) > 0$. Entonces:

1. $\alpha = \neg\beta$ con $\beta \in F$

$$S(\alpha) = \{\alpha\} \cup S(\beta)$$

2. $\alpha = (\beta_1 \star \beta_2)$ con β_1, $\beta_2 \in F$

$$S(\alpha) = \{\alpha\} \cup S(\beta_1) \cup S(\beta_2)$$

4.2 Proposiciones

1. X_1, ..., X_n es una c.f $\Rightarrow$ X_1, ..., X_j es una c.f con $j < i$
2. $C(\alpha) = 0 \Rightarrow \alpha \in VAR$
3. $C(\alpha) > 0 \Rightarrow \alpha = \neg\beta$ con $\beta \in F$
4. $C(\alpha) > 0 \Rightarrow \alpha = \beta_1 \star \beta_2$ con β_1, $\beta_2 \in F$, $\star \in \{\vee, \wedge, \rightarrow\}$

4.3 Teoremas

4.3.1 Teorema 4.A

$$\alpha \in FORM \Leftrightarrow \exists X_1, ..., X_n = \alpha \text{ es c.f}$$

4.3.2 Teorema 4.B

Sea $\alpha \in F$. Entonces:

1. $p(\alpha) = 0$
2. Si $\bullet$ es un conectivo binario que aparece en α $\Rightarrow$ la expresión E a la izquierda de $\bullet$ en α verifica que $p(E) > 0$

4.3.3 Teorema 4.C

$$\#S(\alpha) \leq long(\text{cf de } \alpha)$$

4.3.4 Unicidad de escritura

Sea $\alpha \in F$, $c(\alpha) > 0$. Entonces:

1. $\exists! \, \beta \in F / \, \alpha = \neg\beta$ ó
2. Existen únicos β_1, $\beta_2 \in F$, $\star \in \{\vee, \wedge, \rightarrow\} / \, \alpha = (\beta_1 \star \beta_2)$

5 Lógica Proposicional - Parte 2

5.1 Definiciones Previas

Definición 1. Satisfacible

1. Sea $\alpha \in F$, v val. Decimos que v *satisface* α si $v(\alpha) = 1$
2. Sea $\alpha \in F$. Decimos que α es satisfacible si $\exists v$ val/ $v(\alpha) = 1$

Definición 2. Consecuencia

Sea $\Gamma \subseteq F$, $\alpha \in F$.
Decimos que α es consecuencia de Γ si:

$$(v(\Gamma) = 1 \;\Rightarrow\; v(\alpha) = 1)\; \forall v \text{ val}$$

o lo que es lo mismo:

$$\alpha \notin C(\Gamma) \text{ si } \exists v \text{ val}/\, v(\Gamma) = 1 \text{ y } v(\alpha) = 0$$

Definición 3. Conjunto de Fórmulas Independiente

Γ es un conjunto de fórmulas independiente si:

$$\forall \alpha \in \Gamma \text{ se tiene que } \alpha \notin C(\Gamma - \{\alpha\})$$

Definición 4. Base

Sea Γ un conjunto. Decimos que Γ *es base* si:

1. Γ es independiente
2. Γ es maximal respecto a la independencia. Es decir:

$$\Sigma \text{ conjunto independiente y } \Gamma \subseteq \Sigma \;\Rightarrow\; \Gamma = \Sigma$$

Definición 5. Conjunto Finitamente Satisfacible

Sea Γ un conjunto.
Decimos que Γ *es finitamente satisfacible* si todo subconjunto finito de Γ es satisfacible.

Notación: f.s

Definición 6. Literal

Se llama *literal* a las variables o a las variables negadas.

5.2 Proposiciones

Sean Γ_1, $\Gamma_2 \subset F$

1. $\Gamma \subset C(\Gamma)$
2. $C(C(\Gamma)) = C(\Gamma)$
3. $C(F) = F$
4. $\Gamma_1 \subset \Gamma_2 \Rightarrow C(\Gamma_1) \subset C(\Gamma_2)$
5. $C(\Gamma_1) \cup C(\Gamma_2) \subset C(\Gamma_1 \cup \Gamma_2)$
6. $C(\Gamma_1 \cap \Gamma_2) \subset C(\Gamma_1) \cap C(\Gamma_2)$
7. Si $C(\Gamma_1) = \Gamma_2$ y $C(\Gamma_2) = \Gamma_1 \Rightarrow \Gamma_1 = \Gamma_2$
8. $C(\Gamma_1 \cup \Gamma_2) \subset C(\Gamma_1) \cup C(\Gamma_2) \Leftrightarrow \Gamma_1 \subset C(\Gamma 2)$ ó $\Gamma_2 \subset C(\Gamma_1)$
9. $C(\Gamma_1 \cap \Gamma_2) = C(\Gamma_1) \cap C(\Gamma_2) \Leftrightarrow \alpha \in C(\Gamma_1)$ y $\beta \in C(\Gamma_2) \rightarrow (\alpha \vee \beta) \in C(\Gamma_1 \cap \Gamma_2)$
10. $\Gamma' \subseteq \Gamma$ y Γ satisfacible $\Rightarrow \Gamma'$ satisfacible

5.3 Teoremas

5.3.1 Teorema 5.A

Sea $\Gamma \subseteq F$, $\alpha \in F$. Entonces:

$$\alpha \in C(\Gamma) \Leftrightarrow \Gamma \cup \{\neg\alpha\} \text{ es insatisfacible}$$

5.3.2 Teorema 5.B

Sea $\Gamma = \{\gamma_1, ..., \gamma_n\} \subseteq F, \alpha \in F$. Entonces:

$$\alpha \in C(\Gamma) \Leftrightarrow ((\gamma_1, ..., \gamma_n) \rightarrow \alpha) \text{ es tautología}$$

5.3.3 Teorema de la Deducción (Versión Semántica)

Sea $\Gamma \subseteq F$, α, $\beta \in F$. Entonces:

$$(\alpha \rightarrow \beta) \in C(\Gamma) \Leftrightarrow \beta \in C(\Gamma \cup \{\alpha\})$$

5.3.4 Lema

Sea Γ f.s., $p_i \in VAR$. Entonces:

$$\Gamma \cup \{p_i\} \text{ es f.s. ó } \Gamma \cup \{\neg p_i\} \text{ es f.s.}$$

5.3.5 Teorema de Compacidad

$$\Gamma \text{ es satisfacible} \Leftrightarrow \Gamma \text{ es f.s.}$$

5.3.6 Equivalencia T. Compacidad

Son equivalentes:

1. Γ es satisfacible $\Leftrightarrow$ Γ es f.s.
2. Γ es insatisfacible $\Leftrightarrow$ $\exists \Gamma_0 \subseteq \Gamma$ finito insatisfacible
3. $\alpha \in C(\Gamma) \Rightarrow \exists \Gamma' \subseteq \Gamma, \Gamma'$ finito/ $\alpha \in C(\Gamma')$

6 Lógica Proposicional - Teoría Axiomática

6.1 Definiciones Previas

Definición 1. Teoría Axiomática

Una "Teoría Axiomática" es un conjunto es axiomas unido a un conjunto de reglas.
De esta manera, se definen los siguientes axiomas:
Sean α, β, $\gamma \in F$

$$\begin{aligned}
&\text{AX 1: } (\alpha \to (\beta \to \alpha)) \\
&\text{AX 2: } ((\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))) \\
&\text{AX 3: } ((\neg\alpha \to \neg\beta) \to ((\neg\alpha \to \beta) \to \alpha))
\end{aligned}$$

Observación: Los axiomas son tautologías.

Definición 2. Regla MODUS PONENS (MP)

$$\begin{array}{l}
(\alpha \to \beta) \\
\alpha \\
---- \\
\beta
\end{array}$$

Definición 3. Sistema Axiomático

Está formado por AX1, AX2, AX3 y la regla MP.

Definición 4. Prueba

Sea $\alpha \in F$.
Una prueba para α es una sucesión finita de fórmulas α_1, α_2, ..., α_k/

1. $\alpha_k = \alpha$
2. α_j es un axioma ó se obtiene aplicando MP a α_i y α_t con i, $t < j$. O lo que es lo mismo:

$$\alpha_t = (\alpha_i \to \alpha_j) \text{ con } 1 \leq j \leq k$$

Definición 5. Demostrable

Sea $\alpha \in F$.
Decimos que α es *demostrable* si existe una prueba de α. En este caso α se llama *teorema.*

Definición 6. Deducible

Sea $\Gamma \subseteq F$, $\alpha \in F$.
Decimos que α *se deduce de* Γ si existe una sucesión finita de fórmulas α_1, α_2, ..., $\alpha_n = \alpha$ que verifica que:

$$\begin{aligned}
&\alpha_i \in \Gamma \text{ ó} \\
&\alpha_i \text{ es un axioma ó} \\
&\alpha_i \text{ se obtiene aplicando MP a } \alpha_j = (\alpha_k \to \alpha_i) \text{ y } \alpha_k \text{ con } j,\, k < i
\end{aligned}$$

Notación: $\Gamma \vdash \alpha$

Definición 7. Consistente

Sea $\Gamma \subseteq F$.
Decimos que Γ *es consistente* si:

$$\nexists \phi \in F / \Gamma \vdash \phi \text{ y } \Gamma \vdash \neg\phi$$

y decimos que Γ *es inconsistente* si:

$$\exists \phi \in F / \Gamma \vdash \phi \text{ y } \Gamma \vdash \neg\phi$$

Definición 8. Maximal Consistente

Sea $\Gamma \subseteq F$.
Decimos que Γ *es maximal consistente* (mc) si:

1. Γ es consistente
2. Se cumple que:
$$\forall \phi \in F : \phi \in \Gamma \text{ ó } \Gamma \cup \{\phi\} \text{ es inconsistente}$$

Definición 9. Sistema Axiomático Consistente

Un sistema axiomático S *es consistente* si:

$$\nexists \phi \in F / \vdash_S \phi \text{ y } \vdash_S \neg\phi$$

6.2 Proposiciones

Sea $\Gamma \subseteq F$

1. Γ satisfacible $\Leftrightarrow$ Γ es consistente
2. $\Gamma \cup \{\neg\phi\}$ es inconsistente $\Leftrightarrow$ $\Gamma \vdash \phi$
3. $\Gamma \cup \{\phi\}$ es inconsistente $\Leftrightarrow$ $\Gamma \vdash \neg\phi$
4. Si $\Gamma \vdash \alpha$. Entonces:
$$\Gamma \vdash \beta \Leftrightarrow \Gamma \cup \{\alpha\} \vdash \beta \text{ con } \alpha, \beta \in F$$
5. Γ es mc $\Rightarrow$ $\phi \in \Gamma \veebar \neg\phi \in \Gamma \quad \forall \phi \in F$
6. Γ mc, entonces: $\Gamma \vdash \phi \Leftrightarrow \phi \in \Gamma$

6.3 Teoremas

6.3.1 Teorema 6.A

Sea $\alpha \in F$. Entonces:

$$\alpha \text{ es demostrable } \Rightarrow \alpha \text{ es tautología}$$

6.3.2 Lema

Sean Γ, Σ, $\rho \subseteq F$

$$\Gamma \vdash \rho \;\Rightarrow\; \Gamma \cup \Sigma \vdash \rho$$

6.3.3 Teorema de la Deducción (Versión Axiomática)

Sea $\Gamma \subseteq F; \alpha,\, \beta \in F$. Entonces:

$$\Gamma \vdash (\alpha \to \beta) \;\Leftrightarrow\; \Gamma \cup \{\alpha\} \vdash \beta$$

6.3.4 Teorema de Correctitud y Completitud

Sea $\Gamma \subseteq F$, $\alpha \in F$. Entonces:
Correctitud:

$$\Gamma \vdash \alpha \;\Rightarrow\; \alpha \in C(\Gamma)$$

Completitud

$$\alpha \in C(\Gamma) \;\Rightarrow\; \Gamma \vdash \alpha$$

De esta manera:

$$\alpha \in C(\Gamma) \;\Leftrightarrow\; \Gamma \vdash \alpha$$

6.3.5 Lema de Lindenbaum

$$\Gamma \text{ es consistente } \Rightarrow\; \exists \Gamma' \text{ mc}/\, \Gamma \subseteq \Gamma'$$

6.3.6 Teorema 6.B

$$\Gamma \text{ satisfacible } \Leftrightarrow\; \Gamma \text{ es consistente}$$

6.3.7 Teorema 6.C

El sistema axiomático de la Teoría Axiomática que vimos es consistente.

7 Lógica de 1er Orden - Sintaxis

7.1 Definiciones Previas

Definición 1. Alfabeto de la Lógica de 1er Orden

El alfabeto de la Lógica de 1er orden es:

$$A = VAR \cup \text{CONECTIVOS} \cup \{(,)\} \cup \mathcal{F} \cup \mathcal{C} \cup \mathcal{P}$$

donde:

1. $VAR = \{x_1, x_2, ..., x_k\}$.
   Notación: x_i, y_i, z_i, ...

2. CONECTIVOS $= \{\vee, \wedge, \rightarrow, \neg\} \cup \{ \underbrace{\forall}_{\substack{\text{cuantificador}\\\text{universal}}}, \underbrace{\exists}_{\substack{\text{cuantificador}\\\text{existencial}}} \}$

3. $\mathcal{F}$ = es un conjunto cuyos elementos se llaman *símbolos de función.*
   Notación: f_i^k, g_i^k, h_i^k, ... donde k es la *ariedad.*

4. $\mathcal{C}$ = es un conjunto cuyos elementos se llaman *símbolos de constantes.*
   Notación: c_i, d_i, k_i, ...

5. $\mathcal{P}$ = es un conjunto cuyos elementos se llaman *símbolos de predicado.* $\mathcal{P} \neq \varnothing$.
   Notación: $\mathcal{P} = P_0^{k_0}, P_1^{k_1}, ..., P_k^{k_k}$

Definición 2. Término

Definimos un *término a partir de un alfabeto* A de la siguiente manera:

1. Toda variable es un término.

2. Toda constante es un término.

3. Si t_1, t_2, ..., t_k son términos y $f^k \in \mathcal{F}$. Entonces:

   $$f^k(t_1, t_2, ..., t_k) \text{ es un término}$$

4. Cualquier expresión de A^* que se obtiene aplicando finitas veces 1, 2 y 3 es un término.

Notación: $TERM$

Definición 3. Fórmula

Definimos una *fórmula a partir de un alfabeto* A de la siguiente manera:

1. t_1, t_2, ..., $t_k \in TERM$ y $P^k \in \mathcal{P}$. Entonces:

   $$P^k(t_1, t_2, ..., t_k) \text{ es una fórmula y se llama } \textbf{fórmula atómica}$$

2. $\alpha \in Form \Rightarrow \neg\alpha \in Form$

3. $\alpha, \beta \in Form \Rightarrow (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta) \in Form$

4. $\alpha \in Form$, $x \in VAR \Rightarrow \forall x \ \alpha$ es una fórmula.
5. $\alpha \in Form$, $x \in VAR \Rightarrow \exists x \ \alpha$ es una fórmula.
6. Cualquier expresión de A^* que se obtiene aplicando finitas veces 1, 2, 3, 4 y 5 es una fórmula.

Notación: $Form = F$
Observación: En lenguajes de 1er Orden, *sólo podemos cuantificar variables.*

Definición 4. Término cerrado

Un término se llama *cerrado* si no tiene variables.

Definición 5. Variables libres ligadas

1. Una aparición de una variable x en una fórmula está *ligada* si es alcanzada por un cuantificador.
2. Una variable es *libre* en una fórmula si todas sus apariciones son libres.
3. Una variable es *ligada* en una fórmula, si todas sus apariciones son ligadas.

Definición 6. Sentencia/Enunciado

Una fórmula se llama *sentencia/enunciado* si todas sus variables están ligadas.

8 Lógica de 1er Orden - Semántica

8.1 Definiciones Previas

Definición 1. Interpretación

Dado un alfabeto A y un lenguaje $\mathcal{L}$ de 1er Orden, una *interpretación* $\mathcal{I}$ *de un leguaje* $\mathcal{L}$ consiste en:

1. $\mathcal{U} \neq \varnothing$; $\mathcal{U}$ conjunto que se llama *universo*.
2. $c \in \mathcal{C} \Rightarrow c$ se interpreta como $c_I \in \mathcal{U}$.
3. $f^k \in \mathcal{F} \Rightarrow f^k$ se interpreta como una función cuyo dominio es $\mathcal{U}^k$ y el codominio es $\mathcal{U}$.
4. $P^k \in \mathcal{P} \Rightarrow P^k$ se interpreta como una relación k-aria en $\mathcal{U}$, es decir $P^k_{\mathcal{I}} \in \mathcal{U}^k$.

Definición 2. Valuación

Dado un lenguaje de 1er Orden y una interpretación $\mathcal{I}$ de $\mathcal{L}$.
Llamamos *valuación* a una función:

$$v : VAR \rightarrow \mathcal{U}$$

¿Cómo interpretamos los términos?
Sea $\bar{v} : TERM \rightarrow \mathcal{U}_{\mathcal{I}}$ una extensión de v que verifica:

1. $\bar{v}(x) = v(x)$ con $x \in VAR$
2. $\bar{v}(c) = \mathcal{C}_{\mathcal{I}}$ con $c \in \mathcal{C}$
3. $\bar{v}(f^k(t_1, ..., t_k)) = f^k_{\mathcal{I}}(\bar{v}(t_1), \bar{v}(t_k))$ donde $f^k \in \mathcal{F}$; $t_1, ..., t_k \in TERM$

Definición 3. Valuación Modificada

$$v_{x_i=a} : VAR \rightarrow \mathcal{U}_{\mathcal{I}} /\, v_{x_i=a}(x) = \begin{cases} v(x) & \text{si } x \neq x_i \\ a & \text{si } x = x_i \end{cases}$$

Definición 4. Valor de verdad

Dado $\mathcal{L}$ de 1er Orden, $\mathcal{I}$ de $\mathcal{L}$.
Asignamos el valor de verdad a las fórmulas de la siguiente manera:
Sean β, β_1, $\beta_2 \in FORM$; $x \in VAR$

1. $\alpha = P^k(t_1, ..., t_k)$ con $P^k \in \mathcal{P}$, $t_j \in TERM$, $1 \leq j \leq k$. Entonces:

$$V_{\mathcal{I},v} : Form(\mathcal{L}) \rightarrow \{0, 1\} / \quad V_{\mathcal{I},v}(\alpha) = 1 \Leftrightarrow ((\bar{v}(t_1), ..., \bar{v}(t_k)) \in P^k_{\mathcal{I}}$$

2. $\alpha = \neg\beta$. Entonces:

$$V_{\mathcal{I},v}(\alpha) = 1 - V_{\mathcal{I},v}(\beta)$$

3. $\alpha = (\beta_1 \wedge \beta_2)$. Entonces:

$$V_{\mathcal{I},v}(\alpha) = min\{V_{\mathcal{I},v}(\beta_1), V_{\mathcal{I},v}(\beta_2)\}$$

4. $\alpha = (\beta_1 \vee \beta_2)$. Entonces:
$$V_{\mathcal{I},\,v}(\alpha) = max\{V_{\mathcal{I},\,v}(\beta_1),\, V_{\mathcal{I},\,v}(\beta_2)\}$$
5. $\alpha = (\beta_1 \rightarrow \beta_2)$. Entonces:
$$V_{\mathcal{I},\,v}(\alpha) = max\{1 - V_{\mathcal{I},\,v}(\beta_1),\, V_{\mathcal{I},\,v}(\beta_2)\}$$
6. $\alpha = \forall x\, \beta$. Entonces:
$$V_{\mathcal{I},\,v}(\alpha) \;\Leftrightarrow\; V_{\mathcal{I},\,v_{x=a}}(\beta) = 1 \text{ para todo elemento } a \in \mathcal{U}_{\mathcal{I}}$$
7. $\alpha = \exists x\, \beta$. Entonces:
$$V_{\mathcal{I},\,v}(\alpha) \;\Leftrightarrow\; V_{\mathcal{I},\,v_{x=a}}(\beta) = 1 \text{ para algun } a \in \mathcal{U}_{\mathcal{I}}$$

Además, hay algunas equivalencias que pueden sernos útiles:

1. Todos los A son B: $\forall x(A(x) \rightarrow B(x))$
2. Algunos A son B: $\exists x(A(x) \wedge B(x))$
3. Ningún A es B: $\forall x(A(x) \rightarrow \neg B(x)$
4. $\exists x\, \alpha \equiv \neg\forall x\, \neg\alpha$
5. $\forall x\, \alpha \equiv \neg\exists x\, \neg\alpha$

Observación: Si todas las variables estan ligadas (*enunciado*) entonces no terminamos usando las valuaciones.

Definición 5. Fórmula Satisfacible

Sea $\mathcal{L}$ de 1er orden.
Decimos que α es *satisfacible* si $\exists\, \mathcal{I}$ interpretación, v valuación/
$$V_{\mathcal{I},\,v}(\alpha) = 1$$
Notación: $\mathcal{I} \vDash \alpha[v]$

Definicion 6. Fórmula Verdadera/Válida

Sea $\mathcal{L}$ de 1er orden.
Decimos que α es *verdadera o válida* en una interpretación $\mathcal{I}$ si:
$$V_{\mathcal{I},\,v}(\alpha) = 1 \;\; \forall v : VAR \rightarrow \mathcal{U}_{\mathcal{I}}$$
Notación/Nombre: $\mathcal{I}$ *es un modelo para* α

Definición 7. Fórmula Universalmente Válida

Sea $\mathcal{L}$ de 1er Orden.
Decimos que α es *universalmente válida* si:
$$V_{\mathcal{I},\,v}(\alpha) = 1 \;\; \forall v \text{ valuación y } \forall \mathcal{I} \text{ interpretación}$$
Notación: $\vDash \alpha$

Definición 8. Lenguaje con Igualdad

$\mathcal{L}$ es un *lenguaje con igualdad* si tiene un símbolo de predicado binario que obligatoriamente se interpreta con la $=$ (igualdad).

9 Lógica de 1er Orden

9.1 Definiciones Previas

Definición 1. Conjunto Expresable

Sea $\mathcal{L}$ un lenguaje de 1er Orden. Sea $\mathcal{I}$ una interpretación de $\mathcal{L}$ con universo $\mathcal{U}$.
Sea $A \subseteq U$. Decimos que A *es expresable* si $\exists \alpha \in FORM$ con una única variable libre y todas las demas variables ligadas (<u>Notación</u>: $\alpha(x)$, donde x es la variable libre), tal que:

$$V_{\mathcal{I},\, v_{x=a}} \alpha(x) = 1 \;\Leftrightarrow\; x \in A$$

o lo que es lo mismo:

1. $A = \{a \in \mathcal{U} / \, V_{\mathcal{I},\, v_{x=a}} \alpha(x) = 1$
2. Si $x \in A \;\Rightarrow\; V_{\mathcal{I},\, v_{x=a}} \alpha(x) = 1$
   Si $x \notin A \;\Rightarrow\; V_{\mathcal{I},\, v_{x=a}} \alpha(x) = 0$

Definición 2. Elementos distinguibles

Sea $\mathcal{L}$ un lenguaje de 1er Orden. Sea $\mathcal{I}$ una interpretación de $\mathcal{L}$ con universo $\mathcal{U}$.
Sea $a \in \mathcal{U}$, decimos que *a es distinguible* si $\{a\}$ es expresable.

Definición 3. Isomorfismo

Sea $\mathcal{L}$ un lenguaje de 1er orden. Sean $\mathcal{I}_1$ e $\mathcal{I}_2$ interpretaciones de $\mathcal{L}$ con universos $\mathcal{U}_1$ y $\mathcal{U}_2$ respectivamente.
Una función $F : \mathcal{U}_1 \to \mathcal{U}_2$ se llama isomorfismo si:

1. F es biyectiva
2. $c \in C$ y $c_{\mathcal{I}_1} \in \mathcal{U}_1$ y $c_{\mathcal{I}_2} \in \mathcal{U}_2$ sus respectivas interpretaciones
$$F(c_{\mathcal{I}_1}) = c_{\mathcal{I}_2}$$
3. $f^k \in \mathcal{F}$. Entonces:
$$F(f^k_{\mathcal{I}_1}(u_1,\, ...,\, u_k)) = f^k_{\mathcal{I}_2}(F(u_1),\, ...,\, F(u_k))$$
4. $P^k \in \mathcal{P}$. Entonces:
$$(u_1,\, ...,\, u_k) \in P^k_{\mathcal{I}_1} \;\Leftrightarrow\; (F(u_1),\, ...,\, F(u_k)) \in P^k_{\mathcal{I}_2}$$

<u>Notación</u>: $\mathcal{I}_1 \approx \mathcal{I}_2$ ó $\mathcal{I}_1 \approx_F \mathcal{I}_2$
<u>Observación</u>: Si $\#\mathcal{U}_1 \neq \#\mathcal{U}_2$ entonces no puede haber ismorfismo.

9.2 Proposiciones

Sea $\mathcal{L}$ un lenguaje de 1er Orden. Sea $\mathcal{I}$ una interpretación de $\mathcal{L}$ con universo $\mathcal{U}$.
Entonces:

1. El conjunto $\varnothing$ y el universo $\mathcal{U}_{\mathcal{I}}$ son expresables
2. Si $A \subset \mathcal{U}_{\mathcal{I}}$ es expresable $\Rightarrow$ $\mathcal{U}_{\mathcal{I}} - A$ es expresable
3. Sean $A, B \subset \mathcal{U}_{\mathcal{I}}$ expresables $\Rightarrow$ $A \cup B$, $A \cap B$, $A \setminus B$, $A \triangle B$ es expresable

9.3 Teoremas

9.3.1 Lema

Sea $\mathcal{L}$ un lenguaje de 1er Orden. Sean $\mathcal{I}_1$ e $\mathcal{I}_2$ interpretaciones de $\mathcal{L}$. Sea h un isomorfismo de $\mathcal{I}_1$ a $\mathcal{I}_2$. Sea v una valuación en $\mathcal{I}_1$. Entonces:

$$\overline{h \circ v} = h \circ \overline{v}$$

9.3.2 Teorema 9.A

Sea $\mathcal{L}$ un lenguaje de 1er Orden. Sean $\mathcal{I}_1$ e $\mathcal{I}_2$ interpretaciones de $\mathcal{L}$. Sea h un isomorfismo de $\mathcal{I}_1$ a $\mathcal{I}_2$. Sea v una valuación en $\mathcal{I}_1$. Entonces:

$$\mathcal{I}_1 \vDash \alpha[v] \;\Leftrightarrow\; \mathcal{I}_2 \vDash \alpha[h \circ v]$$

9.3.3 Corolario A.1

Sea $\mathcal{L}$ un lenguaje de 1er Orden. Sean $\mathcal{I}_1$ e $\mathcal{I}_2$ interpretaciones de $\mathcal{L}$, $\mathcal{I}_1 \approx_h \mathcal{I}_2$. Y sea α un enunciado de $\mathcal{L}$. Entonces:

$$\mathcal{I}_1 \text{ es un modelo de } \alpha \;\Leftrightarrow\; \mathcal{I}_2 \text{ es un modelo de } \alpha$$

o lo que es lo mismo:

$$\mathcal{I}_1 \vDash \alpha[v] \;\forall v \text{ val en } \mathcal{I}_1 \;\Leftrightarrow\; \mathcal{I}_2 \vDash \alpha[w] \;\forall w \text{ val en } \mathcal{I}_2$$

Apunte: Esto nos puede servir para demostrar que $\mathcal{I}_1 \not\approx \mathcal{I}_2$.

9.3.4 Corolario A.2

Sea $\mathcal{L}$ de 1er Orden. $\mathcal{I}$ interpretación de $\mathcal{L}$ con universo $\mathcal{U}$. $F : \mathcal{U} \to \mathcal{U}$. Entonces:

$$a \in \mathcal{U} \text{ distinguible } \;\Rightarrow\; F(a) = a$$

Apunte: Esto nos puede servir para demostrar que un elemento no es distinguible.

9.3.5 Corolario A.3

Sea $\mathcal{L}$ un lenguaje de 1er Orden y sea $\mathcal{I}$ una interpretación de $\mathcal{L}$ con universo $\mathcal{U}$. Sea $F : \mathcal{U} \to \mathcal{U}$ un isomorfismo. Entonces:

$$a \text{ es expresable } \;\Rightarrow\; F(A) \subseteq A \;\forall A \subseteq \mathcal{U}$$

9.3.6 Corolario A.4

Sea $\mathcal{L}$ un lenguaje de 1er Orden. Sean $\mathcal{I}_1$ e $\mathcal{I}_2$ interpretaciones de $\mathcal{L}$, $\mathcal{I}_1 \approx \mathcal{I}_2$. Entonces:

$$\text{Si } a \in \mathcal{U}_1 \text{ es distinguible en } \mathcal{I}_1 \;\Rightarrow\; h(a) \in \mathcal{U}_2 \text{ es distinguible en } \mathcal{I}_2$$

Apunte: Esto nos puede servir para armar un isomorfismo.

10 Lenguaje S

10.1 Definiciones Previas

Definición 1. Variables

Hay 3 tipos de variables:

1. Variables de entrada: X_1, X_2, ...
2. Variable de salida: Y
3. Variables auxiliares: Z_1, Z_2, ...

Tanto la *variable de salida* como las *variables auxiliares* están **inicializadas en 0**.
El tipo de datos para todas ellas son los $\mathbb{N}$ (no hay que declararlo en ningún lugar).

Definición 2. Etiquetas

Las etiquetas son:

$$A_1,\ B_1,\ C_1,\ D_1,\ E_1,\ A_2,\ B_2,\ C_2,\ D_2,\ E_2,\ A_3,\ ...$$

Definición 3. Instrucciones

Hay 3 tipos:

1. $V \leftarrow V + 1$
2. $V \leftarrow V - 1$
3. $IF\ V \neq 0\ GOTO\ L$ donde $[L]$ es una etiqueta.

Dado que el dominio son los numeros naturales, si al ejecutar $V \leftarrow V - 1$, V estaba en 0, entonces sigue valiendo 0. Es decir: $0 - 1 = 0$

Definición 4. Programa

Es una lista finita de instrucciones I_1, I_2, ..., I_n que se escribe una debajo de la otra.

Definición 5. Macro

Una *Macro* es una pseudo-instrucción que representa un segmento de programa.
Cada vez que en un programa P aparece una macro, hay que reemplazarla por el segmento de programa que representa. Dicho segmento de programa se denomina *expansión de la macro.*

Definición 6. Estado

Un *estado de un programa* P es una lista finita de igualdades de la forma $V = m$, donde V es una varaiable y $m \in \mathbb{N}$.
Hay una única igualdad para cada variable que aparece en P.

Definición 7. Descripción Instantánea (Snapshot/Foto)

Supongamos que un programa P tiene longitud n, es decir tiene n instrucciones I_1, I_2, ..., I_n.
Para un estado σ de P y un $i \in \{1, 2, ..., n, n+1\}$ tenemos que el par (i, σ) es una foto de P, la cual se llama *terminal* si $i = n + 1$.
De esta manera, i dice a que instrucción apunta antes de ser ejecutada. Ademas, en σ estan los valores de las variables antes de ejecutas la instrucción i.
Dada una foto (i, σ) no terminal, se define la foto *sucesora* (j, τ) de la sigueinte manera:

1. Si la i-ésima instrucción es $V \leftarrow V + 1$

$$\Rightarrow\ j = i+1,\ \tau = \sigma \text{ salvo que si } \underbrace{V = m}_{\text{en } \sigma} \Rightarrow \underbrace{V = m+1}_{\text{en } \tau}$$

2. Si la i-ésima instrucción es $V \leftarrow V - 1$

$$\Rightarrow\ j = i+1,\ \tau = \sigma \text{ salvo que si } \begin{cases} \underbrace{V = m > 0}_{\text{en } \sigma} & \Rightarrow \underbrace{V = m-1}_{\text{en } \tau} \\ \underbrace{V = 0}_{\text{en } \sigma} & \Rightarrow \underbrace{V = 0}_{\text{en } \tau} \end{cases}$$

3. Si la i-ésima instrucción es $IF\ V \neq 0\ GOTO\ L$

$$\Rightarrow\ \tau = \sigma,\ j = \begin{cases} i+1 & \text{si } V = 0 \text{ en } \sigma \\ k & \text{si } V \neq 0 \text{ en } \sigma \end{cases}$$

$$\text{siendo } k = \begin{cases} min\{t/\ I_t \text{ esta etiquetado con L}\} & \text{si } A \neq \varnothing \\ n+1 & \text{si ninguna instrucción esta etiquetada con } L \end{cases}$$

Definición 8. Cómputo

Un cómputo de un programa P a partir de una foto $d_1 = (i, \sigma)$ es una lista finita d_1, d_2, ..., d_k de fotos siendo d_{j+1} la foto sucesora de d_j y d_k es la foto terminal.

Definicion 9. Estado Inicial

Sea P un programa y u_1, u_2, ..., u_m números naturales. El *estado inicial* de P para dichos valores es $\sigma = \{X_1 = u_1,\ X_2 = u_2,\ ...,\ X_m = u_m,\ \underbrace{X_j = 0}_{\text{si aparece}}(j > m),\ \underbrace{Z_j = 0}_{\text{si aparece}},\ Y = 0\}$
La descripción inicial es $d_1 = (1, \sigma)$.

Definición 9. Cómputo a partir de un estado inicial

Sea P un programa u_1, ..., u_m de números naturales y σ_1 el estado inicial para ellos.
Existen dos posibilidades:

1. *"El Programa termina ante dichas entradas"*:
   Existe un cómputo de P a partir de $d_1 = (1, \sigma_1)$, es decir existe d_1, ..., d_k terminal:

$$\Psi_p^m(u_1, ..., u_m) = \text{ el valor de } Y \text{ en } d_k$$

2. *"El Programa no termina"*
   No existe un cómputo de P a partir de $d_1 = (1, \sigma_1)$

$$\Psi_p^m(u_1, ..., u_m) = \uparrow$$

Definición 10. Función computable

1. Una función $f : \mathbb{N}^k \to \mathbb{N}$ es *parcialmente computable* si existe un programa P tal que $f = \Psi_p^k$

$$\begin{aligned}
&\text{si } x = (x_1, ..., x_k) \in \mathbb{N}^k \\
&\text{si } x \in Dom(f) \ \Rightarrow \ f(x_1, ..., x_k) = \Psi_m^k(x_1, ..., x_k) \\
&\text{si } x \notin Dom(f) \ \Rightarrow \ \Psi_m^k(x_1, ..., x_k) = \uparrow
\end{aligned}$$

2. Una función $f : \mathbb{N}^k \to \mathbb{N}$ es *computable* si es parcialmente computable y *total* (osea $Dom(f) = \mathbb{N}^k$)

10.2 Macros útiles

10.2.1 Salto Incondicional

La macro es: *GOTO L*
Y la expasión de la macro es la siguiente:

$$\begin{aligned}
&V \leftarrow V + 1 \\
&IF\ V \neq 0\ GOTO\ L
\end{aligned}$$

10.2.2 Asignación de Cero

La macro es: $V \leftarrow 0$
Y la expansión de la macro es la siguiente:

$$\begin{aligned}
[L]\ &V \leftarrow V - 1 \\
&IF\ V \neq 0\ GOTO\ L
\end{aligned}$$

10.2.3 Asignación de Variables

La macro es: $Y \leftarrow X_1$
Y la expansión de la macro es la siguiente:

$$
\begin{array}{rl}
 & Y \leftarrow 0 \\
 & Z_1 \leftarrow 0 \\
[A_1] & IF\ X_1 \neq 0\ GOTO\ B_1 \\
 & GOTO\ C_1 \\
 & X_1 \leftarrow X_1 - 1 \\
 & Y \leftarrow Y + 1 \\
 & Z_1 \leftarrow Z_1 + 1 \\
 & GOTO\ A_1 \\
[C_1] & IF\ Z_1 \neq 0\ GOTO\ D_1 \\
 & GOTO\ E_1 \\
[D_1] & Z_1 \leftarrow Z_1 - 1 \\
 & X_1 \leftarrow X_1 + 1 \\
 & GOTO\ C_1
\end{array}
$$

11 Funciones Recursivas Primitivas

11.1 Definiciones Previas

Definición 1. Esquema Recursivo tipo I (ERI)

Sea $h : \mathbb{N} \to \mathbb{N}$ y $g : \mathbb{N}^2 \to \mathbb{N}$.
Decimos que *h se obtiene a partir de g por un ERI* si se puede escribir de la siguiente manera:

$$\begin{aligned} &h(0) = k \in \mathbb{N} \\ &h(n+1) = g(n,\, h(n)) \end{aligned}$$

Definición 2. Esquema Recursivo tipo II (ERII)

Sea $h : \mathbb{N}^{n+1} \to \mathbb{N}$, $g : \mathbb{N}^{n+2} \to \mathbb{N}$, $q : \mathbb{N}^n \to \mathbb{N}$.
Decimos que *h se obtiene por ERII a partir de g y q* si puede escribirse:

$$\begin{aligned} &h(x_1,\, x_2,\, ...,\, x_n,\, 0) = q(x_1,\, ...,\, x_n) \\ &h(x_1,\, x_2,\, ...,\, x_n,\, y+1) = g(x_1,\, x_2,\, ...,\, x_n,\, y,\, h(x_1,\, x_2,\, ...,\, x_n,\, y)) \end{aligned}$$

Definición 3. Funciones Iniciales

Las siguientes funciones se denominan *funciones iniciales*

1. Cero. $\mathbb{N} \to \mathbb{N} /\, CERO(x) = 0$
2. Sucesor. $\mathbb{N} \to \mathbb{N} /\, SUC(x) = x + 1$
3. Proyección. $\mathbb{N}^n \to \mathbb{N} /\, \pi^n_j(x_1,\, ...,\, x_n) = x_j$ con $1 \leq j \leq n$

Observación: Las funciones iniciales son computables

Definición 4. Recursiva Primitiva (RP)

Una función es RP si es inicial o se obtiene aplicando finitas "operaciones válidas" a las funciones inicales.
Las operaciones válidas son: composición, ERI y ERII.

Definición 5. Predicados RP

Un predicado P^k de k variables es una relación k-aria de números naturales.
Es decir, se asocia a P^k una función denominada *función característica* que se define como sigue:

$$C_{P^k} : \mathbb{N}^k \to \{0,\, 1\} /\, C_{P^k} = \begin{cases} 1 & \text{si } \vec{x} \in P^k \\ 0 & \text{si } \vec{x} \notin P^k \end{cases}$$

Notación: $C_{P^k} = X_{P^k}$
Decimos "P^k *es RP(computable)*" si C_{P^k} es RP(computable)

11.2 Funciones que son RP

1. fact. $\mathbb{N} \to \mathbb{N} / \, fact(x) = x!$
2. Suma. $\mathbb{N}^2 \to \mathbb{N} / \, SUMA(x,\, y) = x + y$
3. Constante. $\mathbb{N} \to \mathbb{N} / \, h_k(x) = k$ con $k \in \mathbb{N}$
4. Producto. $\mathbb{N}^2 \to \mathbb{N} / \, PROD(x,\, y) = x.y$
5. Potencia. $\mathbb{N}^2 \to \mathbb{N} / \, POT(x,\, y) = (x+1)^y$
6. Predecesor. $\mathbb{N} \to \mathbb{N} / \, PRED(x) = \begin{cases} 0 & x = 0 \\ x-1 & x \neq 0 \end{cases}$
7. Resta Truncada. $\mathbb{N}^2 \to \mathbb{N} / \, \dot{-}(x,\, y) = x \dot{-} y = \begin{cases} x-y & x \geq y \\ 0 & x < y \end{cases}$
8. Distancia. $\mathbb{N}^2 \to \mathbb{N} / \, DIST(x,\, y) = |x-y|$
9. Alpha. $\mathbb{N} \to \mathbb{N} / \, \alpha(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$
10. Igualdad. $\mathbb{N}^2 \to \mathbb{N} / \, EQ(x,\, y) = \begin{cases} 0 & x \neq y \\ 1 & x = y \end{cases}$
11. Máximo. $\mathbb{N}^2 \to \mathbb{N} / \, MAX(x,\, y) = \begin{cases} x & x \geq y \\ y & x < y \end{cases}$
12. Impar. $\mathbb{N} \to \mathbb{N} / \, IMPAR(x) = \begin{cases} 0 & \text{si } x \text{ es par} \\ 1 & \text{si } x \text{ es impar} \end{cases}$
13. Mitad. $\mathbb{N} \to \mathbb{N} / \, MITAD(x) = \begin{cases} \frac{x}{2} & \text{si } x \text{ es par} \\ \frac{x-1}{2} & \text{si } x \text{ es impar} \end{cases}$
14. Composición de f n veces. $\mathbb{N}^2 \to \mathbb{N} / \, I_f(n,\, x) = f^n(x)$
15. MCD. $\mathbb{N}^2 \to \mathbb{N} / \, MCD(x, y) =$ MCD entre x e y (máximo común divisor)
16. MCM. $\mathbb{N}^2 \to \mathbb{N} / \, MCM(x, y) =$ MCM entre x e y (mínimo común múltiplo)

11.3 Teoremas

11.3.1 Teorema 11.A

Sea $f : \mathbb{N}^k \to \mathbb{N}$; $g_1,\, g_2,\, ...,\, g_k : \mathbb{N}^m \to \mathbb{N}$; $h : \mathbb{N}^m \to \mathbb{N}$ tal que:

$$h(x_1,\, ...,\, x_m) = f(g_1(x_1,\, ...,\, x_m),\, g_2(x_1,\, ...,\, x_m),\, ...,\, g_k(x_1,\, ...,\, x_m))$$
$$\Leftrightarrow\ h = f \circ (g_1 \times g_2 \times ... \times g_k)$$

Entonces:

Si $f,\, g_1,\, ...,\, g_k$ son (parcialmente) computables $\Rightarrow$ h es (parcialmente) computable

11.3.2 Teorema 11.B

Si h se obtiene a partir de g por un ERI y g es computable $\Rightarrow$ h es computable

11.3.3 Teorema 11.C

Sea h una función que se obtiene por ERII a partir de g y q. Entonces:

$$\text{Si } g \text{ y } q \text{ son computables} \Rightarrow h \text{ es computable}$$

11.3.4 Teorema 11.D

$$\text{Si } f : \mathbb{N}^k \to \mathbb{N} \text{ es RP} \Rightarrow f \text{ es computable}$$

Observaciones:

1. Si f no es total $\Rightarrow$ f no es RP
2. Existen funciones computables que no son RP

11.3.5 Teorema 11.E

$$f \text{ es composición de funciones RP} \Rightarrow f \text{ es RP}$$

11.3.6 Teorema 11.F

1. P^k y Q^k predicados RP(computables) $\Rightarrow$ $(P \cap Q)$ y $\neg P$ son predicados RP(computables)
2. P^k y Q^k predicados RP(computables) $\Rightarrow$ $(P \cup Q)$ y $(P \to Q)$ son predicados RP(computables)

11.3.7 Teorema 11.G

Sean h, $g : \mathbb{N}^k \to \mathbb{N}$ funciones RP(computables) y sea P^n un predicado RP(computable) y sea $f : \mathbb{N}^n \to \mathbb{N}/$

$$f(\vec{x}) = \begin{cases} h(\vec{x}) & \text{si } \vec{x} \in P \\ g(\vec{x}) & \text{si } \vec{x} \notin P \end{cases}$$

Entonces f es RP(computable)

11.3.8 Teorema 11.H

Sean g_1, ..., g_m, $h : \mathbb{N}^n \to \mathbb{N}$ funciones RP(computables). P_1, ..., P_m predicados n-arios RP(computables). $P_i \cap P_j = \varnothing$ si $i \neq j$. Sea $f : \mathbb{N}^n \to \mathbb{N}/$

$$f(\vec{x}) = \begin{cases} g_1(\vec{x}) & \text{si } \vec{x} \in P_1 \\ g_2(\vec{x}) & \text{si } \vec{x} \in P_2 \\ & \vdots \\ g_m(\vec{x}) & \text{si } \vec{x} \in P_m \\ h(\vec{x}) & \text{sino} \end{cases}$$

Entonces f es RP(computable)

11.3.9 Suma y Productoria Acotadas

Sea $f : \mathbb{N}^{n+1} \to \mathbb{N}$ es RP(computable). Sean SA_f, $PA_f : \mathbb{N}^{n+1} \to \mathbb{N}/$

$$SA_f(\vec{x}, y) = \sum_{k=0}^{y} f(\vec{x}, k)$$
$$PA_f(\vec{x}, y) = \prod_{k=0}^{y} f(\vec{x}, k)$$

siendo $\vec{x} = (x_1, ..., x_n)$ con $n \in \mathbb{N}$.

$$\text{Si } n = 0,\ SA_f(y) = \sum_{k=0}^{y} f(k) \text{ y } PA_f(y) = \prod_{k=0}^{y} f(k) \ \Rightarrow\ SA_f \text{ y } PA_f \text{ son RP(computables)}$$

11.3.10 Cuantificadores acotados con $\leq$

Sea P^{k+1} un predicado RP(computable). Dados,

1. Existencial Acotado:

   $$EA_P : \mathbb{N}^{k+1} \to \{0, 1\}/\ EA_P(\vec{x}, y) = \exists t \leq y\ \ C_P(\vec{x}, y)$$

   donde $\vec{x} = (x_1, ..., x_k)$.
   Este predicado es verdadero $\Leftrightarrow$ $C_P(\vec{x}, 0) = 1$ o $C_P(\vec{x}, 1) = 1$ o ... o $C_P(\vec{x}, y) = 1$

2. Universal Acotado:

   $$UA_P : \mathbb{N}^{k+1} \to \{0, 1\}/\ UA_P(\vec{x}, y) = \forall t \leq y\ \ C_P(\vec{x}, y)$$

   donde $\vec{x} = (x_1, ..., x_k)$.
   Este predicado es verdadero $\Leftrightarrow$ $C_P(\vec{x}, 0) = 1$ y $C_P(\vec{x}, 1) = 1$ y ... y $C_P(\vec{x}, y) = 1$

Entonces, EA_P es RP(computable) y UA_P es RP(computable)

11.3.11 Cuantificadores acotados con $<$

Sea P^{k+1} un predicado RP(computable). Dados,

1. Existencial Acotado Estricto:

   $$EAE_P : \mathbb{N}^{k+1} \to \{0, 1\}/\ EAE_P(\vec{x}, y) = \exists t < y\ \ C_P(\vec{x}, y)$$

   donde $\vec{x} = (x_1, ..., x_k)$.
   Este predicado es verdadero $\Leftrightarrow$ $C_P(\vec{x}, 0) = 1$ o $C_P(\vec{x}, 1) = 1$ o ... o $C_P(\vec{x}, y) = 1$

2. Universal Acotado Estricto:

   $$UAE_P : \mathbb{N}^{k+1} \to \{0, 1\}/\ UAE_P(\vec{x}, y) = \forall t < y\ \ C_P(\vec{x}, y)$$

   donde $\vec{x} = (x_1, ..., x_k)$.
   Este predicado es verdadero $\Leftrightarrow$ $C_P(\vec{x}, 0) = 1$ y $C_P(\vec{x}, 1) = 1$ y ... y $C_P(\vec{x}, y) = 1$

Entonces, EAE_P es RP(computable) y UAE_P es RP(computable)

11.3.12 Minimización Acotada

Sea P^{k+1} un predicado RP(computable), y sea

$$MA_P : \mathbb{N}^{n+1} \to \mathbb{N} / \, MA_P(\vec{x},\, y) = \begin{cases} min_{t \leq y} \;\; C_P(\vec{x},\, t) & \text{si existe } t \leq y / \, C_P(\vec{x},\, y) = 1 \\ 0 & \text{sino} \end{cases}$$

Entonces, MA_P es RP(computable)

11.3.13 Minimización No Acotada

Sea P^{k+1} un predicado RP(computable), y sea

$$H(\vec{x}) = min_t \;\; C_P(\vec{x},\, t) = \begin{cases} min\{t \in \mathbb{N} / \, C_P(\vec{x},\, t) = 1\} & \text{si } A \neq \varnothing \\ \uparrow & \text{si } A = \varnothing \end{cases}$$

es parcialmente computable

12 Funciones Computables

12.1 Definiciones Previas

Definición 1. Función par

Sea $< , >: \mathbb{N}^2 \to \mathbb{N}/$

$$< x, y >= 2^x(2y+1) - 1$$

se llama *función par.*

Definición 2. Funciones *left* y *right*

Sean

$$l : \mathbb{N} \to \mathbb{N}/\, l(z) = x \text{ tal que } z =< x, y >$$
$$r : \mathbb{N} \to \mathbb{N}/\, r(z) = y \text{ tal que } z =< x, y >$$

l se llama *función left* y r se llama *función right.*

Definición 3. Numeración de Gödel

Sea $k \in \mathbb{N}$.
Para cada k se define una función

$$[, ...,] : \mathbb{N}^{k+1} \to \mathbb{N}/\, [(a_0, ..., a_k)] = P_0^{a_0}...P_k^{a_k}$$

donde P_i representa el $(i+1)$-primo ($P_0 = 2$, $P_1 = 3$, $P_2 = 5$, ...).
<u>Nombre</u>: $[(a_0, ..., a_k)]$ se llama *número de Gödel*

Definición 4. Longitud de un Número de Gödel

Sea $n \in \mathbb{N}$. Entonces si $n \neq 0 \Rightarrow n = P_0^{\alpha_0}, P_1^{\alpha_1}, P_2^{\alpha_2}, ..., P_k^{\alpha_k}$ con $\alpha_k \neq 0$ se define la *longitud de n* como:

$$long(n) = long[(\alpha_0, ..., \alpha_k)] = k$$

Definición 5. Indicadores de Números de Gödel

1. $\bullet[\bullet] : \mathbb{N}^2 \to \mathbb{N}/\, x[i] = V_{p_i}(x)$
2. $|\bullet| : \mathbb{N} \to \mathbb{N}/\, |n| = \begin{cases} long(n) & n \neq 0 \\ 0 & n = 0 \end{cases}$

Definición 6. Problema de la Parada

Definimos:

$$Halt : \mathbb{N}^2 \to \mathbb{N}/\, Halt(x, y) = \begin{cases} 1 & \text{si el programa de código } y \text{ ante la entrada } x \text{ termina} \\ 0 & \text{sino (es decir } \Psi_P(x) =\uparrow / \sharp P = y) \end{cases}$$

Definición 7. Programas Universales

Para cada $n > 0$ se define:

$$\phi^n : \mathbb{N}^{n+1} \to \mathbb{N} / \, \phi^n(x_1, ..., x_n, e) = \Psi_P^n(x_1, ..., x_n) \text{ tal que } \sharp P = e$$

12.2 Codificación

12.2.1 Codificación de Instrucciones

Variables

Enumeramos las variables en el siguiente orden:

$$Y, X_1, Z_1, X_2, Z_2, X_3, Z_3, ...$$

Es decir, la variable Y está en la posición 1 de la lista, la variable X_1 en la posición 2, etc.

Etiquetas

Enumeramos las etiquetas en el siguiente orden:

$$A_1, B_1, C_1, D_1, E_1, A_2, B_2, C_2, ...$$

Es decir, la etiqueta A_1 tiene la posición 1 de la lista, la etiqueta B_1 tiene la posición 2 de la lista, etc.

Instrucciones

Disponemos de 4 instrucciones:

$$\begin{aligned}
&V \leftarrow V + 1 \\
&V \leftarrow V - 1 \\
&IF\ V \neq 0\ GOTO\ L \\
&V \leftarrow V
\end{aligned}$$

12.2.2 Función $\sharp I$

De esta manera, definimos:

$$\sharp : \{\text{Instrucciones}\} \to \mathbb{N} / \, \sharp I =< a, < b, c >>$$

Donde

a: Está asociado a la etiqueta de la instrucción

$$a = \begin{cases} 0 & \text{si } I \text{ no tiene etiqueta} \\ \sharp L & \text{si } I \text{ tiene adelante la etiqueta } L \end{cases}$$

b: Esta asociado al tipo de instrucción

$$b = \begin{cases} 0 & \text{si } I \text{ es } V \leftarrow V \\ 1 & \text{si } I \text{ es } V \leftarrow V+1 \\ 2 & \text{si } I \text{ es } V \leftarrow V-1 \\ \sharp L + 2 & \text{si } I \text{ es } IF\ V \neq 0\ GOTO\ L \end{cases}$$

c: Está asociado a la variable que aparece en la instrucción

$$C = \sharp V - 1, \text{ siendo } V \text{ la variable que aparece en la instrucción } I$$

12.2.3 Codificación de Programas

12.2.4 Función $\sharp\mathcal{P}$

Dado un programa $\mathcal{P}$ que tiene k instrucciones: I_1, I_2, I_3, ..., I_k definimos:

$$\sharp : \{\text{Programas}\} \to \mathbb{N} / \sharp\mathcal{P}_{I_1,\,...,\,I_k} = [(\sharp I_1,\ ...,\ \sharp I_k)]_k - 1$$

Además, agregamos una **restricción** a los programas en Lenguaje S: *la última instrucción NO puede ser $Y \leftarrow Y$, a menos que sea la única instrucción.*

12.3 Proposiciones

1. $Halt(x, y)^n = Halt(x, y)$ **(demostarlo siempre que se quiera usar)**

12.4 Teoremas

12.4.1 Función Par

La función Par es biyectiva.

12.4.2 Funciones Par, Left y Right

Las funciones Par, Left y Right son RP.

12.4.3 Funciones Números de Gödel

Las funciones que calculan los números de Gödel son inyectivas y RP, pero NO sobreyectivas.

12.4.4 Funciones $\bullet[\bullet]$ y $|\bullet|$

Las funciones $\bullet[\bullet]$ y $|\bullet|$ son RP.

12.4.5 Funciones $\sharp I$ y $\sharp\mathcal{P}$

Las funciones $\sharp I$ (que asigna un número natural a cada instrucción) y $\sharp\mathcal{P}$ (que asigna un número natural a cada programa) son biyectivas.

12.4.6 Teorema 12.A

Existen funciones no computables.

12.4.7 Función Halt

La función $Halt : \mathbb{N}^2 \to \mathbb{N}$ es no computable.

12.4.8 Función ϕ^n

La función ϕ^n es parcialmente computable.

12.5 EXTRA

Tesis de Church

Todos los algoritmos para computar funciones $f : A \subseteq \mathbb{N}^k \to \mathbb{N}$ se pueden programar en lenguaje S. Es decir, si no se puede programar en lenguaje S $\Rightarrow$ no se puede programar en ningún otro lenguaje.

Tesis de Turing

Lo que no se puede resolver con una máquina de Turing no se puede resolver con otra máquina.

Equivalencia Turing y Church

Todo lo que se puede hacer en una máquina de Turing se puede hacer en Lenguaje S, y viceversa.

13 Álgebra de Boole

13.1 Definiciones

Definición 1. Álgebra de Boole

Un *álgebra de Boole* B es un conjunto B en el cual se pueden distinguir dos elementos notados 0 y 1, y hay tres elementos $\vee$, $\wedge$, $\neg$, que verifican las siguientes propiedades:

1. <u>Conmutatividad</u>:

$$x \wedge y = y \wedge x$$
$$x \vee y = y \vee x$$

2. <u>Asociatividad</u>:

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z$$
$$x \vee (y \vee z) = (x \vee y) \vee z$$

3. <u>Idempotencia</u>:

$$x \wedge x = x$$
$$x \vee x = x$$

4. <u>Absorción</u>:

$$x \vee (x \wedge y) = x$$
$$x \wedge (x \vee y) = x$$

5. <u>Distributividad Doble</u>:

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$
$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

6. <u>Elemento Neutro</u>:

$$x \wedge 0 = x$$
$$x \vee 1 = x$$

7. <u>Elemento Absorbente</u>:

$$x \vee 0 = 0$$
$$x \wedge 1 = 1$$

8. <u>Complementación</u>:

$$x \wedge \neg x = 1$$
$$x \vee \neg x = 0$$

Definición 2. Álgebra de Boole de Lindembaum para el cálculo proposicional

Sea $F =$ *conjunto de fórmulas de la lógica proposicional*

$$\alpha \mathcal{R} \beta \;\Leftrightarrow\; \alpha \equiv \beta$$

<u>Observación</u>: $\mathcal{R}$ es de equivalencia
Entonces:

$$B = \langle F/\mathcal{R},\, \underline{\wedge},\, \underline{\wedge},\, \underline{\neg},\, 0 = [p_1 \wedge \neg p_1],\, 1 = [p_1 \vee \neg p_1] \rangle$$

donde para $[\alpha]$ y $[\beta] \in F/\mathcal{R}$, $\underline{\wedge}$, $\underline{\wedge}$, $\underline{\neg}$ se definen de la siguiente manera:

1. $[\alpha] \underline{\wedge} [\beta] = [\alpha \wedge \beta]$
2. $[\alpha] \underline{\vee} [\beta] = [\alpha \vee \beta]$
3. $\underline{\neg}[\alpha] = [\neg \alpha]$

es un álgebra de Boole.

13.2 Propiedad fundamental de un Álgebra de Boole

Sea $B = \langle B,\, \wedge,\, \vee,\, \neg,\, 0,\, 1 \rangle$ un álgebra de Boole cualquiera. Entonces:

$$x \leq y \;\Leftrightarrow\; x \wedge y = x \;\Leftrightarrow\; x \vee y = y$$

<u>Observaciones</u>:

1. La relación $\leq$ es R, A y T $\Rightarrow$ es de orden
2. $0 \leq x \leq 1 \; \forall x \in B$