LÓGICA PARA COMPUTACIÓN

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

LÓGICA PROPOSICIONAL

CONSECUENCIA LÓGICA

UNSL

- ⌘ Argumentar consiste en deducir una conclusión a partir de un conjunto de premisas que se tienen por verdaderas.
- ⌘ Un argumento estará compuesto de unas *premisas* y de una *conclusión*. Puede ser *válido* o puede ser *inválido.*

Ejemplo

Premisa 1: Si estudio entonces aprobaré el examen de francés.

Premisa 2: No he estudiado.

Conclusión: No aprobaré el examen de francés.

- ⌘ Un argumento se considera *válido* si y sólo si no es posible encontrar una valuación tal que las premisas sean *verdaderas* y la conclusión *falsa*.

- Decimos que una valuación *v satisface una fórmula P* si

$$v(P)=\top$$

- Decimos que *v satisface un conjunto* $S \subseteq$ **Form** *si* v satisface *cada fórmula* de S, es decir, si $S = \{Q_1, Q_2, \ldots Q_n\}$ se cumple

$$v(Q_i) = \top$$

- Decimos que una fórmula P es *satisfacible* si existe alguna valuación v que la satisfaga.

- P se dice *insatisfacible* si no existe alguna valuación que la satisfaga.

- Decimos que una valuación *v satisface una fórmula P* si

$$v(P)=\top$$

- Decimos que *v satisface un conjunto* $S \subseteq$ **Form** *si* v satisface *cada fórmula* de S, es decir, si $S = \{Q_1, Q_2, \ldots Q_n\}$ se cumple

$$v\,(Q_i) = \top$$

- **Observación:**

Toda valuación v satisface al conjunto vacío (∅)

Ninguna valuación v satisface a **Form**

- Sean $S \subseteq \mathbf{Form}$ y $P \in \mathbf{Form}$. P es una *Consecuencia de S* si *toda valuación v que satisface a S también satisface a P.*

- P es una *Consecuencia de S* se denota: $\boldsymbol{S \models P}$

- El conjunto de todas las Consecuencias de S se denota $Con(S)$ y se define:

$$Con(S) = \{P \in Form \ / \ S \models P\}$$

- **Observación:** $\boldsymbol{Con(\emptyset)}$ es el conjunto de las Tautologías

- ***Teorema de la Deducción*** **(forma semántica)**

  Sean P, Q fórmulas y sea $S \subseteq \mathbf{Form}$ *entonces*

  $$Q \in Con(S \cup \{P\}) \quad \textit{si sólo si} \quad (P \rightarrow Q) \in Con(S)$$

- **Corolario**

  Para todo par de fórmulas P, Q se tiene que

  $\{P\} \models Q$ si sólo si $(P \rightarrow Q)$ es una *tautología*.

  $$\emptyset \cup \{P\} \models Q$$

  $$\emptyset \models (P \rightarrow Q)$$

⌘ *Teorema :*

Para todo $S \subseteq \mathbf{Form}$ *y para toda fórmula* P *se tiene que :*

$$S \models P \quad \text{si sólo si} \quad S \cup \{\neg P\} \text{ es insatisfacible}$$

Observación

Si $S = \varnothing$ entonces una fórmula P es *tautología* si sólo si $\neg P$ es *insatisfacible*

$$\varnothing \models P \quad \text{si sólo si} \quad \{\neg P\} \text{ es insatisfacible}$$

⌘ Observaciones:

- Un conjunto finito y no vacío de fórmulas es *satisfacible* si sólo si la *conjunción* de sus fórmula es satisfacible.

$$S = \{ P, Q, \ldots, Z \} \text{ es } \textit{satisfacible} \Leftrightarrow (P \wedge Q \wedge \ldots \wedge Z) \text{ es } \textit{satisfacible}$$

- Una fórmula P tal que $var(P) \subseteq \{x_1, \ldots, x_n\}$ es *satisfacible* si sólo si existe al menos una n-upla $a = (a_1, \ldots, a_n) \in \mathrm{B}^n$ tal que

$$\mathrm{T_P}(a_1, \ldots, a_n) = \top$$

Consecuencia:

La Tabla de Verdad nos permite saber si un conjunto finito de fórmulas es satisfacible o no, al igual que si una fórmula es consecuencia de conjunto finito fórmulas.

⌘ *Método*:

- Dado S = { $P, Q, \ldots, Z$ }, construimos la fórmula formada por la *conjunción* de todas las fórmulas:

$$P \wedge Q \wedge \ldots \wedge Z$$

- Construimos la Tabla de Verdad para la conjunción.
- Hasta encontrar una fila cuya última *columna* sea $\boxed{\top}$.

Si encontramos tal fila, el conjunto S es ***satisfacible****; Sino, es* ***insatisfacible***

⌘ *Método*:

- Dado S = { P, Q, …, Z }, construimos la fórmula formada por la *conjunción* de todas las fórmulas:

$$P \land Q \land \ldots \land Z$$

- Construimos la Tabla de Verdad para la conjunción.
- Hasta encontrar una fila cuya última *columna* sea $\boxed{\top}$.

⌘ **Observación**:

El método tiene un aspecto negativo, cuando la fórmula tiene ***n*** variables proposicionales, podemos hallar $\mathbf{2^n}$ valuaciones.

Peor caso: en la última fila aparece el $\top$

⌘ *Método:*

- Dado $S = \{ Q, R, \ldots, Z \}$ y la fórmula P, construimos la fórmula formada por la conjunción de todas las fórmulas de S y $\neg P$:

$$Q \wedge R \wedge \ldots \wedge Z \wedge \neg P$$

- Construimos la Tabla de Verdad para la conjunción.

- Si para todas las filas en su última columna encontramos $\boxed{\perp}$, entonces $S \cup \{\neg P\}$ es ***insatisfacible***.

Por lo tanto, *P es consecuencia del conjunto S,*

$$S \models P$$

- Una manera de representar los razonamientos utiliza la siguiente notación:

$$\frac{\textit{Premisas}}{\textit{Conclusión}}$$

- La ***conclusión*** de un razonamiento es una proposición que se deriva de otras proposiciones llamadas ***premisas***.

- Un argumento es *lógicamente válido* si y solo si su conclusión es una *consecuencia lógica* de sus premisas.

- Si un argumento, cuya conclusión es Q y cuya única premisa es P *es lógicamente válido*, entonces se dice que P *implica lógicamente a* Q.

- Analicemos el argumento del ejemplo anterior:

Ejemplo

Premisa 1: Si estudio entonces aprobaré el examen de francés.

Premisa 2: No he estudiado.

Conclusión: No aprobaré el examen de francés.

- Para comprobar si es válido hay que ver si la conclusión es una *consecuencia* de sus premisas.
  - Primero formalizamos el razonamiento:

Premisa 1: $(P \to Q)$

Premisa 2: $\neg P$

Conclusión: $\neg Q$

- Hacemos la tabla de verdad de la conjunción de las premisas y la negación de la conclusión: $P_1 \wedge P_2 \wedge \neg C$:

$$(((P \rightarrow Q) \wedge \neg P) \wedge \neg \neg Q)$$

P_1: $(P \rightarrow Q)$

P_1: $\neg P$

C: $\neg Q$

P	Q	$(((P \rightarrow Q) \wedge \neg P) \wedge \neg\neg Q)$
$\top$	$\top$	$\bot$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

⌘ Se ha demostrado que el argumento *NO es válido* y esta tabla es la *prueba* de eso.

Vamos a ver otro método que nos permita analizar fórmulas y argumentos. Para eso veamos estos conceptos.

⌘ ***Lema 1:***

Sean P, Q y R fórmulas y sea v una valuación, entonces se tienen las siguientes propiedades:

R¬) v satisface a $\neg\neg P$ si sólo si v satisface a P

$$\frac{\neg\neg P}{P}$$

Lógica Proposicional

R∨) v satisface a $(P \vee Q)$ si sólo si v satisface a P o v satisface a Q

$$\frac{(P \vee Q)}{P \mid Q}$$

Al menos una de las conclusiones es verdadera

R¬∨) v satisface a $\neg(P \vee Q)$ si sólo si

v satisface a $\neg P$ y v satisface a $\neg Q$

$$\frac{\neg(P \vee Q)}{\neg P, \neg Q}$$

Ambas conclusiones son verdaderas

R ∧) v satisface a $(P \wedge Q)$ si sólo si v satisface a ambas a P y a Q

$$\frac{(P \wedge Q)}{P \;\; , Q}$$

R¬∧) v satisface a $\neg(P \wedge Q)$ si sólo si

v satisface a $\neg P$ o v satisface a $\neg Q$

$$\frac{\neg(P \wedge Q)}{\neg P \mid \neg Q}$$

R $\rightarrow$) v satisface a $(P \rightarrow Q)$ si sólo si

v satisface a $\neg P$ o v satisface a Q

$$\frac{(P \rightarrow Q)}{\neg P \mid Q}$$

R$\neg\rightarrow$) v satisface a $\neg(P \rightarrow Q)$ si sólo si

v satisface a P y v satisface a $\neg Q$

$$\frac{\neg(P \rightarrow Q)}{P \ , \neg Q}$$

⌘ Las reglas de tipo *A* son aquellas que tienen:

una o dos conclusiones separadas por una coma.

Ejemplo:

$$\frac{\neg(P \to Q)}{P, \neg Q}$$

Significado:

Una valuación v satisface a la premisa si sólo si

*v satisface a **todas** las conclusiones.*

Reglas de tipo A son: $R\neg$, $R\neg\vee$, $R\neg\to$ y $R\wedge$

- Las reglas de tipo *B* son aquellas que tienen:

  dos conclusiones separadas por una barra vertical

Ejemplo: $$\frac{\neg(P \wedge Q)}{\neg P \mid Q}$$

Significado:

Una valuación v satisface a la premisa si sólo si

v satisface ***por lo menos una*** *de las conclusiones.*

Reglas de tipo *B* son: $\boxed{R\vee,\ R\rightarrow\ \text{y}\ R\neg\wedge}$

⌘ ***Lema 2:***

Toda fórmula que NO es una variable proposicional o negación de una variable proposicional, es de la forma de la premisa de una (y sólo una) de las reglas:

R¬, R¬∨ , R ∨, R ∧, R ¬ ∧ , R → y R¬ →

Demostración:

Si P no es una variable proposicional, sólo puede ser una de:

1. $P = \neg T$
2. $P = (Q \vee R)$
3. $P = (Q \wedge R)$
4. $P = (Q \rightarrow R)$

En los casos 2, 3 y 4 P es de la forma de la premisa de **R** ∨**, R** ∧**, R** →

En 1 $(P = \neg T)$, si T es una variable proposicional entonces P es la *negación* de una variable proposicional. Sino sólo puede ser uno de:

1. $T = \neg Q$ *entonces* $P = \neg T = \neg\neg Q$
2. $T = (Q \vee R)$ *entonces* $P = \neg T = \neg (Q \vee R)$
3. $T = (Q \wedge R)$ *entonces* $P = \neg T = \neg (Q \wedge R)$
4. $P = (Q \rightarrow R)$ *entonces* $P = \neg T = \neg (Q \rightarrow R)$

Donde P es de la forma de la premia de $\mathbf{R}\neg$, $\mathbf{R}\neg\vee$, $\mathbf{R}\neg\wedge$ y $\mathbf{R}\neg\rightarrow$

c.q.d.

Estas reglas proveen un método alternativo a las Tablas de Verdad para determinar si una fórmula es satisfacible.

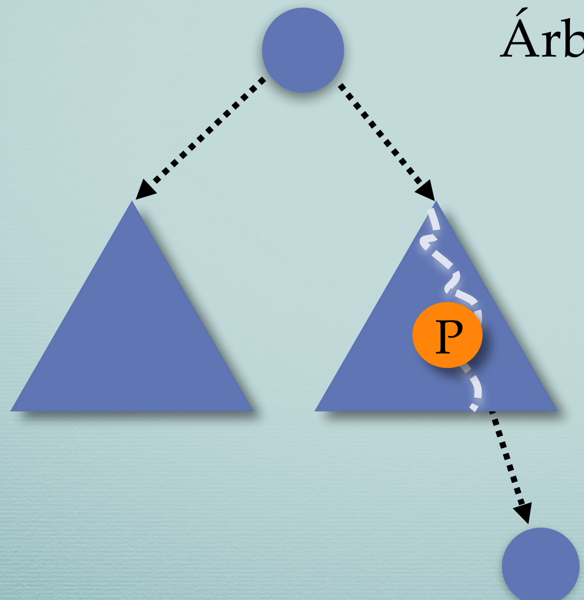
- Se llama ***Árbol de fórmulas*** a un árbol cuyos nodos son fórmulas.

- Sea $\mathcal{A}$ un árbol de fórmulas. Un árbol $\mathcal{A}'$ se dice una ***Extensión Inmediata de*** $\mathcal{A}$, si $\mathcal{A}'$ se obtiene agregando como hoja, la conclusión de una de las reglas

$$R\neg,\ R\vee,\ R\neg\vee,\ R\wedge,\ R\neg\wedge,\ R\rightarrow \text{ y } R\neg\rightarrow$$

aplicadas a una fórmula P que figura en $\mathcal{A}$, del siguiente modo:

⌘ Si la fórmula P es de tipo ***A*** entonces $\mathcal{A}'$ se obtiene agregando un nuevo nodo con ***una de las conclusiones*** de P como único sucesor del nodo terminal de una rama de $\mathcal{A}$ en donde figure P.

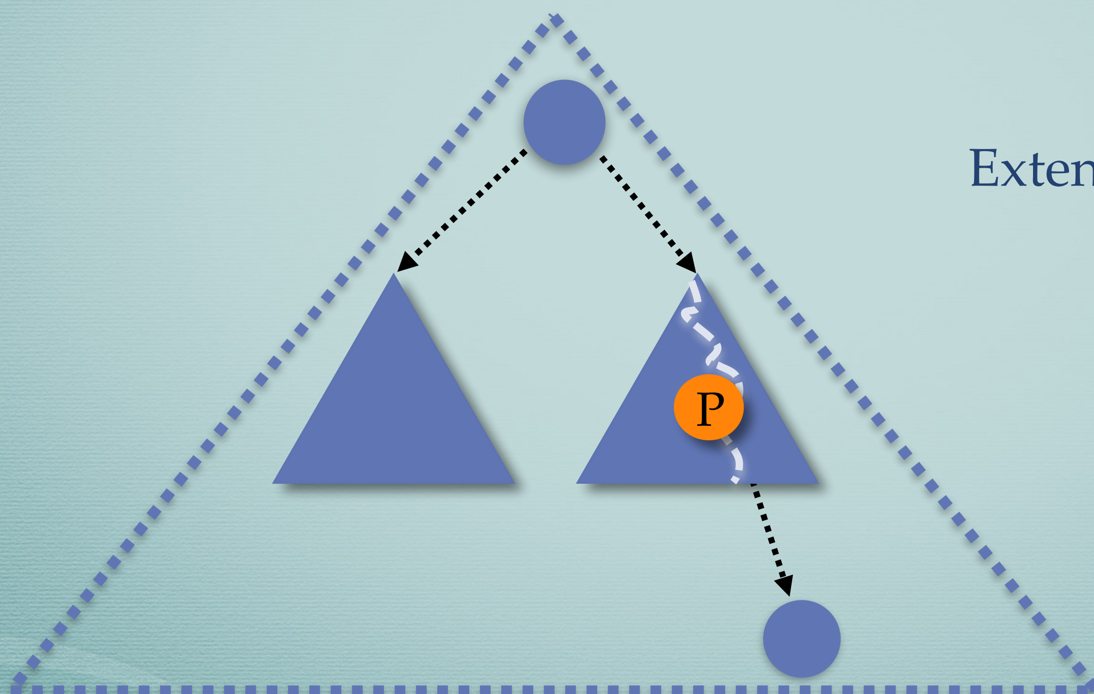


Árbol de fórmulas: $\mathcal{A}$

P tiene una o dos conclusiones: Q , R

Agregamos ***una***: Q o R

⌘ Si la fórmula P es de tipo ***A*** entonces $\mathcal{A}'$ se obtiene agregando un nuevo nodo con ***una de las conclusiones*** de P como único sucesor del nodo terminal de una rama de $\mathcal{A}$ en donde figure P.



Extensión Inmediata $\mathcal{A}'$

Ejemplo:

Sea el Árbol de Fórmulas $\mathcal{A}$

$$P = \neg(\overbrace{((p_0 \vee p_1) \rightarrow p_2)}^{Q} \rightarrow \overbrace{((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2))}^{R})$$

$$\frac{\neg(Q \rightarrow R)}{Q\ ,\neg R}$$

$$Q = ((p_0 \vee p_1) \rightarrow p_2)$$

Extensión Inmediata $\mathcal{A}$'

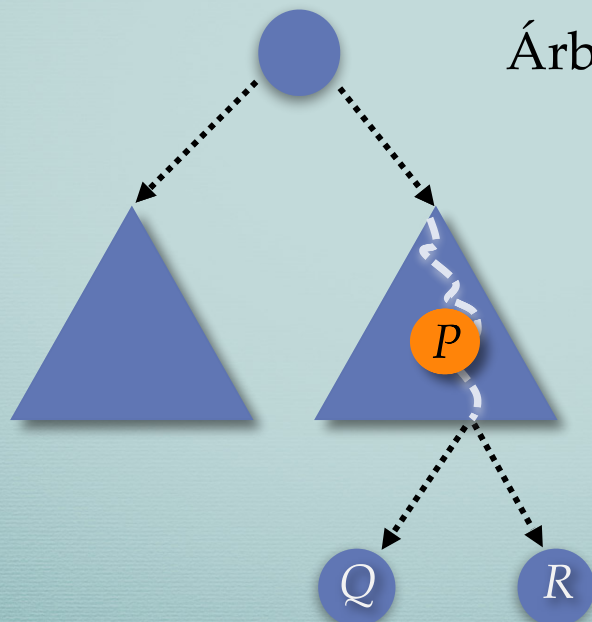
Árbol de Fórmulas $\mathcal{A}$

$$\neg\ R = \neg\ ((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2))$$

Extensión Inmediata $\mathcal{A}$'

Árbol de Fórmulas $\mathcal{A}$

- Si la fórmula es de tipo ***B*** entonces $\mathcal{A}'$ se obtiene agregando ***dos*** nuevos nodos, correspondientes a ***cada una*** de las conclusiones de P, como sucesores del nodo terminal de una rama de $\mathcal{A}$ en donde figure P.

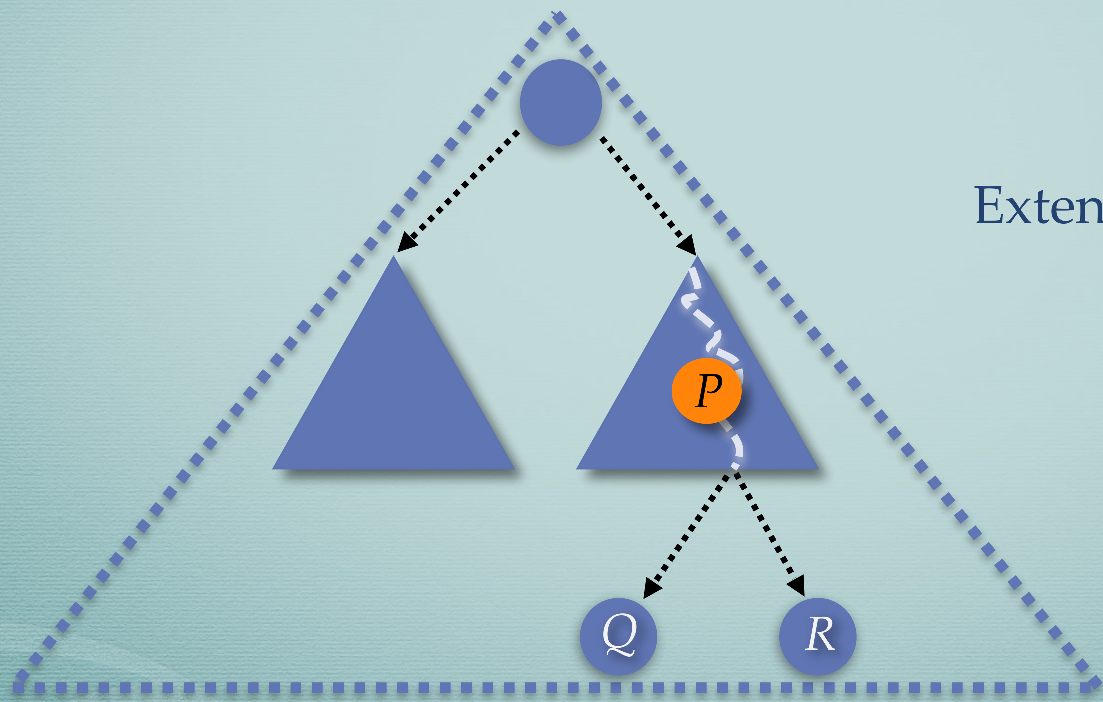


Árbol de fórmulas: $\mathcal{A}$

P tiene dos conclusiones: Q / R

Agregamos ***ambas*** Q y R

- Si la fórmula es de tipo ***B*** entonces $\mathcal{A}'$ se obtiene agregando ***dos*** nuevos nodos, correspondientes a ***cada una*** de las conclusiones de P, como sucesores del nodo terminal de una rama de $\mathcal{A}$ en donde figure P.



Extensión Inmediata $\mathcal{A}'$

Lógica Proposicional

Consecuencia

Árbol de Fórmulas $\mathcal{A}$

$$\neg(((p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2)))$$

$$Q = ((p_0 \vee p_1) \to p_2)$$

$$\frac{(Q_1 \to Q_2)}{\neg Q_1 \mid Q_2}$$

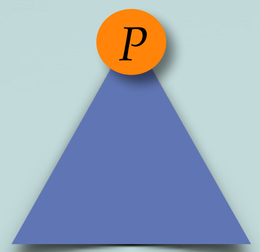
$$\neg R = \neg((p_0 \to p_2) \wedge (p_1 \to p_2))$$

$$Q_1 = \neg(p_0 \vee p_1) \qquad Q_2 = p_2$$

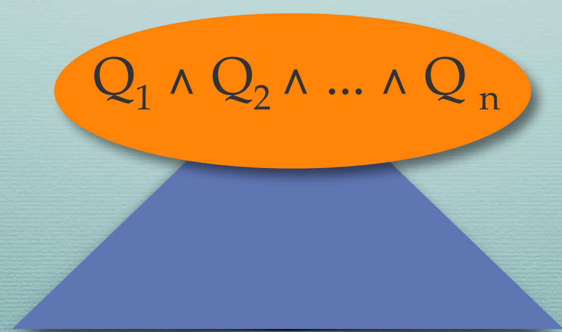
Extensión Inmediata $\mathcal{A}'$

Lógica Proposicional

- Un árbol de fórmulas es un *Árbol de Refutación de una fórmula* P si sólo si se obtiene haciendo un número finito de extensiones inmediatas a partir del árbol cuyo único nodo es P.



- Un *Árbol de Refutación de un conjunto finito de fórmulas* $S = \{Q_1, Q_2, \ldots, Q_n\}$ es el Árbol de Refutación de la conjunción de todas las fórmulas de S.



- Una rama de un árbol de fórmulas se dice *cerrada* si *contiene a la vez*:

a una fórmula y a su negación

- En caso contrario se dice que la rama es *abierta*.

- Un árbol de fórmulas es *cerrado* si *todas sus ramas son cerradas*.

- Una rama de un árbol de fórmulas se dice *saturada* si el conjunto que contiene las fórmulas de sus nodos es *saturado*.

- Un conjunto S de fórmulas se dice *saturado* si sólo si satisface las siguientes condiciones:

  Rs_0) Una fórmula y su negación *no pueden estar en S* simultáneamente.

  Rs_1) Si una fórmula de tipo A está en *S*, entonces *sus dos conclusiones* están en *S* (una en el caso de R¬).

  Rs_2) Si una fórmula de tipo B está en *S*, entonces *al menos una* de sus dos conclusiones están en *S*.

Un conjunto saturado no puede ser cerrado

⌘ Un Árbol de Refutación se dice *completo* si cada una de sus ramas es :

Cerrada o *Saturada*

Una rama saturada no puede ser cerrada

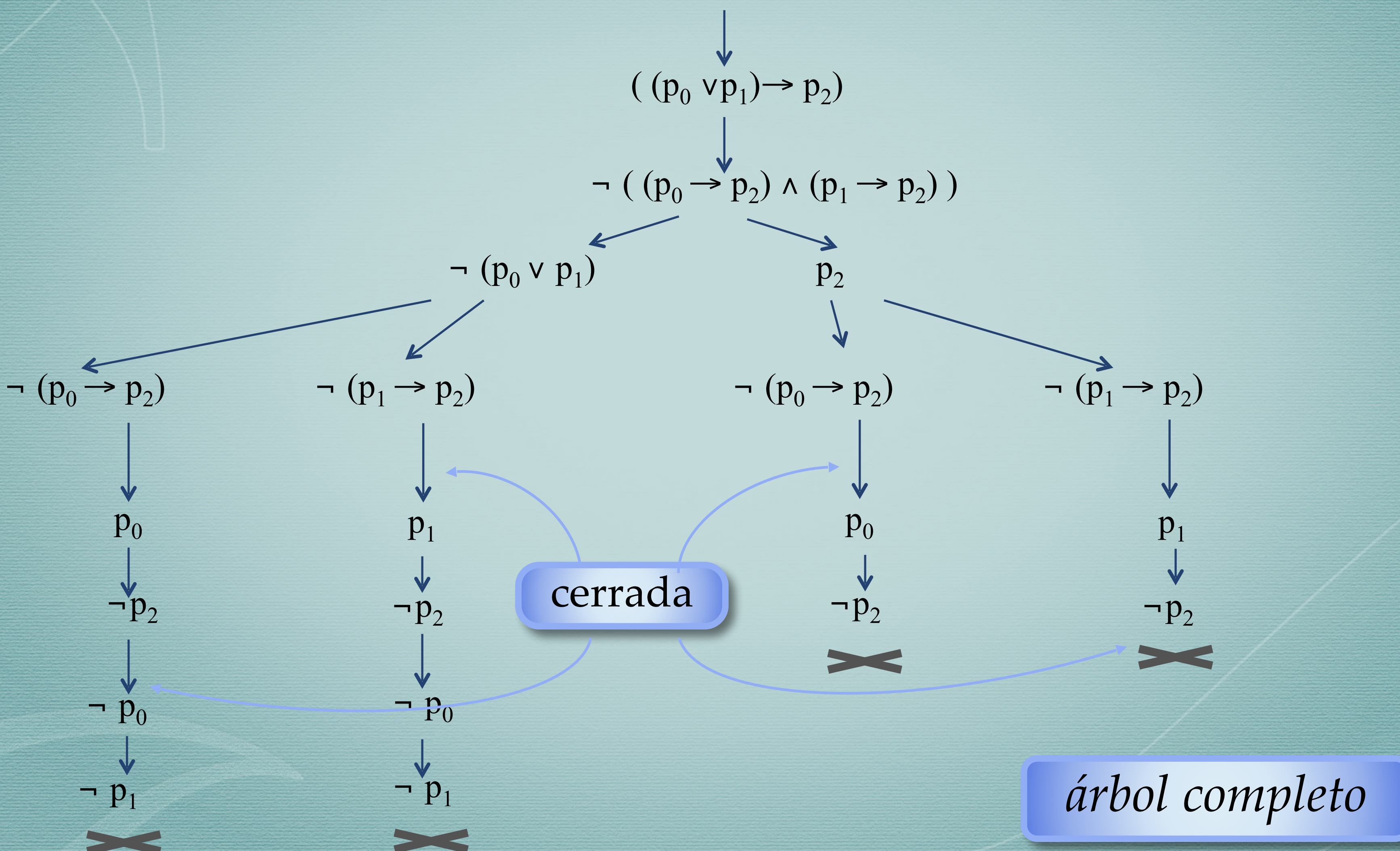
Observación:

Todo Árbol de Refutación cerrado es un ***árbol completo.***

⌘ Lema 2:

Toda fórmula P tiene por lo menos un Árbol de Refutación completo.

Ejemplos: $\neg(((p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2)))$



Lógica Proposicional

Ejemplos:

$$\neg((p_0 \wedge p_1) \rightarrow \neg(p_0 \vee \neg p_1))$$

$$\downarrow$$

$$(p_0 \wedge p_1)$$

$$\downarrow$$

$$\neg\neg(p_0 \vee \neg p_1)$$

$$\downarrow$$

$$(p_0 \vee \neg p_1)$$

$$\downarrow$$

$$p_0$$

$$\downarrow$$

$$p_1$$

$$\swarrow \qquad \searrow$$

$$p_0 \qquad\qquad \neg p_1$$

saturada

cerrada

árbol completo

Ejemplos:

$\neg(((p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2)))$

$((p_0 \vee p_1) \to p_2)$

$\neg((p_0 \to p_2) \wedge (p_1 \to p_2))$

$\neg(p_0 \vee p_1)$ | p_2

$\neg(p_0 \to p_2)$ | $\neg(p_1 \to p_2)$ | $\neg(p_0 \to p_2)$ | $\neg(p_1 \to p_2)$

p_0 | p_1

$\neg p_2$ | $\neg p_2$

abiertas

cerradas

No saturadas

No es un árbol completo

Ejemplo 1: $P = (\,(\,(p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2)\,)\,)$

P es tautología ?

Averiguemos si $\neg P$ es satisfacible o no:

$$\neg P = \neg(\,\underbrace{(\,(p_0 \vee p_1) \to p_2)}_{Q} \to \underbrace{(\,(p_0 \to p_2) \wedge (p_1 \to p_2)\,)}_{R}\,)$$

R$\neg\to$

$$\frac{\neg(Q \to R)}{Q\ ,\neg R}$$

$$Q = (\,(p_0 \vee p_1) \to p_2\,)$$

$$\neg R = \neg\,(\,(p_0 \to p_2) \wedge (p_1 \to p_2)\,)$$

R$\to$

$$\frac{(Q \to R)}{\neg Q \mid R}$$

$$Q_1 = \neg\,(p_0 \vee p_1) \qquad Q_2 = p_2$$

Lógica Proposicional

$$\neg P = \neg(((p_0 \vee p_1) \rightarrow p_2) \rightarrow ((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2)))$$

$$Q = ((p_0 \vee p_1) \rightarrow p_2)$$

R¬∧

$$\frac{\neg(R_1 \wedge R_2)}{\neg R_1 \mid \neg R_2}$$

$$\neg R = \neg((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2))$$

$$Q_1 = \neg(p_0 \vee p_1) \qquad Q_2 = p_2$$

$$\neg R_1 = \neg(p_0 \rightarrow p_2) \quad \neg R_2 = \neg(p_1 \rightarrow p_2) \quad \neg R_1 = \neg(p_0 \rightarrow p_2) \quad \neg R_2 = \neg(p_1 \rightarrow p_2)$$

R¬→

$$\frac{\neg(R_{11} \rightarrow R_{12})}{R_{11}, \neg R_{12}} \qquad \frac{\neg(R_{21} \rightarrow R_{22})}{R_{21}, \neg R_{22}}$$

Lógica Proposicional

$\neg P = \neg(((p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2)))$

$Q = ((p_0 \vee p_1) \to p_2)$

$\neg R = \neg((p_0 \to p_2) \wedge (p_1 \to p_2))$

$Q_1 = \neg(p_0 \vee p_1)$ $\quad$ $Q_2 = p_2$

$\neg R_1 = \neg(p_0 \to p_2)$ $\quad$ $\neg R_2 = \neg(p_1 \to p_2)$ $\quad$ $\neg R_1 = \neg(p_0 \to p_2)$ $\quad$ $\neg R_2 = \neg(p_1 \to p_2)$

$R_{11} = p_0$ $\quad$ $R_{21} = p_1$ $\quad$ $R'_{11} = p_0$ $\quad$ $R'_{21} = p_1$

$R_{12} = \neg p_2$ $\quad$ $R_{22} = \neg p_2$ $\quad$ $R'_{12} = \neg p_2$ $\quad$ $R'_{22} = \neg p_2$

R¬→

$Q_{11} = \neg p_0$ $\quad$ $Q_{11} = \neg p_0$

$Q_{12} = \neg p_1$ $\quad$ $Q_{12} = \neg p_1$

$$\frac{\neg(R_{11} \to R_{12})}{R_{11}, \neg R_{12}}$$

$$\frac{\neg(R_{21} \to R_{22})}{R_{21}, \neg R_{22}}$$

$$\frac{\neg(Q_{11} \vee Q_{12})}{\neg Q_{11}, \neg Q_{12}}$$

Ejemplo 1:

Partimos suponiendo que v satisfacía a $\neg P$, por aplicación de las reglas llegamos una ***contradicción***.

Tiene todas las ramas *cerradas*. Es decir todos los caminos se "*cierran*".

Luego, $\neg P$ es *insatisfacible*.

Conclusión:

P es tautología

Ejemplo 2: $P = ((p_0 \wedge p_1) \rightarrow \neg(p_0 \vee \neg p_1))$

Averigüemos si $\neg P$ es satisfacible o no:

$$\neg P = \neg((p_0 \wedge p_1) \rightarrow \neg(p_0 \vee \neg p_1))$$

R$\neg\rightarrow$

$$\frac{\neg(Q \rightarrow R)}{Q\ ,\neg R}$$

$$Q = (p_0 \wedge p_1)$$

$$\neg R = \neg\neg(p_0 \vee \neg p_1)$$

R$\neg$

$$\frac{\neg\neg R_1}{R_1}$$

$$R_1 = (p_0 \vee \neg p_1)$$

Ejemplo 2:

$$\neg P = \neg(\,(p_0 \wedge p_1) \rightarrow \neg\,(p_0 \vee \neg\, p_1))$$

$$Q = (p_0 \wedge p_1)$$

$$\neg R = \neg\neg\,(p_0 \vee \neg\, p_1)$$

$$R_1 = (p_0 \vee \neg\, p_1)$$

R ∧

$$\frac{(Q_1 \wedge Q_2)}{Q_1\ ,Q_2}$$

$$Q_1 = p_0$$

$$Q_2 = p_1$$

R∨

$$\frac{(R_{11} \vee R_{12})}{R_{11} \mid R_{12}}$$

$$R_{11} = p_0 \qquad R_{12} = \neg p_1$$

Ejemplo 2:

Tiene una rama derecha "*cerrada*" y una rama izquierda "*saturada*", es decir, no hay más reglas para aplicar en ella.

Comenzando desde el nodo hijo hacia arriba, encontramos que $\neg P$ es *satisfacible*.

Conclusión:

P no es tautología

⌘ Lema:

Si P es una fórmula satisfacible, entonces todo Árbol de Refutación de P tiene *al menos una rama abierta*.

⌘ Corolario:

Las siguientes propiedades son equivalentes para toda fórmula P:

(i) P es Tautología

(ii) $\neg P$ tiene tiene un *Árbol de Refutación cerrado*

(iii) Todo Árbol de Refutación completo de $\neg P$ *es cerrado*

⌘ Teorema:

Las siguientes propiedades son equivalentes para todo conjunto finito y no vacío $S \subseteq$ **Form**:

(i) S es satisfacible

(ii) Todo Árbol de Refutación de S tiene *por lo menos una rama abierta*

(iii) Existe un Árbol de Refutación completo de S *con una rama abierta*

⌘ Corolario:

Las siguientes propiedades son equivalentes para todo conjunto finito y no vacío $S \subseteq \mathbf{Form}$:

(i) S es Insatisfacible

(ii) S tiene un Árbol de Refutación *cerrado*

(iii) Todo Árbol de Refutación completo de S es *cerrado*

Lógica Proposicional

⌘ Conclusiones:

- Obtuvimos un método alternativo a las Tablas de Verdad, para determinar si una fórmula es satisfacible y/o Tautología.
- Si la fórmula es satisfacible podemos encontrar una valuación que la satisfaga, siguiendo una rama abierta.

$$
\begin{array}{ccc}
 & \neg((p_0 \wedge p_1) \rightarrow \neg(p_0 \vee \neg p_1)) & \\
 & \downarrow & \\
 & (p_0 \wedge p_1) & \\
 & \downarrow & \\
 & \neg\neg(p_0 \vee \neg p_1) & \\
 & \downarrow & \\
 & (p_0 \vee \neg p_1) & \\
 & \downarrow & \\
 & p_0 & \\
 & \downarrow & \\
 & p_1 & \\
\swarrow & & \searrow \\
p_0 & & \neg p_1 \\
 & & \times
\end{array}
$$

⌘ Conclusiones:

- Obtuvimos un método alternativo a las Tablas de Verdad, para determinar si una fórmula es satisfacible y/o Tautología.
- Si la fórmula es satisfacible podemos encontrar una valuación que la satisfaga, siguiendo una rama abierta.
- Si S = Ø entonces $P \in \mathrm{Con}\,(S)$ si sólo si la fórmula $\neg P$ tiene un *Árbol de Refutación cerrado*.
- Dado $S = \{ Q_1, \ldots, Q_n\} \subseteq$ **Form**, decimos que *P es consecuencia de S* si sólo si

$$Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n \wedge \neg P$$

Es *insatisfacible*, es decir tiene un *Árbol de Refutación cerrado*.

PREGUNTAS .....