Lógica Proposicional

Lógica

- Disciplina matemática
- Disciplina formal
  - Se razona sobre la estructura de las cosas
- Se quiere estudiar el razonamiento, y no las verdades contingentes
- Se quiere estudiar la noción de consecuencia
  - **Si** (ocurre que, se sabe que...) **A es verdadero**,
  - **Entonces** (ocurre que, se sabe que, se puede probar que...) **B es verdadero**.

Los silogismos del Bercho

- Todos los perros ladran
- Bercho es un perro
- Luego, Bercho ladra

- Todos los perros ladran
- Bercho no es un gato
- Luego, Bercho ladra

- Todos los gatos ladran
- Tolomeo es un gato
- Luego, Tolomeo ladra

- Todos los P son L
- B es un P
- Luego, B es un L

Reglas del club escocés

1. Todo miembro no escocés deberá usar medias rojas.
2. Todo miembro que use medias rojas deberá usar un kilt.
3. Los miembros casados no podrán salir los sábados.
4. Los miembros saldrán los sábados si y sólo si son escoceses.
5. Todo miembro que use kilt deberá ser escocés y casado.
6. Todo miembro escocés deberá usar kilt.

Usted fue recientemente contratado como portero del club y llega una persona a la puerta.

¿Bajo qué condiciones lo dejaría pasar?

Una mano al portero

- Si viene alguien con medias rojas
  - debe usar kilt (2)
  - debe ser escocés y casado (5)
  - debe salir los sábados (4)
  - no debe estar casado (3)
- Si viene alguien sin medias rojas
  - debe ser escocés (1)
  - debe usar kilt (6)
  - ... y sigue como antes

Validez de un razonamiento

- Del conjunto de reglas...
  - ¿se deduce que un escocés entra?
  - ¿se deduce que un escocés no entra?
  - ¿se deduce que nadie entra?
- En general: De un conjunto de hipótesis...
  - ¿se deduce cierta conclusión?
- Justificar validez del razonamiento
  - Un razonamiento es válido si siempre que las hipótesis son verdaderas, la conclusión es verdadera.
- ¿Cuándo vamos a considerar inválido un razonamiento?
  - Cuando permita obtener conclusiones falsas a partir de hipótesis verdaderas.

Nociones detrás de “Razonamiento”

- En la idea intuitiva de razonamiento que se está manejando, hay al menos tres elementos involucrados:
  - Una noción de “Verdad”: hay cosas que son verdaderas y otras que son falsas.
  - Una noción de “proceso”: mediante algún mecanismo somos capaces de decidir que si determinadas cosas son ciertas, entonces otras cosas deben ser ciertas también.
  - Una noción de “correctitud” o “validez” del proceso, dada por la forma del planteo.

Justificación de la validez del razonamiento

- Dos maneras diferentes de justificar:
  - Justificar que la veracidad de las hipótesis implica la veracidad de la conclusión (semántica)
  - Dar una demostración que pruebe a la conclusión a partir de las hipótesis, a través de pasos debidamente justificados (sintáctica)
- ¿Ambas formas de justificar son equivalentes?

Lógica Proposicional - Plan

1. Sintaxis de la lógica proposicional
   – El conjunto inductivo PROP, sustitución.
2. Semántica de la Lógica Proposicional
   – Valores de verdad, valuaciones, equivalencia de proposiciones, consecuencia lógica, tautologías.
3. Pruebas en deducción natural
   – Derivación, consecuencia sintáctica, teorema.
4. Metateoría de la lógica proposicional
   – Consistencia y completitud de la lógica proposicional.

Proposiciones

- Proposición: Una oración afirmativa de la cual se puede decir que es verdadera o falsa.
- Ejemplos de Proposiciones:
  - Ayer llovió en Paysandú.
  - El Sol gira alrededor de la Tierra.
  - $2 . 3 = 3 + 3$
  - 3 es primo.
  - El sucesor de 3 es primo.

Más proposiciones...

- **Si** ayer llovió en Paysandú, **entonces** el intendente se mojó.
- El Sol gira alrededor de la Tierra o la Tierra gira alrededor del Sol.
- 2 . 3 = 6 **y** 6 es impar.
- 3 **no** es primo.
- **Hay** un número natural que es par y es primo.
- **Todo** entero par mayor que cuatro es la suma de dos números primos.

Proposiciones para los Razonamientos

- Observaciones:
  - Los razonamientos utilizan proposiciones simples y complejas que pueden ser verdaderas o falsas.
  - Se necesita la posibilidad de verificar los razonamientos sobre esas proposiciones.

- ¿Cómo se va a formalizar esto?
  - Definiendo un lenguaje de proposiciones en forma inductiva: ***Prop***.
  - Definiendo funciones por recursión primitiva sobre Prop que asocian valores de verdad: ***Valuaciones (v:Prop→{0,1})***.
  - Introducción de dos nociones de razonamiento: una basada en los valores de verdad y otra en la sintaxis.
  - Verificar que las nociones coinciden.

1. Sintaxis de la Lógica Proposicional

Alfabeto: Σ_{PROP}

Def 1.1.1 [Σ_{PROP}]

El alfabeto de la lógica proposicional es el conjunto Σ_{PROP} que consiste de:

i) letras de proposición $p_0, p_1, p_2, \ldots$

ii) conectivos $\perp, \neg, \wedge, \vee, \rightarrow, \leftrightarrow$

iii) símbolos auxiliares: (,)

<u>Notación</u> :

$C = \{\wedge, \vee, \rightarrow, \leftrightarrow\}$

$AT = \{\perp, p_0, p_1, p_2, \ldots\}$

$P = \{p_0, p_1, p_2, \ldots\}$

PROP

<u>Def 1.1.2</u> [PROP]

PROP es el conjunto definido inductivamente por :

i) $p_i \in$ PROP para todo $i \in \mathbf{N}$

ii) $\bot \in$ PROP

$\left.\right\}$ fórmulas atómicas

iii) Si $\alpha \in$ PROP y $\beta \in$ PROP entonces:

$(\alpha \land \beta) \in$ PROP

$(\alpha \lor \beta) \in$ PROP

$(\alpha \to \beta) \in$ PROP

$(\alpha \leftrightarrow \beta) \in$ PROP

iv) Si $\alpha \in$ PROP entonces $(\neg\alpha) \in$ PROP

Usamos letras griegas como metavariables: variables para hablar de formulas de PROP

- **Los objetos de PROP se llaman *fórmulas proposicionales***

PROP - Ejemplos

- Algunas palabras de PROP:
  - p_0
  - $(p_1 \to p_3)$
  - $\perp$
  - $((p_1 \to p_2) \vee (\perp \wedge (\neg\, p_5)))$

- Algunas palabras que no están en PROP:
  - (p_0)
  - $(p_1 \to)$
  - $p_1 \to \perp$

Principio de inducción primitiva para PROP

Teorema 1.1.3 [ppio. inducción primitiva para PROP]

Sea P una propiedad sobre PROP que cumple

pb) - $P(p_i)$ para todo $i \in N$

- $P(\perp)$

pi) - Si $P(\alpha)$ y $P(\beta)$ entonces $P((\alpha \square \beta))$, para $\square \in C$

- Si $P(\alpha)$ entonces $P((\neg\alpha))$

Entonces, para toda $\alpha \in$ PROP, se cumple $P(\alpha)$

HINT: Considere $X = \{\alpha \in \text{PROP} : P(\alpha)\}$

Secuencia de formación y Subfórmula

Def 1.1.4 [secuencia de formación y subfórmula]

a. Una secuencia α_0 , α_1 ,.... α_n de elementos de $\Sigma_{PROP}{}^*$ es una secuencia de formación para α ssi $\alpha_n = \alpha$ y para todo $k \leq n$ se cumple:

- $\alpha_k \in AT$ (o sea, $\alpha_k = \perp$ o $\alpha_k = p_i$ para $i \in N$), o bien
- $\alpha_k = (\alpha_i \ \square\ \alpha_j)$ para $\square \in C$ y $i, j < k$, o bien
- $\alpha_k = (\neg \alpha_j)$ para algún $j < k$

b. Dadas $\varphi, \alpha \in PROP$, φ es subfórmula de α ssi:

- $\alpha = \varphi$, o bien
- $\alpha = (\varphi_1 \square \varphi_2)$ con $\square \in C$ y φ subfórmula de φ_1 o de φ_2
- $\alpha = (\neg\, \varphi_1)$ y φ es subfórmula de φ_1

Esquema de Recursión Primitiva para PROP

Para definir f : PROP $\rightarrow$B debo considerar:

- f (p_i) = ... (para todo i$\in$N)
- f ($\perp$) = ...
- f(($\alpha \wedge \beta$)) = ... f (α) .. α.. f (β) .. β ..
- f(($\alpha \vee \beta$)) = ... f (α) .. α.. f (β) .. β ..
- f(($\alpha \rightarrow \beta$)) = ... f (α) .. α.. f (β) .. β ..
- f(($\alpha \leftrightarrow \beta$)) = ... f ($\alpha$) .. α.. f (β) .. β ..
- f(($\neg\alpha$)) = ... f (α).. α..

• <u>Ejemplo</u>: definir cant (α)

Esquema de Recursión Primitiva para PROP

H) Sean H_{at}, $H_\square$ $(\square \in C)$ y $H_\neg$ tales que:

H_{at} : AT $\rightarrow$B

$H_\square$: PROP x B x PROP x B $\rightarrow$ B para $\square \in C$

$H_\neg$: PROP x B $\rightarrow$ B

T) existe una única función f: PROP $\rightarrow$B tal que:

$f(\alpha) = H_{at}(\alpha)$ para $\alpha \in AT$

$f((\alpha \square \beta)) = H_\square(\alpha, f(\alpha), \beta, f(\beta))$ para $\square \in C$

$f((\neg\alpha)) = H_\neg(\alpha, f(\alpha))$

Ejemplos de funciones recursivas

- Long : PROP $\rightarrow$ N
  - cantidad de símbolos de una fórmula proposicional
- Tree : PROP $\rightarrow$ Arbol
  - árbol sintáctico
- Rango : PROP $\rightarrow$ N
  - profundidad del átomo más interior o del árbol
- Atoms : PROP $\rightarrow$ P(AT)
  - conjunto de átomos de una fórmula proposicional

$$\text{Atoms }(p_i) = \{p_i\}$$

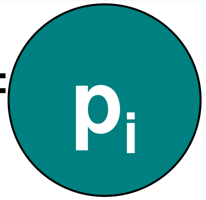
$$\text{Atoms }(\bot) = \{\bot\}$$

$$\text{Atoms}(\ (\alpha \,\square\, \beta)\ ) = \text{Atoms}(\alpha) \cup \text{Atoms}(\beta) \quad , \square \in C$$

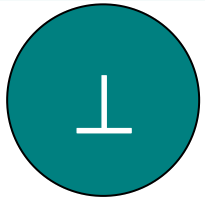
$$\text{Atoms}(\ (\neg\alpha)\ ) = \text{Atoms}(\alpha)$$

Tree

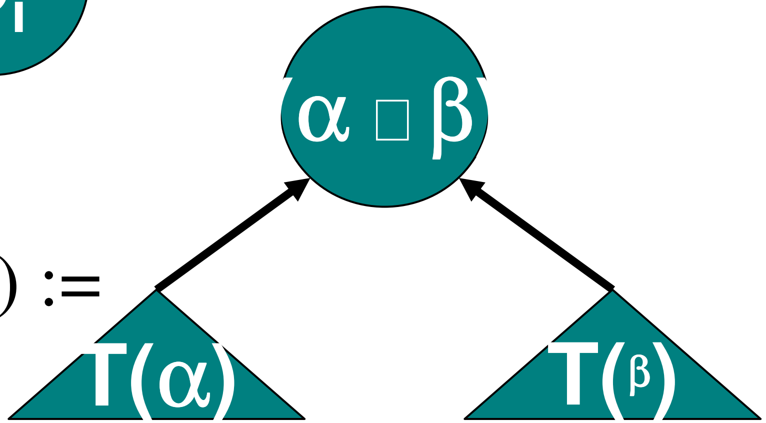
•Tree (p_i) :=



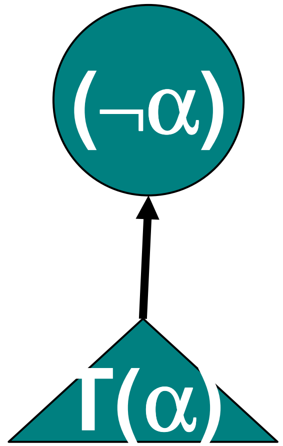
•Tree ($\perp$) :=



•Tree (($\alpha \square \beta$)) :=



•Tree (($\neg\alpha$)) :=



Note que los hijos se distinguen. Una cosa es la izquierda, y otra la derecha.

Subfórmulas

- **Def.1.1.4.b** Dadas $\varphi, \alpha \in$ PROP, φ es subfórmula de α ssi:
  - $\alpha = \varphi$, o bien
  - $\alpha = (\varphi_1 \square\, \varphi_2)$ con $\square \in C$ y φ subfórmula de φ_1 o de φ_2
  - $\alpha = (\neg\, \varphi_1)$ y φ es subfórmula de φ_1
- **Def.1.1.7** Se define recursivamente la función
  - Sub : PROP $\rightarrow$ $\wp$(PROP) como
  - Sub $(\varphi) = \{\varphi\}$, si φ es atomica
  - Sub $((\varphi_1 \square\, \varphi_2)) = \mathrm{Sub}(\varphi_1) \cup \mathrm{Sub}(\varphi_2) \cup \{(\varphi_1 \square\, \varphi_2)\}$
  - Sub $((\neg\, \varphi_1)) = \mathrm{Sub}(\varphi_1) \cup \{(\neg\, \varphi_1)\}$

Def.1.1.4.b vs Def.1.1.7

- **Teorema:**
  - φ es subfórmula de α **ssi** $\varphi \in \mathrm{Sub}(\alpha)$
- Dicho de otra forma

  $(\underline{\forall}\alpha \in \mathrm{PROP} ::$

  $\{\varphi \in \mathrm{PROP} : \varphi \text{ es subfórmula de } \alpha\} = \mathrm{Sub}(\alpha))$
- ¿Qué mecanismo de prueba podemos usar?
- Si usamos inducción, ¿en cuál conjunto?
- Si usamos inducción, ¿en cuál (meta)variable?

Propiedad y Base

- Propiedad a probar. $P(\alpha) :=$
  - $\{\varphi \in PROP : \varphi$ es subfórmula de $\alpha\} = Sub(\alpha)$
- BASE:
  - **T**) Probar $P(\alpha)$, cuando α atómica
- Dem:
  - φ es subfórmula de α (Hipótesis)
  - $\varphi = \alpha$ (Def.1.1.4.b)
  - $\varphi \in Sub(\alpha)$ (Def.1.1.7)
  - Luego,
    - $\{\varphi \in PROP : \varphi$ es subfórmula de $\alpha\} \subseteq Sub(\alpha)$ (Lqqd)

¡Falta probar la otra inclusión !

Un Paso Inductivo

- PI_1)
  $\{\varphi \in$ PROP : φ es subfórmula de $\alpha\} = \text{Sub}(\alpha)$
  - **H**) $P(\alpha)$
  - **T**) $P((\neg\, \alpha))$
- Dem:
  - $\varphi \in \text{Sub}((\neg\, \alpha))$ (Hipótesis)
  - $\varphi = (\neg\, \alpha)$ o $\varphi \in \text{Sub}(\alpha)$ (1.7)
  - **Caso $\varphi = (\neg\, \alpha)$:** φ es subfórmula de $(\neg\, \alpha)$ (1.4.b)
  - **Caso $\varphi \in \text{Sub}(\alpha)$:**
    - φ es subfórmula de α (HI)
    - φ es subfórmula de $(\neg\, \alpha)$ (1.4.b) (lqqd)

¡Falta probar la otra inclusión !

¿Qué se hizo hasta ahora?

- Probamos $P(\alpha)$, cuando α atómica (BASE)
  - En realidad, falta la mitad de esa prueba.
- Probamos $P((\neg\ \alpha))$, teniendo como hipótesis adicional que $P(\alpha)$
  - En realidad, falta la mitad de esa prueba.
- FALTA Probar $P((\varphi_1 \square \varphi_2))$, teniendo como hipótesis adicional que $P(\varphi_1)$ y $P(\varphi_2)$
- En este punto se está en condiciones de aplicar el Principio de Inducción Primitiva.

Aplicacion del Principio de Induccion Primitiva

- El PIP es el (meta)teorema que justifica

  $(\underline{\forall}\alpha \in PROP :: P(\alpha))$

- T) $(\underline{\forall}\alpha \in PROP :: \{\varphi \in PROP : \varphi$ es subfórmula de $\alpha\} = Sub(\alpha))$
- Dem.
  - Las demostraciones anteriores probaron que la propiedad cumple con las hipótesis del P.I.P de Prop, por lo que se deduce que la propiedad se cumple para todo elemento de Prop. (Lqqd)

Prop y Secuencias de Formación

<u>Teorema</u> <u>1.1.5</u>

PROP es el conjunto de todas palabras de $\Sigma_{PROP}{}^{*}$ que tienen secuencia de formación.

<u>Corolario</u>:

Sea P una propiedad de $\Sigma_{PROP}{}^{*}$. Para demostrar que:

Para todo $\alpha \in$ PROP se cumple $P(\alpha)$

podemos hacer la prueba:

– por inducción primitiva en α

– por inducción en la longitud de la secuencia de formación de α

Funciones recursivas: sustitución

$\alpha[\varphi / p_i]$ denota la fórmula obtenida de sustituir todas las ocurrencias de p_i en la fórmula α por la fórmula φ.

Se define por recursión primitiva en α.

<u>Def</u> [sustitución de una variable por una fórmula]

_ [_/_] : PROP x PROP x P $\rightarrow$ PROP

$$\bot\,[\varphi / p_i] = \bot$$

$$p_k\,[\varphi / p_i] = \begin{cases} \varphi & \text{si } i = k \\ p_k & \text{si } i \neq k \end{cases}$$

$$(\alpha \square \beta)\,[\varphi / p_i] = (\alpha\,[\varphi / p_i] \square \beta\,[\varphi / p_i]),\ \square \in C$$

$$(\neg\alpha)\,[\varphi / p_i] = (\neg\,\alpha[\varphi / p_i])$$

Convenciones sintácticas

- Omitimos paréntesis alrededor de la negación
- $\neg$ es más fuerte que $\wedge$
- $\wedge$ y $\vee$ tienen la misma precedencia y son más fuertes que $\rightarrow$
- $\rightarrow$ y $\leftrightarrow$ tienen igual precedencia
- Conectivos de igual precedencia se asocian a la derecha

Ejemplos:

- $p_1 \rightarrow p_3$ se lee como $(p_1 \rightarrow p_3)$
- $(p_1 \rightarrow p_2) \vee \perp \wedge \neg p_5$ se lee como $((p_1 \rightarrow p_2) \vee (\perp \wedge \neg p_5))$
- $p_1 \rightarrow p_3 \rightarrow p_2$ se lee como $(p_1 \rightarrow (p_3 \rightarrow p_2))$

Traducción al lenguaje lógico

- Las proposiciones simples se traducen como letras de proposición (elementos de P).
- Ejemplos:
  - Ayer llovió en Paysandú → p_0
  - El intendente se mojó → p_1
  - El Sol gira alrededor de la Tierra → p_2
  - 2 . 3 = 6 → p_3
  - 6 es impar → p_4
  - El sucesor de 3 es primo → p_5

Traducción al lenguaje Lógico (cont.)

- Las proposiciones compuestas se traducen usando los conectivos

Ejemplos:

- Si ayer llovió en Paysandú, entonces
- el intendente se mojó ➔ $(p_0 \to p_1)$
- 2 . 3 = 6 y 6 es impar ➔ $(p_3 \wedge p_4)$
- 6 no es impar ➔ $(\neg\, p_4)$.

Traducción al lenguaje Lógico (cont.)

- Algunas proposiciones no tienen una buena traducción a PROP
  – Hay un número natural que es par
  – Todo entero par mayor que cuatro es la suma de dos números primos.

(vamos a necesitar un lenguaje más expresivo: más adelante...)