Parte II: Lógica Proposicional

16 de septiembre de 2015

El conjunto $Prop$

- El miércoles pasado definimos la sintaxis de la lógica proposicional.
- Asumimos un conjunto numerable $\mathcal{V}$ de variables proposicionales y definimos los átomos como $At = \mathcal{V} \cup \{\bot\}$. Las proposiciones eran ciertas palabras sobre $\Sigma = At \cup \{\neg, \vee, \wedge\} \cup \{(,)\}$.
- Al conjunto $Prop$ lo definimos inductivamente:

$A \in At$ Si $A \in At$, entonces $A \in Prop$;
$(\neg P)$ Si $P \in Prop$, entonces $(\neg P) \in Prop$;
$(P \wedge Q)$ Si $P \in Prop$ y $Q \in Prop$, entonces $(P \wedge Q) \in Prop$.
$(P \to Q)$ Si $P \in Prop$ y $Q \in Prop$, entonces $(P \to Q) \in Prop$.
$(P \vee Q)$ Si $P \in Prop$ y $Q \in Prop$, entonces $(P \vee Q) \in Prop$.
$(P \leftrightarrow Q)$ Si $P \in Prop$ y $Q \in Prop$, entonces $(P \vee Q) \in Prop$.

Recursión en $Prop$

- El hecho de definir $Prop$ inductivamente nos permite definir funciones de manera recursiva.
- Para definir $f : Prop \to X$ debemos definir:

$$f(P) = \begin{cases} x_i & \text{si } P = p_i \\ y & \text{si } P = \bot \end{cases}$$

- Y tenemos varios casos recursivos, uno para cada conectivo; en el caso de la negación:

$$f((\neg P)) = \ldots f(P) \ldots$$

- Para las fórmulas de la forma $(P_1 \square P_2)$, podemos utilizar la llamada recursiva tanto en P_1 como en P_2:

$$f((P_1 \square P_2)) = \ldots f(P_1) \ldots f(P_2) \ldots$$

Recursión, ejemplos

- Cantidad de conectivos en una proposición.
- Cantidad de símbolos "(" y ")" en una proposición:
- Sustitución del símbolo p_i por la proposición Q:

$$
\begin{aligned}
-\,[Q/p_i] &: Prop \to Prop \\
p_j\,[Q/p_i] &= \begin{cases} Q & \text{si } i = j \\ p_j & \text{si } i \neq j \end{cases} \\
\bot\,[Q/p_i] &= \bot \\
(\neg\, P)\,[Q/p_i] &= (\neg\, P\,[Q/p_i]) \\
(P_1 \,\square\, P_2)\,[Q/p_i] &= (P_1\,[Q/p_i] \,\square\, P_2\,[Q/p_i])
\end{aligned}
$$

Inducción en sub-fórmulas

- Así como podemos definir funciones recursivamente, también podemos probar propiedades sobre $Prop$ utilizando *inducción*.
- Para probar que todo $P \in Prop$ satisface un predicado A, entonces alcanza con probar:

  $P \in At$ $A(P)$, para todo $P \in At$;
  $(\neg P)$ Si $A(P)$, entonces $A((\neg P))$;
  $(P \square Q)$ Si $A(P)$ y $A(Q)$, entonces $A((P \square Q))$.

- La clase pasada utilizamos este principio para probar: "para toda $P \in Prop$, $paren(P) = 2 * con(P)$".

Semántica

- Las proposiciones representan afirmaciones, es decir podemos asignarles un valor de verdad.
- Por ejemplo, establecimos que $\perp$ representaba la proposición siempre falsa.
- Para otras, *hoy llueve* (representada por alguna p_i) no podemos fijar su valor de verdad de una vez y para siempre.
- Si tenemos una proposición formada por un conectivo, $(\neg p_1)$, entonces su valor será **verdadera** si el valor de p_1 es falsa, y viceversa.
- En general, el valor de $(P \square Q)$, dependerá del valor de P y del de Q. Esos valores dependerán a su vez del valor de las variables que aparezcan en ellas.

Tabla de verdad

- En una tabla de verdad esa dependencia se hace patente.
- Cada línea de la tabla de verdad muestra una asignación de valores a las variables proposicionales:

p_0	p_1	p_2	$(\neg p_0)$	$((\neg p_0) \wedge p_1)$	$(((\neg p_0) \wedge p_1) \rightarrow p_2)$
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	1	1	0
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

Semántica

- Los valores de las columnas que no son variables queda determinada unívocamente de las anteriores.
- Una *asignación* es una función $f : \mathcal{V} \to \{0, 1\}$.
- En la tabla de verdad, ¿listamos todas las asignaciones?
- Más adelante veremos por qué alcanza con algunas asignaciones.

Semántica

- Dada una asignación f, el valor de verdad de una proposición se define recursivamente:

$$
\begin{aligned}
[\![-]\!]_f &: \mathit{Prop} \to \mathbb{N} \\
[\![p_i]\!]_f &= f\, p_i \\
[\![\bot]\!]_f &= 0 \\
[\![(\neg P)]\!]_f &= 1 - [\![P]\!]_f \\
[\![(P_1 \wedge P_2)]\!]_f &= \mathrm{mín}([\![P_1]\!]_f, [\![P_2]\!]_f) \\
[\![(P_1 \to P_2)]\!]_f &= \mathrm{máx}(1 - [\![P_1]\!]_f, [\![P_2]\!]_f) \\
[\![(P_1 \vee P_2)]\!]_f &= \mathrm{máx}([\![P_1]\!]_f, [\![P_2]\!]_f) \\
[\![(P_1 \leftrightarrow P_2)]\!]_f &= 1 - (\mathrm{máx}([\![P_1]\!]_f, [\![P_2]\!]_f) - \mathrm{mín}([\![P_1]\!]_f, [\![P_2]\!]_f))
\end{aligned}
$$

Coincidencia

Antes habíamos resaltado que en la tabla de verdad no había asignaciones, pues nos fijábamos sólo en el valor para algunas variables proposicionales. Esto lo podemos realizar gracias al siguiente

Teorema (de coincidencia)

Si $f, f' : \mathcal{V} \to \{0, 1\}$ *coinciden en las variables que ocurren en* P*, entonces* $[\![P]\!]_f = [\![P]\!]_{f'}$.

Por inducción en P.

Lema (de sustitución)

Sea f *una asignación, tal que* $[\![Q_1]\!]_f = [\![Q_2]\!]_f$*. Entonces* $[\![P\,[Q_1/p_i]]\!]_f = [\![P\,[Q_2/p_i]]\!]_f$.

Por inducción en sub-fórmulas. En el pizarrón.

Validez

- Dada una asignación, una proposición P puede tener valor de verdad 0 o 1
- Si para la asignación f, $[\![P]\!]_f = 1$, decimos que f *satisface* P.
- P es una *tautología* ó *válida* si es satisfecha por toda asignación.
- Sea $\Gamma \subseteq Prop$, decimos que f *es una asignación* de Γ, si f satisface P, para toda $P \in \Gamma$.
- ¿Existe alguna asignación de $Prop$?

Consecuencia lógica

- Si P es una tautología, escribimos $\models P$.
- Decimos que P es consecuencia lógica de Γ si para toda asignación f de Γ, $[\![P]\!]_f = 1$. Lo escribimos

$$\Gamma \models P$$

- Como toda asignación f es del vacío, entonces $\models P$ es lo mismo que $\emptyset \models P$.

Consecuencia lógica

- $\models (P \to P)$.
- Si $P \in \Gamma$, entonces $\Gamma \models P$.
- $\{P, (P \to Q)\} \models Q$.
- $\not\models p_1$.

Teorema (de sustitución)

Si $\models (Q_1 \leftrightarrow Q_2)$, *entonces* $\models (P\,[Q_1/p_i] \leftrightarrow P\,[Q_2/p_i])$.

Sea f una asignación, podemos ver que $[\![(Q_1 \leftrightarrow Q_2)]\!]_f = 1$ sii $[\![Q_1]\!]_f = [\![Q_2]\!]_f$. Utilizando el teorema de sustitución, sabemos $[\![P\,[Q_1/p_i]]\!]_f = [\![P\,[Q_2/p_i])]\!]_f$. Por lo tanto, $[\![(P\,[Q_1/p_i] \leftrightarrow P\,[Q_1/p_i])]\!]_f$.

Completitud funcional

- Una tabla de verdad con n renglones describe una función $\{0,1\}^n \to \{0,1\}$.
- Dada una función $F\colon \{0,1\}^n \to \{0,1\}$ existe una proposición P tal que la tabla de verdad de P sea justamente la función F?
- Por ejemplo:

p_0	p_1	p_2	
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

- Notemos que podemos construir la proposición:

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

Completitud funcional

- Notemos que sólo utilizamos $\neg$, $\wedge$, $\vee$ y que podemos hacerlo para cualquier función.
- Decimos que un conjunto de conectivos es *funcionalmente completo*, si podemos definir cualquier función $\{0,1\}^n \to \{0,1\}$ como una proposición que sólo utilice esos conectivos.
- Si sabemos que un conjunto es funcionalmente completo, entonces podemos probar que otro conjunto lo es, utilizando equivalencias lógicas.
- Ejemplo, como $[\![(P \vee Q)]\!]_f = [\![((\neg P) \wedge (\neg Q))]\!]_f$, entonces $\{\neg, \wedge\}$ es funcionalmente completo.