Lógica Computacional 2020-1 Lógica Proposicional

Sara Doris Montes Incin

6 de febrero de 2020

Contenido

Lógica Proposicional

2 Lógica Proposicional en Haskell

Semántica LP

Lógica Proposicional

¿Qué es la lógica proposicional?

Es un sistema formal:

- Lenguaje formal
- Axiomas
- Reglas de inferencia
- Semántica formal

¿Para qué queremos la lógica Proposicional?

Decidir si un argumento es correcto o no



Lenguaje formal

```
 \begin{array}{l} \mathsf{LP} ::= \mathit{VarProp} \mid \mathit{ConstProp} \\ \mathsf{LP} ::= (\neg \ \mathsf{LP}) \mid (\mathsf{LP} \land \mathsf{LP}) \mid (\mathsf{LP} \lor \mathsf{LP}) \mid (\mathsf{LP} \Rightarrow \mathsf{LP}) \mid (\mathsf{LP} \Leftrightarrow \mathsf{LP}) \\ \mathsf{VarProp} ::= \mathit{Var}_1 \mid \mathit{Var}_2 \mid \mathit{Var}_3 \mid ... \mid \mathit{Var}_n \\ \mathsf{ConstProp} ::= \mathsf{True} \mid \mathsf{False} \\ \end{array}
```

Sintaxis

Más en especifico la sintaxis que usaremos está dada de la siguiente manera:

```
LP ::=<ProposicionAtomica> | \negLP | (LP \land LP)| (LP \lor LP) | (LP \Rightarrow LP) | (LP \Leftrightarrow LP)
```

- <ProposicionAtomica> ::= T | F | <VariableProposicional>
- <VariableProposicional>::= v <Indice>
- <Indice>::= [$i|i \in N$]

Teorema Interdefinibilidad

Todas las conectivas se pueden definir en términos de:

- -
- y alguna de las siguientes:
 - ^
 - \
 - $\bullet \Rightarrow$

Definición de LP en Haskell

```
-- Tipo de dato indice

type Indice = Int

-- Tipo de dato fórmula

data LP = Var Indice

| T | F | Neg LP

| And LP LP | Or LP LP

| Imp LP LP deriving (Eq, Show)
```

¿Por qué no usamos los operadores de Haskell?

Porque necesitamos que las expresiones NO se evalúen



Función sobre LP

```
Número de Operadores \begin{array}{l} num0p :: LP \to Int \\ num0p \; phi \; = \; case \; phi \; of \\ T \to 0 \\ F \to 0 \\ Var \; v \to 0 \\ Neg \; alpha \to \; num0p \; alpha \; + \; 1 \\ And \; alpha \; beta \to \; num0p \; alpha \; + \; num0p \; beta \; + \; 1 \\ \end{array}
```

Quita Implicaciones

Firma de la función: $quitaImp:: LP \rightarrow LP$

Semántica lógica proposicional

¿Cómo sabemos si una fórmula es verdadera? Se expresa en tablas de verdad

Cada operador tiene su tabla de verdad

Entonces el valor de verdad depende de los valores de verdad asignados a las variables proposicionales y de los conectivos utilizados

Evaluación

Es una función e: $VarProp \rightarrow \{True, False\}$

Ejemplo:

$$e(Var_1) = True$$

e satisface una fórmula de LP

Se define recursivamente. Sea $\phi \in \mathsf{LP}$

- **1** Si $\phi =$ True, entonces $e \models \phi$
- 2 Si ϕ = False, entonces $e \not\models \phi$
- **3** Si $\phi = v \in VarProp$, entonces $e \models v$ si e(v) = 1
- Si $\phi = \neg \alpha$ donde $\alpha \in \mathsf{LP}$, entonces $e \models \phi$ sii $e \not\models \phi$
- **3** Si $\phi = \alpha \land \beta$ donde $\alpha, \beta \in \mathsf{LP}$, entonces $e \models \phi$ sii $e \models \alpha$ y $e \models \beta$

Ejemplo

Sea
$$\phi = Var_1 \wedge Var_2$$
 y $e(Var_1) = \text{True y } e(Var_2) = \text{False}$

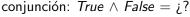
$$e$$
 satisface a ϕ ($e \models \phi$)?

Nos fijamos en la definición anterior:

¿En cuál caso caemos? ¡En el 5! Ya que el operador es una conjunción.

$$e \models Var_1 y e \models_2$$

Para que $e \models Var_1$ por definición se debe cumplir que $e(Var_1) = \text{True y sí}$ se cumple, pero $e(Var_2) = \text{False}$, y por la tabla de verdad de la conjunción: $True \land False = \div ?$



Modelo

Conjunto de evaluaciones

