Lógica Computacional 2020-1
Lógica Proposicional

Sara Doris Montes Incin

6 de febrero de 2020

Contenido

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

LP::= *VarProp* | *ConstProp*
LP::= (¬ LP) | (LP ∧ LP) | (LP ∨ LP) | (LP ⇒ LP) | (LP ⇔ LP)
VarProp::= Var_1 | Var_2 | Var_3 | ... | Var_n
ConstProp::= True | False

Sintaxis

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

LP ::=<ProposicionAtomica> | $\neg$LP | (LP $\wedge$ LP)| (LP $\vee$ 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:

- $\neg$
- y **alguna** de las siguientes:
  - $\wedge$
  - $\vee$
  - $\Rightarrow$

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

```
numOp :: LP → Int
numOp phi = case phi of
T → 0
F → 0
Var v → 0
Neg alpha → numOp alpha + 1
And alpha beta → numOp alpha + numOp beta + 1
```

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 $\phi \in$ LP

1. Si $\phi =$ True, entonces $e \models \phi$
2. Si $\phi =$ False, entonces $e \not\models \phi$
3. Si $\phi =$ v $\in$ VarProp, entonces $e \models$ v si e(v) $= 1$
4. Si $\phi = \neg\alpha$ donde $\alpha \in$ LP, entonces $e \models \phi$ sii e $\not\models \phi$
5. Si $\phi = \alpha \wedge \beta$ donde $\alpha, \beta \in$ LP, entonces $e \models \phi$ sii e $\models \alpha$ y $e \models \beta$

Ejemplo

Sea $\phi = Var_1 \land Var_2$ y $e(Var_1) =$ True y $e(Var_2) =$ 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 $e(Var_1) =$ True y sí se cumple, pero $e(Var_2) =$ False, y por la tabla de verdad de la conjunción: $True \land False =$ ¿?

Modelo

Conjunto de evaluaciones