LENGUAJE LOGIC

especificaciones del lenguaje

Equipo 1

Sintaxis La sintaxis concreta del lenguaje se especifica mediante una gramática libre de contexto. Una producción de la forma β → ∝ establece que la cadena ∝ se deriva a partir de la cadena no-terminal β.

Sintaxis concreta

$Logic := atom$
$atom := \{A - Z\} \cup \{a - z\}$
$Logic := \neg Logic$
$Logic := (Logic \wedge Logic)$
$Logic := (Logic \vee Logic)$
$Logic := (Logic \rightarrow Logic)$

Sintaxis abstracta

l-atom (var)
l-not (expr)
l-and (exp1 exp2)
l-or (exp1 exp2)
l-impl (exp1 exp2)

Semántica La interpretación de expresiones del lenguaje Logic de acuerdo a su sintaxis abstracta, consiste en afirmaciones de la forma (value-of expr ρ) = v, donde expr es una expresión del lenguaje, ρ un entorno y v un valor expresado.

Para definir un entorno ρ, es preciso presentar primero al entorno vacío ρ_0 , el cual actúa como punto de partida para ampliar un entorno ρ vinculando una variable x con un valor v, indicado como [x : v]ρ. También se dispone de un método para aplicar estos entornos a las variables, representado por ρ(x). Los entornos funcionan como funciones que asocian variables a sus valores denotados, y ρ0 permanece sin definir para cualquier variable.

Interpretación de expresiones

(value-of (l-atom var) ρ) = (bool-expressed (denoted->bool val))

(value-of (l-not expr) ρ) = (if (equal? #t (expressed->bool (value-of expr env)))
(bool-expressed #f)
(bool-expressed #t))

(value-of (l-and exp1 exp2) ρ) = (if (equal? #t (and (expressed->bool (value-of exp1 env)) (
expressed->bool (value-of exp2 env))))
(bool-expressed #t)
(bool-expressed #f))

(value-of (l-or exp1 exp2) ρ) = (if (equal? #t (or (expressed->bool (value-of exp1 env))
(expressed->bool (value-of exp2 env))))
(bool-expressed #t)
(bool-expressed #f))

(value-of (l-impl exp1 exp2) ρ) = (if (equal? #f (expressed->bool (value-of exp2 env)))
(bool-expressed #f)
(bool-expressed #t))