



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Lenguajes de Programacion Examen Parcial V



■ Edgar Montiel Ledesma
317317794

■ Carlos Daniel Cortes Jimenez
420004846

1. Sean A, B, C, D, E con la siguiente relacion de subtipado:

$$E \leq B$$

$$D \leq B$$

$$B \leq A$$

$$C \leq A$$

Para los siguientes subtipados diga si son validos o no. En caso de ser validos, muestre la derivacion formal, en caso de ser invalidos, añada una sola regla que la haga valida y haga la derivacion formal.

a) $((A + B) \rightarrow \{y : E, x : D\}) + ((A \times A) \rightarrow E) \leq ((A + B) \rightarrow \{x : C\}) + ((A \times B) \rightarrow C)$

- Esta expresión es válida, y la derivación formal es la siguiente:

$$\begin{aligned} & ((A+B) \rightarrow \{y:E, x:D\}) + ((A \times A) \rightarrow E) \\ & \leq ((A+B) \rightarrow \{x:C\}) + ((A \times B) \rightarrow C) \end{aligned}$$

- Por regla de subtipado:

$$(A+B) \rightarrow \{y:E, x:D\} \leq (A+B) \rightarrow \{x:C\}$$

- Por regla de subtipado:

$$(A \times A) \rightarrow E \leq (A \times B) \rightarrow C$$

- Por reglas de subtipado anteriores:

$$A \leq B \text{ (de } B \leq A)$$

- Por regla de subtipado:

$$A \times A \leq A \times B$$

- Por reglas de subtipado anteriores:

$$E \leq B \text{ (de } B \leq A)$$

- Por regla de subtipado:

$$(A \times A) \rightarrow E \leq (A \times B) \rightarrow C$$

- Por regla de subtipado:

$$\begin{aligned} & (A+B) \rightarrow \{y:E, x:D\} + (A \times A) \rightarrow E \\ & \leq (A+B) \rightarrow \{x:C\} + (A \times B) \rightarrow C \end{aligned}$$

b) $(E \times D \times A \rightarrow \{x : A\}) \rightarrow C \leq (B \times A \rightarrow \{x : A, y : D\}) \rightarrow A$

Esta expresión no es inválida. Para hacerla válida, agregaremos la regla de subtipado $A \leq C$.

- o La derivación formal es la siguiente:

$$\begin{array}{l} (E \times D \times A \rightarrow \{x:A\}) \rightarrow C \\ \leq (B \times A \rightarrow \{x:A, y:D\}) \rightarrow A \end{array}$$

- o Por regla de subtipado:

$$E \times D \times A \rightarrow \{x:A\} \leq B \times A \rightarrow \{x:A, y:D\}$$

- o Por reglas de subtipado anteriores:

$$A \leq C \text{ (nueva regla agregada)}$$

- o Por regla de subtipado:

$$E \times D \times A \rightarrow \{x:A\} \leq B \times A \rightarrow \{x:A, y:D\}$$

- o Por regla de subtipado:

$$(E \times D \times A \rightarrow \{x:A\}) \rightarrow C \leq (B \times A \rightarrow \{x:A, y:D\}) \rightarrow A$$

Ahora, con la adición de la regla $A \leq C$, la expresión se vuelve válida.

2. Considera los tipos Rectangulo parametrizado por la altura y el ancho y Cuadrado parametrizado por el tamaño de uno de los lados.

- a) Define los tipos anteriores mediante tipos suma y producto.

- o Rectangulo: Parametrizado por la altura y el ancho.

$$\text{Rectangulo} = \text{Altura} \times \text{Ancho}$$

- o Cuadrado: Parametrizado por el tamaño de uno de los lados.

$$\text{Cuadrado} = \text{Lado}$$

- b) ¿Que relacion de subtipado existe entre estos tipos? es decir ¿cual de estos es el subtipo y cual el supertipo? justifica tu respuesta, una forma de justificar la respuesta es explicar como seria el proceso de coercion entre el subtipo y el supertipo.

Deminimos: Si consideramos que Cuadrado es un caso especial de Rectangulo (un rectángulo con lados iguales), entonces Cuadrado sería el subtipo y Rectangulo el supertipo. Sin embargo, si consideramos que Rectangulo y Cuadrado son tipos independientes, no habría una relación de subtipado entre ellos.

En el primer caso, la coerción sería natural ya que un Cuadrado puede tratarse como un Rectángulo sin ningún problema. La altura y el ancho serían iguales (el tamaño del lado) en el caso de un Cuadrado. En el segundo caso, no habría coerción directa entre los tipos.

- c) De acuerdo a la relacion de subtipado del inciso anterior, ordena los siguientes tipos funcion: Ordenemos los tipos de acuerdo a la relación de subtipado que describimos en el inciso b:

- o Rectangulo \rightarrow Rectangulo - La relación es reflexiva, por lo que es válida.

- o Rectangulo \rightarrow Cuadrado - Esto sería válido si consideramos que Cuadrado es un caso especial de Rectangulo.

- o Cuadrado \rightarrow Rectangulo - Esto también sería válido si consideramos que Cuadrado es un caso especial de Rectangulo.

- Cuadrado \rightarrow Cuadrado - La relación es reflexiva, por lo que es válida.

En resumen, el orden sería:

Cuadrado \rightarrow Rectangulo \rightarrow Rectangulo \rightarrow Cuadrado

Rectangulo \rightarrow Rectangulo

Rectangulo \rightarrow Cuadrado

Cuadrado \rightarrow Rectangulo

Cuadrado \rightarrow Cuadrado

3. El objetivo de este ejercicio es definir el siguiente mini lenguaje de expresiones aritméticas y booleanas MinEAB en JPP:

$$e ::= n \mid \text{true} \mid \text{false} \mid e + e \mid e < e$$

Se busca que se sigan los siguientes puntos.

- a) Defina una clase Expr que incluya los siguientes metodos:

- isAtom que devuelve true si la expresion no tiene subexpresiones propias.
- lsub que devuelve la subexpresion izquierda de una expresion no atomica.
- rsub que devuelve la subexpresion derecha de una expresion no atomica.
- eval que devuelve el valor de la expresion.

Esta clase debe ser abstracta en el sentido de que no tiene atributos y por lo tanto sus metodos no hacen nada pero deben ser definidos. Es decir, se considera a Expr como una interfaz.

- b) Defina una clase NumExpr que extienda a Expr y de forma que sus instancias correspondan a numeros.
- c) Defina una clase BoolExpr que extienda a Expr y de forma que sus instancias correspondan a booleanos.
- d) Defina una clase SumExpr que extienda a Expr y de forma que sus instancias correspondan a sumas.
- e) Defina una clase LTExpr que extienda a Expr y de forma que sus instancias correspondan a comparaciones de orden.
- f) De ejemplos de instancias de cada una de las clases.

Puede suponer definida una clase Value cuyas instancias sean los valores del lenguaje, ya sea naturales o booleanos. Es decir, Value es esencialmente el tipo $\text{Nat} + \text{Bool}$. Ademas de otras clases primitivas con los metodos que requiera, especificandolos previamente. Tambien puede utilizar convenientemente la constante de error en cualquier metodo.