

El lenguaje de Expresiones Aritméticas No Tipificadas \mathcal{T}

José de Jesús Lavalle Martínez

<http://aleteya.cs.buap.mx/~jllavalle/>

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Licenciatura en Ciencias de la Computación
Fundamentos de Lenguajes de Programación
CCOS 255

Otoño 2020

- 1 Introducción
- 2 Sintaxis del lenguaje \mathcal{T}
- 3 Inducción sobre términos
- 4 Semántica
- 5 Ejercicios

El lenguaje utilizado en esta parte del curso contiene solo un puñado de formas sintácticas:

El lenguaje utilizado en esta parte del curso contiene solo un puñado de formas sintácticas:

- las constantes booleanas verdadero y falso,

El lenguaje utilizado en esta parte del curso contiene solo un puñado de formas sintácticas:

- expresiones condicionales,

El lenguaje utilizado en esta parte del curso contiene solo un puñado de formas sintácticas:

- la constante numérica 0,

El lenguaje utilizado en esta parte del curso contiene solo un puñado de formas sintácticas:

- los operadores aritméticos succ (sucesor) y pred (predecesor),

El lenguaje utilizado en esta parte del curso contiene solo un puñado de formas sintácticas:

- y una operación prueba si es cero que devuelve verdadero. cuando se aplica a 0 y falso cuando se aplica a algún otro número.

Estas formas se pueden resumir de forma compacta con la siguiente gramática:

$t ::=$

true

false

if t then t else t

0

succ t

pred t

iszero t

términos:

constante true

constante false

condicional

constante cero

sucesor

predecesor

prueba si es cero

- Un programa en el lenguaje \mathcal{T} es un término construido a partir de las formas dadas por la gramática anterior.

- A continuación se muestran dos ejemplos de programas, junto con los resultados de su **evaluación**:
 - `if false then 0 else 1` \rightarrow 1
 - `iszero (pred (succ 0))` \rightarrow true

Hay varias formas equivalentes de definir la sintaxis de nuestro lenguaje. Ya hemos visto una por medio de la gramática presentada.

Esta gramática es en realidad sólo una notación compacta para la siguiente definición inductiva:

Definición 1 (Términos inductivamente)

El conjunto de términos es el conjunto más pequeño \mathcal{T} tal que:

- 1 $\{\text{true}, \text{false}, 0\} \subseteq \mathcal{T}$,
- 2 Si $t_1 \in \mathcal{T}$ entonces $\{\text{succ } t_1, \text{pred } t_1, \text{iszero } t_1\} \subseteq \mathcal{T}$,
- 3 Si $t_1, t_2, t_3 \in \mathcal{T}$ entonces $\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \in \mathcal{T}$

Definición 2 (Términos por reglas de inferencia)

El conjunto de términos se define mediante las siguientes reglas:

$$\begin{array}{c}
 \overline{\text{true} \in \mathcal{T}} \\
 \frac{t_1 \in \mathcal{T}}{\text{succ } t_1 \in \mathcal{T}}
 \end{array}
 \qquad
 \begin{array}{c}
 \overline{\text{false} \in \mathcal{T}} \\
 \frac{t_1 \in \mathcal{T}}{\text{pred } t_1 \in \mathcal{T}}
 \end{array}
 \qquad
 \begin{array}{c}
 \overline{0 \in \mathcal{T}} \\
 \frac{t_1 \in \mathcal{T}}{\text{iszero } t_1 \in \mathcal{T}}
 \end{array}$$

$$\frac{t_1 \in \mathcal{T} \quad t_2 \in \mathcal{T} \quad t_3 \in \mathcal{T}}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \in \mathcal{T}}$$

Definición 3 (Términos concretamente)

Para cada número natural i , defina un conjunto S_i como sigue:

$$S_0 = \emptyset$$

$$S_{i+1} = \{\text{true, false, 0}\}$$

$$\cup \{\text{succ } t_1, \text{pred } t_1, \text{iszero } t_1 \mid t_1 \in S_i\}$$

$$\cup \{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \mid t_1, t_2, t_3 \in S_i\}.$$

Finalmente, sea

$$S = \bigcup_{i \geq 0} S_i.$$

Podemos definir el tipo T en ML para el lenguaje \mathcal{T} de la siguiente manera:

Podemos definir el tipo T en ML para el lenguaje \mathcal{T} de la siguiente manera:

```
datatype T = t | f | z |  
            sc of T | pd of T | isz of T |  
            ifte of T * T * T;
```

La caracterización del lenguaje \mathcal{T} mediante la Definición 1 nos permite:

La caracterización del lenguaje \mathcal{T} mediante la Definición 1 nos permite:

- Definir funciones recursivas sobre los términos del lenguaje.

La caracterización del lenguaje \mathcal{T} mediante la Definición 1 nos permite:

- Definir funciones recursivas sobre los términos del lenguaje.
- Hacer demostraciones por inducción de propiedades que tienen los términos del lenguaje.

Definición 4

El conjunto de constantes que aparecen en un término t , que escribimos $Consts(t)$, está definido como sigue:

$$Consts(t) = \begin{cases} \{t\} & \text{si } t = \text{true, false o } 0 \\ Consts(t_1) & \text{si } t = \text{succ } t_1, \text{ pred } t_1 \text{ o iszero } t_1 \\ Consts(t_1) \cup Consts(t_2) \cup Consts(t_3) & \text{si } t = \text{if } t_1 \text{ then } t_2 \text{ else } t_3 \end{cases}$$

Definición 5

El tamaño de un término t , que escribimos $size(t)$, está definido como sigue:

$$size(t) = \begin{cases} 1 & \text{si } t = \text{true}, \text{ false o } 0 \\ size(t_1) + 1 & \text{si } t = \text{succ } t_1, \text{ pred } t_1 \text{ o iszero } t_1 \\ size(t_1) + size(t_2) + size(t_3) + 1 & \text{si } t = \text{if } t_1 \text{ then } t_2 \text{ else } t_3 \end{cases}$$

Definición 6

La profundidad de un término t , que escribimos $depth(t)$, está definido como sigue:

$$depth(t) = \begin{cases} 1 & \text{si } t = \text{true, false o } 0 \\ depth(t_1) + 1 & \text{si } t = \text{succ } t_1, \text{ pred } t_1 \text{ o iszero } t_1 \\ \max(depth(t_1) + depth(t_2) + depth(t_3)) + 1 & \text{si } t = \text{if } t_1 \text{ then } t_2 \text{ else } t_3 \end{cases}$$

- Definiremos una semántica operacional para \mathcal{T} .

- Definiremos una semántica operacional para \mathcal{T} .
- Lo haremos mediante una **relación de evaluación** sobre términos.

- Definiremos una semántica operacional para \mathcal{T} .
- Lo haremos mediante una **relación de evaluación** sobre términos.
- Escribiremos la relación mediante $t \rightarrow t'$ y se pronuncia “t se evalúa a t' en un paso”.

Definición 7

La **relación de evaluación en un paso** \rightarrow sobre los términos de \mathcal{T} está definida mediante las siguientes reglas:

$$\frac{}{\text{if true then } t_2 \text{ else } t_3 \rightarrow t_2} \text{E-IFTRUE}$$

$$\frac{}{\text{if false then } t_2 \text{ else } t_3 \rightarrow t_3} \text{E-IFFALSE}$$

$$\frac{t_1 \rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3} \text{E-IF}$$

$$\frac{t_1 \rightarrow t'_1}{\text{succ } t_1 \rightarrow \text{succ } t'_1} \text{E-SUCC}$$

$$\frac{}{\text{pred } 0 \rightarrow 0} \text{E-PREDZERO}$$

$$\frac{}{\text{pred (succ } nv_1) \rightarrow nv_1} \text{E-PREDSUCC}$$

$$\frac{t_1 \rightarrow t'_1}{\text{pred } t_1 \rightarrow \text{pred } t'_1} \text{E-PRED}$$

$$\frac{}{\text{iszero } 0 \rightarrow \text{true}} \text{E-ISZEROZERO}$$

$$\frac{}{\text{iszero (succ } nv_1) \rightarrow \text{false}} \text{E-ISZEROSUCC}$$

$$\frac{t_1 \rightarrow t'_1}{\text{iszero } t_1 \rightarrow \text{iszero } t'_1} \text{E-ISZERO}$$

Definición 8

La relación de **evaluación en múltiples pasos** \rightarrow^* es la cerradura reflexiva y transitiva de la relación de evaluación en un paso. Es decir es la relación más pequeña que cumple las siguientes reglas.

$$\frac{}{t \rightarrow^* t}$$
$$\frac{t \rightarrow t'}{t \rightarrow^* t'}$$
$$\frac{t \rightarrow t'' \quad t'' \rightarrow^* t'}{t \rightarrow^* t'}$$

Implemente en ML las funciones dadas por las Definiciones:

- 1 4 (llámele Consts), 5 (llámele size), 6 (llámele depth), 7 (llámele eval) y 8 (llámele evalast).
- 2 Pruebe sus funciones con el archivo “pruebasSemana07.ml” que encontrará en la página del curso.