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

José de Jesús Lavalle Martínez

`http://aleteya.cs.buap.mx/~jlavalle/`

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

Contenido

El lenguaje $\mathcal{T}$ I

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

El lenguaje $\mathcal{T}$ I

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 $\mathcal{T}$ I

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

- la constante numérica 0,

El lenguaje $\mathcal{T}$ I

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 $\mathcal{T}$ I

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.

El lenguaje $\mathcal{T}$ II

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

t ::=	términos:
true	*constante true*
false	*constante false*
if t then t else t	*condicional*
0	*constante cero*
succ t	*sucesor*
pred t	*predecesor*
iszero t	*prueba si es cero*

El lenguaje $\mathcal{T}$ III

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

El lenguaje $\mathcal{T}$ III

- 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

Sintaxis I

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

Sintaxis I

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

Sintaxis I

Definición 1 (Términos inductivamente)

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

1. $\{\text{true, false, 0}\} \subseteq \mathcal{T}$,
2. Si $\mathrm{t}_1 \in \mathcal{T}$ entonces $\{\text{succ } \mathrm{t}_1, \text{pred } \mathrm{t}_1, \text{iszero } \mathrm{t}_1\} \subseteq \mathcal{T}$,
3. Si $\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3 \in \mathcal{T}$ entonces if t_1 then t_2 else $\mathrm{t}_3 \in \mathcal{T}$

Sintaxis II

Definición 2 (Términos por reglas de inferencia)

El conjunto de términos se define mediante las siguientes reglas:

$$\frac{}{\mathsf{true} \in \mathcal{T}} \qquad \frac{}{\mathsf{false} \in \mathcal{T}} \qquad \frac{}{0 \in \mathcal{T}}$$

$$\frac{\mathsf{t}_1 \in \mathcal{T}}{\mathsf{succ}\ \mathsf{t}_1 \in \mathcal{T}} \qquad \frac{\mathsf{t}_1 \in \mathcal{T}}{\mathsf{pred}\ \mathsf{t}_1 \in \mathcal{T}} \qquad \frac{\mathsf{t}_1 \in \mathcal{T}}{\mathsf{iszero}\ \mathsf{t}_1 \in \mathcal{T}}$$

$$\frac{\mathsf{t}_1 \in \mathcal{T} \quad \mathsf{t}_2 \in \mathcal{T} \quad \mathsf{t}_3 \in \mathcal{T}}{\mathsf{if}\ \mathsf{t}_1\ \mathsf{then}\ \mathsf{t}_2\ \mathsf{else}\ \mathsf{t}_3 \in \mathcal{T}}$$

Definición 3 (Términos concretamente)

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

$$
\begin{aligned}
S_0 &= \emptyset \\
S_{i+1} &= \{\mathsf{true},\ \mathsf{false},\ 0\} \\
&\quad \cup \{\mathsf{succ}\ \mathsf{t}_1,\ \mathsf{pred}\ \mathsf{t}_1,\ \mathsf{iszero}\ \mathsf{t}_1 | \mathsf{t}_1 \in S_i\} \\
&\quad \cup \{\mathsf{if}\ \mathsf{t}_1\ \mathsf{then}\ \mathsf{t}_2\ \mathsf{else}\ \mathsf{t}_3 | \mathsf{t}_1, \mathsf{t}_2, \mathsf{t}_3 \in S_i\}.
\end{aligned}
$$

Finalmente, sea

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

Sintaxis IV

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

Sintaxis IV

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;
```

Inducción sobre términos I

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

Inducción sobre términos I

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

- Definir funciones recursivas sobre los términos del lenguaje.

Inducción sobre términos I

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 $Consts(\mathsf{t})$, está definido como sigue:

$$Consts(\mathsf{t}) = \begin{cases} \{\mathsf{t}\} & \text{si t} = \text{true, false o 0} \\ Consts(\mathsf{t}_1) & \text{si t} = \text{succ } \mathsf{t}_1\text{, pred } \mathsf{t}_1 \text{ o iszero } \mathsf{t}_1 \\ Consts(\mathsf{t}_1) \cup Consts(\mathsf{t}_2) \cup Consts(\mathsf{t}_3) & \text{si t} = \text{if } \mathsf{t}_1 \text{ then } \mathsf{t}_2 \text{ else } \mathsf{t}_3 \end{cases}$$

Inducción sobre términos III

Definición 5

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

$$size(\mathsf{t}) = \begin{cases} 1 & \text{si t} = \text{true, false o 0} \\ size(\mathsf{t}_1) + 1 & \text{si t} = \text{succ } \mathsf{t}_1\text{, pred } \mathsf{t}_1 \text{ o iszero } \mathsf{t}_1 \\ size(\mathsf{t}_1) + size(\mathsf{t}_2) + size(\mathsf{t}_3) + 1 & \text{si t} = \text{if } \mathsf{t}_1 \text{ then } \mathsf{t}_2 \text{ else } \mathsf{t}_3 \end{cases}$$

Inducción sobre términos IV

Definición 6

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

$$depth(\mathsf{t}) = \begin{cases} 1 & \text{si } \mathsf{t} = \mathsf{true},\ \mathsf{false} \text{ o } \mathsf{0} \\ depth(\mathsf{t}_1) + 1 & \text{si } \mathsf{t} = \mathsf{succ}\ \mathsf{t}_1,\ \mathsf{pred}\ \mathsf{t}_1 \text{ o } \mathsf{iszero}\ \mathsf{t}_1 \\ \max(depth(\mathsf{t}_1) + depth(\mathsf{t}_2) + depth(\mathsf{t}_3)) + 1 & \text{si } \mathsf{t} = \mathsf{if}\ \mathsf{t}_1\ \mathsf{then}\ \mathsf{t}_2\ \mathsf{else}\ \mathsf{t}_3 \end{cases}$$

Semántica operacional para $\mathcal{T}$ I

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

Semántica operacional para $\mathcal{T}$ I

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

Semántica operacional para $\mathcal{T}$ I

- 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 evalua a t' en un paso".

Semántica operacional para $\mathcal{T}$ II

Definición 7

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

$$\frac{}{\mathsf{if\ true\ then\ t_2\ else\ t_3} \rightarrow \mathsf{t_2}}\ \text{E-IfTrue} \qquad \frac{}{\mathsf{if\ false\ then\ t_2\ else\ t_3} \rightarrow \mathsf{t_3}}\ \text{E-IfFalse}$$

$$\frac{\mathsf{t_1} \rightarrow \mathsf{t'_1}}{\mathsf{if\ t_1\ then\ t_2\ else\ t_3} \rightarrow \mathsf{if\ t'_1\ then\ t_2\ else\ t_3}}\ \text{E-If} \qquad \frac{\mathsf{t_1} \rightarrow \mathsf{t'_1}}{\mathsf{succ\ t_1} \rightarrow \mathsf{succ\ t'_1}}\ \text{E-Succ}$$

$$\frac{}{\mathsf{pred\ 0} \rightarrow \mathsf{0}}\ \text{E-PredZero} \qquad \frac{}{\mathsf{pred\ (succ\ nv_1)} \rightarrow \mathsf{nv_1}}\ \text{E-PredSucc}$$

$$\frac{\mathsf{t_1} \rightarrow \mathsf{t'_1}}{\mathsf{pred\ t_1} \rightarrow \mathsf{pred\ t'_1}}\ \text{E-Pred} \qquad \frac{}{\mathsf{iszero\ 0} \rightarrow \mathsf{true}}\ \text{E-IszeroZero}$$

$$\frac{}{\mathsf{iszero\ (succ\ nv_1)} \rightarrow \mathsf{false}}\ \text{E-IszeroSucc} \qquad \frac{\mathsf{t_1} \rightarrow \mathsf{t'_1}}{\mathsf{iszero\ t_1} \rightarrow \mathsf{iszero\ t'_1}}\ \text{E-IsZero}$$

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.

$$\frac{}{\text{t} \rightarrow^* \text{t}}$$

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

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

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.