

Compiladores 2024-1. Notas

Miguel Carrillo

Dudas sobre estas notas: consultar en el grupo o por correo.

August 28, 2023

Contents

1	Motivación	2
2	Un lenguaje de programación simple	2
2.1	Sintaxis de L0	2
2.2	Semántica de L0	2
2.2.1	Listas de variables y Estados	2
2.2.2	Estados de un programa en L0	3
2.2.3	Significado de expresiones de L0	4
2.2.4	Significado de instrucciones de L0	4
2.2.5	Significado de programas de L0	5

1 Motivación

El tema de compiladores implica el conocimiento de conceptos sobre fundamentos de Ciencias de la Computación. El tema de compiladores no implica solamente “Lenguajes de programación”.

2 Un lenguaje de programación simple

2.1 Sintaxis de L0

Sintaxis de L0 usando notación BNF.

$$\begin{aligned}\langle Var \rangle &::= x \mid y \mid z \\ \langle VarList \rangle &::= \langle Var \rangle \mid \langle Var \rangle ; \langle VarList \rangle \\ \langle Dig \rangle &::= 0 \mid 1 \mid 2 \mid 3 \\ \langle Exp \rangle &::= \langle Dig \rangle \mid \langle Var \rangle \\ \langle Stm \rangle &::= \langle AsigStm \rangle \mid \text{Halt} \\ \langle AsigStm \rangle &::= \langle Var \rangle := \langle Exp \rangle \\ \langle StmList \rangle &::= \langle Stm \rangle \mid \langle Stm \rangle ; \langle StmList \rangle \\ \langle Prog \rangle &::= \text{VAR } \langle VarList \rangle \text{ PROG } \langle StmList \rangle\end{aligned}$$

2.2 Semántica de L0

2.2.1 Listas de variables y Estados

Definición 2.1 (Listas de variables). Sea $lv \in \langle VarList \rangle$.

1. Si $m, n \in \mathbb{N}$, el conjunto de naturales de m a n se define:
 $[m..n] := \{x \in \mathbb{N} \mid m \leq x \leq n\}$.
2. Con $|lv|$ denotamos el número de elementos de lv .
3. Para $i \in [1..|lv|]$, usamos $lv(i)$ para denotar el elemento i -ésimo de lv .
4. Decimos que lv es una lista sin repeticiones si
 $\forall i, j \in [1..|lv|]: i \neq j \rightarrow lv(i) \neq lv(j)$.

Definición 2.2 (Estados para lv). Sea $lv \in \langle \text{VarList} \rangle$ una lista de variables sin repeticiones.

1. El conjunto de estados para lv , se define:

$$S_{lv} := \prod_{i=1}^{|lv|} \text{Dom}(lv(i)),$$

donde $\text{Dom}(v)$ denota el dominio de $v \in \langle \text{Var} \rangle$.

Intuitivamente, $S_{lv} := \{(b_1, b_2, \dots, b_{|lv|}) \mid b_i \in \text{Dom}(lv(i))\}$.

2. Si $\sigma \in S_{lv}$, usamos $\sigma(i)$ para denotar el componente i -ésimo de σ .

Definición 2.3 (σ con b en la posición de v). Sea $lv \in \langle \text{VarList} \rangle$ una lista de variables sin repeticiones.

Y sean $\sigma \in S_{lv}$ es un estado, $v \in lv$, y $b \in \text{Dom}(v)$.

Definimos el estado “ σ con b en la posición de v ”, $\sigma[v \leftarrow b]$, mediante las posiciones de lv :

$$\forall i[1..|lv|]: \sigma[v \leftarrow b](i) := \begin{cases} b & \text{si } v = lv(i) \\ \sigma(i) & \text{si } v \neq lv(i) \end{cases}$$

Ejemplo 2.1 (Estados para lv). Sea $lv := [x, y, z]$ (todas las variables posibles en L0).

Si asumimos que, para $v \in \{x, y, z\}$ $\text{Dom}(v) = [0..9]$,

entonces, el conjunto de estados para lv está dado por

$$S_{lv} := \{(b_1, b_2, b_3) \mid b_i \in [0..9]\}.$$

Es decir, $\#(S_{lv}) = 10^3$. El número de estados es exponencial respecto al número de variables.

2.2.2 Estados de un programa en L0

Definición 2.4 (Estados de un programa de L0). Sea $p \in \langle \text{Prog} \rangle$ un programa de L0, $p = \text{VAR } lv \text{ PROG } ls$.

El conjunto de estados de p , S_p , se define:

$$S_p := S_{nub(lv)} \cup \{\omega\},$$

donde:

- i) $nub(lv)$ elimina en lv las últimas repeticiones.
- ii) ω es un estado especial producido por Halt.

2.2.3 Significado de expresiones de L0

Definición 2.5 (Significado de expresiones de L0). XXX MEJOR como en la siguiente def.

Sean: $lv \in \langle VarList \rangle$ una lista de variables sin repeticiones, $\sigma \in S_{lv}$, y $e \in \langle Exp \rangle$ una expresión.

El significado de e en σ , $\llbracket e \rrbracket_\sigma$, se define por casos, según e :

1. Si $e \in \langle Dig \rangle$, $\llbracket e \rrbracket_\sigma := e \in \mathbb{N}$.
2. Si $e \in \langle Var \rangle$, $\llbracket e \rrbracket_\sigma := \begin{cases} \sigma(i) & \text{si } \exists i \in [1..|lv|]: e = lv(i) \\ \perp & \text{si no.} \end{cases}$.

Observación: Aquí \perp denota “indefinido”

2.2.4 Significado de instrucciones de L0

Definición 2.6 (Significado de instrucciones en L0). Sea $p \in \langle Prog \rangle$ un programa de L0, $p = \text{VAR } lv \text{ PROG } ls$. Sea S_p el conjunto de estados de p .

Sean $\sigma \in S_p$ un estado de p . Y sea $s \in \langle Stm \rangle$ una instrucción de L0.

El significado s en σ se define por casos según σ y s :

1. Si $\sigma = \omega$, entonces $\llbracket s \rrbracket_\sigma := \omega$.
2. Si $\sigma \neq \omega$, entonces $\llbracket s \rrbracket_\sigma$ se define mediante la estructura de s :
 - (a) Si $s = \text{Halt}$,
entonces $\llbracket s \rrbracket_\sigma := \omega$.
 - (b) Si $s = v := e$ (*con:* $v \in \langle Var \rangle$, y $e \in \langle Exp \rangle$),
entonces $\llbracket s \rrbracket_\sigma := \text{CASES XXX } \sigma[v \leftarrow \llbracket e \rrbracket_\sigma]$.

Definición 2.7 (Significado de la lista de instrucciones de un programa). Sea $p \in \langle Prog \rangle$ un programa de L0, $p = \text{VAR } lv \text{ PROG } ls$. Sea S_p el conjunto de estados de p . Y sea $\sigma \in S_p$ un estado del programa p .

Observe que $ls \in \langle StmList \rangle$ es la lista de instrucciones del programa p .

El significado de ls en σ , $\llbracket ls \rrbracket_\sigma$, se define recursivamente mediante la estructura de ls :

1. Si $ls = s \in \langle Stm \rangle$,
entonces $\llbracket ls \rrbracket_\sigma := \llbracket s \rrbracket_\sigma$

2. Si $ls = s ; ls'$ (con: $s \in \langle Stm \rangle$ y $ls' \in \langle StmList \rangle$),
entonces $\llbracket ls \rrbracket_\sigma := \llbracket ls' \rrbracket_{\sigma'}$, donde $\sigma' = \llbracket s \rrbracket_\sigma$.

2.2.5 Significado de programas de L0

Definición 2.8 (Significado de programas en L0). Sea $p \in \langle Prog \rangle$ un programa de L0, $p = \text{VAR } lv \text{ PROG } ls$.