Compiladores 2024-1. Notas

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

August 25, 2023

Contents

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

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

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-esimo de lv.
4. Decimos que lv es una lista sin repeticiones si
   $\forall i, j \in [1..|lv|] \colon i \neq j \rightarrow lv(i) \neq lv(j)$.

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:

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

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

   Intuitivamente, $S_{lv} := \{(b_1, b_2, \ldots, b_{|lv|}) \mid bi \in \mathrm{Dom}(lv(i))\}$.

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

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

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

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

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

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

Si asumimos que, para $v \in \{x, y, z\}$ $\mathrm{Dom}(v) = [0..9]$,
entonces, el conjunto de estados para lv está dado por
$S_{lv} := \{(b1, b2, b3) \mid bi \in [0..9]\}$.

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

Definición 2.4 (Singificado de expresiones de L0)**.** Sean: $lv \in \langle VarList \rangle$ una lista de variables sin repeticiones, $\sigma \in S_{lv}$, y $e \in \langle \mathrm{Exp} \rangle$ una expresión.

El significado de e en σ, $[\![e]\!]_\sigma$, se define por casos, según e:

1. Si $e \in \langle \mathrm{Dig} \rangle$, $[\![e]\!]_\sigma := e \in \mathbb{N}$.

2. Si $e \in \langle \mathrm{Var} \rangle$, $[\![e]\!]_\sigma := \begin{cases} \sigma(i) & \text{si } \exists i \in [1..|lv|]\colon e = lv(i) \\ \bot & \text{si no.} \end{cases}$.

Definición 2.5 (Estados de un programas de L0)**.** Sea $p \in \langle \mathrm{Prog} \rangle$ un programa de L0, $p = \mathrm{VAR}\ lv\ \mathrm{PROG}\ ls$.

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.

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

Sean $\sigma \in S_p$ un estado de p, y $s \in \langle \text{Stm} \rangle$ una instrucción de L0.

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

1. Si $\sigma = \omega$, entonces $[\![s]\!]_\sigma = \omega$.
2. Si $s = \text{Halt}$, entonces $[\![s]\!]_\sigma = \omega$.
3. Si $s = v := e$, entonces $[\![s]\!]_\sigma = \sigma[v \leftarrow [\![e]\!]]$.
4. Si $ls \in \langle StmList \rangle$, entonces el significado de ls en σ, $[\![ls]\!]_\sigma$, se define recursivamente:
   (a) Si $ls = s \in \langle Stm \rangle$,
   entonces $[\![ls]\!]_\sigma = [\![s]\!]_\sigma$
   (b) Si $ls = s \; ; \; ls'$,
   entonces $[\![ls]\!]_\sigma = [\![ls']\!]_{\sigma'}$, donde $\sigma' = [\![s]\!]_\sigma$.

Definición 2.7 (Significado de programas en L0)**.** Sea $p \in \langle \text{Prog} \rangle$ un programa de L0, $p = \text{VAR } lv \text{ PROG } ls$.