

Teilaufgaben der Übersetzung in Zwischencode

1. Übersetzung von Ausdrücken und Kontrollstrukturen
2. Übersetzung von Blöcken und Prozeduren
3. Übersetzung von Datenstrukturen
4. Inkrementelle Übersetzung von Modulstrukturen

Teilaufgaben der Übersetzung in Maschinencode

- Registerallokation
- Instruktionsanordnung
- Code-Auswahl
- ...

Syntax von PSA

Ganze Zahlen Int : Z

Bezeichner Ide : I

Deklarationen $Decl$: $\Delta ::= \Delta_C \Delta_V$
 $\Delta_C ::= \varepsilon \mid \mathbf{const} \ I_1 = Z_1; \dots; I_n = Z_n$
 $(n \geq 1)$
 $\Delta_V ::= \varepsilon \mid \mathbf{var} \ I_1, \dots, I_n; \quad (n \geq 1)$
(* nur Integervariablen *)

Arithmetische $AExp$: $E ::= Z \mid I \mid (E_1 \ aop \ E_2)$
Ausdrücke
 $(aop \in \{+, -, *, \dots\})$

Boolesche $BExp$: $B ::= E \ relop \ E \mid$
Ausdrücke
 $\mathbf{not} \ B \mid (B \ \mathbf{and} \ B) \mid (B \ \mathbf{or} \ B)$
 $(relop \in \{=, \neq, <, \dots\})$

Anweisungen Cmd : $\Gamma ::= I := E \mid \Gamma_1; \Gamma_2$
 $\mid \mathbf{if} \ B \ \mathbf{then} \ \Gamma \ \mathbf{else} \ \Gamma$
 $\mid \mathbf{while} \ B \ \mathbf{do} \ \Gamma$

Programme $Prog$: $P ::= \Delta \Gamma$

kontextsensitive Bedingungen

- Bezeichner in Δ müssen paarweise verschieden sein.
- Bezeichner in Γ müssen deklariert sein.

Semantik von PSA (Skizze)

Speicherplätze (locations) $Loc := \{\alpha_1, \alpha_2, \alpha_3, \dots\}$
Zustandsraum (states) $S := \{\sigma \mid \sigma : Loc \dashrightarrow \mathbb{Z}\}$
Umgebung (environment) $Env := \{\rho \mid \rho : Ide \dashrightarrow \mathbb{Z} \cup Loc\}$

Deklarationssemantik $\mathcal{D} :: Decl \times Env \dashrightarrow Env$

$$\mathcal{D}[\Delta_C \Delta_V]_\rho := \mathcal{D}[\Delta_V](\mathcal{D}[\Delta_C]\rho)$$

$$\mathcal{D}[\varepsilon]\rho := \rho$$

$$\mathcal{D}[\mathbf{const} \ I_1 = Z_1; \dots; I_n = Z_n]_\rho := \rho[I_1/Z_1, \dots, I_n/Z_n]$$

$$\mathcal{D}[\mathbf{var} \ I_1, I_2, \dots, I_n]_\rho := \rho[I_1/\alpha_1, \dots, I_n/\alpha_n]$$

Semantik arithmetischer Ausdrücke

$\mathcal{E} :: AExp \times Env \times S \dashrightarrow \mathbb{Z}$

$$\mathcal{E}[Z]\rho\sigma := Z$$

$$\mathcal{E}[I]\rho\sigma := \begin{cases} \rho(I) & \text{falls } \rho(I) \in \mathbb{Z} \\ \sigma(\rho(I)) & \text{falls } \rho(I) \in Loc \end{cases}$$

$$\mathcal{E}[(E_1 \text{ op } E_2)]\rho\sigma := \mathcal{E}[E_1]\rho\sigma \text{ [op] } \mathcal{E}[E_2]\rho\sigma$$

Semantik Boolescher Ausdrücke

$\mathcal{B} :: BExp \times Env \times S \dashrightarrow \{\mathbf{true}, \mathbf{false}\}$

analog mit strikten bzw. nicht-strikten Varianten

Semantik von Anweisungen

$\mathcal{C} :: Cmd \times Env \times S \dashrightarrow S$

Programmsemantik $\mathcal{M} :: Prog \times S \dashrightarrow S$

$$\mathcal{M}[\Delta\Gamma]\sigma := \mathcal{C}[\Gamma](\mathcal{D}[\Delta]\rho_\emptyset)\sigma$$