

8/20

Minitest 6

Minh Khue Pham (2579036)

13.07.2021 13:51:07 - 14:00:09

Referenz - Codegenerierung (Ausdrücke)

Syntaxgesteuerte Code-Erzeugung für Ausdrücke

Const $\frac{}{\text{codeR } \Omega (r :: rs) (c : \text{int}) = \text{li } r \ c}$

Indir $\frac{\text{codeR } \Omega (r :: rs) (e : k *) = c}{\text{codeL } \Omega (r :: rs) (*e : k) = c}$

Addr $\frac{\text{codeL } \Omega (r :: rs) (e : k) = c}{\text{codeR } \Omega (r :: rs) (\&e : k *) = c}$

Var $\frac{\Omega x = \delta}{\text{codeL } \Omega (r :: rs) (x : k) = \text{addiu } r \ \$sp \ \delta}$

Binary $\frac{\begin{array}{l} \text{codeR } \Omega (r_1 :: r_2 :: rs) (e_1 : k_1) = c_1 \\ \text{codeR } \Omega (r_2 :: rs) (e_2 : k_2) = c_2 \\ a = \text{Befehl der dem binären Operator } \circ \text{ entspricht} \end{array}}{\text{codeR } \Omega (r_1 :: r_2 :: rs) (e_1 \circ e_2) : k_1 = \begin{array}{l} c_1 \\ c_2 \\ a \quad r_1 \quad r_1 \quad r_2 \end{array}}$

LToR $\frac{\begin{array}{l} \text{codeL } \Omega (r :: rs) (l : k) = c \\ L = \text{Load, der Wert vom Typ } k \text{ lädt} \end{array}}{\text{codeR } \Omega (r :: rs) (l : k) = \begin{array}{l} c \\ L \quad r \quad r \end{array}}$

Multiple Choice (4 Punkte)

Beantworten Sie die folgenden Fragen mit wahr oder falsch. Für jede richtige Antwort bekommen Sie 1 Punkt. Für falsche Antworten bekommen Sie 0 Punkte. Wenn Sie keine Antwort geben bekommen Sie 0.5 Punkte.

English instruction: Answer the following questions with true or false. For every correct answer you obtain 1 point. For incorrect answers you get 0 points. If you do not give an answer you receive 0.5 points.

1. Eine Typumgebung wird in Compilern benutzt, um den Wert einer Variablen nachzuschlagen.

1. A type environment is used in a compiler to look up the value of a variable.

Response:

- ☐ wahr / true
- ☒ falsch / false

2. Ein abstrakter Syntaxbaum enthält immer alle Zeichen die im Originalprogramm vorhanden waren.

2. An abstract syntax tree contains every character that was present in the original program.

Response:

- ☐ wahr / true
- ☒ falsch / false

3. Die Typüberprüfung dient dazu, bereits vor der Ausführung Programme zurückzuweisen, welche nicht korrekt sind.

3. The purpose of type analysis is to reject incorrect programs before their execution.

Response:

- ☒ wahr / true
- ☐ falsch / false

4. Um die minimale Anzahl an Registern zur Ausdrucksauswertung zu nutzen, muss man zuerst den Teilausdruck auswerten, der mehr Register benötigt.

4. To minimize the amount of registers used, one needs to evaluate the subexpression which requires more registers first.

Response:

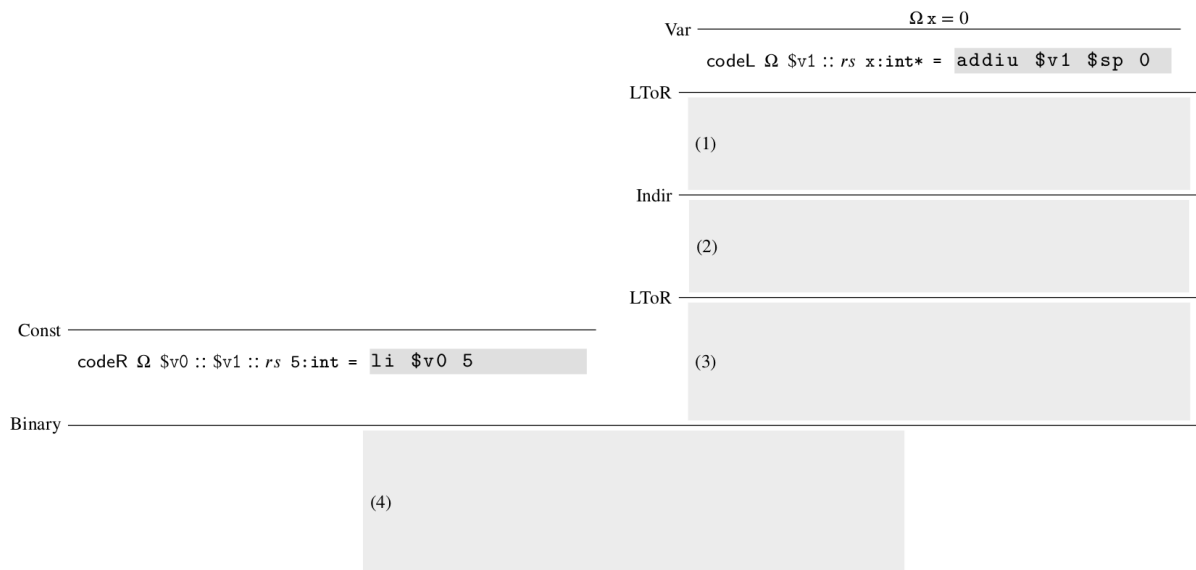
- ☒ wahr / true
- ☐ falsch / false

[[unknown type: ``endAttemptInteraction"]]

Syntaxgesteuerte Codegenerierung (12 Punkte)

Vervollständigen Sie den folgenden Inferenzbaum, welcher die syntaxgesteuerte Codegenerierung für den Ausdruck $5 + *x$ beschreiben soll.

English instruction: Complete the following inference tree, which shall reflect the syntax-driven code generation for the expression $5 + *x$.



Tippen Sie Ihre Antworten hier ein. Für das Sonderzeichen Ω dürfen Sie 'Oo' verwenden.
English instruction: Please enter your answer here. For special character Ω , you may use 'Oo'.

Response:

[[unknown type: ``endAttemptInteraction"]]

4/

Automatische Typanpassung (4 Punkte)

Geben Sie ein Gegenbeispiel für die folgenden Aussagen über die automatische Typanpassungsrelation \leftrightarrow an.

English Version: Give a counter example for the following statements about the automatic type conversion relation \leftrightarrow .

1. Reflexivität: Für alle Typen t gilt $t \leftrightarrow t$. (Reflexivity: For all types t it holds that $t \leftrightarrow t$)

[2 Punkte]

void \leftrightarrow void gilt nicht ✓

2. Wenn $t_1^* \leftrightarrow t_2^*$, dann $t_1 \leftrightarrow t_2$. (If $t_1^* \leftrightarrow t_2^*$, then $t_1 \leftrightarrow t_2$) **[2 Punkte]**

void* \leftrightarrow int*, aber void \leftrightarrow int gilt nicht ✓

Referenz:

$ITy \ni i$	$::=$	char int	Integer-Typ
$PTy \ni p$	$::=$	t^*	Zeiger-Typ
$STy \ni k$	$::=$	p i	Skalarer Typ
$Ty \ni t$	$::=$	k void	Typ

Definition 5.11 (Automatische Typanpassung).

$$\frac{}{i_1 \leftrightarrow i_2} \quad \frac{}{t^* \leftrightarrow t^*} \quad \frac{}{t^* \leftrightarrow \text{void}^*} \quad \frac{}{\text{void}^* \leftrightarrow t^*}$$

[[unknown type: ``endAttemptInteraction'']]

Unknown item: ``endAttemptInteraction''

Unknown item: ``endAttemptInteraction''

Unknown item: ``endAttemptInteraction''