Informatik II für Verkehrsingenieure
Übersetzung $C_0 \longrightarrow AM_0$ (Kapitel 14.3)

Janis Voigtländer

Technische Universität Dresden

Sommersemester 2007

Beispiel

```
#include<stdio.h>
int main()
{ int i,n,s;
  scanf("%i",&n);
  i=1;
  s=0;
  while (i<=n)
   { s=s+i*i;
     i=i+1;
   }
  printf("%d",s);
  return 0;
}
```

Syntaxgesteuerte Übersetzung (I)

⟨Program⟩ ::= `#include <stdio.h> int main()` ⟨Block⟩.

Syntaxgesteuerte Übersetzung (I)

⟨Program⟩ ::= `#include <stdio.h> int main()` ⟨Block⟩.

$\rightsquigarrow$ $\underline{\mathit{trans}}$(`#include <stdio.h> int main()` *block*)

$=$ $\underline{\mathit{blocktrans}}$(*block*)

für alle $\mathit{block} \in W(\langle \text{Block} \rangle)$

Syntaxgesteuerte Übersetzung (I)

⟨Program⟩ ::= `#include <stdio.h> int main()` ⟨Block⟩.

$\rightsquigarrow$ $\underline{\mathit{trans}}$(`#include <stdio.h> int main()` *block*)
$= \underline{\mathit{blocktrans}}(\mathit{block})$
für alle $\mathit{block} \in W(\langle \text{Block} \rangle)$

⟨Block⟩ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;`}.

Syntaxgesteuerte Übersetzung (I)

⟨Program⟩ ::= `#include <stdio.h> int main()` ⟨Block⟩.

$\rightsquigarrow$ $\underline{\mathit{trans}}$(`#include <stdio.h> int main()` *block*)
$= \underline{\mathit{blocktrans}}(\mathit{block})$
für alle $\mathit{block} \in W(\langle\text{Block}\rangle)$

⟨Block⟩ ::= `{` $\hat{[}$⟨VarDeclaration⟩$\hat{]}$ $\hat{[}$⟨StatementSequence⟩$\hat{]}$ `return 0;}`.

$\rightsquigarrow$ $\underline{\mathit{blocktrans}}$(`{`*vardecl statseq* `return 0;}`)
$= \underline{\mathit{stseqtrans}}(\mathit{statseq}, \cdots)$
für alle $\mathit{vardecl} \in \{\varepsilon\} \cup W(\langle\text{VarDeclaration}\rangle)$
und $\mathit{statseq} \in \{\varepsilon\} \cup W(\langle\text{StatementSequence}\rangle)$

Syntaxgesteuerte Übersetzung (I)

$\langle$Program$\rangle$ ::= `#include <stdio.h> int main()` $\langle$Block$\rangle$.

$\rightsquigarrow$ $\underline{\mathit{trans}}$(`#include <stdio.h> int main()` *block*)
$= \underline{\mathit{blocktrans}}(\mathit{block})$
für alle $\mathit{block} \in W(\langle\text{Block}\rangle)$

$\langle$Block$\rangle$ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;}`.

$\rightsquigarrow$ $\underline{\mathit{blocktrans}}$({*vardecl statseq* `return 0;}`)
$= \underline{\mathit{stseqtrans}}(\mathit{statseq}, \cdots)$
für alle $\mathit{vardecl} \in \{\varepsilon\} \cup W(\langle\text{VarDeclaration}\rangle)$
und $\mathit{statseq} \in \{\varepsilon\} \cup W(\langle\text{StatementSequence}\rangle)$

$\langle$StatementSequence$\rangle$ ::= $\langle$Statement$\rangle$ $\hat{\{}\langle$Statement$\rangle\hat{\}}$.

Syntaxgesteuerte Übersetzung (I)

$\langle$Program$\rangle$ ::= `#include <stdio.h> int main()` $\langle$Block$\rangle$.

$\rightsquigarrow$ $\underline{\mathit{trans}}$(`#include <stdio.h> int main()` *block*)
$= \underline{\mathit{blocktrans}}(\mathit{block})$
für alle $\mathit{block} \in W(\langle\text{Block}\rangle)$

$\langle$Block$\rangle$ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;}`.

$\rightsquigarrow$ $\underline{\mathit{blocktrans}}$({*vardecl statseq* `return 0;}`)
$= \underline{\mathit{stseqtrans}}(\mathit{statseq}, \cdots)$
für alle $\mathit{vardecl} \in \{\varepsilon\} \cup W(\langle\text{VarDeclaration}\rangle)$
und $\mathit{statseq} \in \{\varepsilon\} \cup W(\langle\text{StatementSequence}\rangle)$

$\langle$StatementSequence$\rangle$ ::= $\langle$Statement$\rangle$ $\hat{\{}\langle$Statement$\rangle\hat{\}}$.

$\rightsquigarrow$ $\underline{\mathit{stseqtrans}}(\mathit{stat}_1 \ \ldots \ \mathit{stat}_n, \cdots)$
$= \underline{\mathit{sttrans}}(\mathit{stat}_1, \cdots)$
$\ldots$
$\underline{\mathit{sttrans}}(\mathit{stat}_n, \cdots)$
für alle $\mathit{stat}_1, \ldots, \mathit{stat}_n \in W(\langle\text{Statement}\rangle)$

Syntaxgesteuerte Übersetzung (II)

⟨Statement⟩ ::= ⟨Assignment⟩ $\hat{|}$ ⟨IfStatement⟩ $\hat{|}$ ⟨WhileStatement⟩ $\hat{|}$
`scanf("%i",&`⟨Ident⟩`);` $\hat{|}$ `printf("%d",`⟨Ident⟩`);` $\hat{|}$
⟨CompStatement⟩.

Syntaxgesteuerte Übersetzung (II)

⟨Statement⟩ ::= ⟨Assignment⟩ $\hat{|}$ ⟨IfStatement⟩ $\hat{|}$ ⟨WhileStatement⟩ $\hat{|}$
`scanf("%i",&`⟨Ident⟩`);` $\hat{|}$ `printf("%d",`⟨Ident⟩`);` $\hat{|}$
⟨CompStatement⟩.

$\rightsquigarrow$ Fallunterscheidung

Syntaxgesteuerte Übersetzung (II)

⟨Statement⟩ ::= ⟨Assignment⟩ $\hat{|}$ ⟨IfStatement⟩ $\hat{|}$ ⟨WhileStatement⟩ $\hat{|}$
`scanf("%i",&`⟨Ident⟩`);` $\hat{|}$ `printf("%d",`⟨Ident⟩`);` $\hat{|}$
⟨CompStatement⟩.

$\rightsquigarrow$ Fallunterscheidung,
zum Beispiel: $\underline{sttrans}$(`scanf("%i",&`*id*`);`, $\cdots$)
= READ ?
für alle $id \in W(\langle \text{Ident} \rangle)$

Syntaxgesteuerte Übersetzung (II)

⟨Statement⟩ ::= ⟨Assignment⟩ $\hat{|}$ ⟨IfStatement⟩ $\hat{|}$ ⟨WhileStatement⟩ $\hat{|}$
`scanf("%i",&`⟨Ident⟩`);` $\hat{|}$ `printf("%d",`⟨Ident⟩`);` $\hat{|}$
⟨CompStatement⟩.

$\rightsquigarrow$ Fallunterscheidung,
zum Beispiel: $\underline{sttrans}$(`scanf("%i",&`*id*`);`, $\cdots$)
= READ ?
für alle $id \in W(\langle \text{Ident} \rangle)$

Wir brauchen Informationen über die Zuordnung von Bezeichnern (im Programm) zu Speicherplätzen (im HS der AM_0) !

Erzeugung einer Symboltabelle

$\langle$Block$\rangle$ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;}`.

$\rightsquigarrow$ *blocktrans*({*vardecl statseq* `return 0;}`)

$=$ *stseqtrans*(*statseq*, $\cdots$)

für alle *vardecl* $\in \{\varepsilon\} \cup W(\langle$VarDeclaration$\rangle)$

und *statseq* $\in \{\varepsilon\} \cup W(\langle$StatementSequence$\rangle)$

Erzeugung einer Symboltabelle

$\langle \text{Block} \rangle ::= \texttt{\{}\ \hat{[}\langle \text{VarDeclaration} \rangle \hat{]}\ \hat{[}\langle \text{StatementSequence} \rangle \hat{]}\ \texttt{return 0;\}}.$

$\rightsquigarrow \underline{\textit{blocktrans}}(\texttt{\{}\textit{vardecl}\ \textit{statseq}\ \texttt{return 0;\}})$

$= \underline{\textit{stseqtrans}}(\textit{statseq}, \underline{\textit{update}}(\textit{vardecl}), \cdots)$

für alle $\textit{vardecl} \in \{\varepsilon\} \cup W(\langle \text{VarDeclaration} \rangle)$

und $\textit{statseq} \in \{\varepsilon\} \cup W(\langle \text{StatementSequence} \rangle)$

Menge der Symboltabellen:

$\underline{\text{Tab}} = \{\textit{tab} \mid \textit{tab} : W(\langle \text{Ident} \rangle) \rightharpoonup (\{\textit{var}\} \times \mathbb{N})\}$

Erzeugung einer Symboltabelle

$\langle$Block$\rangle$ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;}`.

$\rightsquigarrow$ $\underline{\textit{blocktrans}}(\{\textit{vardecl}\ \textit{statseq}\ \texttt{return 0;}\})$

$= \underline{\textit{stseqtrans}}(\textit{statseq}, \underline{\textit{update}}(\textit{vardecl}), \cdots)$

für alle $\textit{vardecl} \in \{\varepsilon\} \cup W(\langle$VarDeclaration$\rangle)$

und $\textit{statseq} \in \{\varepsilon\} \cup W(\langle$StatementSequence$\rangle)$

Menge der Symboltabellen:

$\underline{\text{Tab}} = \{\textit{tab} \mid \textit{tab} : W(\langle\text{Ident}\rangle) \rightharpoonup (\{\textit{var}\} \times \mathbb{N})\}$

$\langle$VarDeclaration$\rangle$ ::= `int` $\langle$Ident$\rangle$ $\hat{\{}$,$\langle$Ident$\rangle\hat{\}}$;.

Erzeugung einer Symboltabelle

$\langle$Block$\rangle$ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;}`.

$\rightsquigarrow$ $\underline{\textit{blocktrans}}(\{\textit{vardecl}\ \textit{statseq}\ \texttt{return 0;}\})$
$= \underline{\textit{stseqtrans}}(\textit{statseq}, \underline{\textit{update}}(\textit{vardecl}), \cdots)$
für alle $\textit{vardecl} \in \{\varepsilon\} \cup W(\langle$VarDeclaration$\rangle)$
und $\textit{statseq} \in \{\varepsilon\} \cup W(\langle$StatementSequence$\rangle)$

Menge der Symboltabellen:
$\underline{\text{Tab}} = \{\textit{tab} \mid \textit{tab} : W(\langle\text{Ident}\rangle) \rightharpoonup (\{\textit{var}\} \times \mathbb{N})\}$

$\langle$VarDeclaration$\rangle$ ::= `int` $\langle$Ident$\rangle$ $\hat{\{}$,$\langle$Ident$\rangle\hat{\}}$;.

$\rightsquigarrow$ $\underline{\textit{update}}(\varepsilon) = []$ (leere Abbildung)

Erzeugung einer Symboltabelle

$\langle$Block$\rangle$ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;}`.

$\rightsquigarrow$ $\underline{blocktrans}(\{vardecl\ statseq$ `return 0;}`$)$
$= \underline{stseqtrans}(statseq, \underline{update}(vardecl), \cdots)$
für alle $vardecl \in \{\varepsilon\} \cup W(\langle$VarDeclaration$\rangle)$
und $statseq \in \{\varepsilon\} \cup W(\langle$StatementSequence$\rangle)$

Menge der Symboltabellen:
$\underline{\text{Tab}} = \{tab \mid tab : W(\langle\text{Ident}\rangle) \dashrightarrow (\{var\} \times \mathbb{N})\}$

$\langle$VarDeclaration$\rangle$ ::= `int` $\langle$Ident$\rangle$ $\hat{\{}$,$\langle$Ident$\rangle\hat{\}}$;.

$\rightsquigarrow$ $\underline{update}(\varepsilon) = []$ (leere Abbildung)
$\underline{update}($`int` $id_1, \ldots, id_m$;$)$
$= [id_1/(var, 1), \ldots, id_m/(var, m)]$
für alle $id_1, \ldots, id_m \in W(\langle$Ident$\rangle)$

Erzeugung einer Symboltabelle

$\langle$Block$\rangle$ ::= { $\hat{[}\langle$VarDeclaration$\rangle\hat{]}$ $\hat{[}\langle$StatementSequence$\rangle\hat{]}$ `return 0;}`.

$\rightsquigarrow$ $\underline{blocktrans}$({*vardecl statseq* `return 0;}`)
$= \underline{stseqtrans}(statseq, \underline{update}(vardecl), \cdots)$
für alle $vardecl \in \{\varepsilon\} \cup W(\langle\text{VarDeclaration}\rangle)$
und $statseq \in \{\varepsilon\} \cup W(\langle\text{StatementSequence}\rangle)$

Menge der Symboltabellen:
$\underline{\text{Tab}} = \{tab \mid tab : W(\langle\text{Ident}\rangle) \dashrightarrow (\{var\} \times \mathbb{N})\}$

$\langle$VarDeclaration$\rangle$ ::= `int` $\langle$Ident$\rangle$ $\hat{\{}$,$\langle$Ident$\rangle\hat{\}}$;.

$\rightsquigarrow$ $\underline{update}(\varepsilon) =$ [] (leere Abbildung)
$\underline{update}(\texttt{int}\ id_1, \ldots, id_m;)$
$= [id_1/(var, 1), \ldots, id_m/(var, m)]$
für alle $id_1, \ldots, id_m \in W(\langle\text{Ident}\rangle)$

Die Symboltabelle wird von $\underline{stseqtrans}$ aus in weitere Übersetzungsfunktionen propagiert !

Syntaxgesteuerte Übersetzung (III)

$\langle$Statement$\rangle$ ::= $\langle$Assignment$\rangle$ $\hat{|}$ $\langle$IfStatement$\rangle$ $\hat{|}$ $\langle$WhileStatement$\rangle$ $\hat{|}$
`scanf("%i",&`$\langle$Ident$\rangle$`);` $\hat{|}$ `printf("%d",`$\langle$Ident$\rangle$`);` $\hat{|}$
$\langle$CompStatement$\rangle$.

$\rightsquigarrow$ Fallunterscheidung,
zum Beispiel: $\underline{sttrans}$(`scanf("%i",&`id`);`, $tab, \cdots$)
= wenn $tab(id) = (var, n)$, dann READ n;
für alle $id \in W(\langle\text{Ident}\rangle)$ und $tab \in \underline{\text{Tab}}$

Syntaxgesteuerte Übersetzung (III)

⟨Statement⟩ ::= ⟨Assignment⟩ $\hat{|}$ ⟨IfStatement⟩ $\hat{|}$ ⟨WhileStatement⟩ $\hat{|}$
`scanf("%i",&`⟨Ident⟩`);` $\hat{|}$ `printf("%d",`⟨Ident⟩`);` $\hat{|}$
⟨CompStatement⟩.

$\rightsquigarrow$ Fallunterscheidung,
zum Beispiel: $\underline{sttrans}$(`scanf("%i",&`*id*`);`, *tab*, $\cdots$)
= wenn $tab(id) = (var, n)$, dann READ n;
für alle $id \in W($⟨Ident⟩$)$ und $tab \in \underline{\text{Tab}}$

⟨Assignment⟩ ::= ⟨Ident⟩ = ⟨SimpleExpression⟩;.

Syntaxgesteuerte Übersetzung (III)

⟨Statement⟩ ::= ⟨Assignment⟩ $\hat{|}$ ⟨IfStatement⟩ $\hat{|}$ ⟨WhileStatement⟩ $\hat{|}$
`scanf("%i",&`⟨Ident⟩`);` $\hat{|}$ `printf("%d",`⟨Ident⟩`);` $\hat{|}$
⟨CompStatement⟩.

$\rightsquigarrow$ Fallunterscheidung,
zum Beispiel: $\underline{sttrans}$(`scanf("%i",&`id`);`, $tab, \cdots$)
= wenn $tab(id) = (var, n)$, dann READ n;
für alle $id \in W($⟨Ident⟩$)$ und $tab \in \underline{\text{Tab}}$

⟨Assignment⟩ ::= ⟨Ident⟩ = ⟨SimpleExpression⟩;.

$\rightsquigarrow$ $\underline{sttrans}(id = exp;, tab, \cdots)$
= wenn $tab(id) = (var, n)$, dann:
$\underline{simpleexptrans}(exp, tab)$
STORE n;
für alle $id \in W($⟨Ident⟩$)$, $exp \in W($⟨SimpleExpression⟩$)$
und $tab \in \underline{\text{Tab}}$

Syntaxgesteuerte Übersetzung (IV)

⟨SimpleExpression⟩ ::= [+ | -] ⟨Term⟩ { (+ | -) ⟨Term⟩ }.

Syntaxgesteuerte Übersetzung (IV)

$\langle$SimpleExpression$\rangle$ ::= $\hat{[}$ + $\hat{|}$ - $\hat{]}$ $\langle$Term$\rangle$ $\hat{\{}$ $\hat{(}$ + $\hat{|}$ - $\hat{)}$ $\langle$Term$\rangle$ $\hat{\}}$.

$\rightsquigarrow$ $\underline{simpleexptrans}(sign_0\ t_1\ sign_1\ t_2\ \ldots\ sign_{n-1}\ t_n, tab)$

$= \underline{termtrans}(t_1, tab)$
SIGN$_0$
$\underline{termtrans}(t_2, tab)$
SIGN$_1$
$\ldots$
$\underline{termtrans}(t_n, tab)$
SIGN$_{n-1}$
für alle $t_1, \ldots, t_n \in W(\langle\text{Term}\rangle)$, $sign_0 \in \{+,-,\varepsilon\}$,
$sign_1, \ldots, sign_{n-1} \in \{+,-\}$ und $tab \in \underline{\text{Tab}}$,
wobei SIGN$_0 = \varepsilon$, falls $sign_0 \in \{+,\varepsilon\}$
SIGN$_0$ = LIT -1; MUL; falls $sign_0 =$ -
SIGN$_i$ = ADD; falls $sign_i =$ + und $i \geq 1$
SIGN$_i$ = SUB; falls $sign_i =$ - und $i \geq 1$

Syntaxgesteuerte Übersetzung (V)

$\langle \text{Term} \rangle ::= \langle \text{Factor} \rangle \; \hat{\{} \; \hat{(} \; * \; \hat{|} \; / \; \hat{|} \; \% \; \hat{)} \; \langle \text{Factor} \rangle \; \hat{\}}.$

Syntaxgesteuerte Übersetzung (V)

$\langle \text{Term} \rangle ::= \langle \text{Factor} \rangle \; \hat{\{} \; \hat{(} \; * \; \hat{|} \; / \; \hat{|} \; \% \; \hat{)} \; \langle \text{Factor} \rangle \; \hat{\}}.$

$\rightsquigarrow \underline{termtrans}(f_1 \; op_1 \; f_2 \; op_2 \; f_3 \; \ldots \; op_{n-1} \; f_n, tab)$

$= \underline{factortrans}(f_1, tab)$
$\underline{factortrans}(f_2, tab)$
$\text{OP}_1;$
$\underline{factortrans}(f_3, tab)$
$\text{OP}_2;$
$\ldots$
$\underline{factortrans}(f_n, tab)$
$\text{OP}_{n-1};$
für alle $f_1, \ldots, f_n \in W(\langle \text{Factor} \rangle)$, $op_1, \ldots, op_{n-1} \in \{*, /, \%\}$
und $tab \in \underline{\text{Tab}}$,
wobei $\text{OP}_i = \text{MUL}$, falls $op_i = *$
$\text{OP}_i = \text{DIV}$, falls $op_i = /$
$\text{OP}_i = \text{MOD}$, falls $op_i = \%$

Syntaxgesteuerte Übersetzung (VI)

$$\langle \text{Factor} \rangle ::= \langle \text{Ident} \rangle \mathbin{\hat{|}} \langle \text{Number} \rangle \mathbin{\hat{|}} (\langle \text{SimpleExpression} \rangle).$$

Syntaxgesteuerte Übersetzung (VI)

$\langle \text{Factor} \rangle ::= \langle \text{Ident} \rangle \mid \langle \text{Number} \rangle \mid (\langle \text{SimpleExpression} \rangle)$.

$\rightsquigarrow \underline{\mathit{factortrans}}(\mathit{id}, \mathit{tab})$

$=$ wenn $\mathit{tab}(\mathit{id}) = (\mathit{var}, n)$, dann LOAD n;

für alle $\mathit{id} \in W(\langle \text{Ident} \rangle)$ und $\mathit{tab} \in \underline{\text{Tab}}$

Syntaxgesteuerte Übersetzung (VI)

$\langle \text{Factor} \rangle ::= \langle \text{Ident} \rangle \hat{|} \langle \text{Number} \rangle \hat{|} (\langle \text{SimpleExpression} \rangle).$

$\rightsquigarrow \underline{\mathit{factortrans}}(\mathit{id}, \mathit{tab})$
$= \text{wenn } \mathit{tab}(\mathit{id}) = (\mathit{var}, n), \text{ dann LOAD } n;$
für alle $\mathit{id} \in W(\langle \text{Ident} \rangle)$ und $\mathit{tab} \in \underline{\text{Tab}}$

$\underline{\mathit{factortrans}}(z, \mathit{tab})$
$= \text{LIT } z;$
für alle $z \in W(\langle \text{Number} \rangle)$ und $\mathit{tab} \in \underline{\text{Tab}}$

Syntaxgesteuerte Übersetzung (VI)

$\langle \text{Factor} \rangle ::= \langle \text{Ident} \rangle \mid \langle \text{Number} \rangle \mid (\langle \text{SimpleExpression} \rangle)$.

$\rightsquigarrow \underline{\mathit{factortrans}}(\mathit{id}, \mathit{tab})$
$= \text{wenn } \mathit{tab}(\mathit{id}) = (\mathit{var}, n), \text{ dann LOAD } n;$
für alle $\mathit{id} \in W(\langle \text{Ident} \rangle)$ und $\mathit{tab} \in \underline{\text{Tab}}$

$\underline{\mathit{factortrans}}(z, \mathit{tab})$
$= \text{LIT } z;$
für alle $z \in W(\langle \text{Number} \rangle)$ und $\mathit{tab} \in \underline{\text{Tab}}$

$\underline{\mathit{factortrans}}((se), \mathit{tab})$
$= \underline{\mathit{simpleexptrans}}(se, \mathit{tab})$
für alle $se \in W(\langle \text{SimpleExpression} \rangle)$ und $\mathit{tab} \in \underline{\text{Tab}}$

Syntaxgesteuerte Übersetzung (VII)

⟨BoolExpression⟩ ::= ⟨SimpleExpression⟩ ⟨Relation⟩ ⟨SimpleExpression⟩.

⟨Relation⟩ ::= == | != | < | > | <= | >=.

Syntaxgesteuerte Übersetzung (VII)

⟨BoolExpression⟩ ::= ⟨SimpleExpression⟩ ⟨Relation⟩ ⟨SimpleExpression⟩.

⟨Relation⟩ ::= == $\hat{|}$!= $\hat{|}$ < $\hat{|}$ > $\hat{|}$ <= $\hat{|}$ >=.

$\rightsquigarrow$ *boolexptrans*(se_1 *rel* se_2, *tab*)
= *simpleexptrans*(se_1, *tab*)
simpleexptrans(se_2, *tab*)
REL;
für alle $se_1, se_2 \in W(\langle \text{SimpleExpression} \rangle)$,
rel $\in$ {==,!=,<,>,<=,>=} und *tab* $\in$ Tab,
wobei REL = EQ, falls *rel* = ==
REL = NE, falls *rel* = !=
REL = LT, falls *rel* = <
...

Syntaxgesteuerte Übersetzung (VIII)

⟨WhileStatement⟩ ::= `while` (⟨BoolExpression⟩) ⟨Statement⟩.

Syntaxgesteuerte Übersetzung (VIII)

⟨WhileStatement⟩ ::= `while` (⟨BoolExpression⟩) ⟨Statement⟩.

$\rightsquigarrow$ *sttrans*(`while` (*exp*) *stat*, *tab*, $\cdots$)

= *boolexptrans*(*exp*, *tab*)

JMC ?;

sttrans(*stat*, *tab*, $\cdots$)

JMP ?;

für alle $exp \in W(\langle \text{BoolExpression} \rangle)$,

$stat \in W(\langle \text{Statement} \rangle)$ und $tab \in \text{Tab}$

Syntaxgesteuerte Übersetzung (VIII)

⟨WhileStatement⟩ ::= `while` (⟨BoolExpression⟩) ⟨Statement⟩.

$\rightsquigarrow$ *sttrans*(`while` (*exp*) *stat*, *tab*, $\cdots$)

= *boolexptrans*(*exp*, *tab*)
JMC ?;
sttrans(*stat*, *tab*, $\cdots$)
JMP ?;
für alle $exp \in W(\langle \text{BoolExpression} \rangle)$,
$stat \in W(\langle \text{Statement} \rangle)$ und $tab \in \text{Tab}$

Problem:
- keine konkreten Adressen bekannt

Syntaxgesteuerte Übersetzung (VIII)

⟨WhileStatement⟩ ::= `while` (⟨BoolExpression⟩) ⟨Statement⟩.

$\rightsquigarrow$ *sttrans*(`while` (*exp*) *stat*, *tab*, $\cdots$)

= *boolexptrans*(*exp*, *tab*)
JMC ?;
sttrans(*stat*, *tab*, $\cdots$)
JMP ?;
für alle $exp \in W(\langle \text{BoolExpression} \rangle)$,
$stat \in W(\langle \text{Statement} \rangle)$ und $tab \in \text{Tab}$

Problem:

- keine konkreten Adressen bekannt
- hängen unter anderem von Länge des übersetzten Codes für *exp* und *stat* ab

Syntaxgesteuerte Übersetzung (VIII)

⟨WhileStatement⟩ ::= `while` (⟨BoolExpression⟩) ⟨Statement⟩.

$\rightsquigarrow$ *sttrans*(`while` (*exp*) *stat*, *tab*, $\cdots$)

= *boolexptrans*(*exp*, *tab*)
JMC ?;
sttrans(*stat*, *tab*, $\cdots$)
JMP ?;
für alle $exp \in W(\langle \text{BoolExpression} \rangle)$,
$stat \in W(\langle \text{Statement} \rangle)$ und $tab \in \text{Tab}$

Problem:
- keine konkreten Adressen bekannt
- hängen unter anderem von Länge des übersetzten Codes für *exp* und *stat* ab

Lösung:
- zunächst nur abstrakte Adressen, später Nachbearbeitung

Syntaxgesteuerte Übersetzung (VIII)

⟨WhileStatement⟩ ::= `while` (⟨BoolExpression⟩) ⟨Statement⟩.

$\rightsquigarrow$ *sttrans*(`while` (*exp*) *stat*, *tab*, $\cdots$)
= *boolexptrans*(*exp*, *tab*)
JMC ?;
sttrans(*stat*, *tab*, $\cdots$)
JMP ?;
für alle $exp \in W(\langle \text{BoolExpression} \rangle)$,
$stat \in W(\langle \text{Statement} \rangle)$ und $tab \in \text{Tab}$

Problem:
- keine konkreten Adressen bekannt
- hängen unter anderem von Länge des übersetzten Codes für *exp* und *stat* ab

Lösung:
- zunächst nur abstrakte Adressen, später Nachbearbeitung
- „baumstrukturierte Adressen“: Listen über natürlichen Zahlen (Notation 3.2.4.1)

Syntaxgesteuerte Übersetzung (IX)

$\underline{sttrans}(\texttt{while (}exp\texttt{)}\ stat, tab, a)$
$= a.2:\ \underline{boolexptrans}(exp, tab)$
JMC a;
$\underline{sttrans}(stat, tab, a.1)$
JMP $a.2$;
a:

für alle $exp \in W(\langle \text{BoolExpression} \rangle)$,
$stat \in W(\langle \text{Statement} \rangle)$, $tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

Syntaxgesteuerte Übersetzung (X)

$\underline{\textit{blocktrans}}(\{\textit{vardecl}\ \textit{statseq}\ \texttt{return 0;}\})$
$= \underline{\textit{stseqtrans}}(\textit{statseq}, \underline{\textit{update}}(\textit{vardecl}), 1)$
für alle $\textit{vardecl} \in \{\varepsilon\} \cup W(\langle \text{VarDeclaration} \rangle)$
und $\textit{statseq} \in \{\varepsilon\} \cup W(\langle \text{StatementSequence} \rangle)$

Syntaxgesteuerte Übersetzung (X)

$\underline{\mathit{blocktrans}}(\{\mathit{vardecl}\ \mathit{statseq}\ \texttt{return 0;}\})$
$= \underline{\mathit{stseqtrans}}(\mathit{statseq}, \underline{\mathit{update}}(\mathit{vardecl}), 1)$
für alle $\mathit{vardecl} \in \{\varepsilon\} \cup W(\langle\text{VarDeclaration}\rangle)$
und $\mathit{statseq} \in \{\varepsilon\} \cup W(\langle\text{StatementSequence}\rangle)$

$\underline{\mathit{stseqtrans}}(\mathit{stat}_1\ \ldots\ \mathit{stat}_n, \mathit{tab}, a)$
$= \underline{\mathit{sttrans}}(\mathit{stat}_1, \mathit{tab}, a.1)$
$\ldots$
$\underline{\mathit{sttrans}}(\mathit{stat}_n, \mathit{tab}, a.n)$
für alle $\mathit{stat}_1, \ldots, \mathit{stat}_n \in W(\langle\text{Statement}\rangle)$, $\mathit{tab} \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

Syntaxgesteuerte Übersetzung (X)

$\underline{blocktrans}(\{vardecl\ statseq\ \texttt{return 0;}\})$
$= \underline{stseqtrans}(statseq, \underline{update}(vardecl), 1)$
für alle $vardecl \in \{\varepsilon\} \cup W(\langle \text{VarDeclaration} \rangle)$
und $statseq \in \{\varepsilon\} \cup W(\langle \text{StatementSequence} \rangle)$

$\underline{stseqtrans}(stat_1\ \ldots\ stat_n, tab, a)$
$= \underline{sttrans}(stat_1, tab, a.1)$
$\ldots$
$\underline{sttrans}(stat_n, tab, a.n)$
für alle $stat_1, \ldots, stat_n \in W(\langle \text{Statement} \rangle)$, $tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

Noch einige Fälle offen:
$\langle \text{Statement} \rangle ::= \langle \text{Assignment} \rangle \mid \langle \text{IfStatement} \rangle \mid \langle \text{WhileStatement} \rangle \mid$
$\texttt{scanf("\%i",\&}\langle \text{Ident} \rangle\texttt{);} \mid \texttt{printf("\%d",}\langle \text{Ident} \rangle\texttt{);} \mid$
$\langle \text{CompStatement} \rangle$.

Syntaxgesteuerte Übersetzung (XI)

$\underline{sttrans}(\texttt{if (}exp\texttt{)}\ stat, tab, a)$

$=$ $\underline{boolexptrans}(exp, tab)$

JMC a;

$\underline{sttrans}(stat, tab, a.1)$

a:

für alle $exp \in W(\langle \text{BoolExpression} \rangle)$, $stat \in W(\langle \text{Statement} \rangle)$, $tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

Syntaxgesteuerte Übersetzung (XI)

$\underline{sttrans}(\texttt{if (}exp\texttt{)}\ stat, tab, a)$
$=$ $\underline{boolexptrans}(exp, tab)$
JMC a;
$\underline{sttrans}(stat, tab, a.1)$
a:

für alle $exp \in W(\langle \text{BoolExpression} \rangle)$, $stat \in W(\langle \text{Statement} \rangle)$, $tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

$\underline{sttrans}(\texttt{if (}exp\texttt{)}\ stat_1\ \texttt{else}\ stat_2, tab, a)$
$=$ $\underline{boolexptrans}(exp, tab)$
JMC a;
$\underline{sttrans}(stat_1, tab, a.1)$
JMP a.3;
a: $\underline{sttrans}(stat_2, tab, a.2)$
a.3:

für alle $exp \in W(\langle \text{BoolExpression} \rangle)$, $stat_1, stat_2 \in W(\langle \text{Statement} \rangle)$, $tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

Syntaxgesteuerte Übersetzung (XII)

$\underline{sttrans}(\texttt{printf("\%d",}id\texttt{);}, tab, a)$
$=$ wenn $tab(id) = (var, n)$, dann WRITE n;
für alle $id \in W(\langle \text{Ident} \rangle)$, $tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

Syntaxgesteuerte Übersetzung (XII)

$\underline{sttrans}(\texttt{printf("\%d",}id\texttt{);}, tab, a)$
$=$ wenn $tab(id) = (var, n)$, dann WRITE n;
für alle $id \in W(\langle \text{Ident} \rangle)$, $tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

$\underline{sttrans}(\{stat_1 \ \ldots \ stat_n\}, tab, a)$
$= \underline{stseqtrans}(stat_1 \ \ldots \ stat_n, tab, a)$
für alle $stat_1, \ldots, stat_n \in W(\langle \text{Statement} \rangle)$,
$tab \in \underline{\text{Tab}}$ und $a \in \mathbb{N}^*$

Zusammenfassung (I)

$\underline{trans}$(`#include <stdio.h> int main()` $block$)
$= \underline{blocktrans}(block)$

$\underline{blocktrans}$(`{`$vardecl\ statseq$ `return 0;}`)
$= \underline{stseqtrans}(statseq, \underline{update}(vardecl), 1)$

$\underline{update}(\varepsilon) = []$
$\underline{update}(\texttt{int}\ id_1, \ldots, id_m;) = [id_1/(var, 1), \ldots, id_m/(var, m)]$

$\underline{stseqtrans}(stat_1\ \ldots\ stat_n, tab, a)$
$= \underline{sttrans}(stat_1, tab, a.1)$
$\ldots$
$\underline{sttrans}(stat_n, tab, a.n)$

Zusammenfassung (II)

$\underline{sttrans}(id = exp;, tab, a)$
$=$ wenn $tab(id) = (var, n)$, dann:
  $\underline{simpleexptrans}(exp, tab)$
  STORE n;

$\underline{sttrans}(\texttt{if (}exp\texttt{)}\ stat, tab, a)$
$=$ $\underline{boolexptrans}(exp, tab)$
  JMC a;
  $\underline{sttrans}(stat, tab, a.1)$
  a:

$\underline{sttrans}(\texttt{if (}exp\texttt{)}\ stat_1\ \texttt{else}\ stat_2, tab, a)$
$=$ $\underline{boolexptrans}(exp, tab)$
  JMC a;
  $\underline{sttrans}(stat_1, tab, a.1)$
  JMP a.3;
  a: $\underline{sttrans}(stat_2, tab, a.2)$
  a.3:

Zusammenfassung (III)

$\underline{sttrans}$(`while (`exp`)` $stat, tab, a$)
= a.2: $\underline{boolexptrans}(exp, tab)$
JMC a;
$\underline{sttrans}(stat, tab, a.1)$
JMP a.2;
a:

$\underline{sttrans}$(`scanf("%i",&`id`);`$, tab, a$)
= wenn $tab(id) = (var, n)$, dann READ n;

$\underline{sttrans}$(`printf("%d",`id`);`$, tab, a$)
= wenn $tab(id) = (var, n)$, dann WRITE n;

$\underline{sttrans}(\{stat_1 \ \ldots \ stat_n\}, tab, a)$
= $\underline{stseqtrans}(stat_1 \ \ldots \ stat_n, tab, a)$

Zusammenfassung (IV)

$\underline{boolexptrans}(se_1\ rel\ se_2, tab)$
$= \underline{simpleexptrans}(se_1, tab)$
$\underline{simpleexptrans}(se_2, tab)$
REL;
wobei REL = EQ, falls $rel = ==$
$\ldots$

$\underline{simpleexptrans}(sign_0\ t_1\ sign_1\ t_2\ \ldots\ sign_{n-1}\ t_n, tab)$
$= \underline{termtrans}(t_1, tab)$
SIGN_0
$\ldots$
$\underline{termtrans}(t_n, tab)$
SIGN_{n-1}
wobei $\mathrm{SIGN}_0 = \varepsilon$, falls $sign_0 \in \{+, \varepsilon\}$
$\mathrm{SIGN}_0 = \mathrm{LIT}\ -1;\ \mathrm{MUL};$ falls $sign_0 =$ -
$\mathrm{SIGN}_i = \mathrm{ADD};$ falls $sign_i =$ + und $i \geq 1$
$\mathrm{SIGN}_i = \mathrm{SUB};$ falls $sign_i =$ - und $i \geq 1$

Zusammenfassung (V)

$\underline{termtrans}(f_1\ op_1\ f_2\ op_2\ f_3\ \ldots\ op_{n-1}\ f_n, tab)$
$= \underline{factortrans}(f_1, tab)$
$\underline{factortrans}(f_2, tab)$
$\mathrm{OP}_1;$
$\underline{factortrans}(f_3, tab)$
$\mathrm{OP}_2;$
$\ldots$
$\underline{factortrans}(f_n, tab)$
$\mathrm{OP}_{n-1};$
wobei $\mathrm{OP}_i = \mathrm{MUL}$, falls $op_i = *$
$\ldots$

$\underline{factortrans}(id, tab)$
$=$ wenn $tab(id) = (var, n)$, dann LOAD n;

$\underline{factortrans}(z, tab) = \mathrm{LIT}\ z;$

$\underline{factortrans}((se), tab) = \underline{simpleexptrans}(se, tab)$

Beispiel

```
#include<stdio.h>
int main()
{ int i,n,s;
  scanf("%i",&n);
  i=1;
  s=0;
  while (i<=n)
   { s=s+i*i;
     i=i+1;
   }
  printf("%d",s);
  return 0;
}
```

Beispiel — übersetzt

```
        READ 2;
        LIT 1;
        STORE 1;
        LIT 0;
        STORE 3;
1.4.2:  LOAD 1;
        LOAD 2;
        LE;
        JMC 1.4;
        LOAD 3;
        LOAD 1;
        LOAD 1;
        MUL;
        ADD;
        STORE 3;
        LOAD 1;
        LIT 1;
        ADD;
        STORE 1;
        JMP 1.4.2;
1.4:    WRITE 3;
```

Beispiel — übersetzt

```
      READ 2;
      LIT 1;
      STORE 1;
      LIT 0;
      STORE 3;
1.4.2: LOAD 1;
      LOAD 2;
      LE;
      JMC 1.4;
      LOAD 3;
      LOAD 1;
      LOAD 1;
      MUL;
      ADD;
      STORE 3;
      LOAD 1;
      LIT 1;
      ADD;
      STORE 1;
      JMP 1.4.2;
1.4:  WRITE 3;
```

Linearisierung:

1. Durchnumerierung der Befehle, beginnend mit 1

Beispiel — übersetzt

```
      READ 2;
      LIT 1;
      STORE 1;
      LIT 0;
      STORE 3;
1.4.2: LOAD 1;
      LOAD 2;
      LE;
      JMC 1.4;
      LOAD 3;
      LOAD 1;
      LOAD 1;
      MUL;
      ADD;
      STORE 3;
      LOAD 1;
      LIT 1;
      ADD;
      STORE 1;
      JMP 1.4.2;
1.4:  WRITE 3;
```

Linearisierung:

1. Durchnumerierung der Befehle, beginnend mit 1
2. Merken von Paaren aus baumstrukturierter Adresse und numerierter Adresse

Beispiel — übersetzt

```
      READ 2;          LE;              STORE 3;
      LIT 1;           JMC 1.4;         LOAD 1;
      STORE 1;         LOAD 3;          LIT 1;
      LIT 0;           LOAD 1;          ADD;
      STORE 3;         LOAD 1;          STORE 1;
1.4.2: LOAD 1;         MUL;             JMP 1.4.2;
      LOAD 2;          ADD;        1.4: WRITE 3;
```

Linearisierung:

1. Durchnumerierung der Befehle, beginnend mit 1
2. Merken von Paaren aus baumstrukturierter Adresse und numerierter Adresse
3. Anpassen von Sprungbefehlen entsprechend der gemerkten Paare

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$(\ 1\ ,\quad \varepsilon\ ,\ [] \qquad\qquad ,\ 1\ ,\ \varepsilon\ )$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

(1 , ε , [] , 1 , ε)

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$(\ 1\ ,\quad \varepsilon\ ,\ [] \qquad\qquad ,\ 1\ \ ,\ \varepsilon\ )$

$(\ 2\ ,\quad \varepsilon\ ,\ [2/1] \qquad\qquad ,\ \varepsilon\ \ ,\ \varepsilon\ )$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$(\ 1\ ,\ \varepsilon\ ,\ [\ ]\ ,\ 1\ ,\ \varepsilon\ )$
$(\ 2\ ,\ \varepsilon\ ,\ [2/1]\ ,\ \varepsilon\ ,\ \varepsilon\ )$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lll}
(\ 1\ , & \varepsilon\ , [] & ,\ 1\ ,\ \varepsilon\ ) \\
(\ 2\ , & \varepsilon\ , [2/1] & ,\ \varepsilon\ ,\ \varepsilon\ ) \\
(\ 3\ , & 1\ , [2/1] & ,\ \varepsilon\ ,\ \varepsilon\ )
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lll}
(& 1 , & \varepsilon , [] \quad , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] \quad , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] \quad , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lll}
(\ 1\ , & \varepsilon\ , [] & ,\ 1\ ,\ \varepsilon\ ) \\
(\ 2\ , & \varepsilon\ , [2/1] & ,\ \varepsilon\ ,\ \varepsilon\ ) \\
(\ 3\ , & 1\ , [2/1] & ,\ \varepsilon\ ,\ \varepsilon\ ) \\
(\ 4\ , & \varepsilon\ , [1/1,2/1] & ,\ \varepsilon\ ,\ \varepsilon\ )
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;      8: LE;          15: STORE 3;
 2: LIT 1;       9: JMC 21;      16: LOAD 1;
 3: STORE 1;    10: LOAD 3;      17: LIT 1;
 4: LIT 0;      11: LOAD 1;      18: ADD;
 5: STORE 3;    12: LOAD 1;      19: STORE 1;
 6: LOAD 1;     13: MUL;         20: JMP 6;
 7: LOAD 2;     14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llll}
(& 1 , & \varepsilon , [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;      8: LE;          15: STORE 3;
 2: LIT 1;       9: JMC 21;      16: LOAD 1;
 3: STORE 1;    10: LOAD 3;      17: LIT 1;
 4: LIT 0;      11: LOAD 1;      18: ADD;
 5: STORE 3;    12: LOAD 1;      19: STORE 1;
 6: LOAD 1;     13: MUL;         20: JMP 6;
 7: LOAD 2;     14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lll}
(& 1 , & \varepsilon , [] \qquad\qquad , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] \qquad\quad , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] \qquad\quad , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] \quad , \varepsilon , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] \quad , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llll}
(& 1 , & \varepsilon , [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{llll}
(& 1 , & \varepsilon , [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llll}
(& 1 , & \varepsilon , [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{llll}
(& 1 , & \varepsilon , [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{llll}
(& 1 , & \varepsilon , [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

(1 ,	ε , []	, 1	, ε)
(2 ,	ε , [2/1]	, ε	, ε)
(3 ,	1 , [2/1]	, ε	, ε)
(4 ,	ε , [1/1,2/1]	, ε	, ε)
(5 ,	0 , [1/1,2/1]	, ε	, ε)
(6 ,	ε , [1/1,2/1,3/0]	, ε	, ε)
(7 ,	1 , [1/1,2/1,3/0]	, ε	, ε)
(8 ,	1:1 , [1/1,2/1,3/0]	, ε	, ε)

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 1 , & \varepsilon , & [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , & [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , & [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 1 , & \varepsilon , [] & , 1 , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 9 , & 1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 1 , & \varepsilon , [] & , 1 & , \varepsilon) \\
(& 2 , & \varepsilon , [2/1] & , \varepsilon & , \varepsilon) \\
(& 3 , & 1 , [2/1] & , \varepsilon & , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon & , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon & , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 8 , & 1{:}1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 9 , & 1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 2 , & \varepsilon , & [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , & [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 2 , & \varepsilon , & [2/1] & , \varepsilon , \varepsilon) \\
(& 3 , & 1 , & [2/1] & , \varepsilon , \varepsilon) \\
(& 4 , & \varepsilon , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 3 , & 1 , [2/1] & , \varepsilon & , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon & , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon & , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 8 , & 1{:}1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 9 , & 1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 10 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 11 , & 0 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 3 , & 1 , [2/1] & , \varepsilon & , \varepsilon) \\
(& 4 , & \varepsilon , [1/1,2/1] & , \varepsilon & , \varepsilon) \\
(& 5 , & 0 , [1/1,2/1] & , \varepsilon & , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 8 , & 1{:}1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 9 , & 1 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 10 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon) \\
(& 11 , & 0 , [1/1,2/1,3/0] & , \varepsilon & , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;      8: LE;          15: STORE 3;
 2: LIT 1;       9: JMC 21;      16: LOAD 1;
 3: STORE 1;    10: LOAD 3;      17: LIT 1;
 4: LIT 0;      11: LOAD 1;      18: ADD;
 5: STORE 3;    12: LOAD 1;      19: STORE 1;
 6: LOAD 1;     13: MUL;         20: JMP 6;
 7: LOAD 2;     14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 4 , & \varepsilon , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 4 , & \varepsilon , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 5 , & 0 , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 5 , & 0 , & [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{llll}
(& 5 , & 0 , [1/1,2/1] & , \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 9 , & 1 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 11 , & 0 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , [1/1,2/1,3/0] & , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 6 , & \varepsilon , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 7 , & 1 , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , \varepsilon , & \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrl}
(& 6 , & \varepsilon , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 7 , & 1 , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 9 , & 1 , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 11 , & 0 , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , [1/1,2/1,3/0] , \varepsilon , \varepsilon) \\
(& 14 , & 1{:}0 , [1/1,2/1,3/0] , \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 7 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 7 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llllll}
(& 8 , & 1{:}1 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 9 , & 1 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon & , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 9 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{llllll}
(& 9 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{llllll}
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 10 , & \varepsilon , & [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 11 , & 0 , & [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , \varepsilon & , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , \varepsilon & , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , \varepsilon & , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 11 , & 0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;
 2: LIT 1;
 3: STORE 1;
 4: LIT 0;
 5: STORE 3;
 6: LOAD 1;
 7: LOAD 2;
 8: LE;
 9: JMC 21;
10: LOAD 3;
11: LOAD 1;
12: LOAD 1;
13: MUL;
14: ADD;
15: STORE 3;
16: LOAD 1;
17: LIT 1;
18: ADD;
19: STORE 1;
20: JMP 6;
21: WRITE 3;
```

(12 , 1:0 , [1/1,2/1,3/0] , ε , ε)
(13 , 1:1:0 , [1/1,2/1,3/0] , ε , ε)
(14 , 1:0 , [1/1,2/1,3/0] , ε , ε)
(15 , 1 , [1/1,2/1,3/0] , ε , ε)
(16 , ε , [1/1,2/1,3/1] , ε , ε)
(17 , 1 , [1/1,2/1,3/1] , ε , ε)
(18 , 1:1 , [1/1,2/1,3/1] , ε , ε)
(19 , 2 , [1/1,2/1,3/1] , ε , ε)
(20 , ε , [1/2,2/1,3/1] , ε , ε)

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 12 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llllll}
(& 13 , & 1{:}1{:}0 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon & , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llll}
(& 13 , & 1{:}1{:}0 , [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 14 , & 1{:}0 , [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 15 , & 1 , [1/1,2/1,3/0] , \varepsilon & , \varepsilon) \\
(& 16 , & \varepsilon , [1/1,2/1,3/1] , \varepsilon & , \varepsilon) \\
(& 17 , & 1 , [1/1,2/1,3/1] , \varepsilon & , \varepsilon) \\
(& 18 , & 1{:}1 , [1/1,2/1,3/1] , \varepsilon & , \varepsilon) \\
(& 19 , & 2 , [1/1,2/1,3/1] , \varepsilon & , \varepsilon) \\
(& 20 , & \varepsilon , [1/2,2/1,3/1] , \varepsilon & , \varepsilon) \\
(& 6 , & \varepsilon , [1/2,2/1,3/1] , \varepsilon & , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 14 , & 1{:}0 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon \ , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llllll}
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , & \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon , & \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon , & \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 15 , & 1 , & [1/1,2/1,3/0] , & \varepsilon , \varepsilon) \\
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 9 , & 0 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 16 , & \varepsilon , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon) \\
(& 9 , & 0 , & [1/2,2/1,3/1] , & \varepsilon , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 9 , & 0 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 21 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 17 , & 1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 9 , & 0 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 21 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{llllll}
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 9 , & 0 , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 21 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon & , \varepsilon) \\
(& 22 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon & , 1)
\end{array}
$$

Beispiel — linearisiert

```
 1: READ 2;        8: LE;          15: STORE 3;
 2: LIT 1;         9: JMC 21;      16: LOAD 1;
 3: STORE 1;      10: LOAD 3;      17: LIT 1;
 4: LIT 0;        11: LOAD 1;      18: ADD;
 5: STORE 3;      12: LOAD 1;      19: STORE 1;
 6: LOAD 1;       13: MUL;         20: JMP 6;
 7: LOAD 2;       14: ADD;         21: WRITE 3;
```

$$
\begin{array}{lrlll}
(& 18 , & 1{:}1 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 19 , & 2 , & [1/1,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 20 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 6 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 7 , & 2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 8 , & 1{:}2 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 9 , & 0 , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 21 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , \varepsilon) \\
(& 22 , & \varepsilon , & [1/2,2/1,3/1] , & \varepsilon \ , 1)
\end{array}
$$