

Geração de Código:
Algoritmos de Matching de Árvores para Seleção
de Instruções

Seleção de Instruções

- Consiste no problema de selecionar instruções de uma linguagem alvo para implementar as operações de uma representação intermediária
- Por meio de um parser e analisador sintático
 - A gramática para geração de código é, geralmente, ambígua e, com isso, muito propensa a conflitos shift-reduce e reduce-reduce
 - No entanto, o método de parsing é eficiente e bem compreendido

Seleção de Instruções

- Esquemas de matching de árvores
 - Código para a máquina alvo é gerado através da aplicação de uma sequência de regras de matching de árvores a fim de reduzir a árvore de entrada (árvore sintática) a um único nó
 - O processo consiste em associar partes da árvore de entrada (sub-árvore(s)) com os templates previamente estabelecidos no conjunto de regras de matching.

Seleção de Instruções

- Esquemas de matching de árvores
 - O processo é repetido até que a árvore de entrada é reduzida para um único nó ou até que não seja possível nenhum matching entre as sub-árvores da árvore de entrada e os templates.
 - A seqüência de instruções de máquina geradas, a medida que a árvore de entrada é reduzida para um único nó, constitui o código de saída para a máquina alvo

Seleção de Instruções

- Esquemas de matching de árvores
 - Deve-se destacar que a eficiência do processo de geração de código depende da eficiência do algoritmo de matching de árvores
 - A eficiência do código gerado pode depender da ordem em que os templates são utilizados

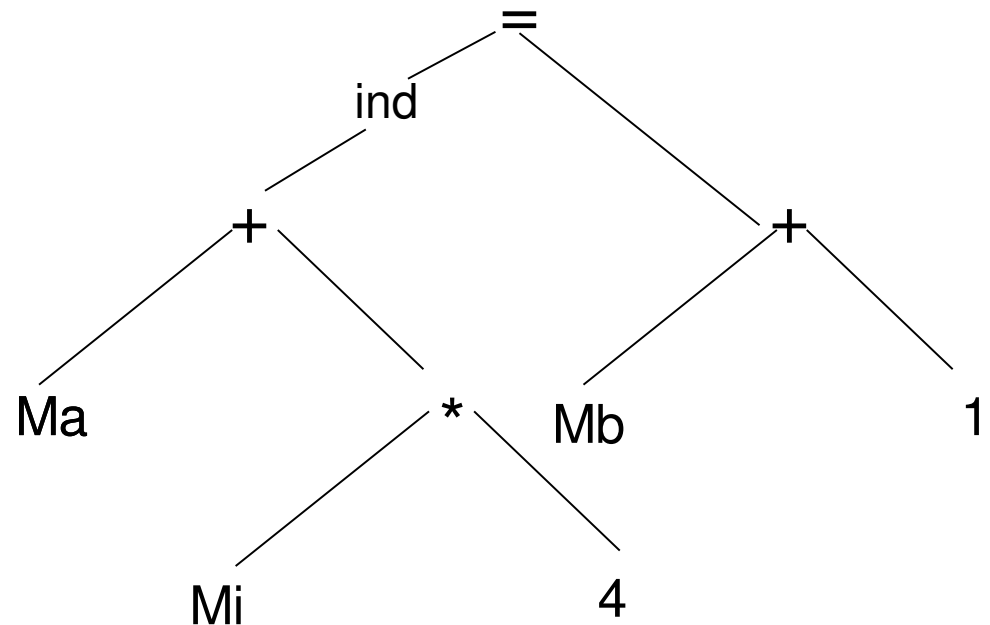
Seleção de Instruções

- Exemplo: Esquemas de matching de árvores

$i=i*4$

$a[i]=b$

$+1$



Código de 3-end

Árvore Sintática

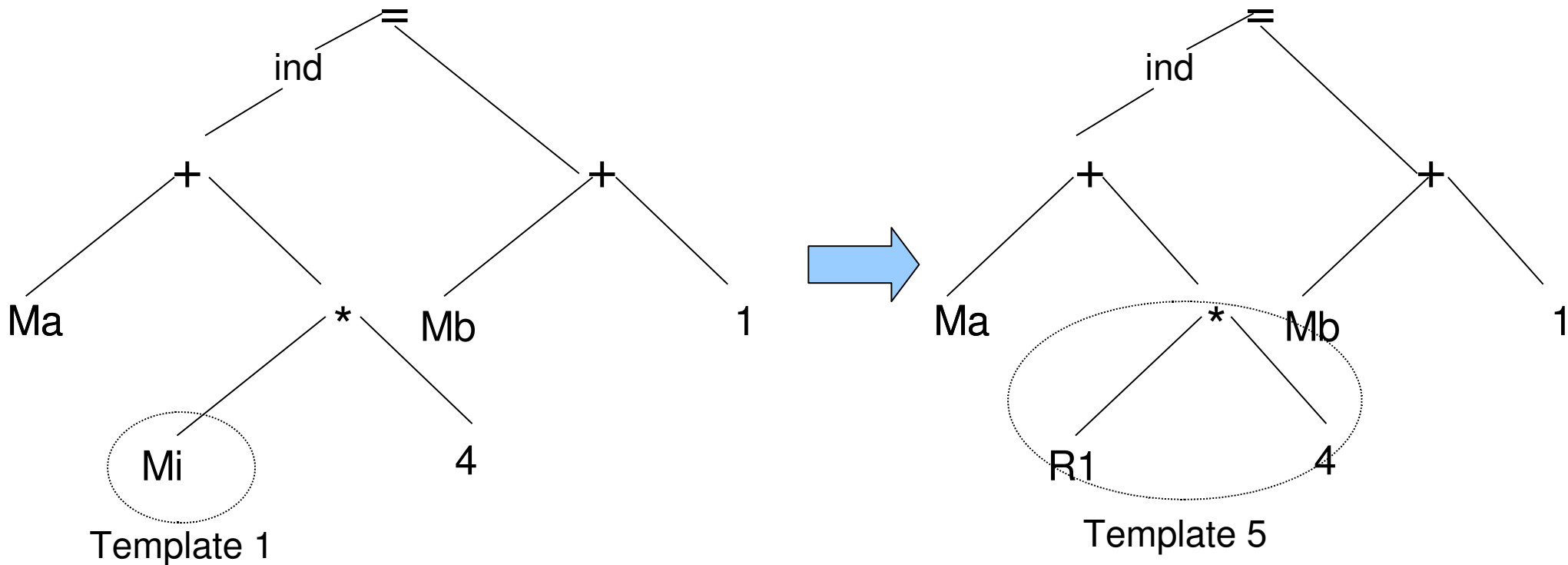
Seleção de Instruções

- Exemplo: Esquemas de matching de árvores

0	$R_i \leftarrow c$	li Ri, c
1	$R_i \leftarrow Mx$	lw Ri, Mx
2	$M \leftarrow \begin{array}{c} \diagup \quad \diagdown \\ \quad \quad \quad = \\ \diagdown \quad \diagup \\ Mx \quad \quad Ri \end{array}$	sw Ri, Mx
3	$R_i \leftarrow \begin{array}{c} \quad \quad \quad + \\ \diagdown \quad \diagup \\ Ri \quad \quad Rj \end{array}$	add Ri, Ri, Rj
4	$R_i \leftarrow \begin{array}{c} \quad \quad \quad + \\ \diagdown \quad \diagup \\ Ri \quad \quad c \end{array}$	addi, Ri, Ri, c
5	$R_i \leftarrow \begin{array}{c} \quad \quad \quad * \\ \diagdown \quad \diagup \\ Ri \quad \quad c \end{array}$	muli Ri, Ri, c
6	$R_i \leftarrow \begin{array}{c} \quad \quad \quad * \\ \diagdown \quad \diagup \\ Ri \quad \quad Rj \end{array}$	mul Ri, Ri, Rj
7	$R_i \leftarrow \begin{array}{c} \quad \quad \quad = \\ \diagdown \quad \diagup \\ ind \quad \quad \\ Ri \quad \quad Rj \end{array}$	sw Rj, Ri
8	$R_i \leftarrow \begin{array}{c} ind \\ \\ Bi \end{array}$	lw Ri, Rj

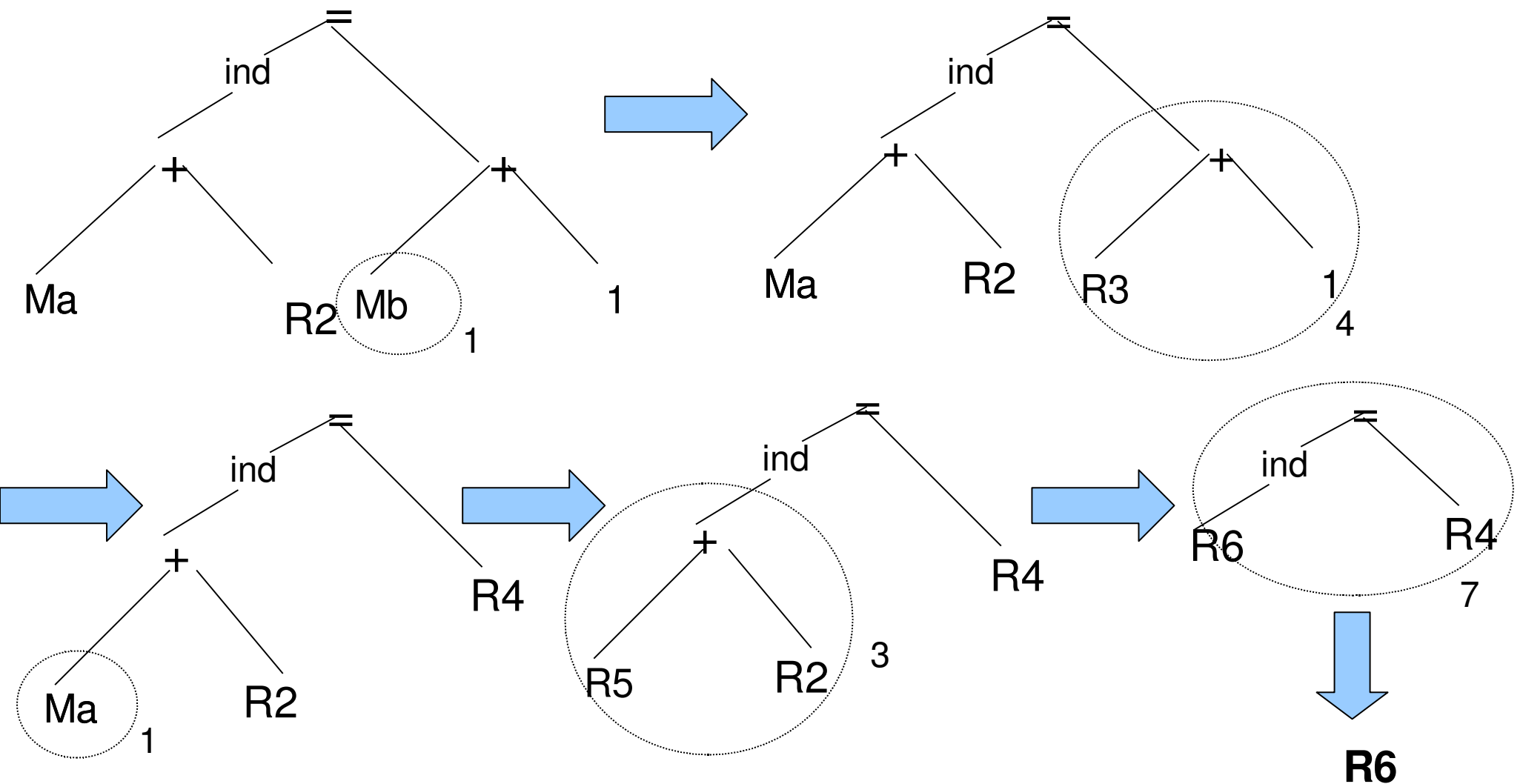
Seleção de Instruções

- Exemplo: Esquemas de matching de árvores



Seleção de Instruções

- Exemplo: Esquemas de matching de árvores



Seleção de Instruções

- Exemplo: Esquemas de matching de árvores

$i=i*4$

$a[i]=b+1$

Código de 3-end

lw R1, Mi

muli R2, R1, 4

lw R3, Mb

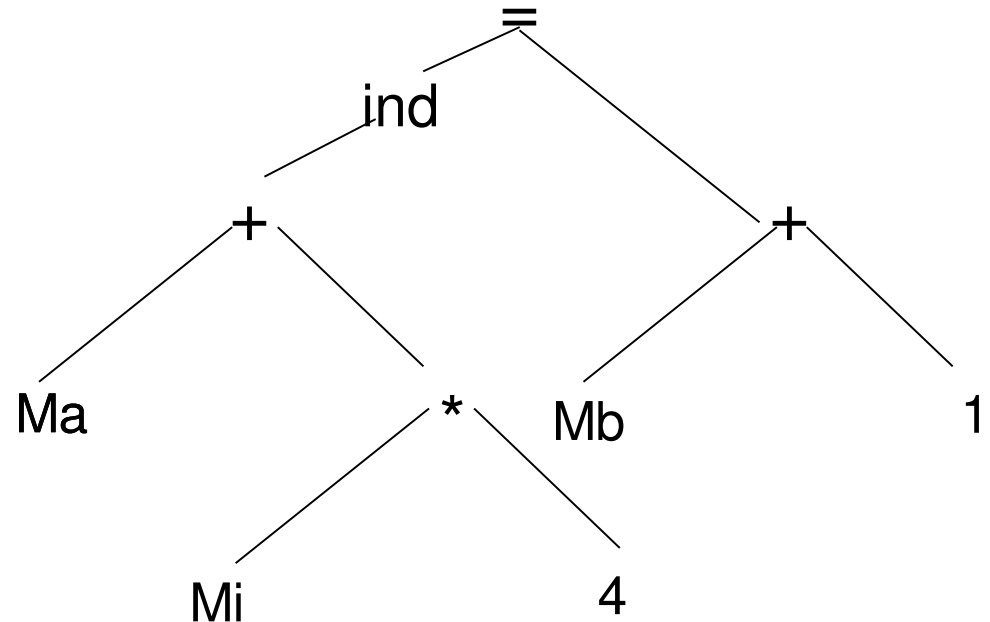
addi R4, R3, 1

lw R5, Ma

add R6, R5, R2

sw R4, R6

Código gerado



Árvore Sintática

Seleção de Instruções

- Exercício: Representar o código a seguir em 3-end, construir a árvore sintática e aplicar a seleção de instruções baseada em matching de árvores
 - a) $x = a * b + c * d$
 - b) $x[i] = y[i] * z[k]$
 - c) $x = x + 1$
- No item c), considere a existência de uma instrução `inc Ri` que incrementa `Ri` em uma unidade.
- Procure executar o matching utilizando os templates já definidos anteriormente.