

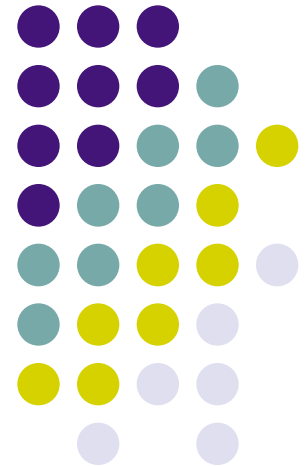
# Processo de Compilação

**TDE06**

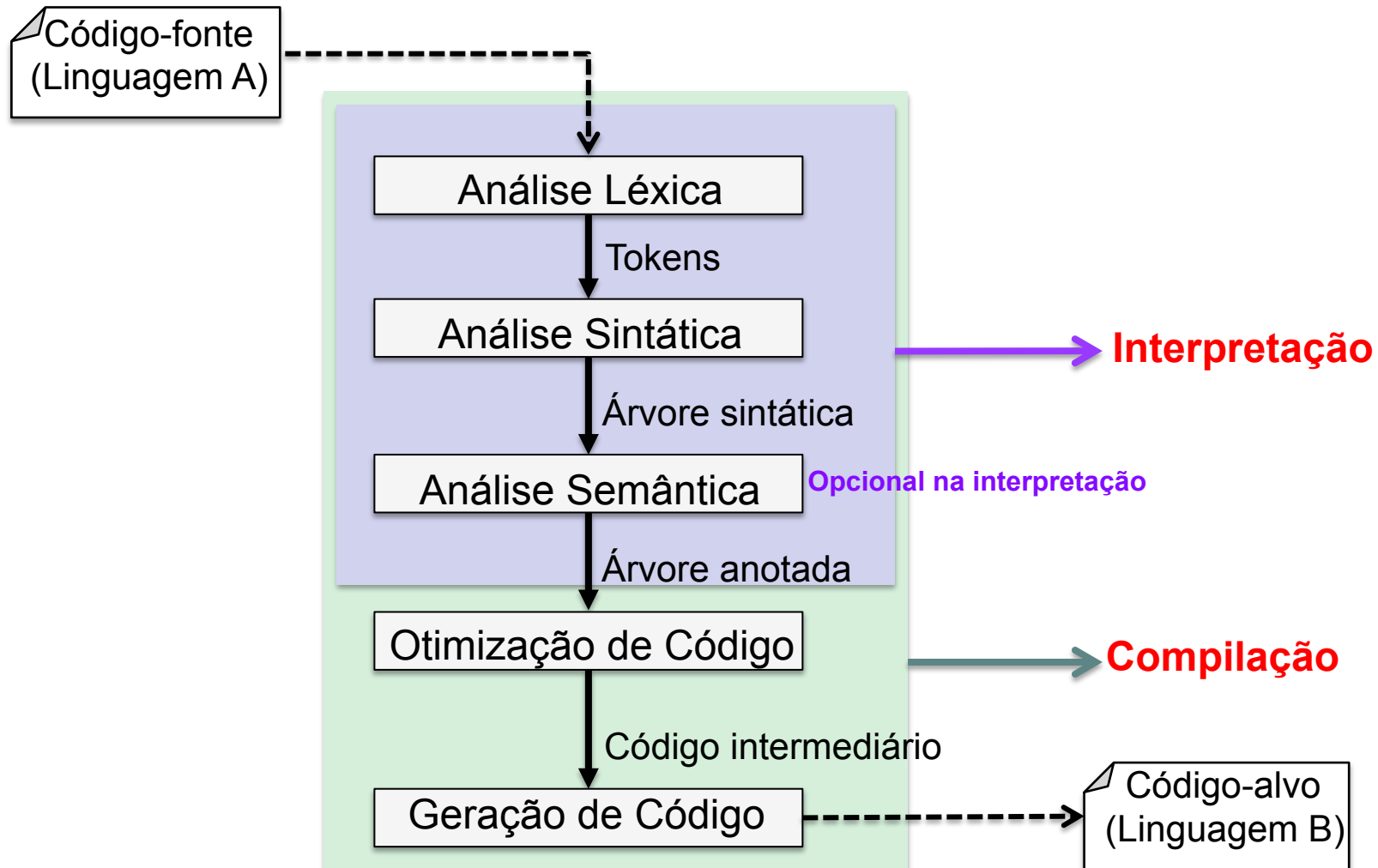
**Gregory Moro Puppi Wanderley**

***Pontifícia Universidade Católica do Paraná (PUCPR)***

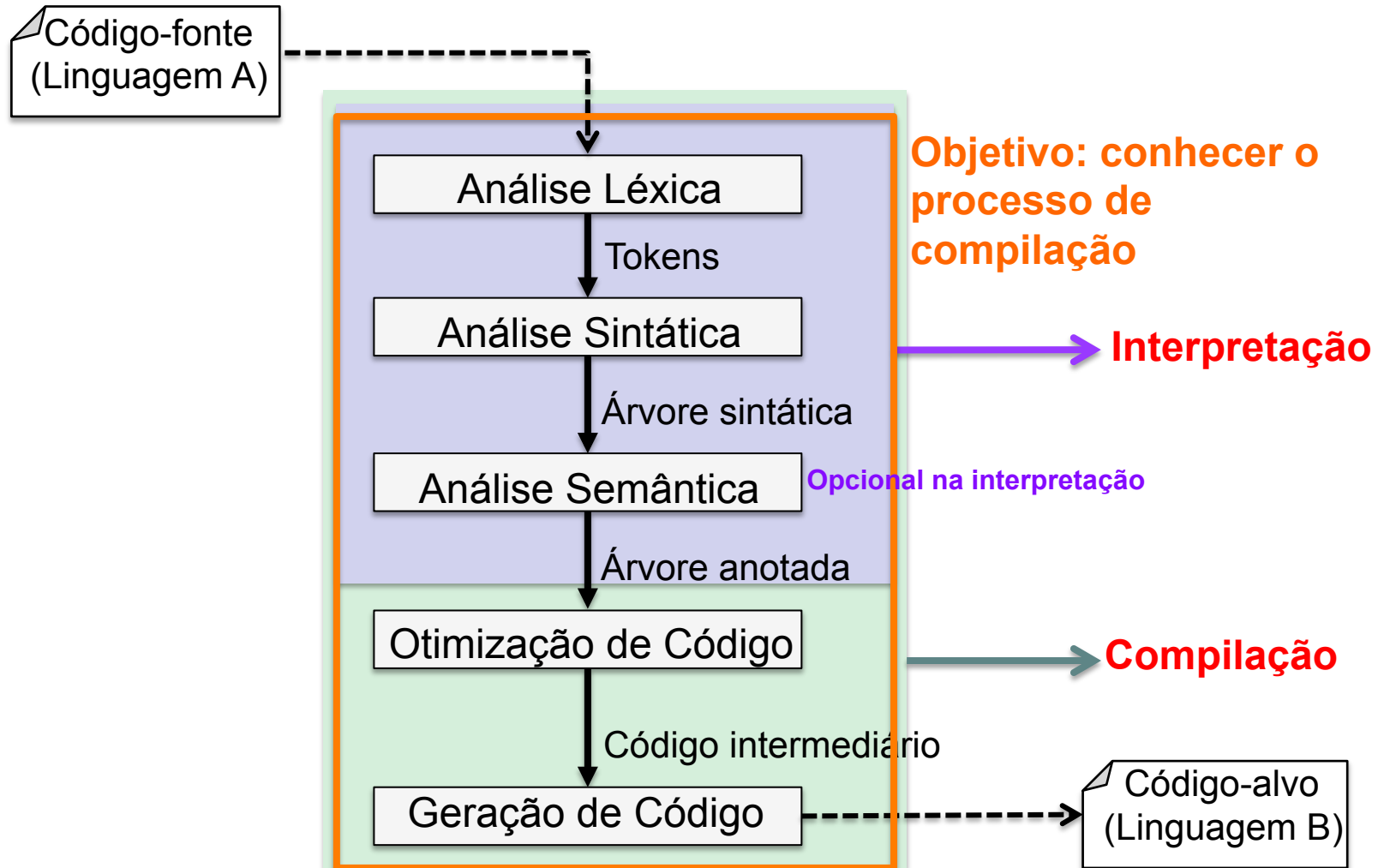
***Bacharelado em Ciência da Computação – 4º Período***

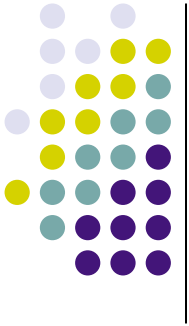


# Contexto



# Contexto





# Compilação

- **Compilador**
  - Recebe como entrada um programa escrito numa **linguagem-fonte**, e produz um programa equivalente numa **linguagem-alvo**.
    - Linguagem-fonte: C, C++, Pascal
    - Linguagem-alvo: linguagem de montagem (assembly), código de máquina (objeto)



# Compilação

- **Compilador**

- Recebe como entrada um programa escrito numa

**linguagem**  
**linguagem**

Linguagem-fonte  
(entrada)

```
int main()  
{  
    printf("Hello world!\n");  
    return 0;  
}
```

lente numa

- Linguagem

- Linguagem  
máquina

**Compilador**

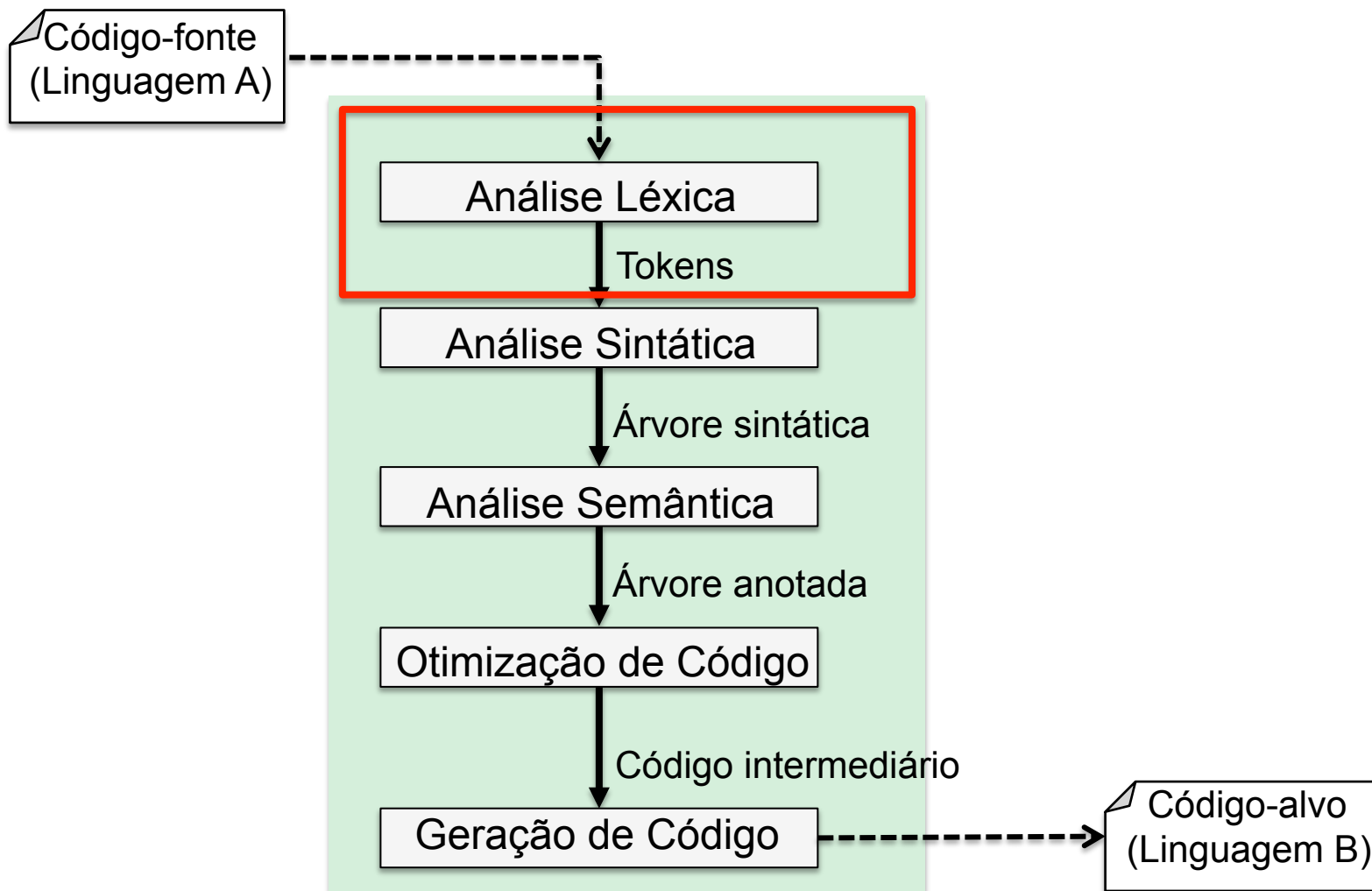
ly), código de

Linguagem-alvo  
(saída)

```
00111100  
10010011  
11100101  
00010010
```



# Análise Léxica





# Análise Léxica

- **Entrada da análise léxica:** código-fonte
- **Objetivo**
  - Ler o código-fonte e identificar sequências de caracteres que constituem unidades léxicas (**tokens**).
    - Tokens são classes de símbolos tais como palavras reservadas, delimitadores, identificadores, etc.



# Análise Léxica

- Exemplo

- Considere a linha de código de um programa em C:

```
a [index] = 4 + 2;
```

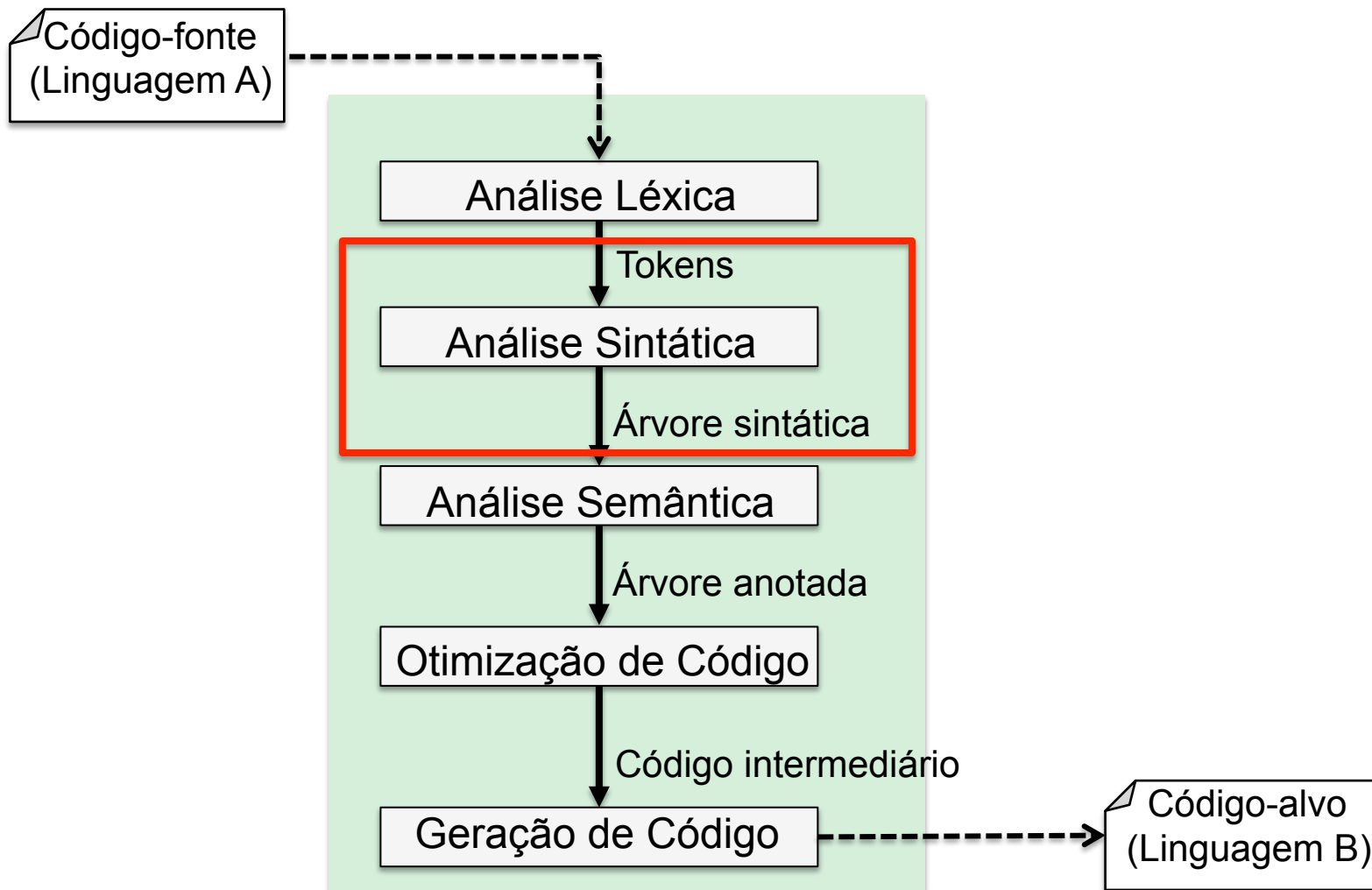
- O código tem 13 caracteres diferentes de espaços, mas só 9 tokens:

▪ a	identificador
▪ [	[
▪ Index	identificador
▪ ]	]
▪ =	=
▪ 4	número
▪ +	signal de adição
▪ 2	número
▪ ;	;





# Análise Sintática





# Análise Sintática

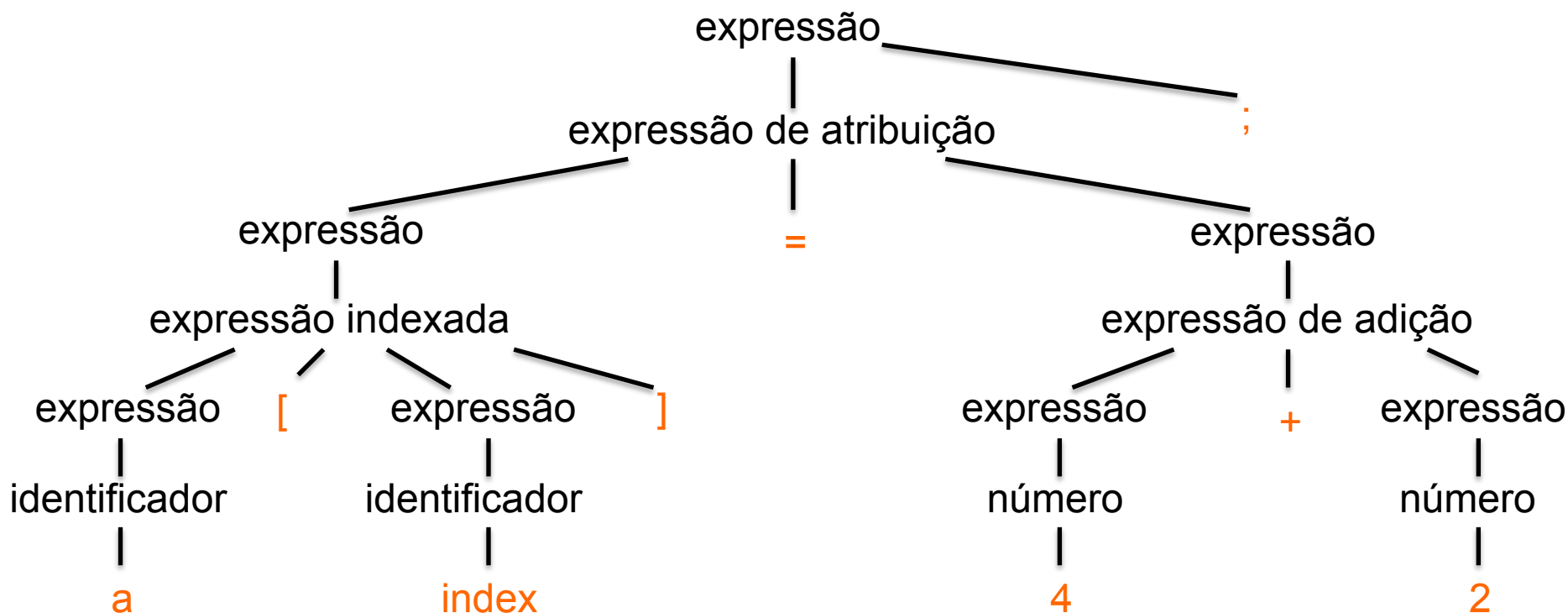
- **Entrada da análise sintática:** cadeia de tokens.
- **Objetivo**
  - Verificar se a estrutura gramatical do programa está correta
    - Isto é, se ela foi formada com base nas **regras gramaticais** da linguagem.



# Análise Sintática

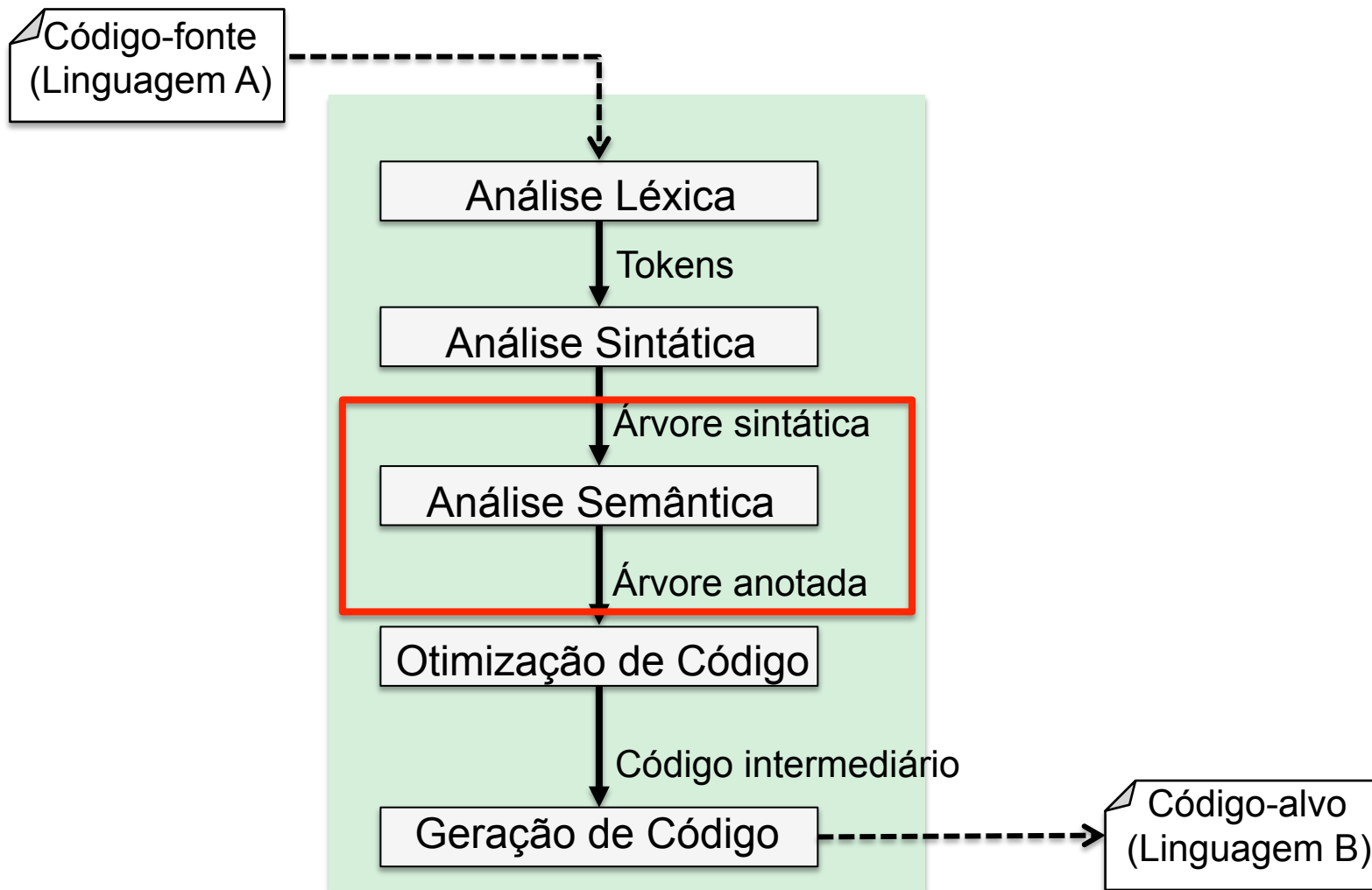
- Exemplo:

- Entrada (lista de tokens): `a`, `[`, `index`, `]`, `=`, `4`, `+`, `2`, `;`





# Análise Semântica





# Análise Semântica

- **Entrada da análise semântica:** árvore sintática.
- **Objetivo**
  - Verificar se as estruturas do programa irão fazer sentido (**significado** correto) durante a execução.
- **Verificação de tipos e declarações de variáveis.**
  - Anotação da árvore sintática.

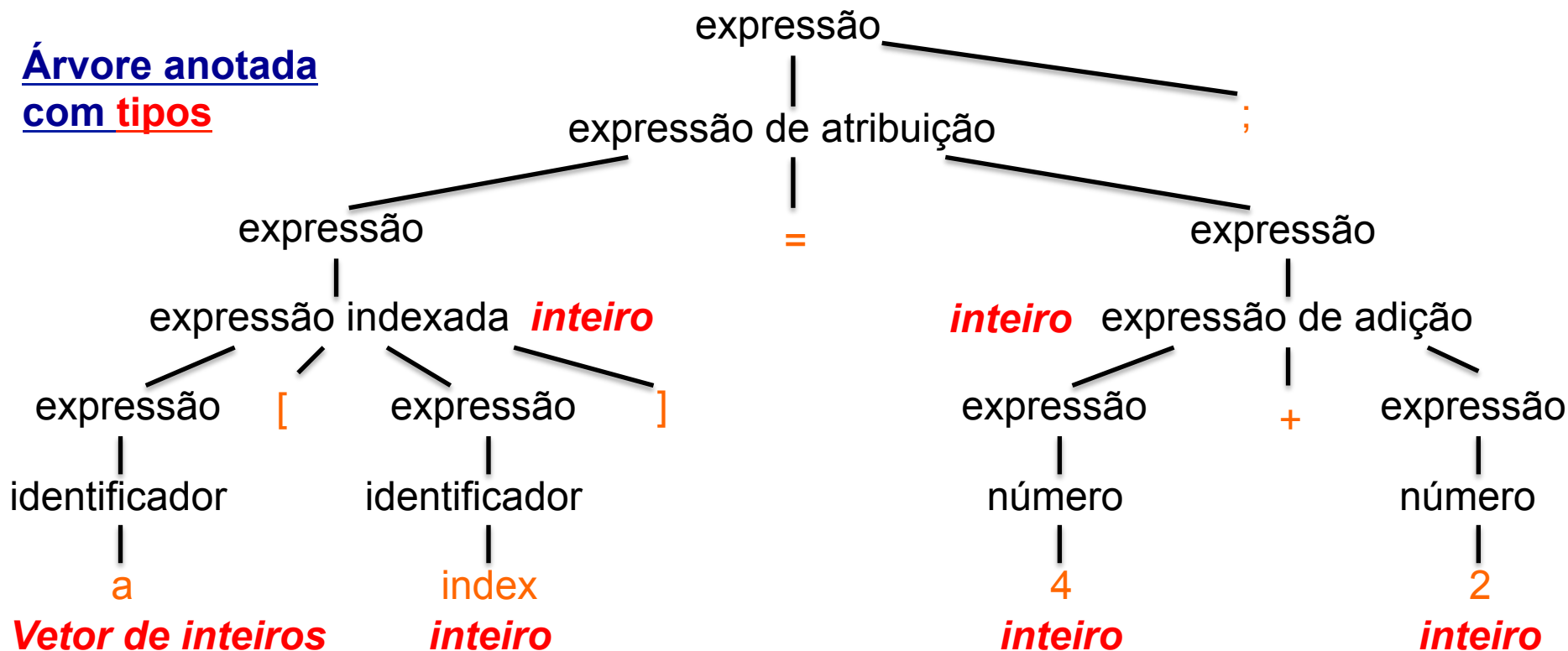


# Análise Semântica

- Exemplo:

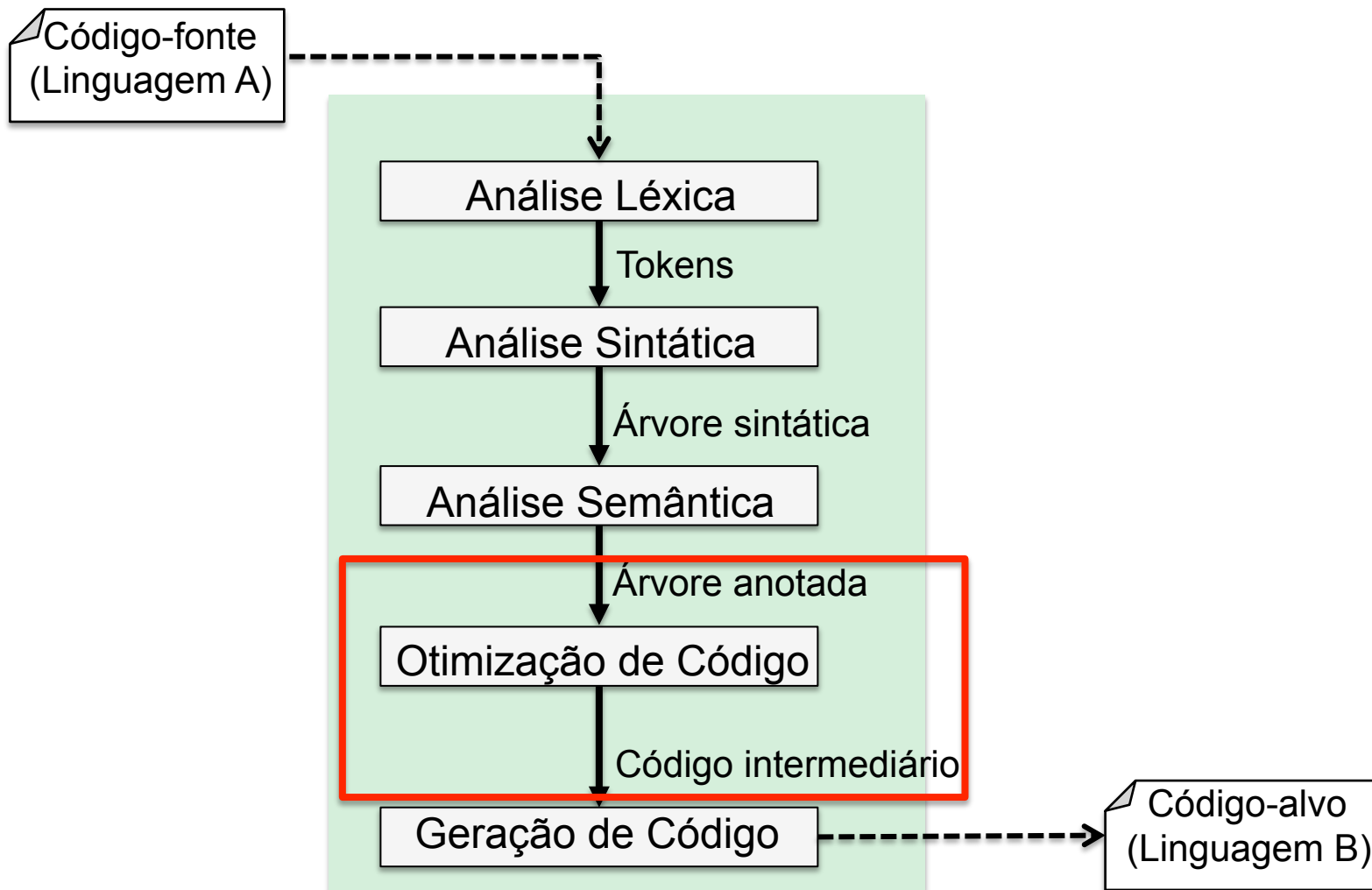
- Entrada: árvore sintática da expressão `a [index] = 4 + 2;`

## Árvore anotada com tipos





# Otimização de Código





# Otimização de Código

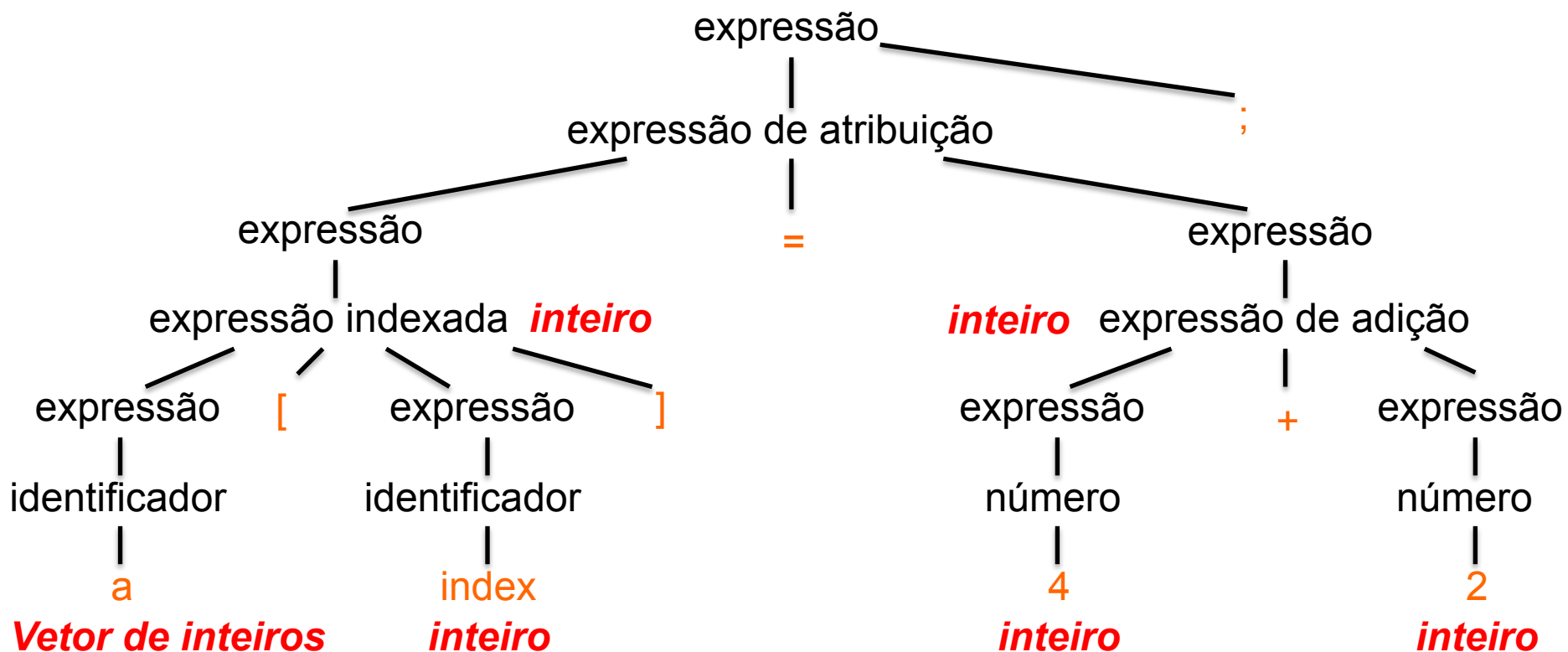
- **Entrada da otimização de código:** árvore sintática anotada.
- **Objetivo**
  - Realizar melhorias (otimizações) no código que está sendo traduzido.
    - Foco na **velocidade** e **tamanho** do código a ser executado.
  - Diversas otimizações podem ser feitas diretamente na árvore anotada.





# Otimização de Código

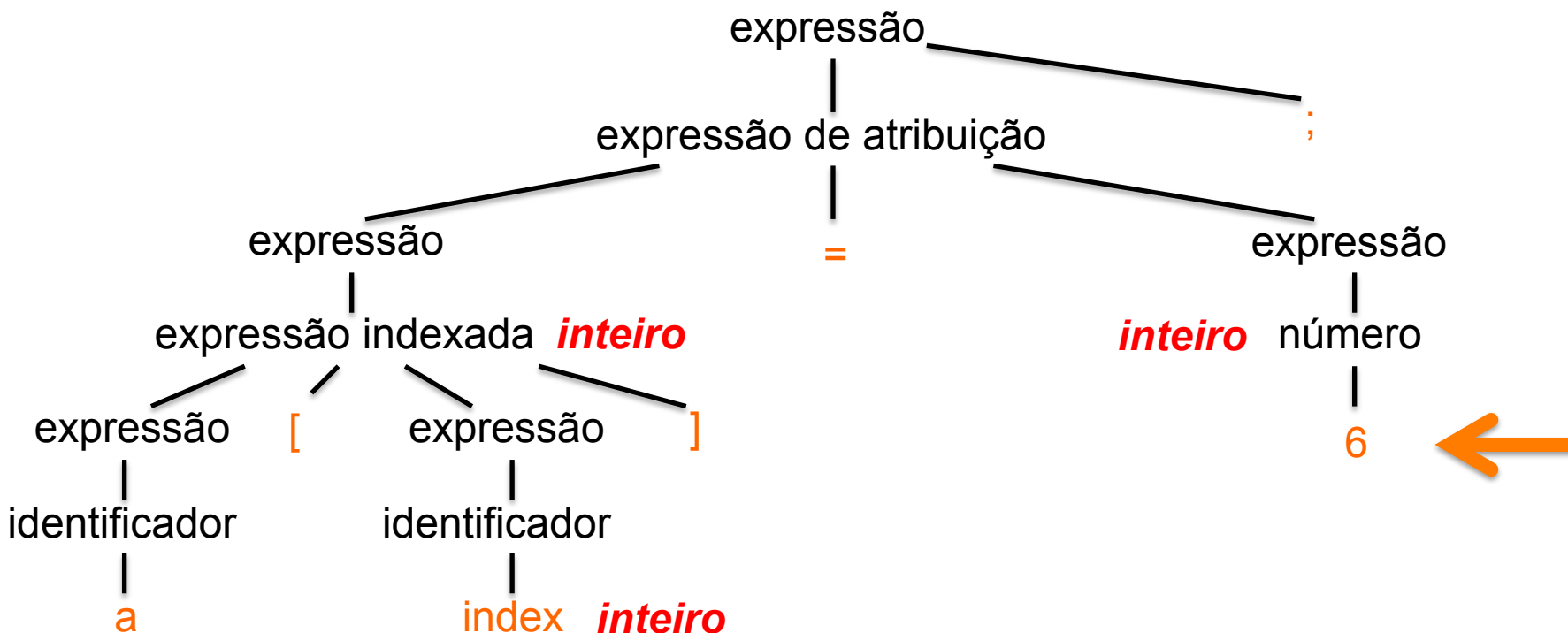
- Exemplo de otimização direta na árvore anotada
  - Entrada: árvore anotada da expressão `a [index] = 4 + 2;`





# Otimização de Código

- Exemplo de otimização direta na árvore anotada
  - A expressão  $4+2$  pode ser pré-computada pelo compilador, evitando realizá-la em tempo de execução.



**Vetor de inteiros**

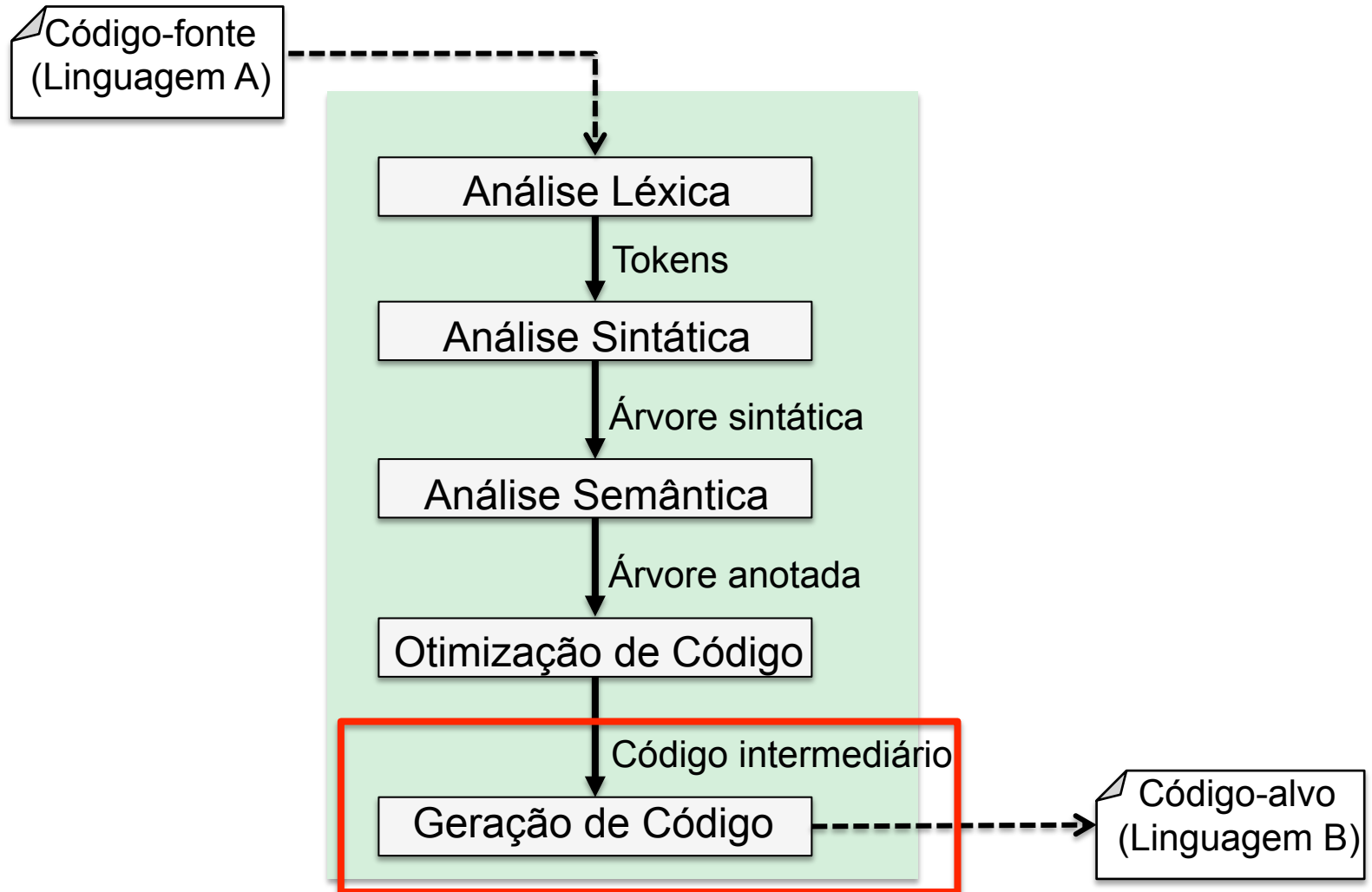


# Otimização de Código

- Em muitos casos pode ser mais fácil otimizar um forma mais linearizada da árvore.
  - Derivação de **código intermediário** mais próximo do formato do código de montagem (ex.: assembly). Exemplo:
    - $t = 4 + 2$
    - $a[\text{index}] = t$
  - Na sequência o código derivado é otimizado:
    - Passo 1:  $t = 6$
    - $a[\text{index}] = t$
    - Passo 2:  $a[\text{index}] = 6$



# Otimização de Código





# Geração de Código

- **Entrada da geração de código:** código intermediário.
- **Objetivo**
  - Gerar a partir do código intermediário o código para a máquina-alvo.
  - As propriedades da máquina-alvo são um dos principais fatores. Exemplo:
    - Representação de dados. Quantos bytes um inteiro ocupará na memória?
  - **Saída do gerador (pode ser):**
    - (a) **Código de montagem** (o qual será posteriormente traduzido para máquina através de um montador).
    - (b) **Código de máquina** direto.



# Geração de Código

- Exemplo

- Entrada: código intermediário otimizado  $a[\text{index}] = 6$
- Saída: código-alvo (ex.: assembly)
  - MOV R0, index
  - MUL R0, 2
  - MOV R1, &a
  - ADD R1, R0
  - MOV \*R1, 6