

Tradução dirigida pela sintaxe

Definições dirigidas pela sintaxe

Prof. Edson Alves

Faculdade UnB Gama

Sumário

1. Construção de árvores sintáticas

Construção de árvores sintáticas para expressões

- ▶ Árvores sintáticas para expressões podem ser construídas de forma semelhante à tradução para notação posfixa

Construção de árvores sintáticas para expressões

- ▶ Árvores sintáticas para expressões podem ser construídas de forma semelhante à tradução para notação posfixa
- ▶ Deve ser construído um nó para cada operação e cada operando

Construção de árvores sintáticas para expressões

- ▶ Árvores sintáticas para expressões podem ser construídas de forma semelhante à tradução para notação posfixa
- ▶ Deve ser construído um nó para cada operação e cada operando
- ▶ Os filhos do nó de um operador ser subárvores que representam as subexpressões que constituem os operandos daquele operador

Construção de árvores sintáticas para expressões

- ▶ Árvores sintáticas para expressões podem ser construídas de forma semelhante à tradução para notação posfixa
- ▶ Deve ser construído um nó para cada operação e cada operando
- ▶ Os filhos do nó de um operador ser subárvores que representam as subexpressões que constituem os operandos daquele operador
- ▶ Cada nó pode ser implementado como um registro com vários campos que caracterizam o nó

Construção de árvores sintáticas para expressões

- ▶ Árvores sintáticas para expressões podem ser construídas de forma semelhante à tradução para notação posfixa
- ▶ Deve ser construído um nó para cada operação e cada operando
- ▶ Os filhos do nó de um operador ser subárvores que representam as subexpressões que constituem os operandos daquele operador
- ▶ Cada nó pode ser implementado como um registro com vários campos que caracterizam o nó
- ▶ O registro de nós que representam operadores devem conter um campo que identifica o operador e os demais campos devem ser ponteiros para os operandos

Construção de árvores sintáticas para expressões

- ▶ Árvores sintáticas para expressões podem ser construídas de forma semelhante à tradução para notação posfixa
- ▶ Deve ser construído um nó para cada operação e cada operando
- ▶ Os filhos do nó de um operador ser subárvores que representam as subexpressões que constituem os operandos daquele operador
- ▶ Cada nó pode ser implementado como um registro com vários campos que caracterizam o nó
- ▶ O registro de nós que representam operadores devem conter um campo que identifica o operador e os demais campos devem ser ponteiros para os operandos
- ▶ As folhas das árvores contém os tokens

Construção de árvores sintáticas para expressões

- ▶ Árvores sintáticas para expressões podem ser construídas de forma semelhante à tradução para notação posfixa
- ▶ Deve ser construído um nó para cada operação e cada operando
- ▶ Os filhos do nó de um operador ser subárvores que representam as subexpressões que constituem os operandos daquele operador
- ▶ Cada nó pode ser implementado como um registro com vários campos que caracterizam o nó
- ▶ O registro de nós que representam operadores devem conter um campo que identifica o operador e os demais campos devem ser ponteiros para os operandos
- ▶ As folhas das árvores contém os tokens
- ▶ O registro de uma folha deve identificar o token e também armazenar um ponteiro para a entrada do token na tabela de símbolos

Funções para a criação de nós da árvore sintática de uma expressão

Cada uma das funções abaixo retorna um ponteiro para o nó criado. Assuma que os operadores são todos binários.

Funções para a criação de nós da árvore sintática de uma expressão

Cada uma das funções abaixo retorna um ponteiro para o nó criado. Assuma que os operadores são todos binários.

1. $\text{CRIARNO}(op, L, R)$: cria um nó de operador cujo rótulo é op , L é o ponteiro do operando à esquerda e R o ponteiro do operando à direita

Funções para a criação de nós da árvore sintática de uma expressão

Cada uma das funções abaixo retorna um ponteiro para o nó criado. Assuma que os operadores são todos binários.

1. $\text{CRIARNO}(op, L, R)$: cria um nó de operador cujo rótulo é op , L é o ponteiro do operando à esquerda e R o ponteiro do operando à direita
2. $\text{CRIARFOLHA}(\text{id}, p)$: cria um nó para um identificador com rótulo **id**, onde p é o ponteiro para o identificador na tabela de símbolos

Funções para a criação de nós da árvore sintática de uma expressão

Cada uma das funções abaixo retorna um ponteiro para o nó criado. Assuma que os operadores são todos binários.

1. $\text{CRIARNO}(op, L, R)$: cria um nó de operador cujo rótulo é op , L é o ponteiro do operando à esquerda e R o ponteiro do operando à direita
2. $\text{CRIARFOLHA}(\text{id}, p)$: cria um nó para um identificador com rótulo **id**, onde p é o ponteiro para o identificador na tabela de símbolos
3. $\text{CRIARFOLHA}(\text{num}, val)$: cria um nó para um número, com rótulo **num**, cujo valor é indicado por val

Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

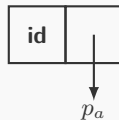
Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$

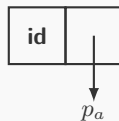
Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$


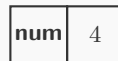
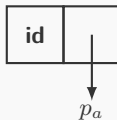
Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{num}, 4)$$


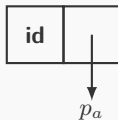
Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{num}, 4)$$


Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{num}, 4)$$
$$p_3 := \text{CRIARNO}(+, p_1, p_2)$$


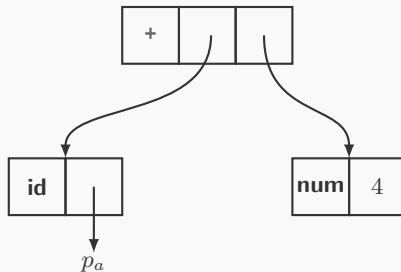
Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\mathbf{num}, 4)$

$p_3 := \text{CRIARNO}(+, p_1, p_2)$



Criação da árvore sintática da expressão $a - 4 + c$

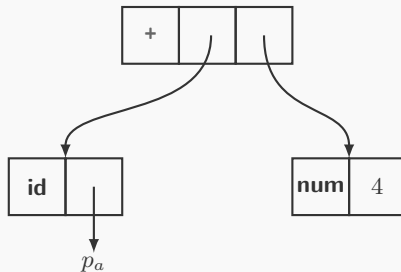
Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\mathbf{num}, 4)$

$p_3 := \text{CRIARNO}(+, p_1, p_2)$

$p_4 := \text{CRIARFOLHA}(\mathbf{id}, p_c)$



Criação da árvore sintática da expressão $a - 4 + c$

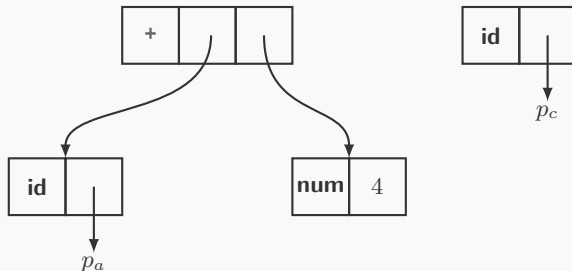
Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{num}, 4)$

$p_3 := \text{CRIARNO}(+, p_1, p_2)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$



Criação da árvore sintática da expressão $a - 4 + c$

Chamadas de funções

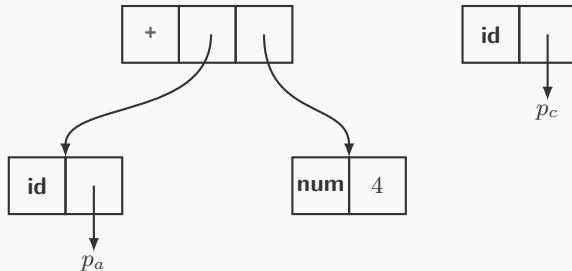
$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{num}, 4)$

$p_3 := \text{CRIARNO}(+, p_1, p_2)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$



Criação da árvore sintática da expressão $a - 4 + c$

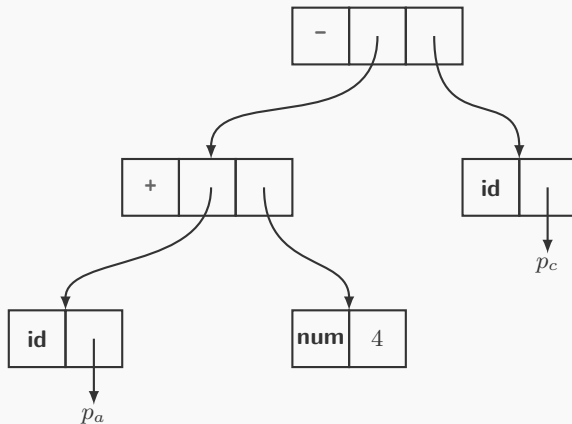
Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$$

$$p_2 := \text{CRIARFOLHA}(\text{num}, 4)$$

$$p_3 := \text{CRIARNO}(+, p_1, p_2)$$

$$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$$

$$p_5 := \text{CRIARNO}(-, p_3, p_4)$$


Definição dirigida pela sintaxe para a construção de árvores sintáticas

- ▶ É possível construir árvores sintáticas para expressões por meio de uma definição S-atribuída

Definição dirigida pela sintaxe para a construção de árvores sintáticas

- ▶ É possível construir árvores sintáticas para expressões por meio de uma definição S-atribuída
- ▶ As regras semânticas agendam as chamadas das funções de criação de nós que irão construir a árvore

Definição dirigida pela sintaxe para a construção de árvores sintáticas

- ▶ É possível construir árvores sintáticas para expressões por meio de uma definição S-atribuída
- ▶ As regras semânticas agendam as chamadas das funções de criação de nós que irão construir a árvore
- ▶ O atributo sintetizado *nptr* controla os ponteiros para os nós retornados pelas funções

Definição dirigida pela sintaxe para a construção de árvores sintáticas

- ▶ É possível construir árvores sintáticas para expressões por meio de uma definição S-atribuída
- ▶ As regras semânticas agendam as chamadas das funções de criação de nós que irão construir a árvore
- ▶ O atributo sintetizado *nptr* controla os ponteiros para os nós retornados pelas funções
- ▶ O atributo *entrada* armazena o endereço de um token na tabela de símbolos e o atributo *val* o valor de um número

Definição dirigida pela sintaxe para a construção de árvores sintáticas

- ▶ É possível construir árvores sintáticas para expressões por meio de uma definição S-atribuída
- ▶ As regras semânticas agendam as chamadas das funções de criação de nós que irão construir a árvore
- ▶ O atributo sintetizado *nptr* controla os ponteiros para os nós retornados pelas funções
- ▶ O atributo *entrada* armazena o endereço de um token na tabela de símbolos e o atributo *val* o valor de um número
- ▶ Estes dois atributos devem ser computados na análise léxica

Definição dirigida pela sintaxe para expressões aritméticas de adição e subtração

Produção	Regra semântica
$E \rightarrow E_1 + T$	$E.nptr := \text{CRIARNO}(+, E_1.nptr, T.nptr)$
$E \rightarrow E_1 - T$	$E.nptr := \text{CRIARNO}(-, E_1.nptr, T.nptr)$
$E \rightarrow T$	$E.nptr := T.nptr$
$T \rightarrow (E)$	$T.nptr := E.nptr$
$T \rightarrow \text{id}$	$T.nptr := \text{CRIARNO}(\text{id}, \text{id.entrada})$
$T \rightarrow \text{num}$	$T.nptr := \text{CRIARNO}(\text{num}, \text{num.val})$

DAG

- ▶ Um grafo direcionado acíclico (*directed acyclic graph* – DAG) é um grafo cujas arestas são direcionadas e que não possui ciclos

DAG

- ▶ Um grafo direcionado acíclico (*directed acyclic graph* – *DAG*) é um grafo cujas arestas são direcionadas e que não possui ciclos
- ▶ Um DAG pode ser usado para identificar subexpressões comuns em uma expressão

DAG

- ▶ Um grafo direcionado acíclico (*directed acyclic graph* – DAG) é um grafo cujas arestas são direcionadas e que não possui ciclos
- ▶ Um DAG pode ser usado para identificar subexpressões comuns em uma expressão
- ▶ De forma similar às árvores sintáticas, um nó representa um operador e seus filhos representam os operandos

DAG

- ▶ Um grafo direcionado acíclico (*directed acyclic graph* – DAG) é um grafo cujas arestas são direcionadas e que não possui ciclos
- ▶ Um DAG pode ser usado para identificar subexpressões comuns em uma expressão
- ▶ De forma similar às árvores sintáticas, um nó representa um operador e seus filhos representam os operandos
- ▶ Se houver uma ou mais expressões comuns, os nós do DAG podem ter “mais de um pai”

DAG

- ▶ Um grafo direcionado acíclico (*directed acyclic graph* – DAG) é um grafo cujas arestas são direcionadas e que não possui ciclos
- ▶ Um DAG pode ser usado para identificar subexpressões comuns em uma expressão
- ▶ De forma similar às árvores sintáticas, um nó representa um operador e seus filhos representam os operandos
- ▶ Se houver uma ou mais expressões comuns, os nós do DAG podem ter “mais de um pai”
- ▶ Nas árvores sintáticas, expressões comuns são duplicadas na árvore

Construção do DAG a partir de uma definição S-atribuída

- ▶ Uma definição S-atribuída para a construção de árvores sintáticas para expressões aritméticas de adições e subtrações pode se adaptada para a construção do DAG

Construção do DAG a partir de uma definição S-atribuída

- ▶ Uma definição S-atribuída para a construção de árvores sintáticas para expressões aritméticas de adições e subtrações pode se adaptada para a construção do DAG
- ▶ De fato, basta modificar o comportamento das funções `CRIARNO()` e `CRIARFOLHA()`

Construção do DAG a partir de uma definição S-atribuída

- ▶ Uma definição S-atribuída para a construção de árvores sintáticas para expressões aritméticas de adições e subtrações pode se adaptada para a construção do DAG
- ▶ De fato, basta modificar o comportamento das funções `CRIARNO()` e `CRIARFOLHA()`
- ▶ Ao invés de criar um novo nó a cada chamada, estas funções devem verificar se os parâmetros passados já não foram usados para construir um nó

Construção do DAG a partir de uma definição S-atribuída

- ▶ Uma definição S-atribuída para a construção de árvores sintáticas para expressões aritméticas de adições e subtrações pode se adaptada para a construção do DAG
- ▶ De fato, basta modificar o comportamento das funções `CRIARNO()` e `CRIARFOLHA()`
- ▶ Ao invés de criar um novo nó a cada chamada, estas funções devem verificar se os parâmetros passados já não foram usados para construir um nó
- ▶ Em caso afirmativo, as funções devem retornar o ponteiro usado anteriormente na criação do nó

Construção do DAG a partir de uma definição S-atribuída

- ▶ Uma definição S-atribuída para a construção de árvores sintáticas para expressões aritméticas de adições e subtrações pode se adaptada para a construção do DAG
- ▶ De fato, basta modificar o comportamento das funções `CRIARNO()` e `CRIARFOLHA()`
- ▶ Ao invés de criar um novo nó a cada chamada, estas funções devem verificar se os parâmetros passados já não foram usados para construir um nó
- ▶ Em caso afirmativo, as funções devem retornar o ponteiro usado anteriormente na criação do nó
- ▶ Caso contrário, deve ser criado um novo nó e o ponteiro criado deve ser armazenado em uma tabela, associado aos parâmetros usados, para consulta posterior

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$

a

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$

a

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_3 := \text{CRIARFOLHA}(\mathbf{id}, p_b)$$

a

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_3 := \text{CRIARFOLHA}(\mathbf{id}, p_b)$$

a

b

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_3 := \text{CRIARFOLHA}(\mathbf{id}, p_b)$$
$$p_4 := \text{CRIARFOLHA}(\mathbf{id}, p_c)$$

a

b

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_3 := \text{CRIARFOLHA}(\mathbf{id}, p_b)$$
$$p_4 := \text{CRIARFOLHA}(\mathbf{id}, p_c)$$

a

b

c

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\mathbf{id}, p_a)$$
$$p_3 := \text{CRIARFOLHA}(\mathbf{id}, p_b)$$
$$p_4 := \text{CRIARFOLHA}(\mathbf{id}, p_c)$$
$$p_5 := \text{CRIARNO}(-, p_3, p_4)$$

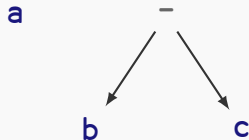
a

b

c

Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$$
$$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$$
$$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$$
$$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$$
$$p_5 := \text{CRIARNO}(-, p_3, p_4)$$


Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

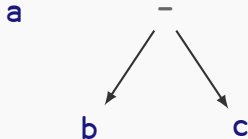
$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$

$p_6 := \text{CRIARNO}(\times, p_2, p_5)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

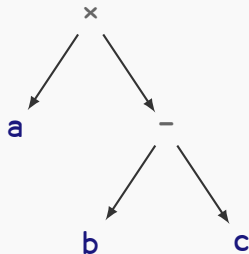
$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$

$p_6 := \text{CRIARNO}(\times, p_2, p_5)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

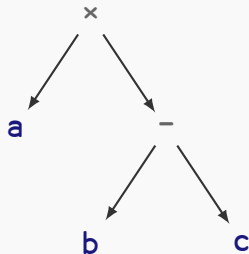
$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$

$p_6 := \text{CRIARNO}(\times, p_2, p_5)$

$p_7 := \text{CRIARNO}(+, p_1, p_6)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

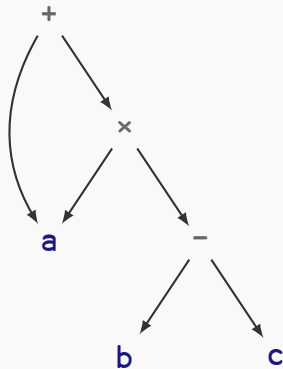
$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$

$p_6 := \text{CRIARNO}(\times, p_2, p_5)$

$p_7 := \text{CRIARNO}(+, p_1, p_6)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

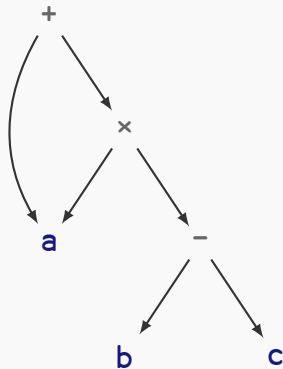
$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$

$p_6 := \text{CRIARNO}(\times, p_2, p_5)$

$p_7 := \text{CRIARNO}(+, p_1, p_6)$

$p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

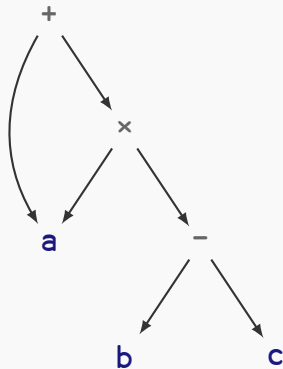
$p_5 := \text{CRIARNO}(-, p_3, p_4)$

$p_6 := \text{CRIARNO}(\times, p_2, p_5)$

$p_7 := \text{CRIARNO}(+, p_1, p_6)$

$p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$

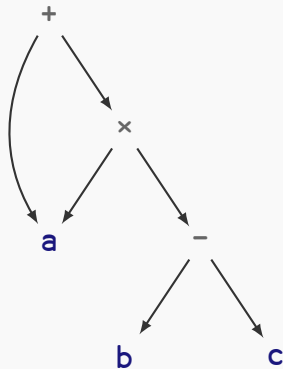
$p_6 := \text{CRIARNO}(\times, p_2, p_5)$

$p_7 := \text{CRIARNO}(+, p_1, p_6)$

$p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_{10} := \text{CRIARNO}(-, p_8, p_9)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$

$p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_5 := \text{CRIARNO}(-, p_3, p_4)$

$p_6 := \text{CRIARNO}(\times, p_2, p_5)$

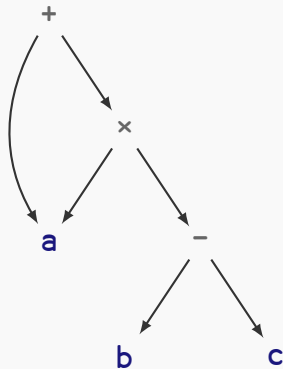
$p_7 := \text{CRIARNO}(+, p_1, p_6)$

$p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$

$p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$

$p_{10} := \text{CRIARNO}(-, p_8, p_9)$

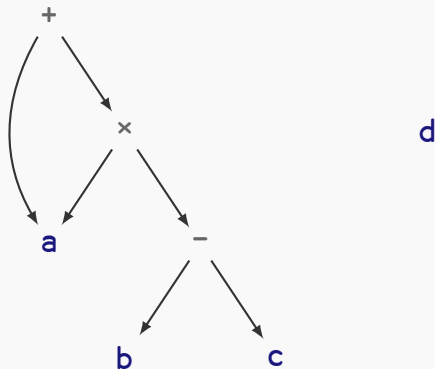
$p_{11} := \text{CRIARFOLHA}(\text{id}, p_d)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

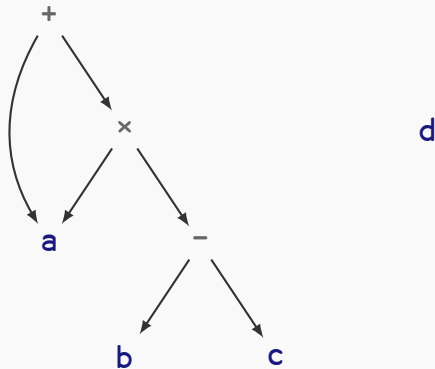
$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_5 := \text{CRIARNO}(-, p_3, p_4)$
 $p_6 := \text{CRIARNO}(\times, p_2, p_5)$
 $p_7 := \text{CRIARNO}(+, p_1, p_6)$
 $p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_{10} := \text{CRIARNO}(-, p_8, p_9)$
 $p_{11} := \text{CRIARFOLHA}(\text{id}, p_d)$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

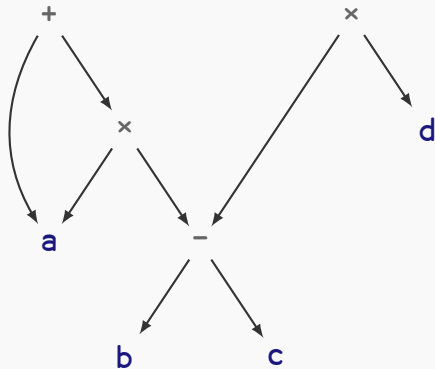
$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_5 := \text{CRIARNO}(-, p_3, p_4)$
 $p_6 := \text{CRIARNO}(\times, p_2, p_5)$
 $p_7 := \text{CRIARNO}(+, p_1, p_6)$
 $p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_{10} := \text{CRIARNO}(-, p_8, p_9)$
 $p_{11} := \text{CRIARFOLHA}(\text{id}, p_d)$
 $p_{12} := \text{CRIARNO}(\times, p_{10}, p_{11})$



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_5 := \text{CRIARNO}(-, p_3, p_4)$
 $p_6 := \text{CRIARNO}(\times, p_2, p_5)$
 $p_7 := \text{CRIARNO}(+, p_1, p_6)$
 $p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_{10} := \text{CRIARNO}(-, p_8, p_9)$
 $p_{11} := \text{CRIARFOLHA}(\text{id}, p_d)$
 $p_{12} := \text{CRIARNO}(\times, p_{10}, p_{11})$



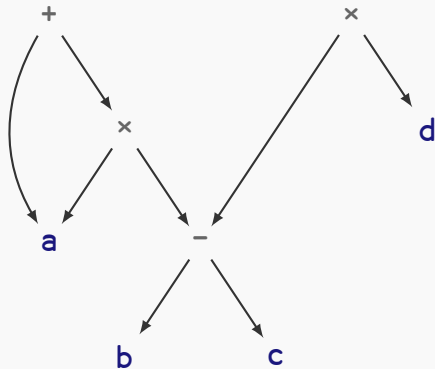
Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

```

 $p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$ 
 $p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$ 
 $p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$ 
 $p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$ 
 $p_5 := \text{CRIARNO}(-, p_3, p_4)$ 
 $p_6 := \text{CRIARNO}(\times, p_2, p_5)$ 
 $p_7 := \text{CRIARNO}(+, p_1, p_6)$ 
 $p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$ 
 $p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$ 
 $p_{10} := \text{CRIARNO}(-, p_8, p_9)$ 
 $p_{11} := \text{CRIARFOLHA}(\text{id}, p_d)$ 
 $p_{12} := \text{CRIARNO}(\times, p_{10}, p_{11})$ 
 $p_{13} := \text{CRIARNO}(+, p_7, p_{12})$ 

```



Criação do DAG para a expressão $a + a \times (b - c) + (b - c) \times d$

Chamadas de funções

$p_1 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_2 := \text{CRIARFOLHA}(\text{id}, p_a)$
 $p_3 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_4 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_5 := \text{CRIARNO}(-, p_3, p_4)$
 $p_6 := \text{CRIARNO}(\times, p_2, p_5)$
 $p_7 := \text{CRIARNO}(+, p_1, p_6)$
 $p_8 := \text{CRIARFOLHA}(\text{id}, p_b)$
 $p_9 := \text{CRIARFOLHA}(\text{id}, p_c)$
 $p_{10} := \text{CRIARNO}(-, p_8, p_9)$
 $p_{11} := \text{CRIARFOLHA}(\text{id}, p_d)$
 $p_{12} := \text{CRIARNO}(\times, p_{10}, p_{11})$
 $p_{13} := \text{CRIARNO}(+, p_7, p_{12})$

