

Primeira Parte - Aquecimento

(0.5) 1. O flex produz:

- A. Analisadores semânticos, em C
- B. Analisadores sintácticos, em Java
- C. Analisadores lexicais, em C
- D. Compiladores para código MIPS

(0.5) 2. Em C ou Java, a sintaxe `a = b = 2 + 1` é válida. Isso acontece porque:

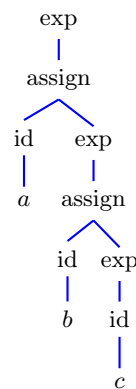
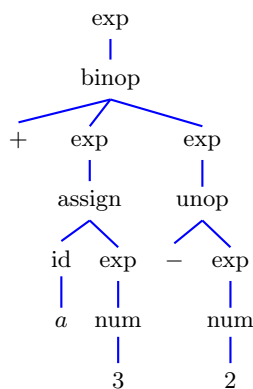
- A. `2 + 1` é uma expressão
- B. `a = b` é o nome de uma variável à qual se afecta `2 + 1`
- C. uma afectação é uma expressão e produz um valor
- D. uma afectação é uma *statement* e tem uma sintaxe abstracta

(2) 3. As APTs das seguintes expressões estão correctas?

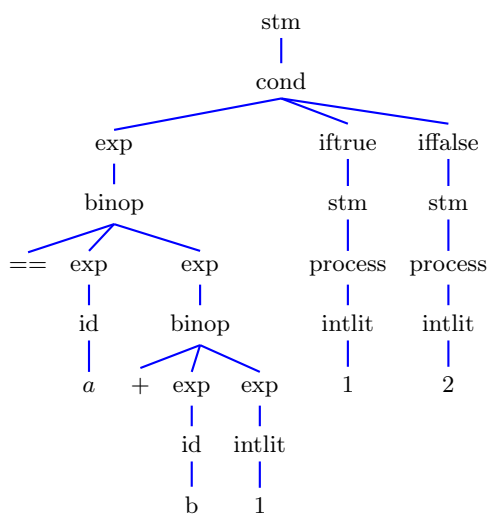
Justifique, no caso de não estarem correctas.

(a) `a = 3 + -2`

(b) `a = b = c`



(1.5) (c) `if a == b+1 then process(1) else process(2)`



Segunda Parte - Aceleração

(2.5) 4. Considere os literais numéricos de vírgula flutuante, presentes na maioria das linguagens de programação:

- têm sempre um ponto decimal (.);
- o zero antes do ponto é opcional (**1.2**, **.234**);
- a parte decimal é obrigatória (**23.**, **0.**);
- são permitidos números em notação científica (**2.443E50**, **2.443e-20**, **.34e+12**):
 - o *E* pode ser maiúsculo ou minúsculo;
 - o número que segue o *E* é sempre inteiro, e pode ter um sinal $+$ ou $-$.

Indique uma expressão regular para reconhecer qualquer literal de vírgula flutuante, nesta forma.

5. Considere um *subset* de uma nova linguagem de programação, *yalang*, com a seguinte sintaxe:

- Tipos atômicos: `int`, `float`, `string`
- Literais (com a forma “habitual”)
- Variáveis (idem)
- Declarações: `variável : tipo`
ou
`variável : tipo = valor` (o valor pode ser qualquer expressão válida)
- Expressões: afectações (símbolo `=`) e operações matemáticas
- Funções pré-definidas (constituem palavras reservadas): `output(expressão)` e `input(variável)` (as chamadas a estas funções são statements, sem qualquer valor)
- O separador de instruções é o `;` (ponto-e-vírgula)

Exemplo:

```
a : int;  
b : string = "Hello, world";  
input(a);  
c : float = a + b;  
output(c * 4 + 1);
```

- (3) (a) Proponha uma gramática para esta linguagem.
- (3) (b) Proponha uma sintaxe abstracta para a linguagem.
- (3) (c) Mostre a APT para o exemplo fornecido, de acordo com a sintaxe abstracta que propôs na alínea anterior.

Terceira Parte - Turbo

(2) 6. Considerando ainda o exemplo da linguagem *yalang* e, assumindo que a análise semântica continua apesar dos erros, diga um resultado possível para a análise semântica do exemplo dado.

7. Relativamente à linguagem *yalang* e à gramática que propôs anteriormente, dê um exemplo de uma declaração:

- (1) (a) com um erro sintáctico.
- (1) (b) com um erro sintáctico que não seja detectado pela gramática.