

1. (5 pontos) Considere a seguinte gramática, em que  $\Sigma = \{0, 1, 2\}$ :

$$S \rightarrow 0S1S \mid 0S \mid 2$$

- (a) (2 pontos) Apresente o conjunto de itens LR(0) para esta gramática.

- (b) (3 pontos) Apresente a tabela SLR para esta gramática.

2. (5 pontos) Considere a seguinte gramática que denota a sintaxe abstrata da linguagem Imp.

$$\begin{aligned} p &\rightarrow blk \\ s &\rightarrow skip \mid t \ x \ init \mid read \ x \mid print \ e \\ &\quad \mid x := e \mid if \ e \ then \ blk \ else \ blk \\ &\quad \mid while \ e \ blk \\ init &\rightarrow := e \mid \bullet \\ blk &\rightarrow slist \\ slist &\rightarrow s \ slist \mid \bullet \\ t &\rightarrow int \mid bool \\ e &\rightarrow v \mid x \mid e \ op \ e \mid not \ e \\ op &\rightarrow + \mid * \mid - \mid / \mid == \mid < \mid \&\& \\ v &\rightarrow n \mid true \mid false \end{aligned}$$

As regras do sistema de tipos de para expressões da linguagem Imp são apresentadas a seguir.

$$\begin{array}{l} \frac{}{\Gamma \vdash n : int} \qquad \frac{b = true \vee b = false}{\Gamma \vdash b : bool} \\ \frac{\Gamma(x) = t}{\Gamma \vdash x : t} \qquad \frac{\Gamma \vdash e : bool}{\Gamma \vdash not \ e : bool} \\ \frac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 + e_2 : int} \qquad \frac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 - e_2 : int} \\ \frac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 * e_2 : int} \qquad \frac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 / e_2 : int} \\ \frac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 == e_2 : bool} \qquad \frac{\Gamma \vdash e_1 : int \quad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1 < e_2 : bool} \\ \frac{\Gamma \vdash e_1 : bool \quad \Gamma \vdash e_2 : bool}{\Gamma \vdash e_1 \&\& e_2 : bool} \end{array}$$

As regras para verificação de comandos são:

$$\begin{array}{l} \frac{}{\Gamma \vdash skip \rightsquigarrow \Gamma} \qquad \frac{\Gamma \vdash e : t}{\Gamma \vdash t \ x := e \rightsquigarrow \Gamma, x : t} \\ \frac{\Gamma \vdash e : \Gamma(x) \quad \exists t. \Gamma(x) = t}{\Gamma \vdash x := e \rightsquigarrow \Gamma \quad \Gamma \vdash read \ x \rightsquigarrow \Gamma} \qquad \frac{\Gamma \vdash e : t}{\Gamma \vdash print \ e \rightsquigarrow \Gamma} \\ \frac{}{\Gamma \vdash \bullet \rightsquigarrow \Gamma} \qquad \frac{\Gamma \vdash s \rightsquigarrow \Gamma' \quad \Gamma' \vdash slist \rightsquigarrow \Gamma_1}{\Gamma \vdash s \ slist \rightsquigarrow \Gamma_1} \\ \frac{\Gamma \vdash e : bool \quad \Gamma \vdash blk_1 \rightsquigarrow \Gamma' \quad \Gamma \vdash blk_2 \rightsquigarrow \Gamma''}{\Gamma \vdash if \ e \ then \ blk_1 \ else \ blk_2 \rightsquigarrow \Gamma} \qquad \frac{\Gamma \vdash e : bool \quad \Gamma \vdash blk \rightsquigarrow \Gamma'}{\Gamma \vdash while \ e \ blk \rightsquigarrow \Gamma'} \end{array}$$

A semântica operacional de Imp é apresentada a seguir.

$$\begin{array}{c}
 \overline{\langle \sigma, n \rangle \Downarrow n} \quad \overline{\langle \sigma, \text{false} \rangle \Downarrow \text{false}} \quad \overline{\langle \sigma, \text{true} \rangle \Downarrow \text{true}} \\
 \\
 \overline{\langle \sigma, x \rangle \Downarrow \sigma(x)} \\
 \\
 \frac{\langle \sigma, e \rangle \Downarrow \text{true}}{\langle \sigma, \text{not } e \rangle \Downarrow \text{false}} \quad \frac{\langle \sigma, e \rangle \Downarrow \text{false}}{\langle \sigma, \text{not } e \rangle \Downarrow \text{true}} \\
 \\
 \frac{\langle \sigma, e_1 \rangle \Downarrow n_1 \quad \langle \sigma, e_2 \rangle \Downarrow n_2}{\langle \sigma, e_1 + e_2 \rangle \Downarrow n_1 \oplus n_2} \quad \frac{\langle \sigma, e_1 \rangle \Downarrow n_1 \quad \langle \sigma, e_2 \rangle \Downarrow n_2}{\langle \sigma, e_1 - e_2 \rangle \Downarrow n_1 \ominus n_2} \\
 \\
 \frac{\langle \sigma, e_1 \rangle \Downarrow n_1 \quad \langle \sigma, e_2 \rangle \Downarrow n_2}{\langle \sigma, e_1 * e_2 \rangle \Downarrow n_1 \otimes n_2} \quad \frac{\langle \sigma, e_1 \rangle \Downarrow n_1 \quad \langle \sigma, e_2 \rangle \Downarrow n_2}{\langle \sigma, e_1 / e_2 \rangle \Downarrow n_1 \div n_2}
 \end{array}$$

Nas regras utilizamos operadores os  $\oplus, \ominus, \otimes$  e  $\div$  representam a adição, subtração, multiplicação e divisão sobre valores numéricos. A seguir, apresentamos as regras para demais operadores e a estrutura é similar.

$$\frac{\langle \sigma, e_1 \rangle \Downarrow n_1 \quad \langle \sigma, e_2 \rangle \Downarrow n_2}{\langle \sigma, e_1 < e_2 \rangle \Downarrow n_1 < n_2} \quad \frac{\langle \sigma, e_1 \rangle \Downarrow n_1 \quad \langle \sigma, e_2 \rangle \Downarrow n_2}{\langle \sigma, e_1 == e_2 \rangle \Downarrow n_1 = n_2} \quad \frac{\langle \sigma, e_1 \rangle \Downarrow b_1 \quad \langle \sigma, e_2 \rangle \Downarrow b_2}{\langle \sigma, e_1 \&e_2 \rangle \Downarrow b_1 \wedge b_2}$$

A semântica de comandos é representada por um conjunto de regras para deduzir  $\langle \sigma, s \rangle \Downarrow \sigma_1$  em que:  $\sigma$  denota o ambiente de execução de entrada,  $s$  o comando a ser executado e  $\sigma_1$  o ambiente produzido pela execução de  $s$ .

$$\begin{array}{c}
 \overline{\langle \sigma, \text{skip} \rangle \Downarrow \sigma} \quad \frac{\langle \sigma, e \rangle \Downarrow v}{\langle \sigma, x := e \rangle \Downarrow \sigma[x \mapsto v]} \quad \frac{\langle \sigma, e \rangle \Downarrow v}{\langle \sigma, t x := e \rangle \Downarrow \sigma[x \mapsto v]} \\
 \\
 \frac{}{\langle \sigma, \bullet \rangle \Downarrow \sigma} \quad \frac{\langle \sigma, s \rangle \Downarrow \sigma_1 \quad \langle \sigma_1, slist \rangle \Downarrow \sigma_2}{\langle \sigma, s \text{ slist} \rangle \Downarrow \sigma_2} \\
 \\
 \frac{\langle \sigma, e \rangle \Downarrow \text{true} \quad \langle \sigma, s_1 \rangle \Downarrow \sigma_1}{\langle \sigma, \text{if } e \text{ then } s_1 \text{ else } s_2 \rangle \Downarrow \sigma_1} \\
 \\
 \frac{\langle \sigma, e \rangle \Downarrow \text{false} \quad \langle \sigma, s_2 \rangle \Downarrow \sigma_2}{\langle \sigma, \text{if } e \text{ then } s_1 \text{ else } s_2 \rangle \Downarrow \sigma_2} \\
 \\
 \frac{\langle \sigma, e \rangle \Downarrow \text{false}}{\langle \sigma, \text{while } e \text{ blk} \rangle \Downarrow \sigma} \\
 \\
 \frac{\langle \sigma, e \rangle \Downarrow \text{true} \quad \langle \sigma, \text{blk} \rangle \Downarrow \sigma_1 \quad \langle \sigma_1, \text{while } e \text{ blk} \rangle \Downarrow \sigma_2}{\langle \sigma, \text{while } e \text{ blk} \rangle \Downarrow \sigma_2}
 \end{array}$$

O objetivo desta questão é adicionar um comando de repetição à linguagem Imp, similar ao comando **for** presente em linguagens como C e Java. Intuitivamente o comando **for** possui a seguinte estrutura:

**for** (inits; cond ; upds) block

em que:

Departamento de Computação  
*BCC328 - Construção de Compiladores I*  
Professor: Rodrigo Ribeiro  
Prova Teórica

DECOM - UFOP  
2º semestre de 2024  
e-mail: rodrigo.ribeiro@ufop.edu.br  
Data: 27/03/2025

- **inits** é um conjunto possivelmente vazio de comandos de inicializações;
  - **cond** é a condição de parada para a repetição;
  - **upds** é um conjunto possivelmente vazio de atualizações a serem executadas ao fim da execução do bloco de comandos **block**.
- (a) (1 ponto) Apresente a modificação da gramática de **Imp** para permitir a escrita de comandos **for**.
- (b) (2 pontos) O comando **for** deve incluir todas as variáveis definidas durante a inicialização no contexto de tipos quando da verificação de tipos da condição de parada (que deve ter tipo booleano), do bloco e das atualizações. Com base no descrito, apresente a regra de verificação semântica para o comando **for**.
- (c) (2 pontos) A execução do comando **for** procede da seguinte forma: primeiro executamos todas as inicializações, em seguida testamos a condição de parada e executamos o bloco de comandos enquanto esta é verdadeira. Ao final de cada iteração do bloco de comandos, executamos as atualizações. Apresente uma regra de semântica operacional para a execução de comandos **for**.