Syntax and Semantics

Written Exam, August 13, 2015, 13:00 - 16:00

Navn:

Stud.nr.:

Read this before you start!

- This exam contains 6 exercises. Each exercise is compulsory and has an equal contribution to the final grade.
- Each exercise should be solved in the space available on the page of the exercise and on the back page. Additional pages might be added if needed - for this contact the exam supervisors. Use page numbers to indicate the order in which the examiner should read your solutions.
- For preparing the final solutions you might use additional paper drafts.
- The solutions of the exercises must be written in English.
- During the exam no written materials are allowed. You are not allowed to use any electronic device.

Exercise 1. 1. Construct an NFA that recognizes the language $L = (01 + 001 + 010)^*$.

2. Is the language L context-free? Motivate your answer by using techniques or theorems learned this semester.

3. Convert the NFA of L to an equivalent DFA.

Exercise 2. Consider the following language over the alphabet $\Sigma = \{a, b, c\}$.

$$L = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}.$$

Is L regular? If yes, show a proof that it is so. If no, provide a proof that it is not so.

Exercises

Exercise 3. Consider the following language over the alphabet $\Sigma = \{a, b\}$.

$$L = \{w \in \{a,b\}^* \mid w \text{ has more } a\text{'s than } b\text{'s}\}$$

Is L context-free? If no, provide a proof that it is not so. If yes, show a proof that it is so.

Exercise 4. Consider the following statements in **Bims**

$$\texttt{while}\ \neg(\underline{2} < \underline{1} + \underline{1})\ \texttt{do}\ (\texttt{if}\ \texttt{x} < \texttt{x}\ \texttt{then}\ \texttt{x} := \underline{2}\ \texttt{else}\ \texttt{skip}) \qquad (S_0)$$

$$\texttt{x} := \underline{0};\ \texttt{while}\ (\texttt{x} \geq \underline{0})\ \texttt{do}\ \texttt{x} := \texttt{x} + \underline{1} \qquad (S_1)$$

$$\texttt{x} := \underline{3};\ \texttt{y} := \underline{4};\ \texttt{while}\ (\texttt{x} \leq \texttt{y})\ \texttt{do}\ (\texttt{x} := \underline{2} * \texttt{x};\ \texttt{y} := \underline{3} * \texttt{y}) \qquad (S_2)$$

1. Prove that there exists $k \geq 0$ such that in the small-step semantics of **Bims**, for any state $s \in \mathbb{S}tates$, $\langle \texttt{S}_0,\, s\rangle \Rightarrow^k \langle \texttt{S}_0,\, s\rangle$;

2. Are S_1 and S_2 semantically equivalent in the big-step semantics of **Bims**? A valid answer must be proven.

3. Are S_1 and S_2 semantically equivalent in the small-step semantics of **Bims**? A valid answer must be proven.

Exercise 5. Consider the following fragment of the language **Bump**, where a denotes an arithmetic expression

$$S ::= \; x := a \mid skip \mid S_1; S_2 \mid \; \text{begin } D_v D_p S \text{ end } \mid \text{call} p(a)$$

$$D_v ::= \; var \; x := a; D_v \mid \varepsilon$$

$$D_p ::= \; proc \; p(x) \; is \; S; D_p \mid \varepsilon.$$

Complete the transition rule below for non-recursive procedure calls using call-by-value mechanism for fully static scope rules, where st, st' represent the initial and the final states respectively, and e_v, e_p the variable and the procedure environments respectively.

$$\frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{e_v, e_p \vdash \langle \text{call} p(a), st \rangle \rightarrow st'}$$

Exercise 6. Consider the language **Bims** in which the Boolean expressions are given by the following grammar:

$$b ::= a_1 \neq a_2 \mid a_1[pd]a_2 \mid b_1 \triangledown b_2 \mid \perp,$$

where a denotes arithmetic expressions, $\neq$ denotes the relation *"not equal"*, $a_1[pd]a_2$ expresses the fact that "*a_1 is a prime number and a divisor of a_2*", $b_1 \triangledown b_2$ denotes the fact that "*if b_2 is true then also b_1 must be true*" and $\perp$ denotes *"false"*.

1. Give a big-step semantics for these Boolean expressions, assuming that you already have the big-step semantics for arithmetic expressions.
2. Give a small-step semantics for these Boolean expressions, assuming that you already have the small-step semantics for arithmetic expressions.