

Syntax and Semantics

Written Exam, June 01, 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. However, do hand in only the pages with the final solutions of the exercises.
- 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.

Exercises

Exercise 1. Consider the following languages over the alphabet $\Sigma = \{a, b\}$.

$$L_1 = \left\{ w \left| \begin{array}{l} w \text{ has an even number of } a\text{'s and an odd number of } b\text{'s,} \\ \text{and } w \text{ does not contain the substring } ab \end{array} \right. \right\},$$

$$L_2 = \{w \mid w \text{ contains exactly one } b\}.$$

1. Prove that L_1 is regular by constructing an automaton that will recognize this language and guarantees that it is regular.
2. Prove that $L_1 \cap L_2$ is regular by constructing an automaton that will recognize this language and guarantees that it is regular.
3. Identify a regular expression that characterizes $L_1 \cap L_2$ by using the algorithm of conversion of an automaton into a regular expression presented at the course. (*If the automaton constructed at point 2. does not correspond to the type of automaton required by the algorithm of conversion, firstly convert your automaton into the required type.*)

Ad hoc solutions will not be considered.

Exercise 2. Consider the following context-free grammar with start variable S over the final alphabet $\Sigma = \{[,], a, :, *\}$:

$$\begin{aligned} S &\rightarrow T \mid S : T \\ T &\rightarrow T * F * T \mid \varepsilon \\ F &\rightarrow a \mid [S] \end{aligned}$$

1. Which of the following strings are described by this grammar and which are not? For those that are described, provide a syntax tree.
 - (a) $**a**a* :$
 - (b) $[*a*]$
 - (c) $:*a* :$
2. Convert the grammar into an equivalent one in Chomsky normal form by using the algorithm presented at the course. Ad hoc solutions will not be considered.
3. Use the construction presented in the course to construct a push-down automaton that recognizes the language generated by this grammar. Ad hoc solutions will not be considered.

Exercise 3. Prove that the language

$$L = \{a^n \mid n \text{ is a prime natural number}\}$$

over the alphabet $\{a\}$ is not a context-free language, where a^n , as usual, denotes the word formed by n consecutive occurrences of a .

Exercise 4. Consider the language **Bims** in which the Boolean expressions are given by the following grammar:

$$b ::= a_1 \neq a_2 \mid a_1[div]a_2 \mid b_1 \leftrightarrow b_2 \mid b_1[eor]b_2 \mid \perp,$$

where a denotes arithmetic expressions, \neq denotes the relation "*not equal*", $a_1[div]a_2$ expresses the fact that " *a_1 is a divisor of a_2* ", \leftrightarrow denotes the classical logical equivalence, $b_1[eor]b_2$ expresses the fact that "*exactly one of b_1 and b_2 is 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. Describe the corresponding transition system using the formal definition provided at the course.

Exercise 5. Consider the language **Bump** with call-by-name parameter passing. Hüttel's book presents transition rule for procedure calls assuming the case of mixed scope rules with dynamic scope rules for variables and static scope rules for procedures. Provide the similar transition rule for procedure calls assuming static scope rules for variables and dynamic scope rules for procedures.

Exercise 6. Consider the following statement in **Bip**.

```
begin
  var x:=2;
  var y:=6;
  proc p is x:=x+1;
  proc q is call p;
  begin
    var x:=8;
    proc p is x:=x+1;
    call q;
    y:=x
  end
end.
```

1. What is the value of y after the statement is executed assuming fully dynamic scope rules for both variables and procedures? Motivate your answer.
2. What is the value of y after the statement is executed assuming dynamic scope rules for procedures and static scope rules for variables? Motivate your answer.
3. What is the value of y after the statement is executed assuming fully static scope rules for both procedures and variables? Motivate your answer.