Syntax and Semantics

Digital Written Exam, June the 12th 2020, 10:00-13:00

Please read the following before solving the exercises.

- This exam contains 5 exercises. Each exercise is compulsory and has an equal contribution to the final grade. The solution has to be composed in English. If you believe that the assignment wording is ambiguous or erroneous, then write down what additional assumption you are using and outline your reasons.
- Each of the five exam exercises has to be solved on a separate A4 piece of paper, i.e., two A4 pages per exercise at most. On each A4 page, state your name, your student number and the exercise(s) which is (are) being solved.
- Solutions have to be uploaded to Digital Exam in form of a single pdf file consisting of at most ten digital pictures of at most ten handwritten A4 pages. Other submission formats will not be considered.
- The submitted digital pictures should be readable, i.e., of sufficient quality (high resolution, enough light, not blurry, etc.).
- Allowed aids are your own notes (made entirely by yourself or as an active participant of a group), lecture slides and exercise sheets, the books of Hans Hüttel and Michael Siepser used during the course. Anything else is rendered illegal, including, in particular, Googling or asking other persons for help.
- In case of emergencies: Students can contact the instructor during the exam by approaching the study secretary, as outlined in the guidelines for online exams. Keep an eye on your student mail for potential announcements during the exam.

Terminology applied in the exam:

- *Provide*: Give something without arguing why it is correct.
- *Prove*: Give a formal proof for the correctness of something.
- *Motivate*: Give an informal argument for the correctness or choice of something.

Last but not least, good luck!

Exercise 1.
Fix the alphabet $\Sigma = \{a, b, c, d\}$ and consider the language

$$L = \{w \in \Sigma^* \mid |w|_a + |w|_b \leq 1\},$$

where $|w|_\sigma = |\{1 \leq i \leq |w| \mid w_i = \sigma\}|$, meaning that $|w|_\sigma$ denotes how many times $\sigma \in \Sigma$ appears in w. Intuitively, any $w \in L$ has at most one a or b, e.g., $acdc \in L$ because $|acdc|_a + |acdc|_b = 1 + 0 = 1$, while $abba \notin L$ because $|abba|_a + |abba|_b = 4$.

1) Provide a DFA with three states that recognizes L.

2) Provide a regular expression that recognizes L.

3) Convert the DFA into a context-free grammar using the algorithm from the course. Ad-hoc solutions will be not considered.

Exercise 2.
Consider the language

$$L' = \{w\tilde{w} \in \{a,b,c,d,e\}^* \mid w \in \{a,b,c,d\}^* \text{ and } \tilde{w} = e^{|w|_a + |w|_b}\},$$

where e^0 is defined as the empty string ε. Intuitively, $w\tilde{w} \in L'$ whenever the $\tilde{w}$ is a sequence of e's whose length coincides with the number of a and b symbols in w. For instance, $acdc, abbaee \notin L'$ but $acdce, abbaeeee \in L'$.

1) Provide a context-free grammar that generates L' and a derivation of $acdce$.

2) Provide a pushdown automaton that recognizes L' and an accepting branch of its computation on input $abcee$.

Exercise 3.
Prove that L' from Exercise 2 is not regular by using the pumping lemma for regular languages.

Exercise 4.

Consider the following grammar for arithmetic expressions $\mathbb{A}$:

$$n ::= 0 \mid s(n) \qquad\qquad a ::= n \mid a + a \mid a * a$$

Intuitively, $s(n)$ is the successor of n, e.g., $s(0)$ and $s(s(0))$ encodes 1 and 2, respectively.

1) Let the set of states be $\mathbb{A}$, the set of final states be $\{0, s(0), s(s(0)), s(s(s(0))), \ldots\}$ and consider the following *small step* transition rules for addition:

$$[r_1]\ s(n_1) + n_2 \Rightarrow n_1 + s(n_2) \qquad\qquad [r_2]\ 0 + a \Rightarrow a$$

$$[r_3]\ a_1 + a_2 \Rightarrow a_2 + a_1 \qquad\qquad [r_4]\ \frac{a_1 \Rightarrow a_1'}{a_1 + a_2 \Rightarrow a_1' + a_2}$$

Prove $s(s(s(0))) + s(s(0)) \Rightarrow^* s(s(s(s(s(0)))))$ using $[r_1], \ldots, [r_4]$. All intermediate computations have to be proven. (Intuitively, this corresponds to deducing 3+2 = 5.)

2) Complete the *small step* semantics from 1) by providing the rules for multiplication.

Note: The complete semantics of $\mathbb{A}$ is intended to be the same as for arithmetic expressions in the language **Bims** from Hans Hüttel's book but with the modified representation of numbers as explained above.

3) Using the transition rules from 1) and 2), prove $s(s(0)) * s(s(0)) \Rightarrow^* s(s(s(s(0))))$. All intermediate computations have to be proven.

Exercise 5.
Consider the following statement in **Bip**.

```
begin
   var x:=6;
   var y:=7;
   proc p is x:=x+2;
   proc q is call p;
   begin
       var x:=3;
       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 static scope rules for both procedures and variables? Motivate your answer.

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

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