Seminar Questions

Language Syntax

1. Let $G = (V, \Sigma, R, S)$ be defined by $V = \{0, 1, a\}$, $\Sigma = \{0, 1\}$, $S = a$, $R = \{a \rightarrow 1a0, a \rightarrow 0a1, a \rightarrow 01, a \rightarrow 10\}$. Describe $L(G)$.
2. Let $G$ be defined by $V = \{a, b, c, d, e\}$, $\Sigma = \{a, b\}$, $S = c$, $R = \{c \rightarrow de, d \rightarrow dab, d \rightarrow e, e \rightarrow acea, e \rightarrow e\}$. Give the parse trees for $\varepsilon$, $ab$, $abaa$.
3. Prove that the BNF grammar:
   $$<expr> ::= <E> | <expr> + <expr> | <expr> * <expr>$$
   is ambiguous. Give a grammar $G1$ which gives $\ast$ higher precedence than $\ast$, and a grammar $G2$ which gives $\ast$ higher precedence than $\ast$.
4. Prove that the BNF grammar:
   $$<S> ::= \text{if} <E> \text{then} <S> | \text{if} <E> \text{then} <S> \text{else} <S>$$
   is ambiguous. Suggest ways in which this problem may be solved.
5. Give the complete trace of a top down parser for the string $1.4$ and the grammar:
   $$<num> ::= <int> | <real>$$
   $$<int> ::= <digit> | <digit><int>$$
   $$<real> ::= <int-part>. <fraction>$$
   $$<int-part> ::= <digit> | <int-part> <digit>$$
   $$<fraction> ::= <digit> | <digit> <fraction>$$
   $$<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$
6. Consider the following BNF grammar with $<S>$ as the start symbol:
   $$<S> ::= a <S> c <B> | <A>$$
   $$<A> ::= c <A> | c$$
   $$<B> ::= d | <A>$$
   a) which of the following strings are in the language generated by this grammar? Draw the parse tree for those strings which are part of the language.
   i) abcd ii) acccbd iii) acccbcc iv) acd v) accv vi) aaaccddccccd
   b) Is the grammar ambiguous? If it is, what is a string that would result in more than one parse tree.
7. For each of the following languages, use a CFG (if possible) to describe a grammar that would accept those and only those sentences in the language.
   a) $L(G) = \{w1w2 | w1$ is any non-empty binary string and $w2$ is the reverse of $w1\}$
   b) $L(G) = \{\text{set of binary palindromes}\}$
8. A grammar described in EBNF can be translated into an equivalent grammar in BNF.
   a) Translate the EBNF rule $<S> ::= [+ | -] <B> \{<B>\}^*$
   b) Translate the EBNF rule $<S> ::= <A> \{<A> | <B>\}^*$
9. Consider the following BNF:
   $$<N> ::= 11 | 1001 | <N>0$$
   a) prove that any string generated by the above grammar is a binary number which is divisible by 3.
   b) is it true that any binary number divisible by 3 is also generated by the grammar?
10. Give a regular grammar to generate identifiers in Pascal, i.e. a letter followed by any number of letters or digits.
11. Consider the programming language IMP presented in class. Which of the following grammars for IMP are ambiguous and why?
   a) Grammar for arithmetic expressions
   b) Grammar for Boolean expressions
   c) Grammar for commands

Semantics and Program Verification

1. indicate which of the following terms imperative, object-oriented, functional, logic, concurrent, are associated to the following languages C, C++, Lisp, ML, Java, Prolog, Pascal.
2. Consider the operational semantics rule for evaluating the meaning of logical conjunction in IMP. A straightforward application of this rule will lead to inefficient evaluation because even if one of the operands is false, it will evaluate the other operand. Propose new operational semantics rule(s) for logical conjunction which remove this problem.

3. Describe the operational semantics of the following program fragment. Show the applications of all the rules. Assume that in the initial state \(X0=1\), \(X1=2\) and \(n=3\).

\[
\begin{align*}
X0 &:= X1; \\
X1 &:= X0 \quad n; \\
n &:= n - 1;
\end{align*}
\]

4. Same as question 3. Assume \(n=0\) initially.

\[
\begin{align*}
n &:= 2; \\
\text{while } n>0 \text{ do} \\
n &:= n - 1.
\end{align*}
\]

5. Which is the weakest formula in the following set

\[
\{x=7, x>5, x>10, x>6 \land x>2\}
\]

6. Evaluate the weakest precondition \(P\) such that

\[
\{P\} x:=3*y+1 \quad \{x<7\}
\]

7. Evaluate the weakest precondition \(P\) such that

\[
\{P\} x:=x+5; \quad y:=y+1 \quad \{x=y^3+2\}
\]

8. Prove the assertion

\[
\{ \text{true} \}
\]

if \(a <= b\) \(m := a; \)
else \(m := b; \)
\{ m = \min(a,b) \}

9. Find an invariant for the program fragment assuming initially \(x=y=0\) and \(a=3\)

\[
\begin{align*}
\text{while true do} \\
x &:= x+a; \\
y &:= y+1;
\end{align*}
\]

10. Find an invariant for the program fragment assuming initially \(n=5\)

\[
\begin{align*}
\text{while } n>0 \text{ do} \\
n &:= n-1
\end{align*}
\]

If the loop terminates, what does the while-rule allow us to conclude?

11. Explain the complete proof for the MULT(A,B) program given in the lecture notes.

12. Consider the following program where initially \(n \geq 0\). Find an invariant for the loop in the program.

\[
\begin{align*}
j &:= 0; \\
x &:= 1; \\
\text{while } j < n \text{ do} \\
j &:= j + 1; \quad x := 2 * x
\end{align*}
\]

13. Prove

\[
\{n \geq 0\}
\]

\[
\begin{align*}
j &:= 0; \\
x &:= 1; \\
\text{while } j < n \text{ do} \\
j &:= j + 1; \quad x := 2 * x
\end{align*}
\]

\[
\{x:=2^n\}
\]

14. Prove

\[
\{\text{true}\}
\]

\[
\begin{align*}
y &:= 1; \\
z &:= 0; \\
\text{while } \{z \downarrow \downarrow x\} \text{ do} \\
z &:= z + 1; \\
y &:= y * z
\end{align*}
\]

\[
\{y=x!\}\]
Logic Programming

15. Given the Prolog program below, show the steps the Prolog interpreter follows to evaluate the query `path(algol60,L)` (show the SLD-tree of the evaluation).

   link(algol60,cpl).
   link(algol60,simula67).
   link(cpl,bcpl).
   link(simula67,cplusplus).
   link(bcpl,c).
   link(simula67,smalltalk80).

   \[ \text{path}(X,X). \]
   \[ \text{path}(X,Y) : \text{link}(X,Z), \text{path}(Z,Y). \]

16. Define a Prolog predicate for `sister(X,Y)` meaning that X is the sister of Y (assume some definitions for `parent/2`, `male/1`, `female/1`).

17. Define a Prolog predicate for `concat(X,Y,Z)` meaning that Z is the concatenation of X and Y, e.g. `concat([1,2,3],[4,5],Z)` returns `Z=[1,2,3,4,5]` (show the SLD-tree of the evaluation).

18. Define a Prolog predicate `sublist(X,Y)` meaning that X is a sublist of Y, e.g.

   \[ \text{sublist}([c,d,e],[a,b,c,d,e,f]) \text{ evaluates to true}. \]

19. Define a predicate `del(X,L,L1)` for deleting an item X from a list L and returning the result in L1.

20. Show the SLD-tree for the query `a(X)` and the program:

   \[ \text{a}(X) : \text{b}(X). \]
   \[ \text{a}(X) : \text{f}(X). \]

21. Same as question 20 but with the program:

   \[ \text{a}(X) : \text{b}(X). \]
   \[ \text{a}(X) : \text{f}(X). \]

22. Consider the following program. What is the answer that Prolog gives to the query `q(X,Y)`.

   \[ \text{p}(a,b). \]
   \[ \text{p}(a,b). \]
   \[ \text{q}(X,Y) : \text{not}(X=Y), \text{p}(M,X), \text{p}(M,Y). \]

23. Given the definition in Prolog for the following predicates:

   a) `quitaultimo(L1,L2)` that evaluates to true if L2 is L1 without the last element.
   b) `añadeultimo(X,L)` that adds an item X at the end of L.
   c) `set(L)` that evaluates to true if L is a set, (i.e. a list without repeated elements) and fails if it is not.
   d) `union(S1,S2,S3)` that evaluates to true if S3 is the union of sets S1 and S2

24. Consider the following program and construct (or sketch) the SLD-tree for the query `p(5)`

   \[ \text{p}(X) : \text{q}(X), \text{p}(Y). \]
   \[ \text{q}(X). \]