Practice test for Matematisk Lingvistik (VT98,980518)

Try to make this practice test without consulting the literature because you are not allowed to use the literature during the final test. You will find the answers for this practice test at the end of this file.

This practice test contains 7 questions that are worth 100 points in total. You can find the value of each question attached to its index character.

1. Consider the sets A={ 1 , 2 , 3 , 4 } and B={ 2 , { 3 } }. We regard the union of these sets as the universe

(a-10) Compute A∩B, A∪B, $\mathcal{P}$(B) and A'. Is B⊆A true?

(b-5) Prove that for any sets C and D $(C \cap D') \cup (D \cap C) = C$. You may use only the following set theoretic laws for this purpose:
2a. $X \cup Y = Y \cup X$
2b. $X \cap Y = Y \cap X$
4a. $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$
4b. $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$
5b. $(X \cup U) = U$
5d. $(X \cap U) = X$
6a. $(X \cup X') = U$
6c. $(X \cap X') = \emptyset$

2. We have A={ 1 , 2 , 3 , 4 }, $R_1 = \{ <x, y> \mid x \epsilon A$ and $y \epsilon A$ and $x = y \}$ and $R_2 = \{ <x, y> \mid x \epsilon A$ and $y \epsilon A$ and $x<y \}$

(a-10) Check for both R_1 and R_2 if they are functions from A to A, onto A or one-to-one.

(b-5) Compute the relation $R_2^{-1} \circ R_1$

3. We have A={ 1 , 2 , 3 , 4 }, R_3 = { <1,2> , <1,3> , <1,4> , <2,3> , <4,2> , <4,3> } and R_4 = { <1,1> , <1,2> , <2,2> , <3,3> , <4,4> }

(a-10) Determine for both R_3 and R_4 if they are reflexive, transitive, symmetric, anti-symmetric or connected.

(b-5) R_3 is an order. Is it a weak or strong order? Give the minimal, maximal, least and greatest elements of R_3.

4. Consider the type 3 grammar G=<V_T,V_N,S,R> with V_T={a,b}, V_N={S}, S=S and R={ <S,aS> , <S,bS> , <S,a> , <S,b> }

(a-9) Draw the tree for the derivation of the string aab by G and give the dominance relation and the precedence relation for this tree.

(b-6) Give an example of a type 0 language that is not context-sensitive. Furthermore give an example of a type 1 language that is not context-free.

5. We define the finite automaton FA as FA=<K,Σ,q_0,F,δ> such that K={q_0,q_1}, Σ={a,b}, q_0=q_0, F={q_0} and δ={ (q_0,a,q_1), (q_0,b,q_0), (q_1,a,q_0), (q_1,b,q_1) }

(a-5) Make a type 3 grammar which produces the language that is accepted by finite automaton FA.

(b-5) Make a regular expression for the language that is produced by finite automaton FA.

(c-5) What happens when you apply the pumping theorem for regular languages to the language consisting of strings which contain as many a's as b's in any order? What conclusion do you draw from this?

6. We define the non-deterministic push-down automaton PDA as PDA=$<K,\Sigma,\Gamma,q_0,F,\Delta>$ such that $K=\{q_0,q_1,q_2\}$, $\Sigma=\{a,b,c\}$, $\Gamma=\{C\}$, $q_0=q_0$, $F=\{q_2\}$ and $\Delta=\{ (q_0,a,e,q_0,C)$, (q_0,e,e,q_1,e), (q_1,b,e,q_1,C), (q_1,e,e,q_2,e), $(q_2,c,C,q_2,e) \}$

(a-5) Give a description of the language that is accepted by the push-down automaton PDA.

(b-5) Give a context-free grammar that produces the language that is accepted by the push-down automaton PDA.

(c-5) Is it possible to construct a regular grammar that produces the language that is accepted by the push-down automaton PDA? Motivate your answer.

7.(a-5) Construct a Turing machine which accepts the language a*b* and replaces all a's on the tape by b's.

(b-5) Construct a type 0 grammar that generates the language $\{ xx \mid x \in \{a,b\}^* \}$.

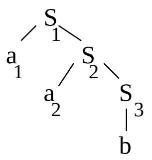
Answers practice test for Matematisk Lingvistik (VT98,980518)

1. (a) A∩B = {2}
   A∪B, = {1,2,3,{3},4}
   $\mathcal{P}$(B) = {∅,{2},{{3}},{2,{3}}}
   A' = {{3}}
   B⊆A is not true because {3} is not an element of A
   (b) $(C \cap D') \cup (D \cap C) = C$ (apply 2b)
   $(C \cap D') \cup (C \cap D) = C$ (apply 4b)
   $C \cap (D' \cup D) = C$ (apply 2a)
   $C \cap (D \cup D') = C$ (apply 6a)
   $C \cap U = C$ (apply 5d)
   $C = C$ (done)

2. (a) R_1 is a function that is onto and one-to-one. R_2 is not a function. It is neither onto nor one-to-one.
   (b) $R_2^{-1} = \{ <x,y> \mid x \epsilon A$ and $y \epsilon A$ and $x > y \}$
   R_1 is the identity relation on A so $R_2^{-1} \circ R_1 = R_2^{-1}$

3. (a) R_3 is transitive, anti-symmetric and connected but it is not reflexive and it is not symmetric. R_4 is reflexive, anti-symmetric and transitive but it is not symmetric and it is not connected.
   (b) R_3 is a strong order because it is irreflexive, anti-symmetric and transitive.
   Maximal element=greatest element=3 and Minimal element=least element=1.

4. (a) In order to distinguish the different S nodes and a nodes I have given them index numbers.



   Dominance relation = $\{ <S_1,S_1> , <S_2,S_2> , <S_3,S_3> , <a_1,a_1> , <a_2,a_2> , <b,b> , <S_1,a_1> , <S_1,S_2> , <S_1,a_2> , <S_1,S_3> , <S_1,b> , <S_2,a_2> , <S_2,S_3> , <S_2,b> , <S_3,b> \}$
   Precedence relation = $\{ <a_1,S_2> , <a_1,a_2> , <a_1,S_3> , <a_1,b> , <a_2,S_3> , <a_2,b> \}$
   (b) A grammar with the rules S→abS, abS→ba is type 0 but not context-sensitive.
   A grammar with the rules S→aBc, aBc→abc is type 1 but not context-free.

5. The finite automaton accepts strings that contain an even number of a's.
   (a) G=$<V_T,V_N,S,R>$ with V_T={a,b}, V_N={S,A}, S=S and R contains the rules:
   S→e (produces the empty string)
   S→bS (we can append as many b's as we want to a correct string)
   S→aA (if we produce an a we need to change the non-terminal)
   A→bA (we can append as many b's as we want to an incorrect string)
   A→aS (if we produce an a the string is correct again and we can change to non-terminal S)

(b) Regular expression: b*(ab*ab*)*
any number of b's followed by any number of a-pairs with possible b's in between.

(c) The language can be handled by the pumping theorem for fal's: x=e, y=ab, z=e will give you strings of the format xy^nz that are in the language for any value of n. We cannot conclude anything from this result. (It would be wrong to conclude here that this language is fal. In fact it is not.)

6. (a) The pda accepts strings of the format $a^kb^lc^m$ in which m=k+l.

(b) $G=<V_T,V_N,S,R>$ with V_T={a,b}, V_N={S,B}, S=S and R contains the rules:
$S\rightarrow e$ (produces the empty string)
$S\rightarrow aSc$ (produces a's and corresponding c's)
$S\rightarrow bBc$ (produces b's and corresponding c's)
$B\rightarrow bBc$ (copy of previous rule. Non-terminal change was necessary to prevent the a's and b's from being mixed)
$B\rightarrow e$ (necessary to remove the B non-terminal)

(c) It is not possible to make a regular grammar for this language because this is not a regular language but a context-free language. It is possible to prove that the language is not regular by using the pumping theorem for fal's (this proof is not necessary here).

7. (a) $TM=<K,\Sigma,s,R>$ such that $K=\{q_0,q_1,q_2\}$, $\Sigma=\{a,b,\#\}$ s=q0 and R=={ (q_0,a,q_1,b), (q_0,b,q_2,R), (q_1,b,q_0,R), (q_2,a,q_2,a), (q_2,b,q_2,R) }.
(The Turing machine replaces each *a* by a *b* and moves from initial state q_0 to q_1 (rule 1). This state is used for moving to the right and going back to state q_0 (rule 3). As soon as the automaton discovers a *b* in the string it will move to q_2 (rule 2) and thereafter it will only accepts *b*'s (rule 4 and 5))

(b) G = < VT,VN,S,R > where VT = {a,b} , VN = {A,B,S,T} , S=S and R = { <S,aAS> , <S,bBS> , <S,T> , <Aa,aA> , <Ba,aB> , <Ab,bA> , <Bb,bB> , <AT,Ta> , <BT,Tb> , <T,e> }.
(The grammar generates a string aAbBaAS with occurrences of aA for every a in x and bB for every B in x (rules 1-3). Then it replaces the S with a T (rule 3) and moves the A's and B's behind the T where they become a's and b's (rules 4-9). Finally the T is removed)