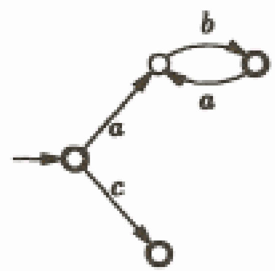
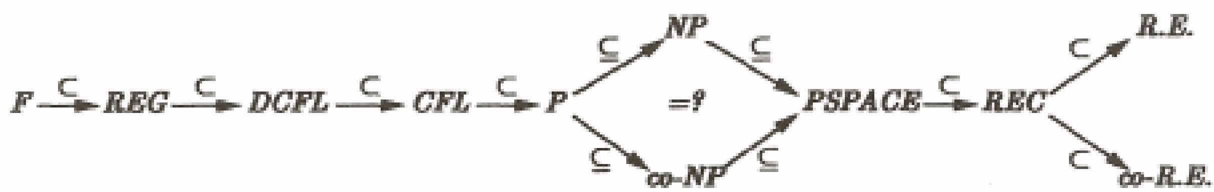
Solutions comprehensive exam Automata and Formal Languages

October 31, 2001

1. *Draw a nondeterministic finite automaton without ε-moves accepting the language* $(\mathbf{ab})^* + \mathbf{c}$.



2.



3. *Consider the following problem: Given descriptions of two Turing machines M and N over a given alphabet, is there a string accepted by both M and N?*
   - *Is this problem decidable?*

   No. By Rice's theorem it is undecidable whether a given Turing machine M accepts a non-empty language. That problem can be reduced to the problem above, by taking N to be a fixed Turing machine that accepts every string over the given alphabet. It follows that also the problem above is undecidable.
   - *Is it recursive enumerable?*

   Yes. Given descriptions of M and N, enumerate all pairs (w, n) with w a word over the given alphabet and n a natural number. For each such pair (w, n) run both M and N on w for n steps. In case both accept, return "Yes".

4. *A clique of size k in an undirected graph G is a subgraph with k nodes, wherein every two nodes are connected by an edge. In particular, there must be an edge between each node in the clique and itself (a self-loop). Let a soft clique be obtained by skipping the requirement that there are self-loops. Thus, a soft clique of size k is a subgraph with k nodes, wherein every two different nodes are connected by an edge. Now every clique is surely a soft clique.*

   CLIQUE is the problem that asks, given an undirected graph G and a number k, whether G has a clique of size k. Likewise, let SOFT CLIQUE be the problem that asks, given an undirected graph G and a number k, whether G has a soft clique of size k.

   Given the fact that the problem CLIQUE is NP-complete, prove that SOFT CLIQUE is NP-complete as well. Be concise.

   The NP-hardness of SOFT CLIQUE can be established by means of a polynomial time reduction from CLIQUE to SOFT CLIQUE. Given an instance of CLIQUE, i.e. an undirected graph G and a number k, create the subgraph G^* of G consisting of all nodes in G with self-loops and all edges of G between such nodes. As any clique in G must lay entirely within G^*, and in G^* any soft clique is a clique (and vice versa), G has a clique of size k if and only of G^* has a soft clique of size k. It remains to be shown that the reduction (i.e. the creation of G^*) can be done in polynomial time. The following algorithm establishes that: pass over the description of G (a list of nodes and edges) once, listing all nodes with self-loops. Then pass over the description again, this time listing all edges between such nodes. This is clearly a polynomial time algorithm.

   Besides showing NP-hardness, it should also be pointed out that SOFT CLIQUE is in the class NP. One way to do that is by means of a polynomial time reduction from SOFT CLIQUE to CLIQUE. Given an instance of SOFT CLIQUE, i.e. an undirected graph G and a number k, create the graph G° out of G by adding a self-loop for every node. This is clearly a polynomial time reduction. G has a soft clique of size k if and only if G° has a clique of size k.

5. *Prove that the language $L = \{xyy \mid x, y \in \{0,1,2\}^*,\ |y| > 0\}$ is not context-free.*
   Context-free languages are closed under intersection with a regular language. Thus, if L is context-free then also $K := L \cap 2(0+1)^*22(0+1)^*2 = \{2y22y2 \mid y \in \{0,1\}^*\}$ is context-free. However, the latter is not context-free, as can be established with the pumping lemma.

   Suppose K is context-free. Let p be the pumping length for this language.
   Then $z := 20^p1^p220^p1^p2 \in L$. So it must be possible to write z as $uvwxy$, such that

   (a) for each $i > 0$, $uv^iwx^iy \in L$,
   (b) $|vx| > 0$, and
   (c) $|vwx| \leq p$.

   In case v or x contain a 2, pumping down (by taking $i = 0$) yields a string that is not in K. In case v and x both lay in the left half of z, pumping down disturbes the matching between the left and right sides of the string. The same holds if v and x both lay in the right half of z. In case v lays to the left, and x to the right of the middle 2's, by condition (c) above, v contains only 1's and x only 0's. Again, pumping down disturbes the matching between the left and right sides of the string. It follows that K, and hence also L, is not context-free.

   Directly using the pumping lemma to show that L is not context-free is rather hard.

12