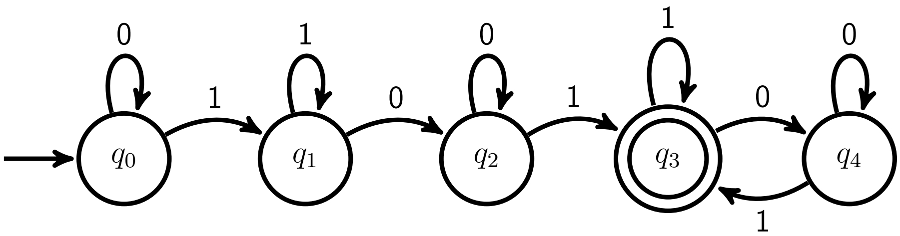
Çankaya University
Department of Computer Engineering
2013 - 2014 Fall Semester

CENG 491 - Formal Languages and Automata

First Midterm Examination

1) Design a DFA for all strings over the alphabet $\Sigma = \{a, b, c\}$ in which there is no aa, no bb and no cc.

2) What language does the following DFA recognize? Describe verbally.

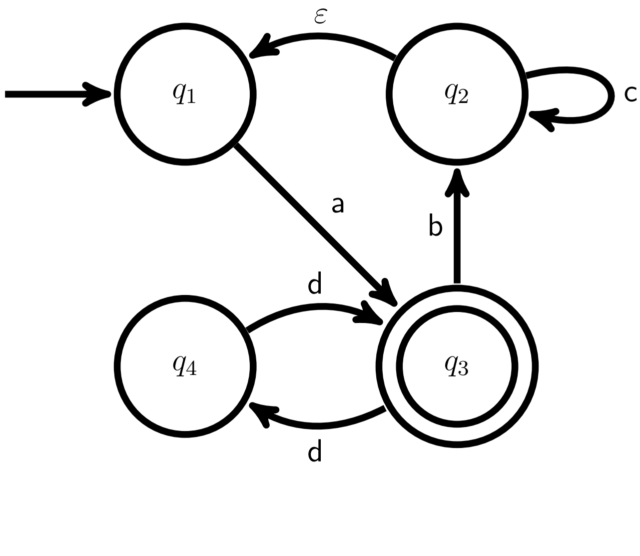


3) Let A be a language over $\Sigma = \{0, 1\}$. $A = \{w \mid w \text{ ends with } 11 \text{ or } 000\}$.

a) Find an NFA that recognizes A.

b) Find an NFA that recognizes A^*.

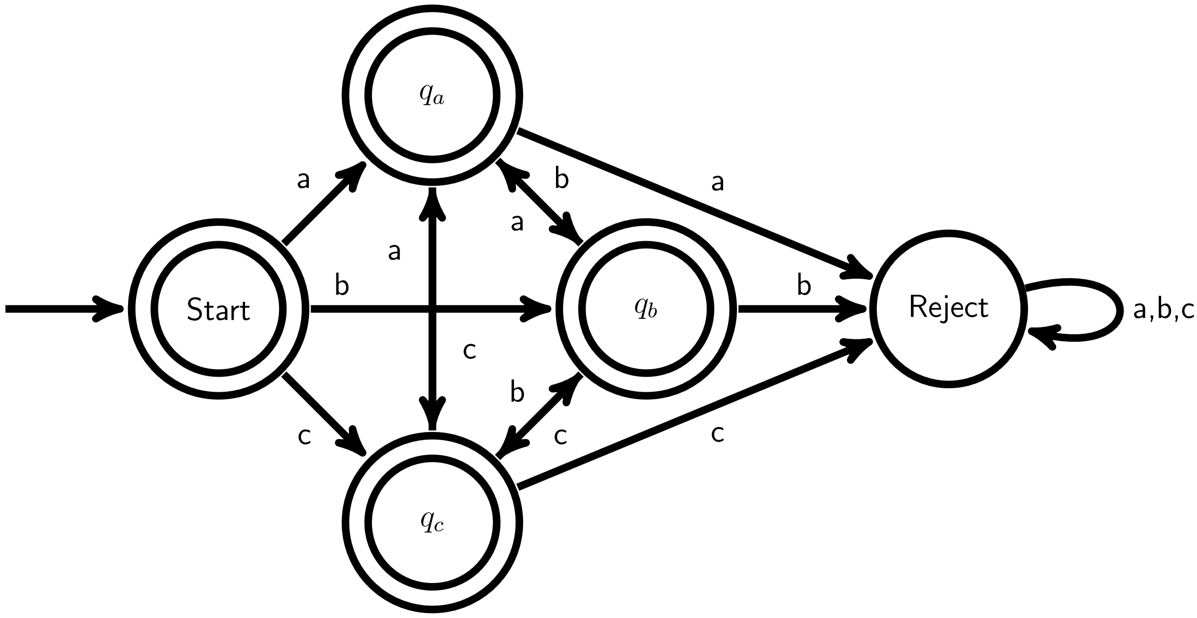
4) Find the regular expression equivalent to the following NFA:



5) Show that the language $A = \{w \mid w = w^R\}$ over $\Sigma = \{0, 1\}$ is non-regular. (Here, w^R shows the reverse of the string w, for example $10111^R = 11101$)

Answers

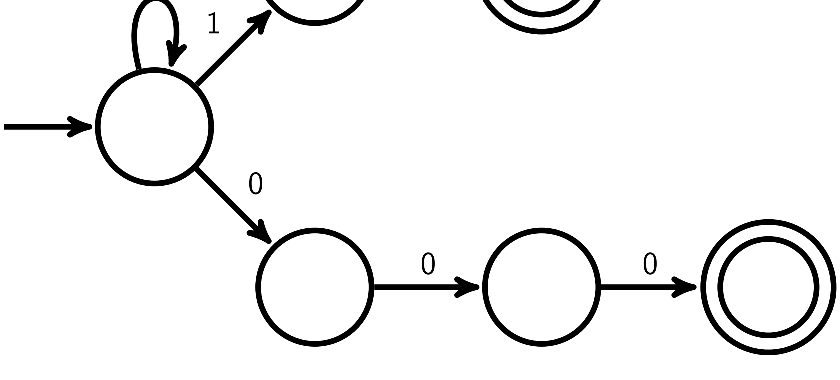
1)

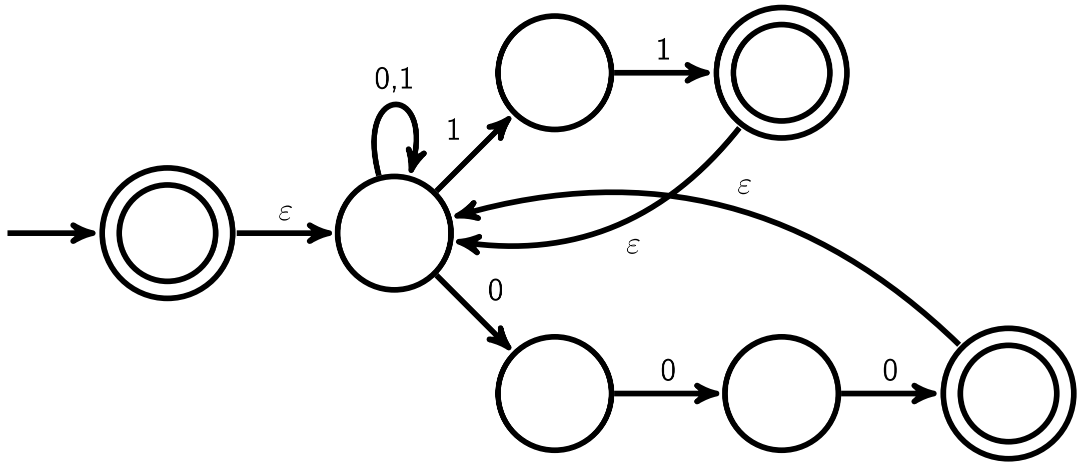


2) Many answers are possible. Some of them, starting with the simplest are:

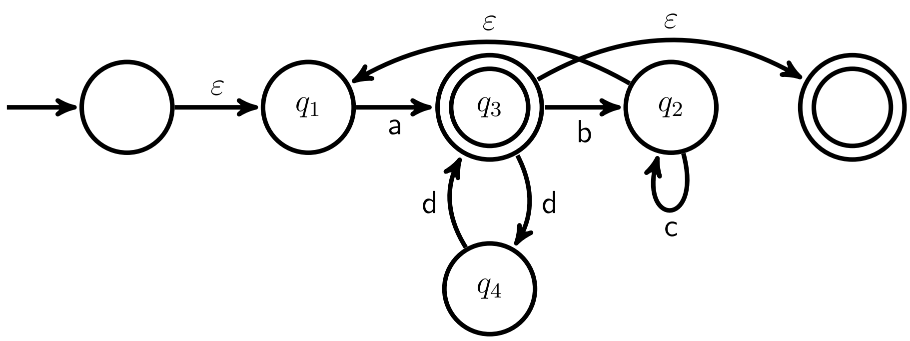
- Contains 10 and ends with 1.
- Contains 10 and 01 and ends with 1.
- Contains at least two symbol changes and ends with 1.
- Contains at least 3 symbols, 1, 0 and 1 in this order (but not necessarily consecutively) and ends with 1.

3)

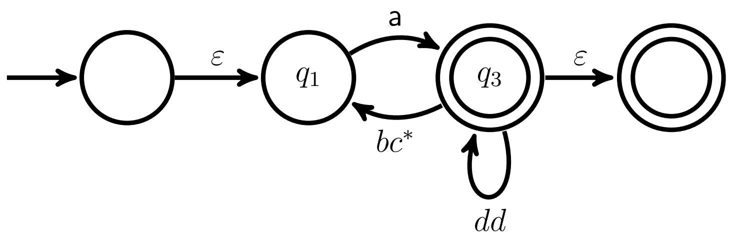




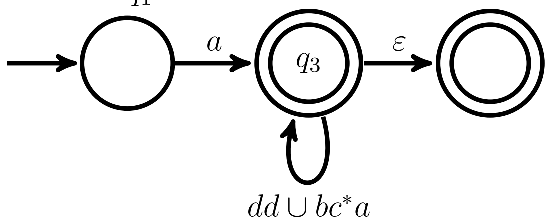
4) GNFA:



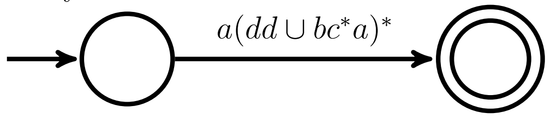
Eliminate q_4 and q_2:



Eliminate q_1:



Finally:



So the answer is: $a(dd \cup bc^*a)^*$

Other possible answers are: $[a(dd)^*bc^*]^*a(dd)^*$ or $a[(dd)^*(bc^*a)*]^*$

5) Suppose A is regular. Let the pumping length be p. Choose $w = 0^p10^p$.

We know that $w = xyz$ and $|xy| \leqslant p$ therefore y consists of zeros only.

$y = 0^k$ where $k \leqslant p$.

$w = xyz \in A$ but $xyyz = 0^{p+k}10^p \notin A$ for any possible choice of y. So there is a contradiction.

A is non-regular by pumping lemma.

Çankaya University
Department of Computer Engineering
2013 - 2014 Fall Semester

CENG 491 - Formal Languages and Automata

Second Midterm Examination

1) Eliminate rules containing ε from the following grammar:

$$S \to AB \mid 00A$$
$$A \to BB \mid 0 \mid A11B \mid \varepsilon$$
$$B \to 1B \mid 1A1$$

2) Give *either* a PDA (pushdown automaton) *or* a CFG (context free grammar) for the language $\{a^n b^m c^m b^n \mid n, m \geqslant 0\}$ over $\Sigma = \{a, b, c\}$

3) Let $A = \{a^n b^{n+1} c^{4n} \mid n \geqslant 0\}$. Show that A is not a context-free language.

4) Describe a Turing machine that recognizes the language A, given in question 3.

5) A Turing machine with doubly infinite tape has a single tape, but its tape is infinite to the left and to the right. The tape is initially blank except for the input. Show that this is equivalent to an ordinary Turing machine.

Bonus) Give a PDA (pushdown automaton) for the language over $\Sigma = \{0, 1\}$ made of strings that contain equal numbers of 0's and 1's.

Answers

1)

$S \to AB \mid 00A \mid B \mid 00$

$A \to BB \mid 0 \mid A11B \mid 11B$

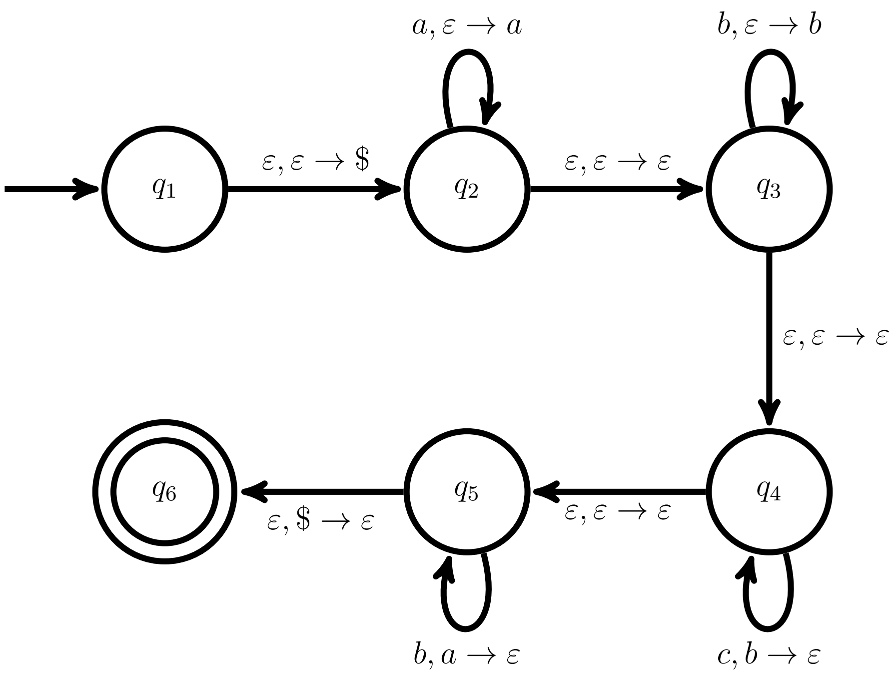
$B \to 1B \mid 1A1 \mid 11$

2)

$S \to aSb \mid T$

$T \to bTc \mid \varepsilon$

OR



3) Suppose A is context free. Let p be the pumping length. Choose w as $w = a^p b^{p+1} c^{4p}$. How can we choose v and y such that $a^p b^{p+1} c^{4p} = uvxyz$?

1) They contain more than one symbol.

In this case, the pumped string uv^2xy^2z will have symbols out of order. For example, it will contain a's after b's. Therefore $uv^2xy^2z \notin A$.

2) They contain a single symbol.

In that case, we can pump at most two of the symbols $\{a, b, c\}$. The remaining one will have the same power in the pumped string. For example, if we choose $v = a$ and $y = b$ we obtain $uv^2xy^2z = a^{n+1}b^{n+2}c^{4n} \notin A$

Therefore we cannot pump this string. By pumping lemma, A is not context free.

4)

1. Sweep from left to right. IF any symbol is out of order (for example, a's after b's) REJECT.
2. Go to start. Search for a.
   IF found, cross it. (Replace by $\times$)
   ELSE, Go to 6.
3. Search for b.
   IF found, cross it.
   ELSE, REJECT.
4. Repeat 4 times:
   Search for c.
      IF found, cross it.
      ELSE, REJECT.
5. Go to 2.
6. Sweep from left to right.
   IF there is one and only one b AND there is no c, ACCEPT.
   ELSE, REJECT.

5) We can divide the double tape into two parts: positive and negative. Then, we can map it into normal Turing tape by zig-zagging and placing positive squares to odd squares and negative squares to even squares as follows:

$$f(n) = \begin{cases} 2n-1 & n>0 \\ -2n & n<0 \end{cases}$$

Double Tape

⋯	−3	−2	−1	1	2	3	⋯

Normal Tape

1	−1	2	−2	3	−3	⋯

OR

We can use the multi-tape idea for double tape and separate the contents by a #:

Double Tape

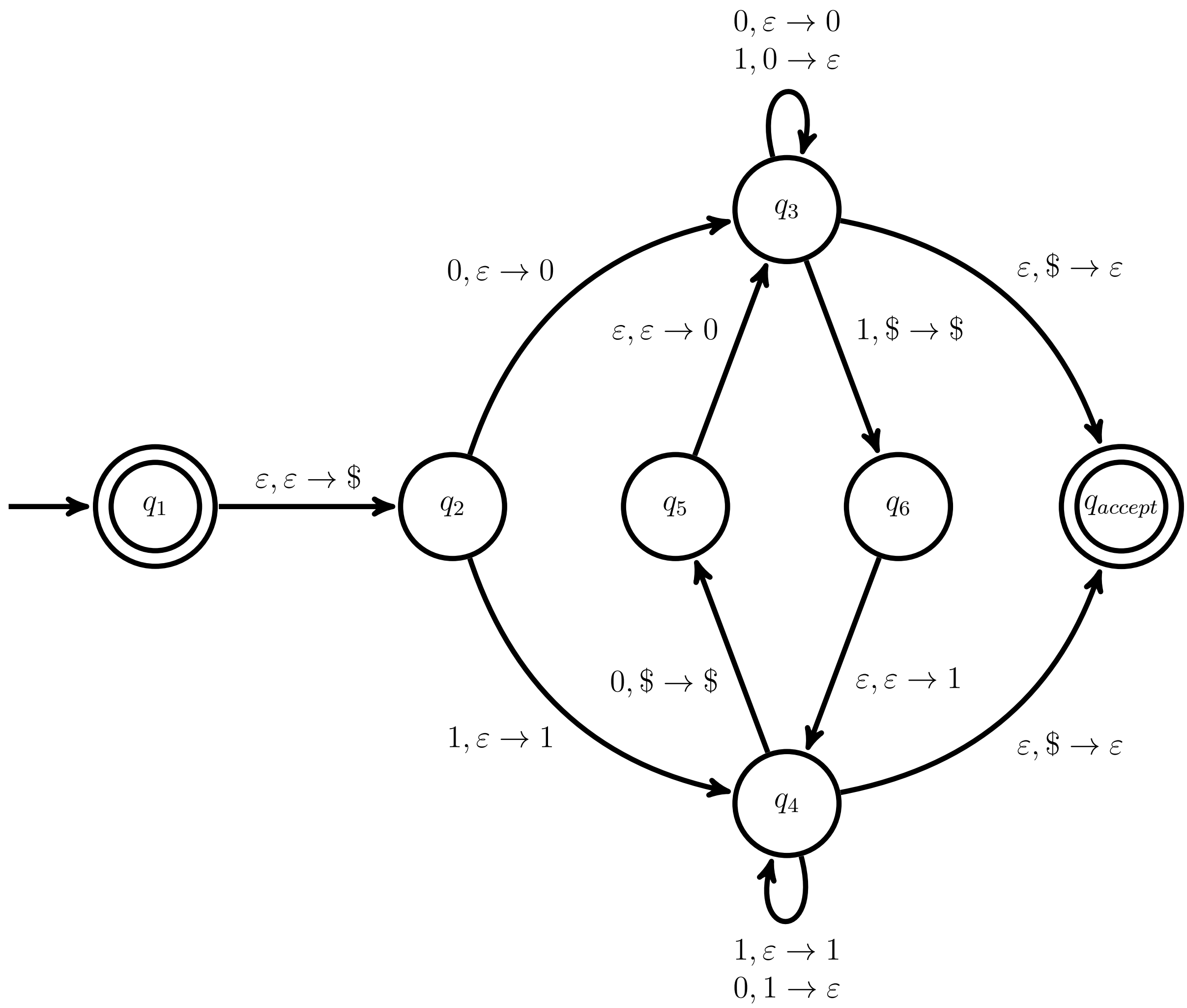
⋯	−3	−2	−1	1	2	3	⋯

Normal Tape

1	2	⋯	n	#	−1	−2	⋯

Note that in this case the # moves as the program proceeds.

Bonus)



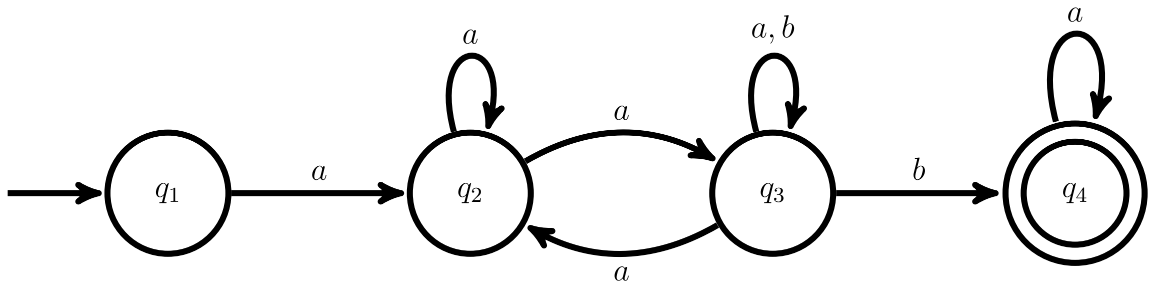
State q_3: Up to now, more 0's than 1's arrived. Stack contains 0's.
State q_4: Up to now, more 1's than 0's arrived. Stack contains 1's.

Çankaya University
Department of Computer Engineering
2013 - 2014 Fall Semester

CENG 491 - Formal Languages and Automata

Final Examination

1) Find an equivalent DFA for the following NFA:



2) Give a CFG that generates the following language over $\Sigma = \{0, 1\}$:

$\{w \mid w$ is of odd length and contains more 0's than 1's.$\}$

3) Define the language L as: $\langle A, k \rangle$ where A is an NFA, k is an integer and A rejects all strings of length $\leqslant k$.

Show that L is decidable. (Hint: Give a TM for L that halts)

4) Let $A =$ the set of all integer coefficient polynomials.
$B =$ the set of all functions $f : \mathbb{N} \to \mathbb{N}$.
$C =$ the set of all infinite strings on $\Sigma = \{0, 1, 2\}$.
Choose one of these sets. Claim that it is countable or uncountable. Prove your claim.

5) TS (Traveling Salesman) PROBLEM: Given n cities, and distances $d(i, j)$ between them, what is the minimum distance of a path that visits each city once?

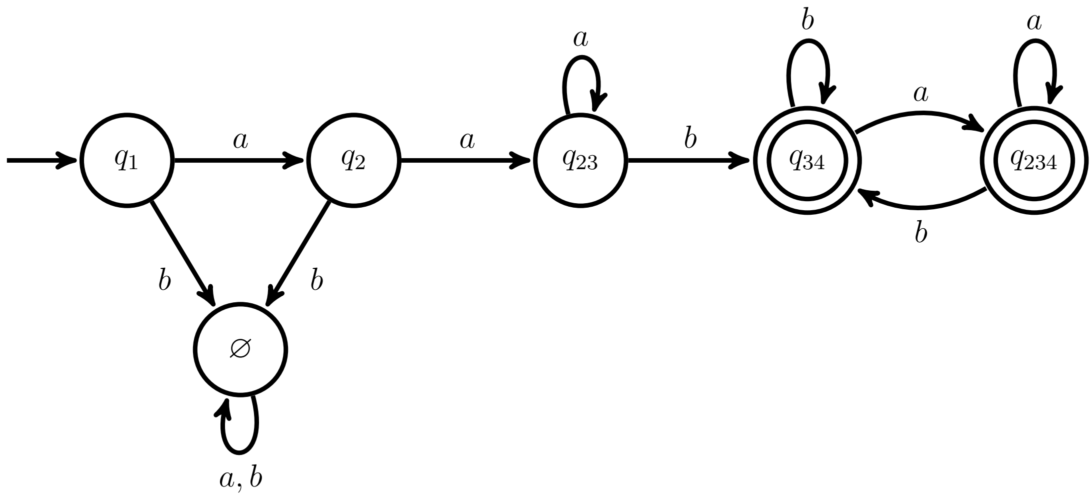
HAMPATH (Hamiltonian Path) PROBLEM: Given a graph G, is there a path that goes through each node exactly once?

a) Reformulate the TS problem as a Yes/No problem. Call it TS2.

b) Given that the HAMPATH problem is NP-complete, show that TS2 is NP-complete.

Answers

1)



2)

$S \to T0T$

$T \to 0T1 \mid 1T0 \mid 0T0 \mid TT \mid \varepsilon$

3)

1. Simulate A on the TM.
2. Mark start state.
3. For $i = 1$ to k
   Mark all states that can be reached from marked states in one step.
   If an accept state is marked, Return REJECT.
   EndFor
4. Return ACCEPT.

Another idea (that is bad style but still works) is to make a list of all strings of length $\leqslant k$, give them one by one to our simulation of A and reject if it accepts any one of them.

4) a) The set $\{\ldots, -1, 0, 1, 2, \ldots\}$ is countable. Therefore there are countably many a_0 terms in the polynomial. Similarly, there are countably many $a_1 x$ terms. In that way, we have to find countable union of countable sets which is countable. We can count them in the same way we counted rational numbers, by counting an infinite matrix.

b) We can think of each function as an infinite string. The set of such strings is uncountable. If we assume there is a list, we can generate an element that is not on the list using Cantor's diagonal idea. This contradicts the assumption that the list contains everything. This is the same idea we used in proving uncountability of real numbers on $[0, 1]$.

c) The same as **b)**.

5) a) TS2 (Traveling Salesman) PROBLEM: Given n cities, and distances $d(i, j)$ between them, and a positive number k, is there a path that visits each city once and has length $\leqslant k$?

b) We have to show that

$$\text{HAMPATH} \leqslant_P \text{TS2}$$

Suppose we can solve any TS2 problem. Given a HAMPATH problem, transform it into TS2 as follows:

- If there is a connection between vertex i and vertex j, set $d(i, j) = 1$.
- If there is no connection between vertex i and vertex j, set $d(i, j) = 10$.
- Set $k = n$ (number of vertices)

If TS2 finds a path of length $\leqslant n$ (actually, we have length $= n$) it is the one we are looking for.

Clearly, this reduction is of polynomial type.

Name-Surname: 03.10.2013

ID Number:

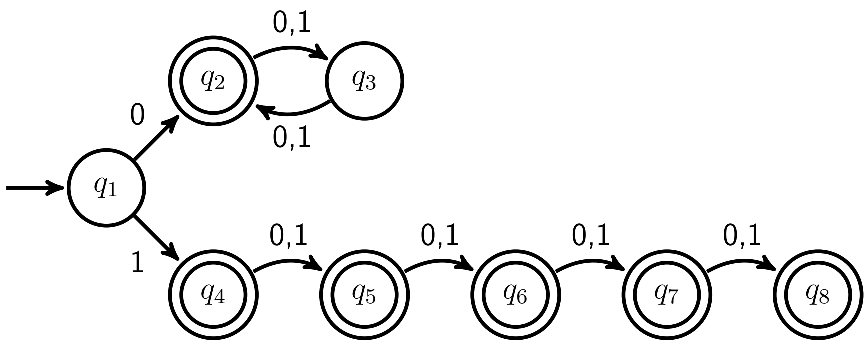
CLASSWORK 1

Give the state diagram of a DFA or an NFA that recognizes the language A over alphabet $\Sigma = \{0, 1\}$ where

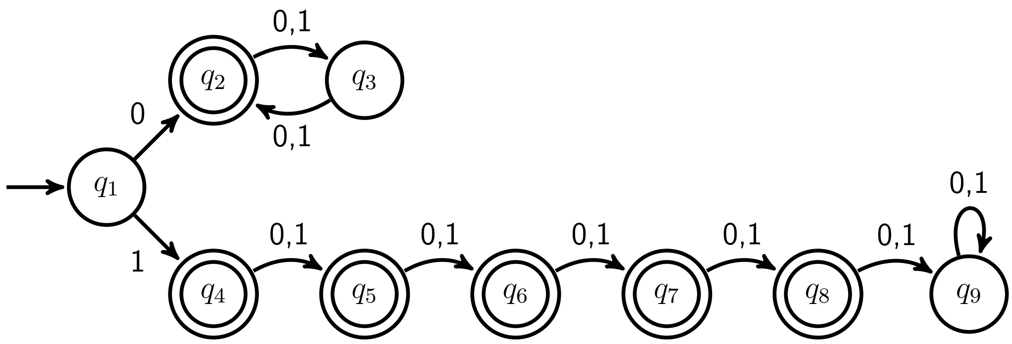
$A = \{w \mid w$ starts with 0 and has odd length, or starts with 1 and has length at most 5$\}$

Answer:

NFA:



DFA:



Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 03.10.2013

ID Number:

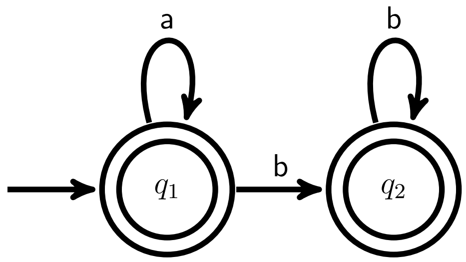
CLASSWORK 1

Give the state diagram of a DFA or an NFA that recognizes the language A over alphabet $\Sigma = \{a, b\}$ where

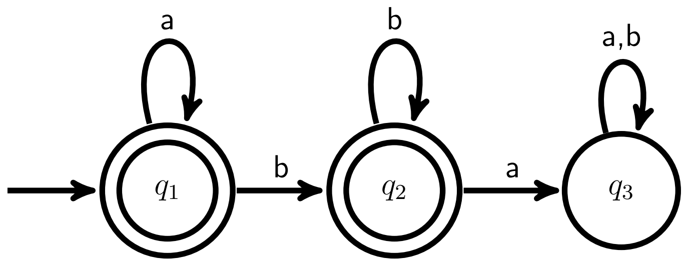
$$A = \{w \mid w = a^*b^*\}$$

Answer:

NFA:



DFA:



Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 10.10.2013

ID Number:

CLASSWORK 2

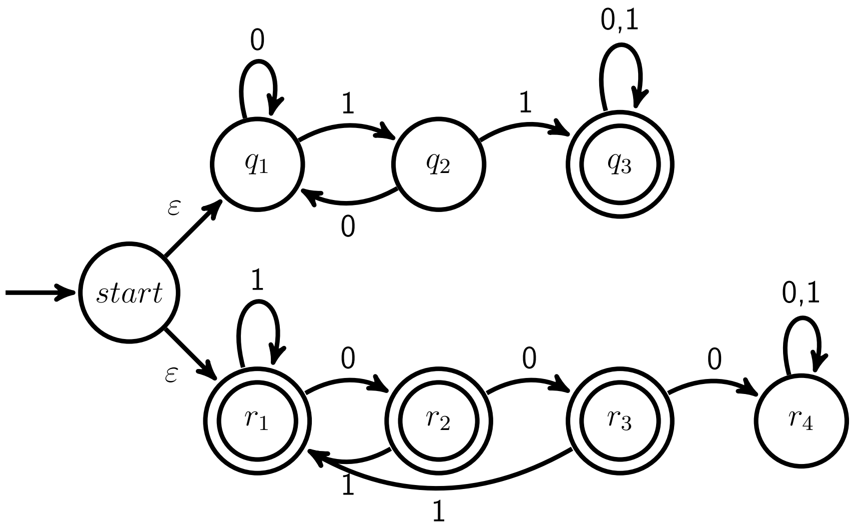
Let A and B be languages over $\Sigma = \{0, 1\}$.

$A = \{w \mid w \text{ contains } 11\}$

$B = \{w \mid w \text{ does not contain } 000\}$

Give the state diagram of an NFA that recognizes $A \cup B$.

Answer:



Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 10.10.2013

ID Number:

CLASSWORK 2

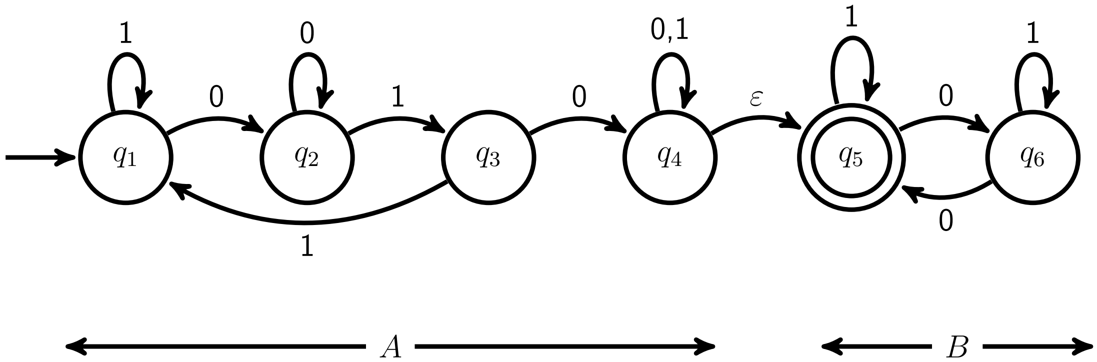
Let A and B be languages over $\Sigma = \{0, 1\}$.

$A = \{w \mid w \text{ contains } 010\}$

$B = \{w \mid w \text{ contains even number of zeros}\}$

Give the state diagram of an NFA that recognizes $A \circ B$.

Answer:



Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 24.10.2013

ID Number:

CLASSWORK 3

Let $\Sigma = \{a, b\}$. A language is described by the regular expression: $(\Sigma^* a \Sigma^* a \cup bb)b^*$

a) Give a string that is in this language.

b) Give a string that is NOT in this language.

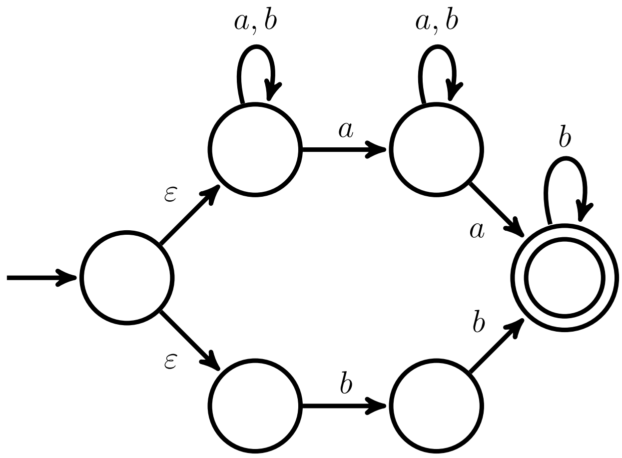
c) Give an NFA that recognizes this language.

Answer:

a) *aa* or *bb* or *aabbbbb* or *bbabababab*

b) *a* or *b* or *bbbba*

c)



Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 24.10.2013

ID Number:

CLASSWORK 3

Let $\Sigma = \{0, 1\}$. A language is described by the regular expression: $(\varepsilon \cup 000 \cup 11)(01)^*$

a) Give a string that is in this language.

b) Give a string that is NOT in this language.

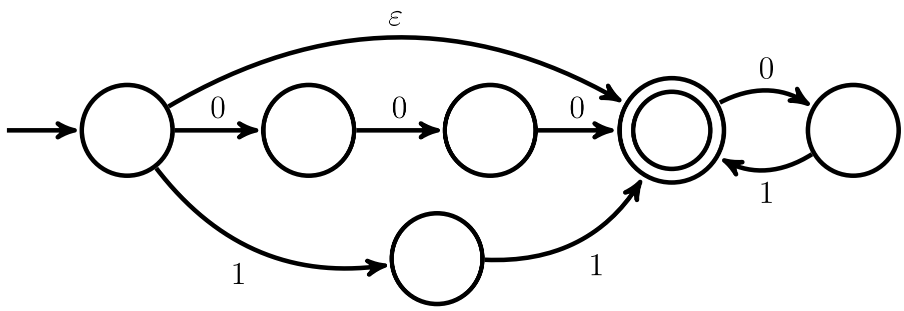
c) Give an NFA that recognizes this language.

Answer:

a) 000 or 11 or 00001 or 010101

b) 11111 or 0000 or 0 or 1.

c)



Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 07.11.2013

ID Number:

CLASSWORK 4

Give a CFG generating the following language over $\Sigma = \{0, 1\}$:

$\{w \mid w$ is of even length and starts and ends with the same symbol.$\}$

Answer:

$S \to 0A0 \mid 1A1 \mid \varepsilon$

$A \to 00A \mid 01A \mid 10A \mid 11A \mid \varepsilon$

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 07.11.2013

ID Number:

CLASSWORK 4

Give a CFG generating the following language over $\Sigma = \{0, 1\}$:

$\{w \mid w$ is of odd length and contains at least two 0's.$\}$

Answer:

$S \rightarrow A0A0A \mid A0B0B \mid B0A0B \mid B0B0A$

$A \rightarrow 0B \mid 1B$

$B \rightarrow \varepsilon \mid 00B \mid 01B \mid 10B \mid 11B$

(Here, A is any string of odd length and B is any string of even length.)

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

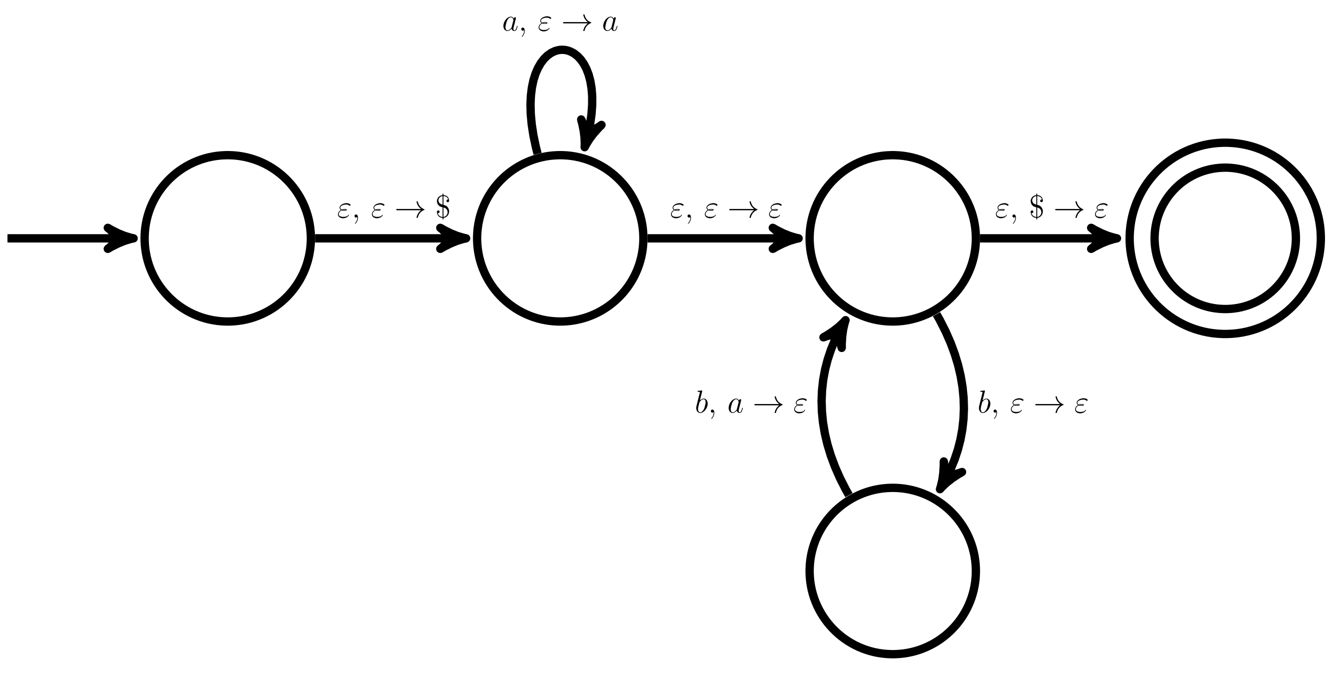
Name-Surname: 14.11.2013

ID Number:

CLASSWORK 5

Give a PDA (pushdown automaton) that recognizes the language $\{a^n b^{2n} \mid n \geqslant 0\}$ over $\Sigma = \{a, b\}$.

Answer:



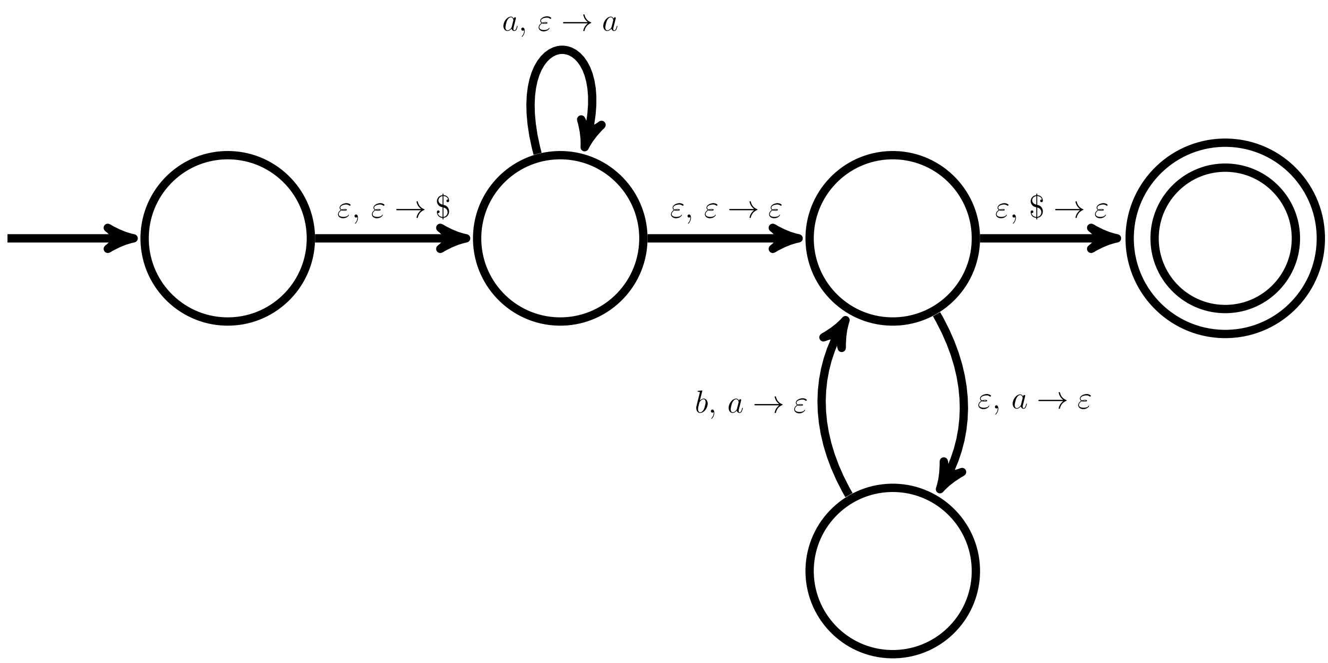
Name-Surname: 14.11.2013

ID Number:

CLASSWORK 5

Give a PDA (pushdown automaton) that recognizes the language $\{a^{2n}b^n \mid n \geqslant 0\}$ over $\Sigma = \{a, b\}$.

Answer:



Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 21.11.2013

ID Number:

CLASSWORK 6

Let $A = \{a^{n!} \mid n \geqslant 0\}$. Use pumping lemma to show that A is not a context-free language.

Answer:

Let the pumping length be p. Choose the string as $s = a^{p!}$. Now divide into five parts as $s = uvxyz$. Any choice for v and y consist of a's only. We know that

$$|vxy| \leqslant p$$

therefore

$$|uv^2xy^2z| \leqslant p! + p$$

Clearly,

$$p! + p < (p+1)!$$

therefore

$$uv^2xy^2z \neq a^{(p+1)!} \quad \Rightarrow \quad uv^2xy^2z \notin A$$

So we cannot pump this string.

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 21.11.2013

ID Number:

CLASSWORK 6

Let $B = \{a^{n^2} \mid n \geqslant 0\}$. Use pumping lemma to show that B is not a context-free language.

Answer:

Let the pumping length be p. Choose the string as $s = a^{p^2}$. Now divide into five parts as $s = uvxyz$. Any choice for v and y consist of a's only. We know that

$$|vxy| \leqslant p$$

therefore

$$|uv^2xy^2z| \leqslant p^2 + p$$

Clearly,

$$p^2 + p < (p+1)^2$$

therefore

$$uv^2xy^2z \neq a^{(p+1)^2} \quad \Rightarrow \quad uv^2xy^2z \notin A$$

So we cannot pump this string.

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 28.11.2013

ID Number:

CLASSWORK 7

Give description of a Turing machine M that, given an input from $\{0, 1, \#\}^*$, eliminates all $\#'s$. For example, given input 011#01#0, the tape should contain 011010 when M halts.

Answer:

1. Move right until meeting the first #. If there is no #, accept.
2. Move right, read symbol, move left, write symbol, move right. (This moves one symbol one unit left, the symbol could be $0, 1$, # or blank).
3. If symbol is blank, go to 4.
   Else, go to 2. (This moves all symbols to the right of # one unit left).
4. Move the head to the start and go to 1.

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 28.11.2013

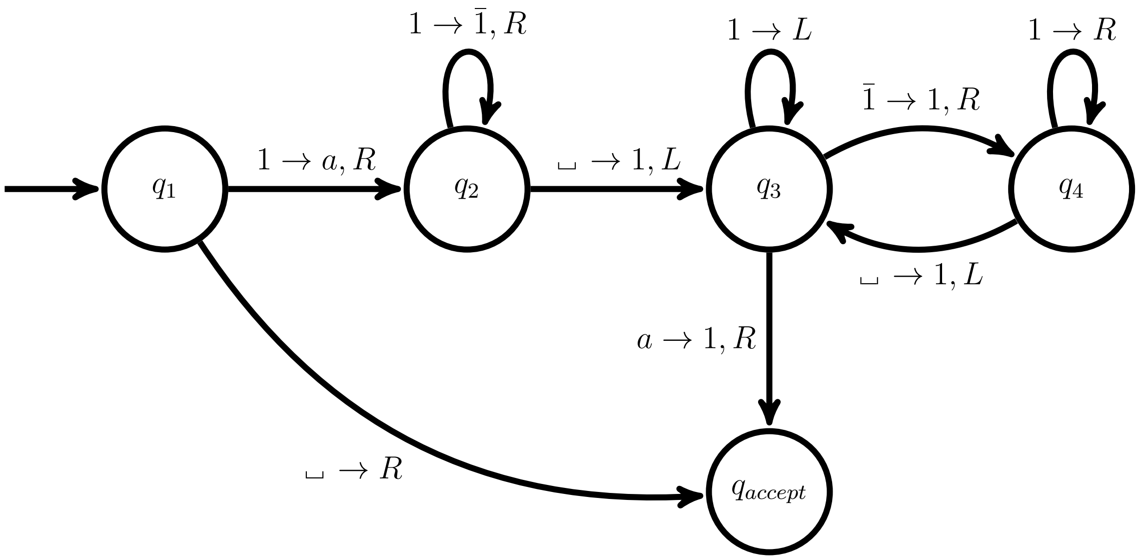
ID Number:

CLASSWORK 7

Give description of a Turing machine M that, given an input from $\{1\}^*$, doubles the number of symbols. For example, given input 111, the tape should contain 111111 when M halts.

Answer:

1. If the first symbol is blank, accept. If it is 1, put a mark on it and move right. Put a mark on all 1's until meeting blank.
2. Move left until meeting the first $\bar{1}$ (marked 1). Erase the mark. If there's no marked 1, accept.
3. Move right until meeting blank. Replace blank with 1.
4. Go to 2.



(Here, a denotes the very first $\bar{1}$)

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 19.11.2013

ID Number:

CLASSWORK 8

Let $E_{DFA} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \varnothing\}$.

In other words, A is a DFA that does not accept any string.

Show that E_{DFA} is a decidable language.

Answer:

The following Turing machine decides this language. (Try to build a path from accept state backwards to start state)

On input A where A is a DFA:

1. Mark the accept states of A. If there is no accept state, ACCEPT.
2. Repeat until no new states get marked:
3. If a state has an arrow pointing to a marked state, mark it.
4. If start state is marked, REJECT; otherwise, ACCEPT.

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 19.11.2013

ID Number:

CLASSWORK 8

Let $E_{CFG} = \{\langle G \rangle \mid G \text{ is a CFG and } L(G) = \varnothing\}$.

In other words, G is a CFG that does not generate any string.

Show that E_{CFG} is a decidable language.

Answer:

The following Turing machine decides this language. (Try to build a path from terminals backwards to S)

On input G, where G is a CFG:

1. Use Chomsky form. Mark all terminal symbols. If there is no terminal, ACCEPT.
2. Repeat until no new variables get marked:
3. Mark any variable A if there is a rule $A \to a$, OR if there is a rule $A \to BC$ and both B and C are marked.
4. If S is not marked, ACCEPT; otherwise, REJECT.

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 26.11.2013

ID Number:

CLASSWORK 9

The VERTEX COVER problem is defined as:

Given a graph and an integer k, is there a collection of k vertices such that each edge is connected to one of the vertices in the collection?

Is VERTEX COVER in NP?

Answer:

Given a solution, we can verify it in polynomial time by the following method:

```
INPUT A total of n vertices, a list of k vertices (cover), a list of ℓ edges.
For i = 1 to k
    For j = 1 to ℓ
        If vertex i is connected to edge j
            Mark edge j
        EndIf
    EndFor
EndFor
For j = 1 to ℓ
    If edge j is unmarked
        Return REJECT
    EndIf
EndFor
Return ACCEPT
```

We know that $k \leqslant n$ and $\ell \leqslant n(n-1)/2$ so this algorithm takes $O(n^3) + O(n^2) = O(n^3)$ time, i.e. polynomial time.

Therefore VERTEX COVER is in NP.

Note: We assume connection between vertices can be checked in $O(1)$ operation.

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 26.11.2013

ID Number:

CLASSWORK 9

The COLORING problem is defined as:

Given a graph and an integer k, is there a way to color the vertices with k colors such that adjacent vertices are colored differently?

Is COLORING in NP?

Answer:

Given a solution, we can verify it in polynomial time by the following method:

```
INPUT A list of n vertices with each one having one of k colors, a list of ℓ edges.
For i = 1 to n − 1
    For j = i + 1 to n
        If vertex i is connected to vertex j AND Color(i)==Color(j)
            Return REJECT
        EndIf
    EndFor
EndFor
Return ACCEPT
```

This algorithm takes $O(n^2)$ time, i.e. polynomial time.

Therefore COLORING is in NP.

Note: We assume connection between vertices can be checked in $O(1)$ operation.

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 07.01.2014

ID Number:

CLASSWORK 10

INDEPENDENT SET problem: Given a graph G and an integer k, is there a subset S of k vertices such that no two vertices in S are connected? (All vertices are independent)

VERTEX COVER problem: Given a graph G and an integer k, is there a subset S of k vertices such that every edge has at least one endpoint in S?

Show that

INDEPENDENT SET $\leqslant_P$ VERTEX COVER

Answer:

Suppose we have an algorithm that decides VERTEX COVER in polynomial time.

INPUT A graph G with n vertices, an integer k.
VERTEXCOVER$(G, n-k)$ // If reject, reject, if accept, accept.

(Note that, if there is a vertex cover of $n-k$ elements, the remaining k elements form an independent set.)

Çankaya University
Department of Computer Engineering

CENG 491 - Formal Languages and Automata

Name-Surname: 07.01.2014

ID Number:

CLASSWORK 10

PARTITION Problem: Given a finite set of integers, determine if it can be partitioned into two sets such that the sum of all integers in the first set equals the sum of all integers in the second set.

SUBSET SUM Problem: Given a set of integers and a number k, determine if there is a subset of these integers whose sum is k.

Show that

PARTITION $\leqslant_P$ SUBSET SUM

Answer:

Suppose we have an algorithm that decides SUBSET SUM in polynomial time.

INPUT A set S of integers, an integer k.

Add all elements of S. Call the result TOTAL.

If TOTAL is odd

    Return REJECT

else

    SUBSETSUM($S, TOTAL/2$)   // If reject, reject, if accept, accept.

EndIf

Prepared by: Dr. Emre Sermutlu (2014)