(1) [6 marks]

Consider the context-sensitive grammar $G = (\{\mathtt{S}\}, \{\mathtt{a}, \mathtt{b}\}, \mathtt{S}, P)$ where P consists of the following productions:

$$\begin{array}{rcl} \mathtt{S} & \rightarrow & \mathtt{abb} \mid \Lambda \\ \mathtt{ab} & \rightarrow & \mathtt{aabbb} \mid \mathtt{ba} \end{array}$$

Precisely define $\mathrm{L}(G)$, the language generated by G.

(2) [20 marks]

Let M be the Turing machine with state set $Q = \{q_0, \ldots, q_7, \mathrm{h_a}, \mathrm{h_r}\}$, input alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \#\}$, tape alphabet $\Gamma = \Sigma \cup \{\Delta, \mathtt{X}\}$, initial state q_0, and the following transition function δ:

δ	Δ	$\mathtt{a}$	$\mathtt{b}$	$\#$	$\mathtt{X}$
q_0	$(q_1, \Delta, \mathrm{R})$				
q_1		$(q_2, \Delta, \mathrm{R})$	$(q_5, \Delta, \mathrm{R})$	$(q_7, \#, \mathrm{R})$	
q_2		$(q_2, \mathtt{a}, \mathrm{R})$	$(q_2, \mathtt{b}, \mathrm{R})$	$(q_3, \#, \mathrm{R})$	
q_3			$(q_4, \mathtt{X}, \mathrm{L})$		$(q_3, \mathtt{X}, \mathrm{R})$
q_4	$(q_1, \Delta, \mathrm{R})$	$(q_4, \mathtt{a}, \mathrm{L})$	$(q_4, \mathtt{b}, \mathrm{L})$	$(q_4, \#, \mathrm{L})$	$(q_4, \mathtt{X}, \mathrm{L})$
q_5		$(q_5, \mathtt{a}, \mathrm{R})$	$(q_5, \mathtt{b}, \mathrm{R})$	$(q_6, \#, \mathrm{R})$	
q_6		$(q_4, \mathtt{X}, \mathrm{L})$			$(q_6, \mathtt{X}, \mathrm{R})$
q_7	$(\mathrm{h_a}, \Delta, \mathrm{S})$				$(q_7, \mathtt{X}, \mathrm{R})$

(Remember that unspecified transitions go to the reject state $\mathrm{h_r}$.)

(i) [8 marks] Draw a transition diagram for M.

(ii) [6 marks] For each of the following strings, say whether it is accepted by M or not:

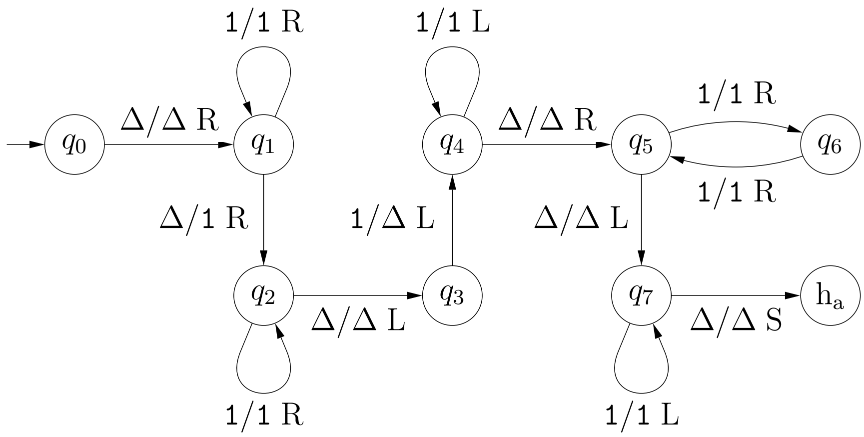
- aab#aab
- aba#bab

(iii) [6 marks] Precisely define $\mathrm{L}(M)$, the language accepted by M.

(3) [12 marks]

Assume that natural numbers $n \in \mathbb{N}$ are represented by strings 1^n (where $1^0 = \Lambda$). Remember that Turing machines compute multi-argument functions from a list of inputs separated by blanks.

Consider the following Turing machine M which computes a binary partial function $f_M : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$:



(i) (a) [2 marks] What is $f_M(1,1)$?

(b) [2 marks] What is $f_M(1,2)$?

(ii) [8 marks] Give the partial function $f_M : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

(4) [12 marks]

Consider the following decision problems:

Halting problem (HP)

Input: A Turing machine M and a string $w \in \Sigma^*$.
Question: Does M reach a halting configuration on input w?

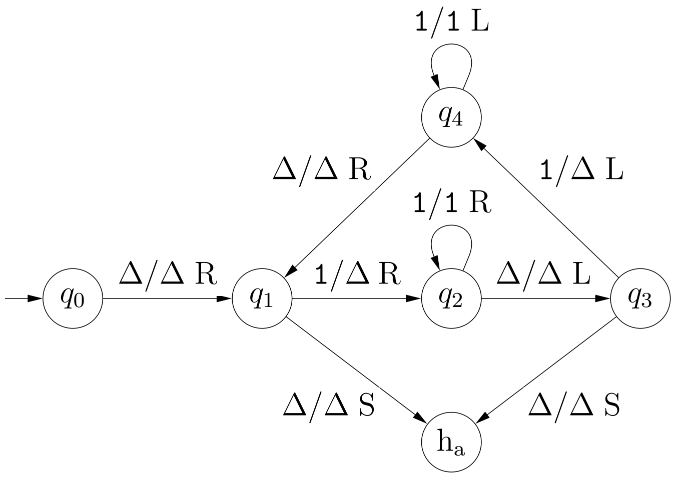
abc-halting problem (abc-HP)

Input: A Turing machine M.
Question: Does M reach a halting configuration on input abc?

Show that the abc-halting problem is undecidable, by reducing HP to abc-HP.

(5) [20 marks]

Consider the following Turing machine M with input alphabet $\{1\}$:



(i) [6 marks] Describe in words the behaviour of M on inputs of length n.

(ii) [4 marks] Give the function s_M, the space complexity of M.

(iii) [10 marks] Give the function τ_M, the time complexity of M, and carefully justify your claim.

(6) [8 marks]

Answer each of the following with "true" or "false". You need not justify your answers.

(i) [2 marks] $2n\log(n) + 128n \in O(n^2)$

(ii) [2 marks] $3^n \in O(2^n)$

(iii) [2 marks] $2^n \in O(n!)$

(iv) [2 marks] $n^n \in O(n!)$

(7) [12 marks]

A boolean expression $C_1 \wedge C_2 \wedge \cdots \wedge C_n$ is in *conjunctive normal form* (CNF) if each conjunct C_i is a disjunction of literals.For example,

$$(\neg \mathtt{x}_1 \vee \mathtt{x}_2 \vee \mathtt{x}_4) \wedge (\mathtt{x}_3 \vee \neg \mathtt{x}_3)$$

is in conjunctive normal form. An expression is a *tautology* if every assignment of truth values to variables makes the expression true. For example, the above expression is not a tautology because the assignment $\mathtt{x}_1, \mathtt{x}_3 \mapsto$ `true`, $\mathtt{x}_2, \mathtt{x}_4 \mapsto$ `false` makes it false. Now consider the following decision problem:

<u>CNF-Tautology problem (CNF-Taut)</u>

Input: A boolean expression C in conjunctive normal form.
Question: Is C a tautology?

Sketch a proof that CNF-Taut is in P.

Hint: show that a CNF expression is a tautology if and only if each disjunction of literals has a particular form. Then describe a Turing machine which checks whether each conjunct has this form and explain why the machine's time complexity is a polynomial.

(8) [10 marks]

Show that if L_1 and L_2 are languages in NP, then $L_1 \cap L_2$ is also in NP.