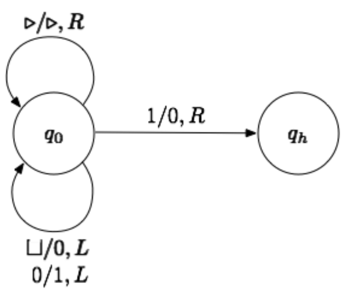
Instructions

- Classroom Problems C2.1–C2.3 will be discussed and solved at the tutorial session on Wed 22 Jan, 14–16, Room T4 (A238). No credit is given for these problems.
- Homework Problems H2.1–H2.3 you should solve on your own, and submit your solutions via the MyCourses interface by the deadline of Tue 28 Jan, 23:59, These problems will be individually graded on a scale of 0–2 points per problem.
- In preparing your solutions to the Homework Problems:
  1. Justify your solutions, be precise, and provide sufficient detail so that it is easy to follow your reasoning.
  2. Submit your solutions as an easily readable, single pdf file, which is either typeset or written in full sentences and clean handwriting.
  3. **[Code of Conduct]** You can discuss the problems with your colleagues and the course's teaching staff, but you must write the presentations of your solutions *independently* and *individually*, without any notes from such discussions.

Classroom Problems

C2.1 Determine the binary representation, according to the encoding scheme presented at Lecture 3, of the single-tape TM with the following transition diagram:



What function on binary input strings does the machine compute?

C2.2 Let $M = (K, \Sigma, \delta, s)$ be a (deterministic or nondeterministic) single-tape Turing machine. How many configurations does M have where the string on its tape has length n? What if M is a transducer (TM with input and output), with $k \geq 0$ worktapes each bounded to length n? (A reasonable upper bound is enough in this case.)

C2.3 Prove, by a reduction from the Halting Problem, that the following problem is undecidable: Given a Turing machine M, does it halt on the empty input string?

Homework Problems

H2.1 Prove that the complexity class **NP** is closed under unions and intersections: that is, if languages A and B are in **NP**, then so are $A \cup B$ and $A \cap B$. (You don't need to describe any Turing machine constructions in extreme detail. It is sufficient to describe the key principles at such a level that a proficient Turing machine programmer could easily implement them in code.) (2 points)

H2.2 Let us define the following complexity classes:

$$\mathbf{PSPACE} = \bigcup_{k>0} \mathbf{SPACE}(n^k),$$
$$\mathbf{EXP} = \bigcup_{k>0} \mathbf{TIME}(2^{n^k}).$$

In other words:

$$\mathbf{PSPACE} = \{L \mid L \text{ is decided by some Turing machine operating in polynomial space}\},$$
$$\mathbf{EXP} = \{L \mid L \text{ is decided by some Turing machine operating in time } 2^{p(n)}, \text{ for some polynomial } p(n)\}.$$

By Theorem 3.1 presented at Lecture 3, we know that $\mathbf{NP} \subseteq \mathbf{EXP}$. Refine this result by proving that in fact $\mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXP}$. (Required detail similar as in previous problem.) (2 points)

H2.3 Prove, by a reduction from the Halting Problem, that the following problem is undecidable: Given a Turing machine M, does it accept all strings x of length $|x| \leq 7$, and only those? (2 points)