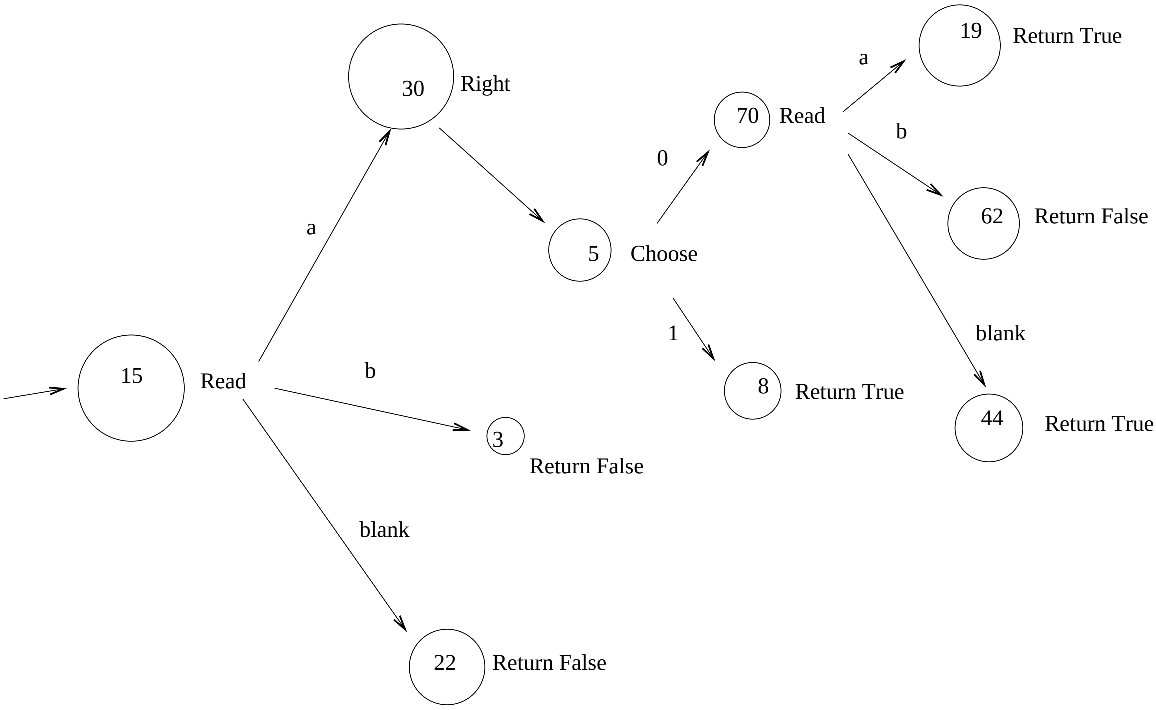
Nondeterministic Turing machines: Problems for Week 8

Exercise 1 What is the language of the following NDTM, over the alphabet $\Sigma = \{a, b\}$? (For a word w, the machine starts with just w on the tape, and the head on the leftmost character, or on a blank if $w = \varepsilon$.)



Solution. All words beginning with a.

Exercise 2 *Are these formulas satisfiable? Justify your answer.*

1. $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q) \wedge p \wedge \neg r$.

__Solution.__ Any truth value assignment that makes this formula true must make the third conjunct p *true, and the fourth conjunct* $\neg r$ *true so* r *will be false. The second conjunct* $\neg p \vee q$ *must be true, but* $\neg p$ *is false so* q *must be true. But then the first conjunct* $(\neg p \vee \neg q \vee r)$ *is false. So this formula is not satisfiable.*

Alternatively, you could explain why this formula is not satisfiable by providing a truth table.

2. $(p \vee q \vee \neg r) \wedge (\neg p \vee q) \wedge p \wedge \neg r$

__Solution.__ The formula is satisfiable. Here is a satisfying truth value assignment: $p =$ *true,* $q =$ *true,* $r =$*false.*

Exercise 3 Consider the following problem: given a set of integers, say whether it has a subset that adds up to 0. For example, if we're given the set $\{12, 2, -7, -8, 3, 14, -5, 1\}$, we could return $\{-7, -8, 1, 14\}$. That's not the only solution, but we're only asked to find one. If we're given the set $\{3, 9, -55, -2\}$ we return "Impossible". Show this problem is in $\mathbf{NP}$.

Hint: You are not expected to give a full Turing machine. Just

- say what a certificate is for this problem,
- explain why it has length polynomial in the input,
- explain why it takes polynomially many steps to check that it is indeed a certificate.

Solution.

A certificate is a subset that adds up to 0. The length of this certificate is linear in the size of the input.

A candidate (which has length $\leqslant n$) consists of at most n words each of length at most n. Adding two words takes $O(n)$ steps (on a two-tape machine), so adding n words takes $O(n^2)$ steps. Checking the sum is zero is $O(n)$ steps. In total, $O(n^2)$ steps, which is polynomial.

(On a Turing machine, there is just one tape, so adding two words takes $O(n^2)$ steps, because we need to mark the current position in each word and move between these marks as we add each digit. So adding n words takes $O(n^3)$ steps, and the overall time is still polynomial. Since the question doesn't specify the kind of machine, and it doesn't affect polynomial status, I'm happy for you to use a second tape here.)

Exercise 4

Suppose there are three boxes numbered $0, 1, 2$ and three bottles, one red, one green and one brown. Each box can accommodate at most two bottles. Let $\phi_{R,i}$ indicate that the red bottle is in space i, and let $\phi_{G,i}$ indicate that the green bottle is in box i, and let $\phi_{B,i}$ indicate that the brown bottle is in box i.

1. Write a formula saying that each bottle is in precisely one box.
2. Write a formula saying that no box contains all three bottles.

First, we want you to write your answers in full, using $\vee, \wedge, \neg, \Rightarrow$. Then, abbreviate your answers using $\bigvee$ and $\bigwedge$. For example,

$$\bigvee_{i\in\{0,1,2,3\}} \phi_i,$$

is an abbreviation for

$$\phi_0 \vee \phi_1 \vee \phi_2 \vee \phi_3.$$

Solution.

1.

$$\begin{aligned}
& (\phi_{R,0} \vee \phi_{R,1} \vee \phi_{R,2}) \wedge \neg((\phi_{R,0} \wedge \phi_{R,1}) \vee (\phi_{R,0} \wedge \phi_{R,2}) \vee (\phi_{R,1} \wedge \phi_{R,2})) \\
\wedge \quad & (\phi_{G,0} \vee \phi_{G,1} \vee \phi_{G,2}) \wedge \neg((\phi_{G,0} \wedge \phi_{G,1}) \vee (\phi_{G,0} \wedge \phi_{G,2}) \vee (\phi_{G,1} \wedge \phi_{G,2})) \\
\wedge \quad & (\phi_{B,0} \vee \phi_{B,1} \vee \phi_{B,2}) \wedge \neg((\phi_{B,0} \wedge \phi_{B,1}) \vee (\phi_{B,0} \wedge \phi_{B,2}) \vee (\phi_{B,1} \wedge \phi_{B,2}))
\end{aligned}$$

Abbreviated solution

$$\bigwedge_{b\in\{R,G,B\}} \left(\bigvee_{i\in\{0,1,2\}} \phi_{b,i} \;\wedge\; \neg \bigvee_{\substack{i,j\in\{0,1,2\}\\ i<j}} (\phi_{b,i} \wedge \phi_{b,j}) \right)$$

2.

$$\neg((\phi_{R,0} \wedge \phi_{G,0} \wedge \phi_{B,0}) \vee (\phi_{R,1} \wedge \phi_{G,1} \wedge \phi_{B,1}) \vee (\phi_{R,2} \wedge \phi_{G,2} \wedge \phi_{B,2}))$$

Abbreviated solution

$$\neg \bigvee_{i\in\{0,1,2\}} \bigwedge_{b\in\{R,G,B\}} \phi_{b,i}$$

Exercise 5 For the alphabet Σ, let L and L' be languages in $\mathbf{NP}$. Show that the language $L \cap L'$ is also in $\mathbf{NP}$. *Hint:* use the "checking machine" definition of $\mathbf{NP}$. You need only describe the machines in outline.

Solution. Say that a certificate for membership of $L \cap L'$ is a certificate for L followed by a certificate for L'. The total length is the sum of the two lengths, hence polynomial. The checking machine for this double certificate begins by checking (in polytime) that the first part is an L-certificate, and then that the second part is an L'-certificate, and returns True if both of these return True, otherwise returns False. This takes polynomial time.