

Homework 2a

Prove that the following problem is PSPACE-complete:

Input: a finite alphabet A , a letter $x_0 \in A$, a finite set X of tuples of the form (a, j, b, k, c) where a, b, c are letters from A and j, k are natural numbers written in unary (i.e., written as a symbol 1 repeated j or k times, respectively; observe that j and/or k can be equal 0);

Question: does there exist an infinite sequence of letters $x_0, x_1, x_2 \dots \in A$ (starting with the given letter) such that for every tuple $(a, j, b, k, c) \in X$ and for every position $i \in \mathbb{N}$, if $x_i = a$ and $x_{i+j} = b$ then $x_{i+j+k} = c$ (in other words, a tuple (a, j, b, k, c) is an implication: if on some position we have a and j positions later we have b , then k more positions later we have c)?

Solution

Proof is split into two parts; 1. shows that given problem belongs to PSPACE, 2. shows it's hardness.

1. For each rule-tuple I will name it's *length* as it's "scope", i.a. 4 for $(a, 1, b, 2, c)$.

Let $MAX := \max(\text{length}(x)), x \in X$. Observe, that with alphabet size of S , it is enough for sequence to have size $S + 1$ to include at least one letter repetition (also, there are S^{S+1} all that sequences).

Let's consider non-deterministic Turing Machine M , that solves given problem the way below:

It creates all legal sequences, each turn checking if it's enough to stop and say, that infinite sequence exists. Let's observe, that we can achieve it the way below:

Remember only last MAX characters (longest tuple scope), beside keep decreasing counter with starting value of $S^{MAX} + 1$. Each time new letter is appended and does not break constraints, counter is decreased. If it does break, then finish run with failure. Let's observe, that if we are still running and reached counter value 0, it means that any configuration of length MAX appeared more than once. Thus we can cycle it and obtain infinite sequence. Because used memory is polynomial, the problem belongs to PSPACE.

2. Let's consider language $L \in PSPACE$ and a Turing Machine M , such that $L(M) = L$ and M working for every word with length n in memory at most $p(n)$, for some polynomial p , and with halting property. Let Q be the set of states of M , Σ it's input

alphabet. I will show that given problem is PSPACE-hard, by reducing L to it. I construct set X of rules as a input of given problem, such that they simulate machine M . Input alphabet for new machine will be $\{\#, \%\} \cup \{a | a \in \Sigma\} \cup \{(q, a) | q \in Q, a \in \Sigma\}$, where (q, a) occurs once per configuration, because it marks head position and current state. Letter x_0 from task description will be $(q_0, w[0])$, where q_0 is initial M 's state and $w[0]$ is first letter of w . For every transition $(q_1, a) \rightarrow (q_2, b, \rightarrow)$ I create rules:

- $((q_1, a), 0, (q_1, a), p(n), b)$
- $\forall_{c \in \Sigma} ((q_1, a), 1, c, p(n), (q_2, c))$

(analogously with \leftarrow and stay-in-place transitions).

Cells not changed by transition are simply copied. It's worth seeing, that list of generated rules is polynomial size, thus reduction is correct.

We add some sets of special rules, that provide condition "accepting run in M exists iff there exists infinite word matching all rules".

- $\{((q_r, a), 0, (q_r, a), 1, S) | a \in \Sigma, S \in \{\#, \%\}\}$.
- $\{((q_a, a), 0, (q_a, a), 1, \#) | a \in \Sigma\}$
- $(\#, 0, \#, 1, \#)$ (for infinite sequence generation).

Let's observe, that rules above make it impossible for infinite sequence to exist, providing we reached rejecting state. Also, reaching accepting state, we are able to generate infinite sequence of $\#$ special signs.

Summary:

We generated polynomial-space reduction of any P-SPACE problem to given problem, thus it is PSPACE-hard, and in association with belonging to PSPACE, it is PSPACE-complete.