CS760 Topics in Computational Complexity	**2024-25 Sem I**

Lecture 9: 30-08-2024

Scribe: Priyanshu Singh, Ravi B Prakash *Lecturer: Rohit Gurjar*

1 From the previous lectures, we know that

We know the following relationships between complexity classes:

$$\mathrm{L} \subseteq NL \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{NPSPACE} \subseteq \mathrm{EXP}$$

By the Space Hierarchy Theorem, we also know that:

$$\mathrm{L} \subsetneq \mathrm{PSPACE} \quad \text{and} \quad \mathrm{P} \subsetneq \mathrm{EXP}$$

2 Savitch's Theorem

Savitch's Theorem states that if we have a non-deterministic Turing machine (NTM) working in space $S(n)$, then:

$$\mathrm{NSPACE}(S(n)) \subseteq \mathrm{SPACE}(S(n)^2)$$

This implies:

$$\mathrm{NL} \subseteq \mathrm{L}^2$$

(Note: In the NSPACE($S(n)$) part, a standard depth-first search (DFS) does not help because we end up with an exponential number of vertices in the configuration graph.)

(For $SPACE(S(n)^2)$, we have a deterministic Turing machine (TM) with space bound by $S(n)^2$.)

2.1 Proof of Savitch's Theorem

Given an input x and a Turing Machine (TM) that is space-bound by $O(S(n))$, the TM has a configuration graph with $2^{O(S(n))}$ nodes. A configuration consists of:

- Head position
- State of the TM
- Work tape contents

We assume $S(n) \geq \log n$. Each edge in the configuration graph represents a transition between configurations based on the input. The main question is: **Is there a path from the starting configuration $C_{\mathbf{start}}$ to an accepting configuration $C_{\mathbf{accept}}$?** This is known as the reachability problem.

We can solve this using the following observation:

- Given two configurations C and C', it is easy to check if there is an edge from C to C' in $O(S(n))$ space. - Using this, we solve the reachability problem by bounding the path length by $2^{O(S(n))}$.

We define the function $\texttt{IsPath}(C, C', i)$: whether there is a path from C to C' of length at most 2^i. Our final answer will be $\texttt{IsPath}(C_{\text{start}}, C_{\text{accept}}, S(n))$. This can be computed recursively as follows:

$$\texttt{IsPath}(C, C', i) \text{ is true if and only if there exists a configuration } C'' \text{ such that:}$$

$$\texttt{IsPath}(C, C'', i-1) \text{ and } \texttt{IsPath}(C'', C', i-1)$$

We check this for all possible configurations C''. The number of configurations is $2^{O(S(n))}$, so this step has exponential time complexity.

2.2 Space Complexity

The space complexity can be computed using the recurrence relation:

$$T(i) = T(i-1) + O(S(n))$$

$T(i)$ represents the space needed at recursion depth i, and i goes up to $S(n)$. Since the depth of recursion is bounded by $S(n)$, the overall space complexity comes out to be:

$$O(S(n)^2)$$

This proves Savitch's Theorem.

2.3 Time Complexity

The time complexity is $2^{O(S(n)^2)}$ because the branching factor is $2^{O(S(n))}$ and the maximum depth of the recursion tree is $S(n)$.

For the reachability problem on an n-node graph, the algorithm has:

- Space complexity: $O((\log n)^2)$
- Time complexity: $O(n^{\log n})$

In contrast a polynomial time algorithm like DFS uses $O(n)$ space. Here, we have a big improvement in space complexity, but then we have to pay the cost in time complexity.

3 PSPACE Completeness

A problem Q is **PSPACE-Complete** if the following conditions hold:

1. Every problem in PSPACE can be reduced to Q in polynomial time (poly-time reduction).
2. Q belongs to PSPACE.

Note: The power of reduction must be strictly less than the complexity class for which we are defining completeness. Otherwise, we would end up with a trivial case.

4 Theorem - QBF Problem is PSPACE-Complete

The Quantified Boolean Formula (QBF) problem is PSPACE-Complete. To prove this, we first note that:
- The SAT problem reduces to QBF, which makes QBF NP-hard.

Before diving into the proof, here are a few interesting examples of PSPACE-Complete problems:

- **Chess**: The game of Chess is PSPACE-Complete (for some generalized forms of the game).
- **POSET Game**: The Partially Ordered Set (POSET) game is PSPACE-Complete.
- **Robotics Problems**: Some problems in robotics also turn out to be PSPACE-Complete. For instance, in certain robotics scenarios, after making every move, the environment (or nature) changes. The objective is to make the right moves in order to "win".
- **Games Against Nature**: Papadimitirou showed that certain games against nature are also PSPACE-Complete. These can be viewed as instances of QBF, where Player 1 is the non-random player, and "nature" acts as a random player.

4.1 Proof of PSPACE-Completeness of QBF

Let us now prove that QBF is PSPACE-Complete.

Consider a language L and say it is accepted by a Turing machine M working in space $S(n)$. Given an input x, we need a polynomial-time reduction to a QBF instance $\Psi_{M,x}$ such that M accepts x if and only if $\Psi_{M,x}$ is true.

We can convert the question "Is there an edge from C to C'?" into a polynomial-size Boolean formula. Let:

$$C \text{ be represented by } (x_1, x_2, \ldots, x_s)$$

$$C' \text{ be represented by } (x'_1, x'_2, \ldots, x'_s)$$

We can write a Boolean formula in variables $x_1, \ldots, x_s, x'_1, \ldots, x'_s$ which expresses whether it is possible to transition from configuration C to C'. This can be built using the transition function of the Turing machine and the input tape.

4.2 Approach 1

Define a QBF $\Psi_i(C, C')$ for $0 \leq i \leq s$, where $\Psi_i(C, C')$ is true if and only if there is a path of length at most 2^i from C to C'. We now define the recursive relation:

$$\Psi_{i+1}(C, C') = \exists C''[\Psi_i(C, C'') \wedge \Psi_i(C'', C')]$$

However, proceeding this way results in formula size growing exponentially with i.

4.3 Improved Approach

Instead, define $\Psi_{i+1}(C, C')$ as:

$$\Psi_{i+1}(C, C') = \exists C'' \forall D_1 \forall D_2 \left[(D_1 = C \wedge D_2 = C'') \vee (D_1 = C'' \wedge D_2 = C')\right] \implies \Psi_i(D_1, D_2)$$

The part of the formula inside the square brackets has size $O(s)$, and we avoid the exponential blow-up.

Thus, the size of $\Psi_{M,x}$ is $O(s(n)^2)$, which implies that QBF is PSPACE-hard.

Note: The same QBF formula works for a non-deterministic Turing machine (NTM) M, implying that QBF is also NPSPACE-hard. Since we know from the previous lecture that QBF belongs to PSPACE and NPSPACE, it follows that:

$$\text{PSPACE} = \text{NPSPACE}$$

This conclusion is supported by Savitch's Theorem.