Space Complexity

Xin Li

lixints@cs.jhu.edu
Topics

• Space Hierarchy Theorem

• PSPACE-Completeness
Space Hierarchy Theorem

• Thm: If $f$ and $g$ are space constructible functions satisfying $f(n)=o(g(n))$, then

\[ \text{SPACE}(f(n)) \subset \text{SPACE}(g(n)) \]

• Proof: almost identical to the time hierarchy theorem (diagonalization). Using the universal TM to simulate $M_x(x)$ using space $g(|x|)$ for time $2^{O(g(|x|))}$, and flip the output if halt (otherwise reject).

• Note the universal TM only has constant space overhead, so can show the above language $\in \text{SPACE}(g(n))$ but $\not\in \text{SPACE}(f(n))$. 
Complete Problems

• We know that $L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXP \subseteq NEXP$.

• Let’s look at some of the relations closely.

• $P \subseteq PSPACE$. 
PSPACE-Completeness

• We don’t know if $P = \text{PSPACE}$, but strongly believe not since $P \subseteq \text{NP} \subseteq \text{PSPACE}$.

• Thus, $P = \text{PSPACE} \implies P = \text{NP}$.

• Similar to NP-completeness, we can define PSPACE-completeness (the hardest language in PSPACE).
**PSPACE-Completeness**

- Recall we say $L' \leq_p L$ if $L'$ is poly-time reducible to $L$.

- **Definition:** A language $L$ is PSPACE-hard if $\forall L' \in \text{PSPACE}, L' \leq_p L$. If in addition $L \in \text{PSPACE}$, say $L$ is PSPACE-complete.

- $P = \text{PSPACE} \iff$ Whether a PSPACE-complete language $\in \text{P}$.
PSPACE-Completeness

• Question: does there exist a PSPACE-complete language?

• Answer: Yes!

• Quantified Boolean Formula.
Quantified Boolean Formula

• A quantified Boolean formula (QBF) is a formula of the following form:

\[ \psi = \forall x_1 \exists x_2 \exists x_3 \ldots \exists x_n \Phi(x_1, x_2, \ldots, x_n) \]

• Where each \( Q_i \) is either \( \forall \) or \( \exists \). \( x_1, x_2, \ldots x_n \) are Boolean variables and \( \Phi \) is a plain (unquantified) Boolean formula.

• \( \psi \) is said to be in the prenex form (can transform any QBF into this form). Unlike in SAT/3SAT, don’t require \( \Phi \) to be a CNF or 3CNF.
Quantified Boolean Formula

• A quantified Boolean formula (QBF) is a formula of the following form:

\[ \psi = Q_1 x_1 \, Q_2 x_2 \, \ldots \, Q_n \, x_n \, \Phi(x_1, x_2, \ldots, x_n) \]

• Since all the variables of \( \psi \) are quantified (no free variable), \( \psi \) is always either True or False.

• Example: \( \forall x \, \exists y \, (x \land y) \lor (\neg x \land \neg y) \) True \quad \forall x \, \forall y \, (x \land y) \lor (\neg x \land \neg y) \) False
Quantified Boolean Formula

• Define the language TQBF to be the set of all quantified Boolean formulas that are true.

• Equivalently, given a QBF $\psi$, decide if $\psi$ is TRUE.

• Thm: TQBF is PSPACE-complete.
TQBF is PSPACE-complete

- Pf: Need to show two things

- 1) $TQBF \in \text{PSPACE}$. 

- 2) $\forall L \in \text{PSPACE}, L \leq_p TQBF$. 
TQBF $\in$ PSPACE

• Given a QBF $\psi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \Phi(x_1, x_2, \ldots x_n)$ with $n$ variables and size $m$ (length of description as a binary string).

• Show a simple recursive algorithm that decides if $\psi$ is true in $O(mn)$ space.

• Consider a more general case where $\psi$ may also contain constants 0 and 1.
Recursive algorithm for TQBF

• Base case: given a QBF \( \psi \), if \( n=0 \), then \( \psi \) can be computed in \( O(m) \) time and space.
  
  \[ \psi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \Phi(x_1, x_2, \ldots x_n) \]

• Recursive step: If \( n>0 \), then consider \( \psi\mid_{x_1=b} \) for \( b=0 \) and \( 1 \), where \( \psi\mid_{x_1=b} \) is the modification of \( \psi \) where \( Q_1 \) is dropped and all occurrences of \( x_1 \) are replaced with \( b \).
Recursive algorithm for TQBF

• Let the ALG be A.

• Recursive step: If \( Q_1 = \forall \), then A outputs 1 if both \( A(\psi|x_1=1) \) and \( A(\psi|x_1=0) \) output 1. If \( Q_1 = \exists \), then A outputs 1 if either \( A(\psi|x_1=1) \) or \( A(\psi|x_1=0) \) outputs 1.

• Crucial point: the space of computing \( A(\psi|x_1=1) \) and \( A(\psi|x_1=0) \) can be reused.
Recursive algorithm for TQBF

- **Recursive step:** If $Q_1 = \forall$, then $A$ outputs 1 if both $A(\psi|x_1=1)$ and $A(\psi|x_1=0)$ output 1. If $Q_1 = \exists$, then $A$ outputs 1 if either $A(\psi|x_1=1)$ or $A(\psi|x_1=0)$ outputs 1.

- For each recursion, needs 1 bit to record the current assignment to the variable, 1 bit to record the output of $A$, and $O(m)$ bits to write the modified formula.

- Let $S_{n,m}$ be the space used. Then $S_{n,m} = S_{n-1,m} + O(m) \Rightarrow S_{n,m} = O(mn)$.

- Can be improved to $O(m+n)$ if instead of writing the new formula each step, do a depth-first type search.
TQBF is PSPACE-hard

• $\forall L \in \text{PSPACE}, L \leq_p \text{TQBF}$.

• A poly time computable function $f: x \in \{0, 1\}^* \rightarrow \text{QBF } \psi$ s.t.
  $x \in L \iff \psi \in \text{TQBF}$

• Let $M$ be a TM that decides $L$ in $S(n)$ space.

• Show how to construct $\psi$ of size $O(S(n)^2)$ in poly-time.
**TQBF is PSPACE-hard**

- Let \( m = O(S(n)) \) be the \# of bits needed to encode a configuration of \( M \).

- Recall: There is an \( O(m) \) size CNF formula \( \Phi_{M, x} \) s.t. \( \forall \) two strings \( C, C' \in \{0,1\}^m \), \( \Phi_{M, x}(C, C') = 1 \) iff \( C, C' \) encode two configurations s.t. \( C' \) can be reached from \( C \) in one step.

- Generalize: we will use \( \Phi_{M, x} \) to come up with a QBF \( \psi \) with polynomially many quantified variables and two unquantified variables \( C, C' \) s.t.

\[
\forall C, C' \in \{0,1\}^m, \psi(C, C') = 1 \text{ iff } \exists \text{ a directed path from } C \text{ to } C' \text{ in the configuration graph } G_{M, x}
\]
TQBF is PSPACE-hard

• QBF $\psi$ with polynomially many quantified variables and two unquantified variables $C$, $C’$ s.t.
  \[ \forall C, C’ \in \{0,1\}^m, \psi(C, C’) = 1 \iff \exists \text{ a directed path from } C \text{ to } C’ \text{ in the configuration graph } G_{M,x} \]

• Plug in $C = C_{\text{start}}$ and $C’ = C_{\text{accept}}$, we get $x \in L \iff \psi \in \text{TQBF}$.

• Define $\psi$ recursively: let $\psi_i(C, C’) = 1 \iff \exists \text{ a directed path from } C \text{ to } C’ \text{ in the configuration graph with length at most } 2^i$. Then $\psi = \psi_m$ and $\psi_0 = \Phi_{M,x}$. 
TQBF is PSPACE-hard

- Define $\psi$ recursively: let $\psi_i(C, C')=1$ iff $\exists$ a directed path from $C$ to $C'$ in the configuration graph with length at most $2^i$.

- Crucial observation: $\exists$ a path from $C$ to $C'$ in the configuration graph with length at most $2^i$ iff $\exists$ a configuration $C''$ with a path of length at most $2^{i-1}$ from $C$ to $C''$ and a path of length at most $2^{i-1}$ from $C''$ to $C'$.

- Thus, $\psi_i(C, C')= \exists C'' \psi_{i-1}(C, C'') \land \psi_{i-1}(C'', C')$. 
TQBF is PSPACE-hard

• $\psi_i(C, C') = \exists C'' \psi_{i-1}(C, C'') \land \psi_{i-1}(C'', C')$

• However, the size of $\psi_i(C, C')$ doubles each time, so the final size could be $2^m$.

• Use additional quantified variables to save on the size, and rewrite the above formula as follows:

• $\exists C'' \forall D^1 \forall D^2 ((D^1 = C \land D^2 = C'') \lor (D^1 = C'' \land D^2 = C')) \Rightarrow \psi_{i-1}(D^1, D^2)$
TQBF is PSPACE-hard

• \( \exists C'' \forall D^1 \forall D^2((D^1 = C \land D^2 = C'') \lor (D^1 = C'' \land D^2 = C')) \Rightarrow \psi_{i-1}(C, C'') \)

• \( A = B: (A \land B) \lor (\neg A \land \neg B) \), \( A \Rightarrow B: \neg A \lor (A \land B) \).

• Can be converted into the prenex form.

• \( \text{Size}(\psi_i) \leq \text{Size}(\psi_{i-1}) + O(m) \) since \( C, C', C'', D^1, D^2 \) all have size \( m \). So \( \text{Size}(\psi) = O(m^2) = O(S(n)^2) \) and the reduction runs in poly-time.
A closer look

• The proof only uses the properties of the configuration graph.

• So does it still work if the TM is non-deterministic?

• Answer: Yes!

• Implication: TQBF is also NPSPACE-hard. However $TQBF \in PSPACE$, this implies $PSPACE=NPSPACE$. 