Computational Complexity; slides 8, HT 2022
PSPACE-completeness and Quantified Boolean Formulae

Prof. Paul W. Goldberg (Dept. of Computer Science, University of Oxford)

HT 2022

Another nice result

Theorem

If P=NP, *then* EXPTIME=NEXPTIME

Suppose X ∈NEXPTIME. Define $pad(X)$ as follows:

$$w \in X \text{ iff } w\square^{2^n} \in pad(X) \quad \text{(where } n = |w|\text{)}$$

We have $pad(X)$ ∈NP: Given a word of the form $w\square^N$,

- Check you have the right number of $\square$'s.
- run the NEXPTIME algorithm on w-prefix (not the $\square$'s).

Hence $pad(X)$ ∈P by assumption.
Then, you can take poly-time algorithm for $pad(X)$, and convert it to algorithm that checks w-prefix, in time exponential in $|w|$.

Savitch's Theorem: PSPACE=NPSPACE

Let M be an NPSPACE TM of interest; want to know whether M can accept w within $2^{p(n)}$ steps.

Proof idea: predicate reachable(C, C', i), satisfied by configurations C, C' and integer i, provided C' is reachable from C within 2^i transitions (w.r.t M).

Note: reachable(C, C', i) is satisfied provided there exists C'' such that
reachable$(C, C'', i-1)$ and reachable$(C'', C', i-1)$

To check reachable$(C_{init}, C_{accept}, p(n))$, try for all configs C'':
reachable$(C_{init}, C'', p(n)-1)$ and reachable$(C'', C_{accept}, p(n)-1)$

Which themselves are checked recursively. Depth of recursion is $p(n)$, need to remember at most $p(n)$ configs at any time. We may assume C_{accept} is unique.

Savitch's Theorem

More generally:

Theorem. (Savitch 1970)
For all (space-constructible) $S : \mathbb{N} \to \mathbb{N}$ such that $S(n) \geq \log n$,

$$\mathsf{NSPACE}(S(n)) \subseteq \mathsf{DSPACE}(S(n)^2).$$

In particular: PSPACE = NPSPACE
EXPSPACE = NEXPSPACE

A PSPACE-complete problem: QBF

c.f. Cook's theorem.
A more general kind of logic problem characterises PSPACE
`https://en.wikipedia.org/wiki/True_quantified_Boolean_formula`

A Quantified Boolean Formula is a formula of the form

$$Q_1 X_1 \dots Q_n X_n \varphi(X_1, \dots, X_n)$$

where

- the Q_i are quantifiers $\exists$ or $\forall$
- φ is a CNF formula in the variables $X_1, \dots, X_n$ and atoms 0 and 1

Example

$$\exists X_1 \forall X_2 \exists X_3 \forall X_4 \forall X_5 \Big((X_1 \vee 0 \vee \neg X_5) \wedge (\neg X_2 \vee 1 \vee \neg X_5) \wedge (X_2 \vee X_3 \vee X_4) \Big)$$

Quantified Boolean Formulae

Consider the following problem:

QBF	
Input:	A QBF formula φ.
Question:	Is φ true?

Observation: For any propositional formula φ:

φ is satisfiable if, and only if, $\exists X_1 \dots \exists X_n \varphi$ is true.

$X_1, \dots, X_n$: Variables occurring in φ

Consequence: QBF is NP-hard.

Similarly, QBF is also co-NP-hard.

Theorem: QBF is in PSPACE

Proof: Given $\varphi := Q_1 X_1 \ \dots Q_n X_n \psi$, letting $m := |\psi|$

Eval-QBF(φ)

if $n = 0$ Accept if ψ evaluates to true. Reject otherwise.

if $\varphi := \exists X \psi'$
construct $\varphi_1 := \psi'[X \mapsto 1]$
if Eval-QBF(φ_1) evaluates to true, accept.
else construct $\varphi_0 := \psi'[X \mapsto 0]$ (reuse space in Eval-QBF(φ_1))
return **Eval-QBF**(φ_0)

if $\varphi := \forall X \psi'$
construct $\varphi_1 := \psi'[X \mapsto 1]$
if Eval-QBF(φ_1) evaluates to false, reject.
else construct $\varphi_0 := \psi'[X \mapsto 0]$ (reuse space in Eval-QBF(φ_1))
return **Eval-QBF**(φ_0)

Theorem: QBF is in PSPACE

Proof: Given $\varphi := Q_1 X_1 \ \ldots Q_n X_n \psi$, letting $m := |\psi|$

Eval-QBF(φ)

if $n = 0$ Accept if ψ evaluates to true. Reject otherwise.

if $\varphi := \exists X \psi'$
- construct $\varphi_1 := \psi'[X \mapsto 1]$
- **if Eval-QBF**(φ_1) evaluates to true, accept.
- **else** construct $\varphi_0 := \psi'[X \mapsto 0]$ (reuse space in Eval-QBF(φ_1))
  - return **Eval-QBF**(φ_0)

if $\varphi := \forall X \psi'$
- construct $\varphi_1 := \psi'[X \mapsto 1]$
- **if Eval-QBF**(φ_1) evaluates to false, reject.
- **else** construct $\varphi_0 := \psi'[X \mapsto 0]$ (reuse space in Eval-QBF(φ_1))
  - return **Eval-QBF**(φ_0)

Space complexity: Algorithm uses $\mathcal{O}(nm)$ tape cells.
(At depth d of recursion tree, remember d simplified versions of φ; can be improved to $\mathcal{O}(n+m)$ by remembering φ and d bits...)

Theorem: QBF is NPSPACE-hard

Let $\mathcal{L} \in$ NPSPACE. We show $\mathcal{L} \leq_p \mathrm{QBF}$.

Let $M := (Q, \Sigma, \Gamma, q_0, \Delta, F_a, F_r)$ be a TM deciding $\mathcal{L}$
such that M never uses more than $p(n)$ cells.

For each input $w \in \Sigma^*$, $|w| = n$, we construct a formula $\varphi_{M,w}$ such that

M accepts w if, and only if, $\varphi_{M,w}$ is true.

Theorem: QBF is NPSPACE-hard

Let $\mathcal{L} \in$ NPSPACE. We show $\mathcal{L} \leq_p \mathrm{QBF}$.

Let $M := (Q, \Sigma, \Gamma, q_0, \Delta, F_a, F_r)$ be a TM deciding $\mathcal{L}$ such that M never uses more than $p(n)$ cells.

For each input $w \in \Sigma^*$, $|w| = n$, we construct a formula $\varphi_{M,w}$ such that

M accepts w if, and only if, $\varphi_{M,w}$ is true.

Describe configuration $(q, p, a_1 \dots a_{p(n)})$ by a set

$$\mathcal{V} := \{X_q, Y_i, Z_{a,i} : q \in Q, \quad a \in \Gamma, \quad 0 \leq i < p(n)\}$$

of variables and the truth assignment β defined as

$$\beta(X_s) := \begin{cases} 1 & s = q \\ 0 & s \neq q \end{cases} \qquad \beta(Y_s) := \begin{cases} 1 & s = p \\ 0 & s \neq p \end{cases} \qquad \beta(Z_{a,i}) := \begin{cases} 1 & a = a_i \\ 0 & a \neq a_i \end{cases}$$

NPSPACE-Hardness of QBF

Consider the following formula $\textsc{Conf}(\mathcal{V})$ with free variables

$$\mathcal{V} := \{X_q, Y_i, Z_{a,i} : q \in Q, \quad a \in \Gamma, \quad 0 \leq i < p(n)\}$$

$$\textsc{Conf}(\mathcal{V}) := \bigvee_{q \in Q} \Big(X_q \wedge \bigwedge_{q' \neq q} \neg X_{q'} \Big) \quad \wedge \quad \bigvee_{p \leq p(n)} \Big(Y_p \wedge \bigwedge_{p' \neq p} \neg Y_{p'} \Big) \wedge$$

$$\bigwedge_{1 \leq i \leq p(n)} \bigvee_{a \in \Gamma} \big(Z_{a,i} \wedge \bigwedge_{b \neq a \in \Gamma} \neg Z_{b,i} \big)$$

Definition. For any truth assignment β of $\mathcal{V}$ define config$(\mathcal{V}, \beta)$ as

$$\{(q, p, w_1 \dots w_{p(n)}) : \beta(X_q) = \beta(Y_p) = \beta(Z_{w_i,i}) = 1, \forall i \leq p(n)\}$$

Lemma

If β satisfies $\textsc{Conf}(\mathcal{V})$ then $|config(\mathcal{V}, \beta)| = 1$.

NPSPACE-hardness of QBF

Definition. For an assignment β of $\mathcal{V}$ we defined $\text{config}(\mathcal{V}, \beta)$ as

$$\{(q, p, w_1 \dots w_{p(n)}) : \beta(X_q) = \beta(Y_p) = \beta(Z_{w_i,i}) = 1, \forall i \leq p(n)\}$$

Lemma

If β satisfies $\textsc{Conf}(\mathcal{V})$ then $|\mathit{config}(\mathcal{V}, \beta)| = 1$.

Remark. β may be defined on other variables than those in $\mathcal{V}$.

$\text{config}(\mathcal{V}, \beta)$ is a potential configuration of M, but it might not be reachable from the start configuration of M on input w.

Conversely: Every configuration $(q, p, w_1 \dots w_{p(n)})$ induces a satisfying assignment.

NPSPACE-Hardness of QBF

Consider the following formula $\textsc{Next}(\mathcal{V}, \mathcal{V}')$ defined as

$\textsc{Conf}(\mathcal{V}) \wedge \textsc{Conf}(\mathcal{V}') \wedge \textsc{Nochange}(\mathcal{V}, \mathcal{V}') \wedge \textsc{Change}(\mathcal{V}, \mathcal{V}')$.

$$\textsc{Nochange} := \bigwedge_{1 \le p \le p(n)} \Big(Y_p \Rightarrow \bigwedge_{\substack{i \neq p \\ a \in \Gamma}} (Z_{a,i} \leftrightarrow Z'_{a,i}) \Big)$$

$$\textsc{Change} := \bigwedge_{1 \le p \le p(n)} \Big((Y_p \wedge X_q \wedge Z_{a,p}) \Rightarrow \bigvee_{(q,a,q',b,m) \in \Delta} (X'_{q'} \wedge Z'_{b,p} \wedge Y'_{\text{``}p+m\text{''}}) \Big)$$

Lemma

For any assignment β defined on $\mathcal{V}, \mathcal{V}'$:

$$\beta \textit{ satisfies } \textsc{Next}(\mathcal{V}, \mathcal{V}') \iff \textit{config}(\mathcal{V}, \beta) \vdash_M \textit{config}(\mathcal{V}', \beta)$$

NPSPACE-hardness of QBF

Define $\textsc{Path}_i(\mathcal{V}_1, \mathcal{V}_2)$:

M starting on config$(\mathcal{V}_1, \beta)$ can reach config$(\mathcal{V}_2, \beta)$ in $\leq 2^i$ steps.

For $i = 0$: $\quad \textsc{Path}_0 \quad := \quad \mathcal{V}_1 = \mathcal{V}_2 \quad \vee \quad \textsc{Next}(\mathcal{V}_1, \mathcal{V}_2)$

NPSPACE-hardness of QBF

Define $\textsc{Path}_i(\mathcal{V}_1, \mathcal{V}_2)$:

M starting on config$(\mathcal{V}_1, \beta)$ can reach config$(\mathcal{V}_2, \beta)$ in $\leq 2^i$ steps.

For $i = 0$***:*** $\quad \textsc{Path}_0 \quad := \quad \mathcal{V}_1 = \mathcal{V}_2 \quad \vee \quad \textsc{Next}(\mathcal{V}_1, \mathcal{V}_2)$

For $i \to i+1$***:***

Idea: $\textsc{Path}_{i+1}(\mathcal{V}_1, \mathcal{V}_2) := \exists \mathcal{V} \Big[\textsc{Conf}(\mathcal{V}) \wedge \textsc{Path}_i(\mathcal{V}_1, \mathcal{V}) \wedge \textsc{Path}_i(\mathcal{V}, \mathcal{V}_2) \Big]$

NPSPACE-hardness of QBF

Define $\textsc{Path}_i(\mathcal{V}_1, \mathcal{V}_2)$:

M starting on config$(\mathcal{V}_1, \beta)$ can reach config$(\mathcal{V}_2, \beta)$ in $\leq 2^i$ steps.

For $i = 0$: $\quad \textsc{Path}_0 \quad := \quad \mathcal{V}_1 = \mathcal{V}_2 \quad \vee \quad \textsc{Next}(\mathcal{V}_1, \mathcal{V}_2)$

For $i \to i+1$:

Idea: $\textsc{Path}_{i+1}(\mathcal{V}_1, \mathcal{V}_2) := \exists \mathcal{V} \Big[\textsc{Conf}(\mathcal{V}) \wedge \textsc{Path}_i(\mathcal{V}_1, \mathcal{V}) \wedge \textsc{Path}_i(\mathcal{V}, \mathcal{V}_2) \Big]$

Problem: $|\textsc{Path}_i| = \mathcal{O}(2^i)$ (Reduction would use exp. time/space)

NPSPACE-hardness of QBF

Define $\textsc{Path}_i(\mathcal{V}_1, \mathcal{V}_2)$:

M starting on config$(\mathcal{V}_1, \beta)$ can reach config$(\mathcal{V}_2, \beta)$ in $\leq 2^i$ steps.

For $i = 0$***:*** $\quad \textsc{Path}_0 \quad := \quad \mathcal{V}_1 = \mathcal{V}_2 \quad \vee \quad \textsc{Next}(\mathcal{V}_1, \mathcal{V}_2)$

For $i \to i + 1$***:***

Idea: $\textsc{Path}_{i+1}(\mathcal{V}_1, \mathcal{V}_2) := \exists \mathcal{V} \Big[\textsc{Conf}(\mathcal{V}) \wedge \textsc{Path}_i(\mathcal{V}_1, \mathcal{V}) \wedge \textsc{Path}_i(\mathcal{V}, \mathcal{V}_2) \Big]$

Problem: $|\textsc{Path}_i| = \mathcal{O}(2^i)$ (Reduction would use exp. time/space)

New Idea:

$$\textsc{Path}_{i+1}(\mathcal{V}_1, \mathcal{V}_2) := \exists \mathcal{V}\ \textsc{Conf}(\mathcal{V}) \wedge \forall \mathcal{Z}_1 \forall \mathcal{Z}_2 \Big(\Big(\begin{matrix} (\mathcal{Z}_1 = \mathcal{V}_1 \wedge \mathcal{Z}_2 = \mathcal{V}) \\ (\mathcal{Z}_1 = \mathcal{V} \wedge \mathcal{Z}_2 = \mathcal{V}_2) \end{matrix} \vee \Big) \to \textsc{Path}_i(\mathcal{Z}_1, \mathcal{Z}_2) \Big)$$

NPSPACE-hardness of QBF

Define $\text{PATH}_i(\mathcal{V}_1, \mathcal{V}_2)$:

M starting on config$(\mathcal{V}_1, \beta)$ can reach config$(\mathcal{V}_2, \beta)$ in $\leq 2^i$ steps.

For $i = 0$**:** $\quad \text{PATH}_0 \quad := \quad \mathcal{V}_1 = \mathcal{V}_2 \quad \vee \quad \text{NEXT}(\mathcal{V}_1, \mathcal{V}_2)$

For $i \to i + 1$**:**

Idea: $\text{PATH}_{i+1}(\mathcal{V}_1, \mathcal{V}_2) := \exists \mathcal{V} \Big[\text{CONF}(\mathcal{V}) \wedge \text{PATH}_i(\mathcal{V}_1, \mathcal{V}) \wedge \text{PATH}_i(\mathcal{V}, \mathcal{V}_2) \Big]$

Problem: $|\text{PATH}_i| = \mathcal{O}(2^i)$ (Reduction would use exp. time/space)

New Idea:

$$\begin{aligned} \text{PATH}_{i+1}(\mathcal{V}_1, \mathcal{V}_2) := \quad & \exists \mathcal{V}\ \text{CONF}(\mathcal{V}) \wedge \\ & \forall \mathcal{Z}_1 \forall \mathcal{Z}_2 \Big(\Big(\begin{matrix} (\mathcal{Z}_1 = \mathcal{V}_1 \wedge \mathcal{Z}_2 = \mathcal{V}) \\ (\mathcal{Z}_1 = \mathcal{V} \wedge \mathcal{Z}_2 = \mathcal{V}_2) \end{matrix} \ \vee \Big) \to \text{PATH}_i(\mathcal{Z}_1, \mathcal{Z}_2) \Big) \end{aligned}$$

Lemma

For any assignment β defined on $\mathcal{V}_1, \mathcal{V}_2$: If β satisfies $\text{PATH}_i(\mathcal{V}_1, \mathcal{V}_2)$, then config$(\mathcal{V}_2, \beta)$ is reachable from config$(\mathcal{V}_1, \beta)$ in $\leq 2^i$ steps.

NPSPACE-hardness of QBF

$\mathbf{Path}_i(\mathcal{V}_1, \mathcal{V}_2)$:

M starting on config$(\mathcal{V}_1, \beta)$ can reach config$(\mathcal{V}_2, \beta)$ in $\leq 2^i$ steps.

Start and end configuration:

$$\textsc{Start}(\mathcal{V}) \quad := \quad \textsc{Conf}(\mathcal{V}) \wedge X_{q_0} \;\wedge Y_0 \;\wedge \bigwedge_{i=0}^{n-1} Z_{w_i,i} \wedge \bigwedge_{i=n}^{p(n)} Z_{\Box,i}$$

$$\textsc{End}(\mathcal{V}) \quad := \quad \textsc{Conf}(\mathcal{V}) \;\wedge\; \bigvee_{q \in F_a} X_q$$

Lemma

Let C_{start} be starting configuration of M on input w.

1. *β satisfies $\textsc{Start}$ if, and only if, config$(\mathcal{V}, \beta) = C_{start}$*
2. *β satisfies $\textsc{End}$ if, and only if, config$(\mathcal{V}, \beta)$ is an accepting stop configuration. (not nec reachable from C_{start})*

NPSPACE-hardness of QBF

$\mathbf{Path}_i(\mathcal{V}_1, \mathcal{V}_2)$:

M starting on config$(\mathcal{V}_1, \beta)$ can reach config$(\mathcal{V}_2, \beta)$ in $\leq 2^i$ steps.

Start and end configuration:

$$\textsc{Start}(\mathcal{V}) \;:=\; \textsc{Conf}(\mathcal{V}) \wedge X_{q_0} \;\wedge Y_0 \;\wedge \bigwedge_{i=0}^{n-1} Z_{w_i,i} \wedge \bigwedge_{i=n}^{p(n)} Z_{\Box,i}$$

$$\textsc{End}(\mathcal{V}) \;:=\; \textsc{Conf}(\mathcal{V}) \;\wedge\; \bigvee_{q \in F_a} X_q$$

Lemma

Let C_{start} be starting configuration of M on input w.

1. *β satisfies* $\textsc{Start}$ *if, and only if, config$(\mathcal{V}, \beta) = C_{start}$*
2. *β satisfies* $\textsc{End}$ *if, and only if, config$(\mathcal{V}, \beta)$ is an accepting stop configuration.* *(not nec reachable from C_{start})*

Putting it all together: M accepts w if, and only if,

$$\varphi_{M,w} := \exists \mathcal{V}_1 \, \exists \mathcal{V}_2 \; \textsc{Start}(\mathcal{V}_1) \wedge \textsc{End}(\mathcal{V}_2) \wedge \textsc{Path}_{p(n)}(\mathcal{V}_1, \mathcal{V}_2)$$ is true.

NPSPACE-hardness of QBF (to conclude)

Theorem

QBF *is* NPSPACE-*hard.*

Proof. Let $\mathcal{L} \in$ NPSPACE, we show $\mathcal{L} \leq_p \mathrm{QBF}$.

Let $M := (Q, \Sigma, q_0, \Delta, F_a, F_r)$ be a TM deciding $\mathcal{L}$. M never uses more than $p(n)$ cells.

For each input $w \in \Sigma^*$, $|w| = n$, we construct (in poly time!) a formula $\varphi_{M,w}$ such that

M accepts w if, and only if, $\varphi_{M,w}$ is true.

Glossed over some detail: $\varphi_{M,w}$ is not in prenex form, can be manipulated into that. Also, quantifiers don't alternate $\forall/\exists/\forall/\exists\ldots$; that also can be fixed...

To conclude

We have a “natural” PSPACE-complete problem

“natural” (slightly vague definition): the problem does not arise in the study of PSPACE, it has separate interest.

obvious analogy with SAT being complete for NP

Next: how to use this to prove various other problems are also PSPACE-complete.