COMP3630/6360: Theory of Computation
Semester 1, 2022
The Australian National University

Space Complexity

This lecture covers Chapter 11 of HMU: Other Complexity Classes

- PSPACE completeness
- Quantified Boolean Formulae
- QBF is PSPACE complete

Additional Reading: Chapter 11 of HMU.

PSPACE completeness

Definition 10.1

A problem L is *PSPACE hard* if there is a polytime reduction from any PSPACE problem to L.

A problem L is *PSPACE complete*, if it is PSPACE hard and in PSPACE.

Q. Why polytime, and not polyspace reductions?

Observation.

Let L be a PSPACE complete problem.

1. If $L \in P$, then P = PSPACE.
2. if $L \in NP$, then NP = PSPACE.

Quantified Boolean Formulae

Definition 10.2

If V is a set of variables, then the set of *quantified boolean formulae* over V is given by:

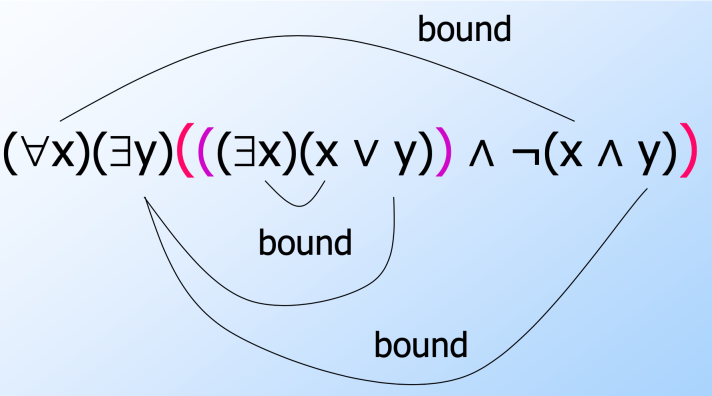
- Every variable $v \in V$ is a QBF, and so are *tt* and *ff*
- If ϕ, ψ are QBF, then so are $\phi \wedge \psi$ and $\phi \vee \psi$
- If ϕ is a QBF, then so is $\neg\phi$.
- If ϕ is a QBF and $v \in V$ is a variable, then $\exists v \phi$ and $\forall v \psi$ are QBF.

Definition 10.3

In a QBF ϕ, an occurrence variable v is *bound* if it is in the scope of a quantifier $\forall v$ or $\exists v$. The variable v is *free* otherwise.

If $x \in \{tt, ff\}$ is a truth value, then $\phi[x/v]$ is the result of replacing all *free* occurrences of v with x.

Example



Evaluation of QBFs

Observation.

A QBF ϕ without free variables can be evaluated to a truth value:

- $\text{eval}(\forall v \phi) = \phi[tt/x] \wedge \phi[ff/x]$
- $\text{eval}(\exists v \phi) = \phi[tt/x] \vee \phi[ff/x]$

and quantifier-free formulae without free variables can be evaluated.

QBFs versus boolean formulae.

- a boolean formula ϕ in variables $v_1, \ldots, v_n$ is satisfiable if $\exists v_1 \exists v_2 \ldots \exists v_n \phi$ evaluates to true.
- ϕ is a tautology if $\forall v_1 \forall v_2 \ldots \forall v_n \phi$ evaluates to true.

Definition 10.4

The QBF problem is the problem of determining whether a given quantified boolean formula without free variables evaluates to true:

$$\text{QBF} = \{\phi \mid \phi \text{ a true QBF without free variables}\}$$

QBFs vs Boolean Formulae

- evaluating a boolean formula without free variables is in P.
- $(\forall v \phi) \rightsquigarrow \phi[\mathit{tt}/x] \wedge \phi[\mathit{ff}/x]$
- $(\exists v \phi) \rightsquigarrow \phi[\mathit{tt}/x] \vee \phi[\mathit{ff}/x]$
- the resulting formula may be exponentially large
- but this shows that QBF is in EXPTIME.

Q. Can we do better?

QBF is in PSPACE

Main Idea.

- to evaluate $\forall v \phi$, *don't* write out $\phi[tt/v] \wedge \phi[ff/v]$.
- instead, evaluate $\phi[tt/v]$ and $\phi[ff/v]$ in sequence.
- avoids exponential space blowup

```
Algorithm evalqbf (phi) = case phi of
- tt: return tt
- phi /\ psi: if evalqbf(phi) then evalqbf(psi) else false
- forall v phi: if evalqbf(phi[tt/v]) then evalqbf (phi[ff/v]) else false
- (other cases analogous)
```

Analysis.

- Given QBF ϕ of size n:
- at most n recursive calls active
- each call stores a partially evaluated QBF of size n
- total space requirement $\mathcal{O}(n^2)$

QBF is PSPACE-complete

Proof IdeaNote.

Let L be in PSPACE.

- Then L is accepted by a polyspace bounded TM with bound $p(n)$
- If $w \in L$, then M accepts in $\leq c^{p(n)}$ moves
- construct QBF ϕ: ‘there is a sequence of $c^{p(n)}$ ID’s that accepts w
- use recursive doubling to express this in polytime.

The Gory Detail

Variables.

- Need $\mathcal{O}(p(n))$ variables to represent ID:
- $y_{j,A} = tt$ iff the j-th symbol of the ID is A, $1 \leq j \leq p(n) + 1$ tuples.

Structure of the QBF.

$$\phi = (\exists I_0)(\exists I_f) S \wedge N \wedge F \wedge U$$

- I_0 and I_f are initial / accepting IDs
- S says that $I_0 = q_0 w$
- F says that I_f is accepting
- U says that every ID has at most one symbol per position
- N says that there is a sequence of ID's of length $\leq c^{p(n)}$ from I_0 to I_f.
- S, F, and U are as in Cook's theorem.

Recursive Doubling

- $N = N(I_0, I_f)$: have sequence of length $\leq c^{p(n)}$ from I_9 to I_f.
- Detour: $N_0(I, J) = I \vdash^* J$ in ≤ 1 steps: as for Cook's theorem
- Detour: $N_i(I, J) = I \vdash^* J$ in $\leq 2^i$ steps:

$$N_i(I, J) = (\exists K)(\forall P)(\forall Q)[(P, Q) = (I, K) \vee (P, Q) = (K, J) \rightarrow N_{i-1}(P, Q)]$$

  - Could also say $(\exists K)(N_{i-1})(I, K) \wedge N_{i-1}(K, J))$
  - this would write out N_{i-1} twice, doubling formula size at each step
  - above trick is key step in proof to keep formula size small
- Let $N(I_0, I_f) = N_k(I_0, I_f)$ where $2^k \geq c^{p(n)}$ (note $k \in \mathcal{O}(p(n))$)
- each N_i can be written in $\mathcal{O}(p(n))$ many steps, plus the time to write N_{i-1}
- so $\mathcal{O}(p(n)^2)$ overall

By construction, $\phi = tt$ iff M accepts w.