More about PSPACE

with respect to $\leq_m^P$

Thm: QBF is PSPACE-complete.

the language of all true quantified Boolean formulas (all variables bound by quantifiers)

Proof: (1) QBF $\in$ PSPACE by the following recursive algorithm:

- $QBF(\forall x\, \varphi(x)) = QBF(\varphi(0))\ \&\ QBF(\varphi(1))$
- $QBF(\exists x\, \varphi(x)) = QBF(\varphi(0)) \vee QBF(\varphi(1))$
- $QBF(\varphi \vee \psi) = QBF(\varphi) \vee QBF(\psi)$
- $QBF(\varphi\ \&\ \psi) = QBF(\varphi)\ \&\ QBF(\psi)$
- $QBF(\neg\varphi) = \neg QBF(\varphi)$

$O(n)$ levels of recursion, $O(n)$ space per level. $O(n^2)$ space

(2) QBF is PSPACE-hard: consider $L \in$ PSPACE, TM M deciding L and its config. graph G.

- vectors of variables $\bar{x}$ encoding vertices ... $O(poly(n))$ bits [log of $|G|$]
- formula $\varphi(\bar{x},\bar{y}) \equiv (\bar{x}=\bar{y}) \vee (x,y) \in E(G)$
  - can construct poly-sized circuit as in proof of Cook-Levin thm. & then reduce the circuit to an existentially-quantified formula as in Circuit-SAT $\leq_m^P$ SAT.
- Mimic proof of Savitch's thm.

Failed attempt: $\varphi_k(\bar{x},\bar{y}) \equiv \exists \bar{z}\,(\varphi_{k-1}(\bar{x},\bar{z})\ \&\ \varphi_{k-1}(\bar{z},\bar{y}))$

Double recursion $\Rightarrow$ formula size grows exponentially!

Better: $\varphi_k(\bar{x},\bar{y}) \equiv \exists \bar{z}\ \forall \bar{a}\ \forall \bar{b}\,\big(((\bar{a}=\bar{x}\ \&\ \bar{b}=\bar{z}) \vee (\bar{a}=\bar{z}\ \&\ \bar{b}=\bar{y})) \Rightarrow \varphi_{k-1}(\bar{a},\bar{b})\big)$

- G has size $O(2^{poly(n)}) \Rightarrow \log(\text{path len}) \in poly(n) \Rightarrow$ recursion has $poly(n)$ levels, formula size grows to $poly(n)$.

Intuition: PSPACE is the class of strategies for 2-player games with perfect information:

($\exists$ player 1 opening move) ($\forall$ player 2's ~~resp~~ response) ($\exists$ player 1's counter-response) ($\forall$ ...) ...

Examples: these are known to be PSPACE-complete

- graph coloring game: undirected graph, finite set of k colors each player colors an uncolored vertex, ~~every~~ no edge must have both ends of the same color
- graph path game: building path edge by edge, ~~first each~~ ~~player has his target,~~ ~~2nd player~~ vertices must not repeat
- variants of Go, Checkers &c. (generalized to $N \times N$ boards)
- Sokoban (1-player, but enough internal state, which restricts future moves, but can be modified)

Alternating Turing Machine (ATM)

3 kinds of states:
- deterministic: config is accepting $\Leftrightarrow$ next config is accepting
- existential: config is accepting ~~accepts~~ $\Leftrightarrow$ $\exists$ non-det. choice ~~leading s.t. the rest~~ leading to accepting config.
- universal: is accepting $\Leftrightarrow$ $\forall$ non.-det. choice leads to accepting config.

We will require all computations to halt.

ATM → classes ATIME(f), AP ... [we won't define space-bounded classes as we require all computations to halt & alarm clocks don't help]

<u>Theorem:</u> AP = PSPACE.

<u>Proof:</u> $\supseteq$ is easy: QBF $\in$ AP since we can execute quantifiers using corresponding state types.
So if $L \leq_m^p$ QBF, we can first compute the reduction and then solve QBF.

$\subseteq$: simulate the ATM recursively as in QBF $\in$ PSPACE.

- configurations take $O(poly(n))$ space – all computations are poly-time, so they are poly-space, too.
- recursion depth is bounded by time of the ATM.

Polynomial Hierarchy

Consider following restrictions of QBF:

- Σ_k-formulas: $\underbrace{(\exists x_1 \dots \exists x_k)(\forall \dots)(\exists \dots) \dots}_{k \text{ groups of quantifiers, starting with } \exists} \underbrace{\varphi(-)}_{\text{quantifier-free formula}}$
- Π_k-formulas: similar, but starting with $(\forall \dots)$

negation of a Σ_k-formula can be written as a Π_k-formula & vice versa

- Σ_k-SAT := $\{ \underset{\substack{\uparrow \\ \text{encoding of}}}{\langle \varphi \rangle} \mid \varphi \text{ is a true } \Sigma_k\text{-formula}\}$ ... similarly Π_k-SAT.
- Σ_1-SAT is SAT (for general formulas, not only CNF), Π_1-SAT is TAUT.

We can use this to define new classes which generalize NP:

- $\Sigma_k^P := \{ L \mid L \leq_m^P \Sigma_k\text{-SAT}\}$ ... $\Sigma_1^P = NP$, $\Sigma_0^P = P$
- $\Pi_k^P := \{ L \mid L \leq_m^P \Pi_k\text{-SAT}\}$ ... $\Pi_1^P = \text{co-}NP$, $\Pi_0^P = P$, $\Pi_k^P = \text{co-}\Sigma_k^P$

we defined classes using a problem complete for them

- generally, the following inclusions hold:

$$\begin{tikzcd}
 & \Sigma_1^P = NP \arrow[r] \arrow[dr] & \Sigma_2^P \arrow[r] \arrow[dr] & \Sigma_3^P \arrow[r] \arrow[dr] & \dots \\
\Sigma_0^P = \Pi_0^P = P \arrow[ur] \arrow[dr] & & & & \\
 & \Pi_1^P = \text{co-}NP \arrow[r] \arrow[ur] & \Pi_2^P \arrow[r] \arrow[ur] & \Pi_3^P \arrow[r] \arrow[ur] & \dots
\end{tikzcd}$$

This is akin to the arithmetical hierarchy, but the inclusions are not known to be strict.

- $PH := \bigcup_k \Sigma_k^P = \bigcup_k \Pi_k^P$
- since every Σ_k/Π_k-SAT reduces trivially to QBF, we have PH $\subseteq$ PSPACE (not known to be strict)

<u>Example:</u> "given Bool. formula φ, find the shortest ψ s.t. $\forall \bar{x}\ \varphi(\bar{x}) \Leftrightarrow \psi(\bar{x})$" $\in \Sigma_2^P$
(in fact, it's Σ_2^P-complete)

<u>Remark:</u> Definition using oracle machines also works:

$\Sigma_{k+1}^P := NP[\Sigma_k^P] = NP[\Pi_k^P]$

$\Pi_{k+1}^P := \text{co-}NP[\Sigma_k^P] = \text{co-}NP[\Pi_k^P]$

we can add $\Delta_{k+1}^P := P[\Sigma_k^P] = P[\Pi_k^P]$ ...

$$\begin{tikzcd}
\Sigma_k \arrow[dr] \arrow[rr, bend left] & & \Sigma_{k+1} \\
 & \Delta_{k+1} \arrow[ur] \arrow[dr] & \\
\Pi_k \arrow[ur] \arrow[rr, bend right] & & \Pi_{k+1}
\end{tikzcd}$$

this leads to the same hierarchy

Remark: We can also define Σ_k^P & Π_k^P using alternative TMs:

- Σ_k-TIME$(f) := \{ L \mid L$ can be decided by an ATM running in time $\leq f(|\text{input}|)$ which performs at most $k-1$ quantifier changes, starting with $\exists \}$
- $\Sigma_k^P = \Sigma_k$-TIME(poly(n))

Collapse of PH

- $P = NP \iff P = PH$
- $NP = \text{co-}NP \iff NP = PH$
- if $\Sigma_j^P = \Sigma_{j+1}^P$, then $\Sigma_j^P = \Sigma_k^P$ for all ~~...~~ $k > j$ ... so $PH = \Sigma_j^P$] we say that PH collapsed to the j-th level
- if $\Sigma_j^P = \Pi_j^P$, then $\Sigma_{j+1}^P = \Sigma_j^P$ (we can reduce $\exists \bar{x} \forall \bar{y}\, \varphi(\dots)$ to $\exists \bar{x} \exists \bar{y}\, \psi(\dots)$)

this is weaker than $P = NP$, but still open

Note: If graph isomorphism is in P, then $PH = \Sigma_2^P$. [proof non-trivial]

SPACE CO-CLASSES

Unlike non-deterministic time classes, non-det. space classes are known to be closed under complement.

Theorem (Immerman–Szelepcsényi): $NSPACE(s(n)) = \text{co-}NSPACE(s(n))$ for all space-constructible functions $s(n) \geq \log n$.

Proof: We design a non-deterministic algorithm for non-reachability in config. graphs. More generally, we'll calculate $R_i :=$ #vertices reachable from source by walk of len $\leq i$.

$V_i :=$ set of these vertices

Then modify graph by adding edges from target to all vertices, so (~~...~~ tgt reachable from src) $\iff R_n = n$ for $n = $#vertices.

- $R_0 = 1$
- $R_{i-1} \to R_i$: For all $v \in V$:
  - For all $w \in V_{i-1}$:
    - if $(w,v) \in E$ or $v = w$: $R_i \leftarrow R_i + 1$
- Enumeration of V_i:

```
t <- 0
For all u ∈ V:
    If guess that u ∈ V_i:
        If ∄ walk src → u of length ≤ i: REJECT
        t <- t+1
If t ≠ R_{i-1}: REJECT
```

if we don't guess correctly, either the path doesn't exist or $t < R_{i-1}$ at the end

guess the path using non-determinism & check that it's valid

Space needed:
- $O(1)$ variables for vertices & counters
- R_i and R_{i-1}

} $O(\log |V|)$ space, where $|V| = 2^{O(s(n))}$, so this is in $NSPACE(s(n))$.