Theorem 9.1 $\overline{\text{REACH}} \in \text{NL}$

Proof: Fix G, let $N_d = \left|\left\{v \mid \text{distance}(s,v) \le d\right\}\right|$

Claim: The following problems are in NL:

1. $\text{DIST}(x,d)$: distance$(s,x) \le d$
2. $\text{NDIST}(x,d;m)$: if $m = N_d$ then $\neg\text{DIST}(x,d)$

Proof:

1. Guess the path of length $\le d$ from s to x.
2. Guess m vertices, $v \ne x$, with $\text{DIST}(v,d)$.

```
c := 0;
for v := 1 to n do {    // nondeterministically
    ( DIST(v, d) && v ≠ x; c++ )    ||
    ( no-op )
}
if (c == m) then ACCEPT
```

□

Claim: We can compute N_d in NL.

Proof: By induction on d.

Base case: $N_0 = 1$

Inductive step: Suppose we have N_d.

1. $c := 0;$
2. **for** $v := 1$ to n **do** { // nondeterministically
3. ($\mathrm{DIST}(v, d+1); c++$) ||
4. $(\forall z\,(\mathrm{NDIST}(z, d; N_d) \;\vee\; (z \neq v \wedge \neg E(z, v))))$
5. }
6. $N_{d+1} := c$

□

$$G \in \overline{\mathrm{REACH}} \quad \Leftrightarrow \quad \mathrm{NDIST}(t, n; N_n)$$

□

Theorem 9.2 [Immerman-Szelepcsényi] *If* $s(n) \geq \log n$, *Then,* NSPACE$s(n)$ = *co*-NSPACE$s(n)$

Proof: Let $A \in$ NSPACE$s(n)$; $A = \mathcal{L}(N)$

$$w \in A \quad \Leftrightarrow \quad \mathrm{CompGraph}(N, w) \in \mathrm{REACH}$$

$$|w| = n; \qquad |\mathrm{CompGraph}(N, w)| = 2^{O(s(n))}$$

Testing if CompGraph$(N, w) \in \overline{\mathrm{REACH}}$ takes space,

$$\begin{aligned} \log(|\mathrm{CompGraph}(N, w)|) &= \log(2^{O(s(n))}) \\ &= O(s(n)) \end{aligned}$$

□

PSPACE

$$\text{PSPACE} \quad = \quad \text{DSPACE}[n^{O(1)}] \quad = \quad \text{NSPACE}[n^{O(1)}]$$

- PSPACE consists of what we could compute with a feasible amount of hardware, but with no time limit.
- PSPACE is a large and very robust complexity class.
- With polynomially many bits of memory, we can search any implicitly-defined graph of exponential size. This leads to complete problems such as reachability on exponentially-large graphs.
- We can search the game tree of any board game whose configurations are describable with polynomially-many bits and which lasts at most polynomially many moves. This leads to complete problems concerning winning strategies.

PSPACE-Complete Problems

Def: The **quantified satisfiability problem** (QSAT) is the set of true formulas of the following form:

$$\Psi \quad = \quad Q_1 x_1\, Q_2 x_2 \cdots Q_r x_r\, (\varphi)$$

For any boolean formula φ on variables $\overline{x}$,

$$\varphi \in \text{SAT} \quad \Leftrightarrow \quad \exists \overline{x}\,(\varphi) \in \text{QSAT}$$

$$\varphi \notin \text{SAT} \quad \Leftrightarrow \quad \forall \overline{x}\,(\neg\varphi) \in \text{QSAT}$$

Thus QSAT logically contains SAT and $\overline{\text{SAT}}$.

Fact 9.3 QSAT *is* PSPACE*-complete.*

Proof: QSAT is in PSPACE because we can use n bits of space to systematically search the 2^n possible assignments to the variables of the formula φ to check if it is true. For example, if $\varphi = \exists x_1 \psi(x_1)$, then to check if φ is true, we first check if $\psi(0)$ is true. If so, the answer is "yes"; else we reuse the same space to accept iff $\psi(1)$ is true. Thus each new variable only requires one more bit of space.

To show that QSAT is PSPACE hard, we reduce SUCCINCT REACH to QSAT. The main point is that given a succinct representation of a graph, G, we must write the quantified boolean formula $\varphi(G)$ which will be true iff there is a path in G of length at most 2^n from s to t.

Inductively, we write the formula $P_d(x, y)$ meaning that there is a path from x to y of length at most d, inductively as follows:

$$P_d(x, y) \quad \equiv \quad \exists z (P_{d/2}(x, z) \wedge P_{d/2}(z, y)) \;.$$

Note that x, y, and z are written with n bits each. Next, using a universal boolean quantifier, we shorten the formula by writing $P_{d/2}$ only once:

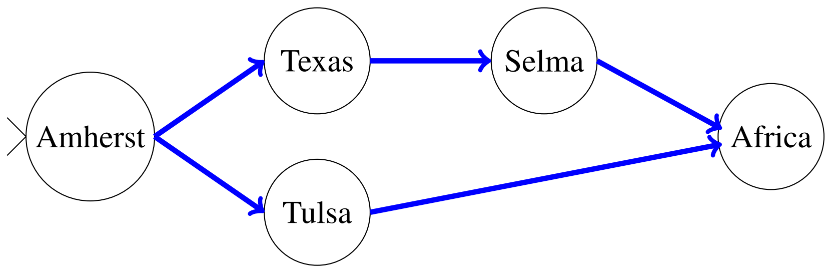
$$P_d(x, y) \quad \equiv \quad \exists z\, \forall b\, (\exists x', y'.((b \wedge x' = x \wedge y' = z) \vee (\neg b \wedge x' = z \wedge y' = y)))\, (P_{d/2}(x', y'))$$

For the base case, $d = 1$, we have to show that a boolean circuit can be evaluated by a formula in QSAT. This formula simply guesses the values of all the gates of the circuit and asserts that they are evaluated correctly and the root is true, i.e., evaluates to 1. □

Geography is a two-person game.

1. E "chooses" the start vertex v_1.
2. A chooses v_2, having an edge from v_1
3. E chooses v_3, having an edge from v_2, etc.

No vertex may be chosen twice. Whoever moves last wins.



Let GEOGRAPHY be the set of positions in geography games s.t. $\exists$ has a winning strategy.

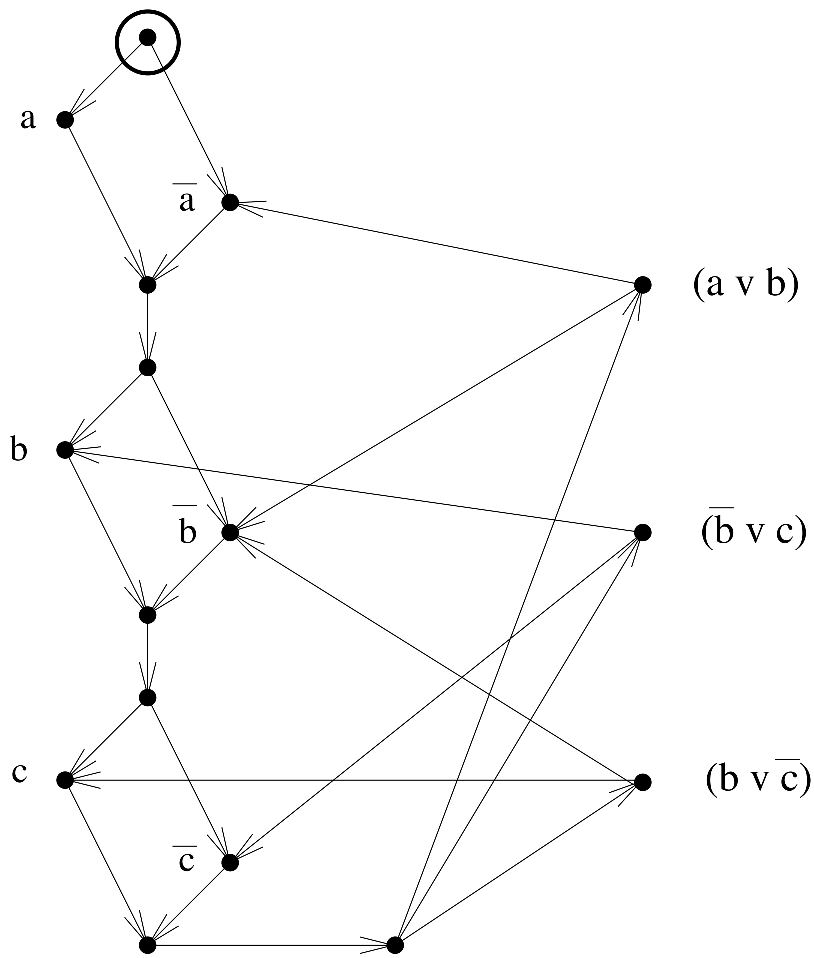
Proposition 9.4 GEOGRAPHY *is* PSPACE-*complete.*

Proof: GEOGRAPHY $\in$ PSPACE: search the polynomial-depth game tree. A polynomial-size stack suffices.

Show: QSAT $\leq$ GEOGRAPHY

Given formula, φ, build graph G_φ s.t. $\exists$ chooses existential variables; $\forall$ chooses universal variables.

$$\varphi \equiv \exists a \forall b \exists c$$
$$[(a \vee b) \wedge$$
$$(\bar{b} \vee c) \wedge$$
$$(b \vee \bar{c})]$$



□

Definition 9.5 A **succinct** representation of a graph is $G(n, C, s, t) = (V, E, s, t)$ where C is a boolean circuit with $2n$ inputs and

$$V = \{w \mid w \in \{0,1\}^n\}$$
$$E = \{(w, w') \mid C(w, w') = 1\}$$

□

$$\text{SUCCINCT REACH} = \{(n, C, s, t) \mid G(n, C, s, t) \in \text{REACH}\}$$

Proposition 9.6 SUCCINCT REACH $\in$ PSPACE

Why?

Remember Savitch's Thm:

$$\text{REACH} \in \text{NSPACE}[\log n] \subseteq \text{DSPACE}[(\log n)^2]$$

$$\text{SUCCINCT REACH} \in \text{NSPACE}[n] \subseteq \text{DSPACE}[n^2] \subseteq \text{PSPACE}$$

□

Proposition 9.7 SUCCINCT REACH *is* PSPACE-*complete.*

Proof: Let M be a DSPACE$[n^k]$ TM, input w, $n = |w|$

$$M(w) = 1 \quad \leftrightarrow \quad \text{CompGraph}(M, w) \in \text{REACH}$$

$$\text{CompGraph}(n, w) = (V, E, s, t)$$

$$V = \{\text{ID} = \langle q, h, p \rangle \mid q \in \text{States}(N), h \leq n, |p| \leq c\,n^k\}$$
$$E = \{(\text{ID}_1, \text{ID}_2) \mid \text{ID}_1(w) \xrightarrow[M]{} \text{ID}_2(w)\}$$
$$s = \text{initial ID}$$
$$t = \text{accepting ID}$$

□

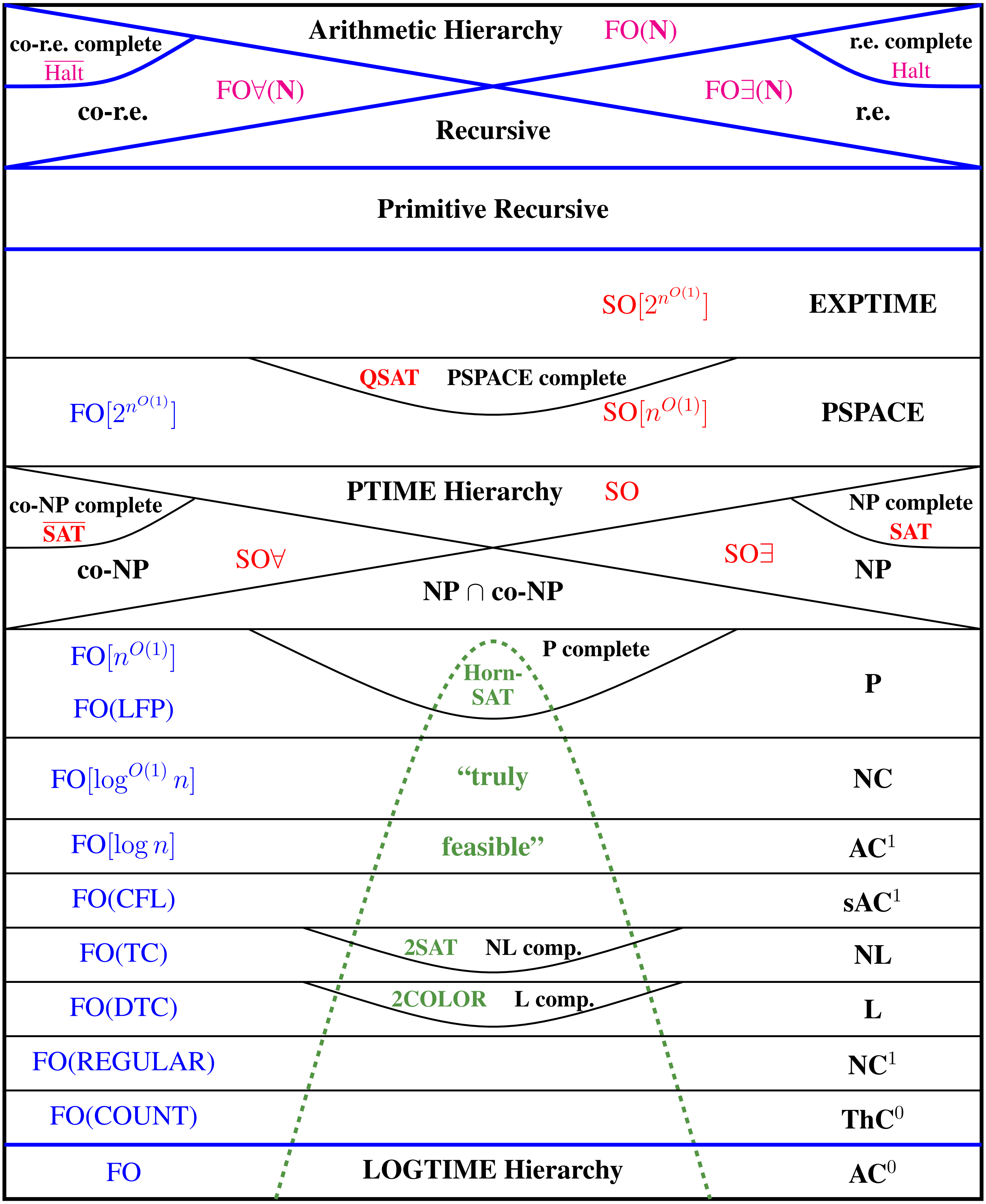
Succinct Representation of CompGraph(n, w):

$$
\begin{aligned}
V &= \left\{\text{ID} = \langle q, h, p\rangle \;\middle|\; q \in \text{States}(N), h \le n, |p| \le c\,n^k\right\} \\
E &= \left\{(\text{ID}_1, \text{ID}_2) \;\middle|\; \text{ID}_1(w) \underset{M}{\longrightarrow} \text{ID}_2(w)\right\}
\end{aligned}
$$

Let $V = \{0,1\}^{c'n^k}$

Build circuit C_w: on input $u, v \in V$, accept iff $u \underset{M}{\longrightarrow} v$.

$$M(w) = 1 \quad \Leftrightarrow \quad G(c'n^k, C_w, s, t) \in \text{SUCCINCT REACH} \qquad \square$$


Arithmetic Hierarchy
FO(N)
co-r.e. complete
$\overline{\text{Halt}}$
r.e. complete
Halt
FO∀(N)
FO∃(N)
co-r.e.
r.e.
Recursive
Primitive Recursive
$SO[2^{n^{O(1)}}]$
EXPTIME
QSAT
PSPACE complete
$FO[2^{n^{O(1)}}]$
$SO[n^{O(1)}]$
PSPACE
PTIME Hierarchy
SO
co-NP complete
$\overline{\text{SAT}}$
NP complete
SAT
co-NP
SO∀
SO∃
NP
NP ∩ co-NP
$FO[n^{O(1)}]$
P complete
Horn-SAT
FO(LFP)
P
$FO[\log^{O(1)} n]$
"truly
NC
$FO[\log n]$
feasible"
AC^1
FO(CFL)
sAC^1
FO(TC)
2SAT
NL comp.
NL
FO(DTC)
2COLOR
L comp.
L
FO(REGULAR)
NC^1
FO(COUNT)
ThC^0
FO
LOGTIME Hierarchy
AC^0