Theorem 9.1 \( \text{REACH} \in \text{NL} \)

**Proof:** Fix \( G \), let \( N_d = \{|v| \text{ distance}(s, v) \leq d|\} \)

**Claim:** The following problems are in \( \text{NL} \):

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

**Proof:**

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

\[
c := 0;
\text{for } v := 1 \text{ to } n \text{ do } \{ \quad // \text{nondeterministically} \\
(\text{DIST}(v, d) \&\& v \neq x; c++) \quad || \\
(\text{no-op})
\}
\]

if \( c == m \) then ACCEPT

\[\square\]
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. ( DIST($v, d + 1$); $c +$ ) ||
4. ( $\forall z$ (NDIST($z, d; N_d$) $\vee$ ($z \neq v \land \neg E(z, v)$)))
5. }
6. $N_{d+1} := c$

$G \in \text{REACH} \iff \text{NDIST}(t, n; N_n)$

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

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

$$w \in A \iff \text{CompGraph}(N, w) \in \text{REACH}$$

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

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

$$\log(|\text{CompGraph}(N, w)|) = \log(2^{O(s(n))}) = O(s(n))$$
PSPACE

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

- 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:

\[ \Psi = Q_1 x_1 Q_2 x_2 \cdots Q_r x_r (\varphi) \]

For any boolean formula \( \varphi \) on variables \( \bar{x} \),

\[
\begin{align*}
\varphi \in \text{SAT} & \iff \exists \bar{x} (\varphi) \in \text{QSAT} \\
\varphi \not\in \text{SAT} & \iff \forall \bar{x} (\neg \varphi) \in \text{QSAT}
\end{align*}
\]

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

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

We will prove this later.
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, $\varphi$, build graph $G_\varphi$ s.t. $\exists$ chooses existential variables; $\forall$ chooses universal variables.

$$
\varphi \equiv \exists a \forall b \exists c \\
[(a \lor b) \land \\
(b \lor c) \land \\
(b \lor \overline{c})]
$$
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
\[
V = \{ w \mid w \in \{0, 1\}^n \} \\
E = \{ (w, w') \mid C(w, w') = 1 \}
\]

\[ \square \]

**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:

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

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

\[ \square \]

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

**Proof:** Let \( M \) be a **DSPACE**[\( n^k \)] TM, input \( w \), \( n = |w| \)

\[ M(w) = 1 \iff \text{CompGraph}(M, w) \in \text{REACH} \]

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

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

\[ \square \]
Succinct Representation of \text{CompGraph}(n, w):

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

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

Build circuit \( C_w \): on input \( u, v \in V \), accept iff \( u \xrightarrow{M} v \).

\[ M(w) = 1 \iff G(c'n^k, C_w, s, t) \in \text{SUCCINCT REACH} \]
Arithmetic Hierarchy

Recursive

Halt

r.e. complete

Arithmetic Hierarchy

FO(N)

r.e.

co-r.e.

Halt

FO\forall(N)

Recursive

FO\exists(N)

co-r.e. complete

Recursive

Halt

Halt

CO-r.e.

Recursive

Halt

FO

FO

\exists

FO

\forall

FO

SO

\[2^{n^{O(1)}}\]

EXPTIME

QSAT

PSPACE complete

SO

\[n^{O(1)}\]

PSPACE

PTIME Hierarchy

SO

NP complete

SAT

co-NP complete

SAT

PTIME Hierarchy

SO

NP

co-NP

NP \cap co-NP

NP

NP \cap co-NP

NP

FO\[n^{O(1)}\]

Horn-SAT

P complete

P

FO\[n^{O(1)}\]

“truly feasible”

AC^1

FO\[\log^{O(1)} n\]

NC

FO\[\log n\]

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