Finite Model Theory / Descriptive Complexity:

Theorem: $\mathrm{FO} \;\subseteq\; \mathbf{L} \;=\; \mathbf{DSPACE}[\log n]$

Fagin's Theorem: $\mathbf{NP} = \mathrm{SO}\exists$.

$$\mathcal{A} \models \Phi \quad \Leftrightarrow \quad N(\mathrm{bin}(\mathcal{A})) = 1$$

$$\Phi \quad = \quad (\exists C_0^{2k} \cdots C_{g-1}^{2k} \Delta^k)(\forall \bar{x})\psi$$

ψ is quantifier-free.

	Space							
	0	1	$\bar{s}$	$n-1$	n		n^k-1	Δ
Time 0	$\langle q_0, w_0 \rangle$	w_1	$\cdots$	w_{n-1}	$\sqcup$	$\cdots$	$\sqcup$	δ_0
1	w_0	$\langle q_1, w_1 \rangle$	$\cdots$	w_{n-1}	$\sqcup$	$\cdots$	$\sqcup$	δ_1
	$\vdots$	$\vdots$	$\vdots$			$\vdots$		$\vdots$
$\bar{t}$			$\boxed{a_{-1}}\boxed{a_0}\boxed{a_1}$					δ_t
$\bar{t}+1$			$\boxed{b}$					δ_{t+1}
	$\vdots$	$\vdots$	$\vdots$			$\vdots$		$\vdots$
n^k-1	$\langle q_f, 1 \rangle$	$\sqcup$	$\cdots$	$\sqcup$	$\sqcup$	$\cdots$	$\sqcup$	

Accepting computation of N on input $w_0 w_1 \cdots w_{n-1}$

Theorem 20.1 (Cook-Levin Theorem)

SAT *is* **NP**-*complete.*

(This theorem was proved roughly simultaneously by Steve Cook in the USA and Leonid Levin in the USSR, *before* Fagin proved his theorem. We'll prove Cook-Levin as a corollary of Fagin's Theorem, somewhat contrary to history. But note that the proof of Cook-Levin in Sipser, for example, is almost the same as our proof of Fagin.)

Proof: Let $B \in \mathbf{NP}$. By Fagin's theorem,

$$B = \{\mathcal{A} \mid \mathcal{A} \models \Phi\}$$

$$\Phi = (\exists C_0^{2k} \cdots C_{g-1}^{2k} \Delta^k)(\forall x_1 \cdots x_t)\psi(\bar{x})$$

with ψ quantifier-free and CNF,

$$\psi(\bar{x}) = \bigwedge_{j=1}^{r} T_j(\bar{x})$$

with each T_j a disjunction of literals.

Let $\mathcal{A}$ be arbitrary, with $n = \|\mathcal{A}\|$.

Define a boolean formula $\varphi(\mathcal{A})$ as follows:

boolean variables:

$$C_i(e_1, \ldots, e_{2k}), \Delta(e_1, \ldots, e_k), \qquad i = 1, \ldots, g, e_1, \ldots, e_{2k} \in |\mathcal{A}|$$

clauses:

$$T_j(\bar{e}), \quad j = 1, \ldots, r, \bar{e} \in |\mathcal{A}|^t$$

$T'_j(\bar{e})$ is $T_j(\bar{e})$ with atomic numeric or input predicates, $R(\bar{e})$, replaced by **true** or **false** according as they are true or false in $\mathcal{A}$. Occurrences of $C_i(\bar{e})$, and $\Delta(\bar{e})$ are considered boolean variables.

$$\Phi \equiv (\exists C_0^{2k} \cdots C_{g-1}^{2k} \Delta^k)(\forall x_1 \cdots x_t) \bigwedge_{j=1}^{r} T_j(\bar{x})$$

$$\varphi(\mathcal{A}) \equiv \bigwedge_{e_1, \ldots, e_t \in |\mathcal{A}|} \bigwedge_{j=1}^{r} T'_j(\bar{e})$$

$$\mathcal{A} \in B \qquad \Leftrightarrow \qquad \mathcal{A} \models \Phi \qquad \Leftrightarrow \qquad \varphi(\mathcal{A}) \in \mathrm{SAT} \spadesuit$$

Proposition 20.2

$$\text{3-SAT} \quad = \quad \{\varphi \in \text{CNF-SAT} \mid \varphi \textit{ has} \leq 3 \textit{ literals per clause}\}$$

3-SAT *is* **NP**-*complete.*

Proof: Show SAT $\leq$ 3-SAT.

Example:

$$C \quad = \quad (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_7)$$

$$C' \equiv (\ell_1 \vee \ell_2 \vee d_1) \wedge (\overline{d_1} \vee \ell_3 \vee d_2) \wedge (\overline{d_2} \vee \ell_4 \vee d_3) \wedge (\overline{d_3} \vee \ell_5 \vee d_4) \wedge (\overline{d_4} \vee \ell_6 \vee \ell_7)$$

Claim: $C \in \text{SAT} \quad \Leftrightarrow \quad C' \in \text{3-SAT}$

In general, just do this construction for each clause independently, introducing separate dummy variables for each cluase. The AND of all the new 3-variable clauses is satisfiable iff the AND of all the old clauses is. ♠

What about reducing 3-SAT to SAT?

Can we do it?

Easily! The *identity function* serves as a reduction, because every 3-SAT instance is also a SAT instance with the same answer. This is an example of the general phenomenon of one problem being *a special case* of another. Here's another example:

Definition 20.3 A graph is *levelled* if its nodes are labelled with integers and every edge from a vertex labelled i goes to a vertex labelled $i + 1$. The problem *LEVELLED-REACH* is the set of levelled graphs such that there is a path from s to t. ♠

Proposition 20.4 LEVELLED-REACH *is complete for* **NL** *under log-space reductions.*

Proof: LEVELLED-REACH is a special case of REACH and so clearly LEVELLED-REACH $\leq$ REACH. We'll see the other direction below. ♠

But what does it prove to reduce 3-SAT **to** SAT**?**

Not much – only the fact that 3-SAT is in **NP** or that LEVELLED-REACH is in **NL**, neither of which was hard to prove anyway. To prove that a special case of a general problem is complete for some class, we have two options:

1. Reduce the general problem to the specific one, or
2. Show that the completeness proof for the general case can be adapted to always yield an instance of the special case

For example, with LEVELLED-REACH the first method would be to reduce REACH to LEVELLED-REACH directly. This can be done by taking the arbitrary directed graph G and making a new levelled graph out of n copies of G, with an edge from (u, i) to $(v, i+1)$ whenever (u, v) is an edge of G.

The second method would be to show that when we map an arbitrary **NL** problem to a REACH instance, we can make sure that we get a LEVELLED-REACH instance. (If we put a clock on the TM's worktape, for example, the configuration graph becomes levelled with the clock value as the level number.)

Proposition 20.5 3-COLOR *is* **NP**-*complete.*

Proof: Show 3-SAT $\leq$ 3-COLOR.

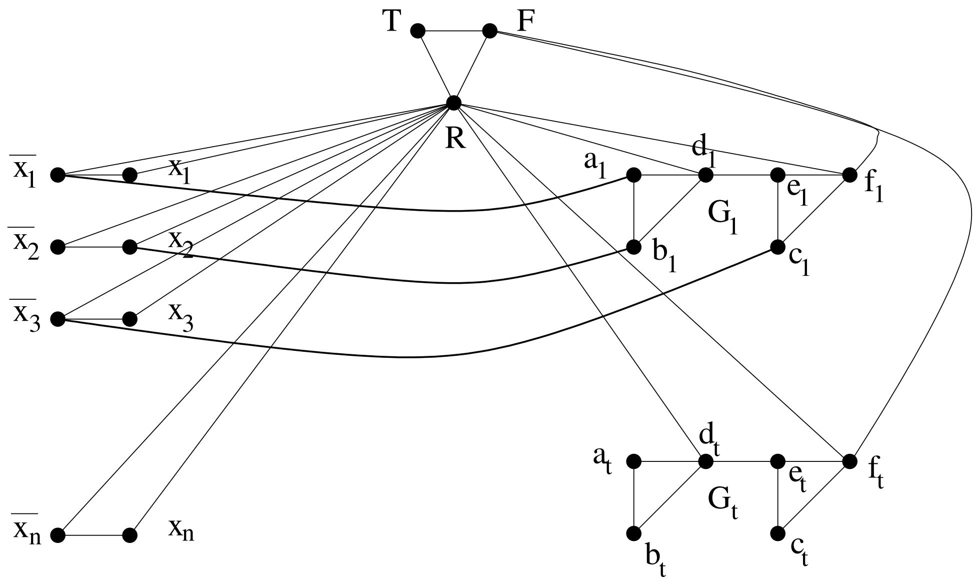
$$\varphi \quad = \quad C_1 \ \wedge \ C_2 \ \wedge \ \cdots \ \wedge \ C_t \quad \in \quad \text{3-CNF}$$

$$\text{VAR}(\varphi) \quad = \quad \{x_1, x_2, \ldots, x_n\}$$

Must build graph $G(\varphi)$ s.t.

$$\varphi \in \text{3-SAT} \qquad \Leftrightarrow \qquad G(\varphi) \in \text{3-COLOR}$$

Working assumption: 3-SAT requires $2^{\epsilon n}$ time.



G_1 encodes clause $C_1 = (\overline{x_1} \vee x_2 \vee \overline{x_3})$

Claim: Triangle a_1, b_1, d_1 serves as an "or"-gate:

d_1 may be colored "true" iff at least one of its inputs $\overline{x_1}, x_2$ is colored "true". Similarly, the output f_1 may be colored "true" iff at least one of d_1 and the third input, $\overline{x_3}$ is colored "true".

f_i can only be colored "true".

A three coloring of the literals can be extended to color G_i iff the corresponding truth assignment makes C_i true. ♠

Proposition 20.6 CLIQUE *is* **NP**-*complete.*

Proof:

Show SAT $\leq$ CLIQUE.

$$\varphi \quad = \quad C_1 \ \wedge \ C_2 \ \wedge \ \cdots \ \wedge \ C_t \quad \in \quad \text{CNF}$$

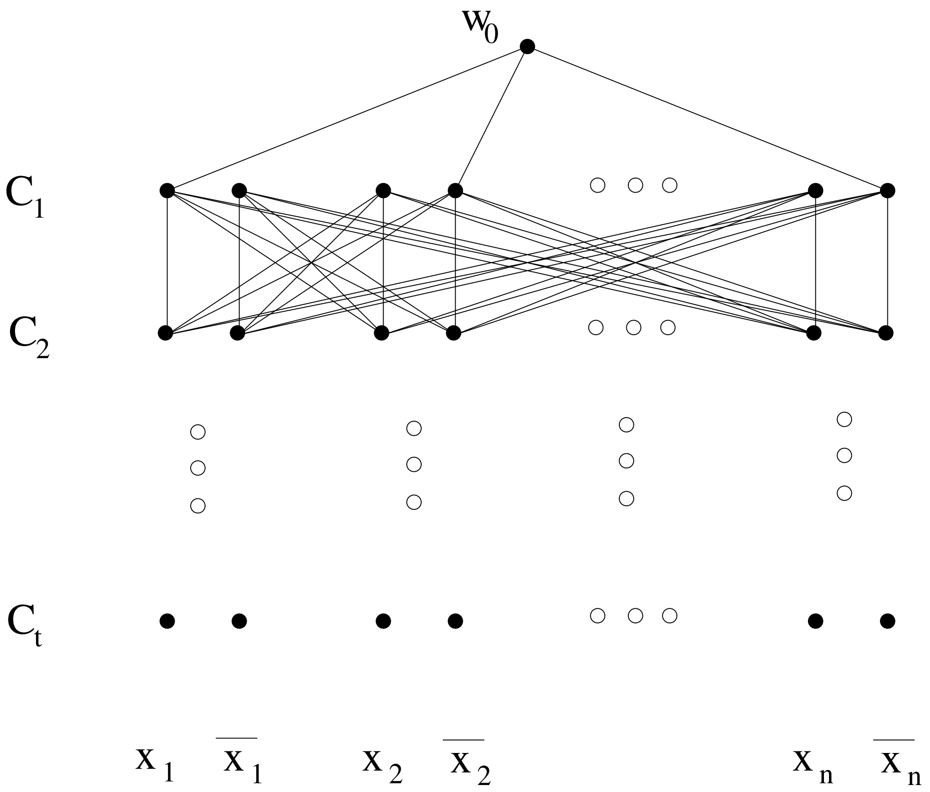
$$\text{VAR}(\varphi) \quad = \quad \{x_1, x_2, \ldots, x_n\}$$

Must build graph $g(\varphi)$ s.t.

$$\varphi \in \text{SAT} \qquad \Leftrightarrow \qquad g(\varphi) \in \text{CLIQUE}$$

$$L \ = \ \{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n\}; \quad C \ = \ \{c_1, \ldots, c_t\}$$

$$\begin{aligned}
g(\varphi) \ &= \ (V^{g(\varphi)}, E^{g(\varphi)}, k^{g(\varphi)}) \\
V^{g(\varphi)} \ &= \ (C \times L) \cup \{w_0\} \\
E^{g(\varphi)} \ &= \ \{(\langle c_1, \ell_1\rangle, \langle c_2, \ell_2\rangle) \ \mid \ c_1 \neq c_2 \text{ and } \overline{\ell}_1 \neq \ell_2\} \ \cup \\
&\quad \ \{(w_0, \langle c, \ell\rangle), (\langle c, \ell\rangle, w_0) \ \mid \ \ell \text{ occurs in } c\} \\
k^{g(\varphi)} \ &= \ t + 1
\end{aligned}$$



$$g(\varphi), \quad C_1 = (x_1 \vee \overline{x_2} \vee \overline{x_n})$$

$$
\begin{aligned}
V^{g(\varphi)} &= (C \times L) \cup \{w_0\} \\
E^{g(\varphi)} &= \{(\langle c_1, \ell_1\rangle, \langle c_2, \ell_2\rangle) \mid c_1 \neq c_2 \text{ and } \overline{\ell}_1 \neq \ell_2\} \cup \\
&\quad \{(w_0, \langle c, \ell\rangle), (\langle c, \ell\rangle, w_0) \mid \ell \text{ occurs in } c\} \\
k^{g(\varphi)} &= t + 1
\end{aligned}
$$

$$
\begin{aligned}
V^{g(\varphi)} &= (C \times L) \cup \{w_0\} \\
E^{g(\varphi)} &= \{(\langle c_1, \ell_1\rangle, \langle c_2, \ell_2\rangle) \mid c_1 \neq c_2 \text{ and } \overline{\ell}_1 \neq \ell_2\} \ \cup \\
&\quad \{(w_0, \langle c, \ell\rangle), (\langle c, \ell\rangle, w_0) \mid \ell \text{ occurs in } c\} \\
k^{g(\varphi)} &= t + 1
\end{aligned}
$$

$$
(\varphi \in \text{SAT}) \quad \Leftrightarrow \quad (g(\varphi) \in \text{CLIQUE})
$$

Claim: $g \in F(\mathbf{L})$

Proposition 20.7 *Subset Sum is* **NP**-*Complete.*

$$\left\{m_1, \ldots, m_r, T \ \in \ \mathbf{N} \ \middle| \ (\exists S \subseteq \{1, \ldots, r\})(\sum_{i \in S} m_i \ = \ T)\right\}$$

Show 3-SAT $\leq$ Subset Sum.

$$\begin{array}{rcl} \varphi & \equiv & C_1 \ \wedge \ C_2 \ \wedge \ \cdots \ \wedge \ C_t \quad \in \quad \text{3-CNF} \\ \mathrm{VAR}(\varphi) & = & \{x_1, x_2, \ldots, x_n\} \end{array}$$

Build $f \in F(\mathbf{L})$ such that for all φ,

$$\varphi \in \text{3-SAT} \qquad \Leftrightarrow \qquad f(\varphi) \in \text{ Subset Sum}$$

	x_1	x_2	$\cdots$	x_n	C_1	C_2	$\cdots$	C_t	
T	1	1	$\cdots$	1	3	3	$\cdots$	3	
x_1	1	0	$\cdots$	0	1	0	$\cdots$	1	$C_1 = (x_1 \vee \overline{x_2} \vee x_3)$
$\overline{x_1}$	1	0	$\cdots$	0	0	1	$\cdots$	0	
x_2	0	1	$\cdots$	0	0	1	$\cdots$	1	$C_2 = (\overline{x_1} \vee x_2 \vee x_n)$
$\overline{x_2}$	0	1	$\cdots$	0	1	0	$\cdots$	0	
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$C_t = (x_1 \vee x_2 \vee \overline{x_n})$
x_n	0	0	$\cdots$	1	0	1	$\cdots$	0	
$\overline{x_n}$	0	0	$\cdots$	1	0	0	$\cdots$	1	
a_1	0	0	$\cdots$	0	1	0	$\cdots$	0	
b_1	0	0	$\cdots$	0	1	0	$\cdots$	0	
a_2	0	0	$\cdots$	0	0	1	$\cdots$	0	
b_2	0	0	$\cdots$	0	0	1	$\cdots$	0	
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	
a_t	0	0	$\cdots$	0	0	0	$\cdots$	1	
b_t	0	0	$\cdots$	0	0	0	$\cdots$	1	

Knapsack

Given n objects:

object	o_1	o_2	$\cdots$	o_n	
weight	w_1	w_2	$\cdots$	w_n	≥ 0
value	v_1	v_2	$\cdots$	v_n	

W = max weight I can carry in my knapsack.

Optimization Problem:

choose $S \subseteq \{1, \ldots, n\}$

to maximize $\sum\limits_{i \in S} v_i$

such that $\sum\limits_{i \in S} w_i \ \leq \ W$

Decision Problem:

Given $\bar{w}, \bar{v}, W, V$, can I get total value $\geq V$ while total weight is $\leq W$?

Proposition 20.8 *Knapsack is* **NP**-*Complete.*

Proof: Let $I = \langle m_1, \ldots m_n, T \rangle$ be an instance of Subset Sum.

Problem: $(\exists ? S \subseteq \{1, \ldots, n\})\big(\sum_{i \in S} m_i \ = \ T\big)$

Let $f(I) = \langle m_1, \ldots m_n, m_1, \ldots, m_n, T, T \rangle$ be an instance of Knapsack.

Claim: $\quad I \in$ Subset Sum $\quad \Leftrightarrow \quad f(I) \in$ Knapsack

$$(\exists S \subseteq \{1, \ldots, n\})\big(\sum_{i \in S} m_i \ = \ T\big)$$

$$\Leftrightarrow$$

$$(\exists S \subseteq \{1, \ldots, n\})\big(\sum_{i \in S} m_i \ \geq \ T \quad \wedge \quad \sum_{i \in S} m_i \ \leq \ T\big) \spadesuit$$

Fact 20.9 *Even though Knapsack is* **NP**-*Complete there is an efficient dynamic programming algorithm that can closely approximate the maximum possible* V.