601 Lect 19: Recall Last Time: Finite Model Theory

Def: **FO** is the set of first-order definable decision problems on finite structures.

Def: **SO**$\exists$ is the set of second-order-existential definable decision problems on finite structures.

Thm: **FO** $\subseteq$ **L** $=$ **DSPACE**$[\log n]$

Fagin's Thm: **NP** $=$ **SO**$\exists$.

Cook's Thm: SAT is NP-complete.

Proof: Let $B \in$ **NP**. By Fagin's theorem,

$$B \;=\; \{\mathcal{A} \mid \mathcal{A} \models \Phi\}; \qquad \Phi \;\equiv\; \exists C_0^{2k} \cdots C_{g-1}^{2k} \Delta^k \, \forall x_1 \cdots x_t \, (\alpha(\overline{x}))$$

with α quantifier-free and CNF,

$$\alpha(\bar{x}) \quad = \quad \bigwedge_{i=1}^{r} \bigvee_{j=1}^{s} \lambda_{i,j}(\bar{x})$$

where each $\lambda_{i,j}$ is a literal.

$\mathcal{A}$ arbitrary, $n = \|\mathcal{A}\|$, Define boolean formula $\varphi_{\mathcal{A}}$:

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

literals: $\lambda_{i,j}(\bar{e})$, $i = 1, \ldots, r$, $j = 1, \ldots, s$, $\bar{e} \in |\mathcal{A}|^t$

$\lambda'_{i,j}(\bar{e})$ is $\lambda_{i,j}(\bar{e})$ with $R(\bar{e})$, replaced by $\top$ or $\bot$ according as $\mathcal{A} \models R(\bar{e})$; $C_i(\bar{e})$, and $\Delta(\bar{e})$ are just boolean variables.

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

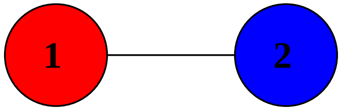
$$\varphi(\mathcal{A}) \equiv \bigwedge_{e_1,\ldots,e_t \in |\mathcal{A}|} \bigwedge_{i=1}^{r} \bigvee_{j=1}^{s} \lambda'_{i,j}(\bar{e})$$

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

Example: Fagin's Theorem implies Cook's Theorem

$$\Phi_{\text{2-color}} \equiv \exists R^1 \exists B^1 \forall x, y \left[(R(x) \vee B(x)) \right.$$
$$\wedge \quad (\neg E(x,y) \vee \neg R(x) \vee \neg R(y))$$
$$\left. \wedge \quad (\neg E(x,y) \vee \neg B(x) \vee \neg B(y)) \right]$$

G



boolean variables: r_1, r_2, b_1, b_2

$$\begin{aligned}
\varphi_G \equiv\ & (r_1 \vee b_1) \wedge (\top \vee \overline{r_1} \vee \overline{r_1}) \wedge (\top \vee \overline{b_1} \vee \overline{b_1}) \\
& (r_1 \vee b_1) \wedge (\bot \vee \overline{r_1} \vee \overline{r_2}) \wedge (\bot \vee \overline{b_1} \vee \overline{b_2}) \\
& (r_2 \vee b_2) \wedge (\bot \vee \overline{r_2} \vee \overline{r_1}) \wedge (\bot \vee \overline{b_2} \vee \overline{b_1}) \\
& (r_2 \vee b_2) \wedge (\top \vee \overline{r_2} \vee \overline{r_2}) \wedge (\top \vee \overline{b_2} \vee \overline{b_2})
\end{aligned}$$

Simplifies to: $(r_1 \vee b_1) \wedge (r_2 \vee b_2) \wedge (\overline{r_1} \vee \overline{r_2}) \wedge (\overline{b_1} \vee \overline{b_2})$

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

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

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

Example: $C \quad \equiv \quad (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_7)$

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

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

Do this construction for each clause independently. $\square$

Prop: 3-COLOR **is NP-complete.**

Proof: Saw last time that 3-COLOR $\in$ **SO**$\exists$ $=$ **NP**.

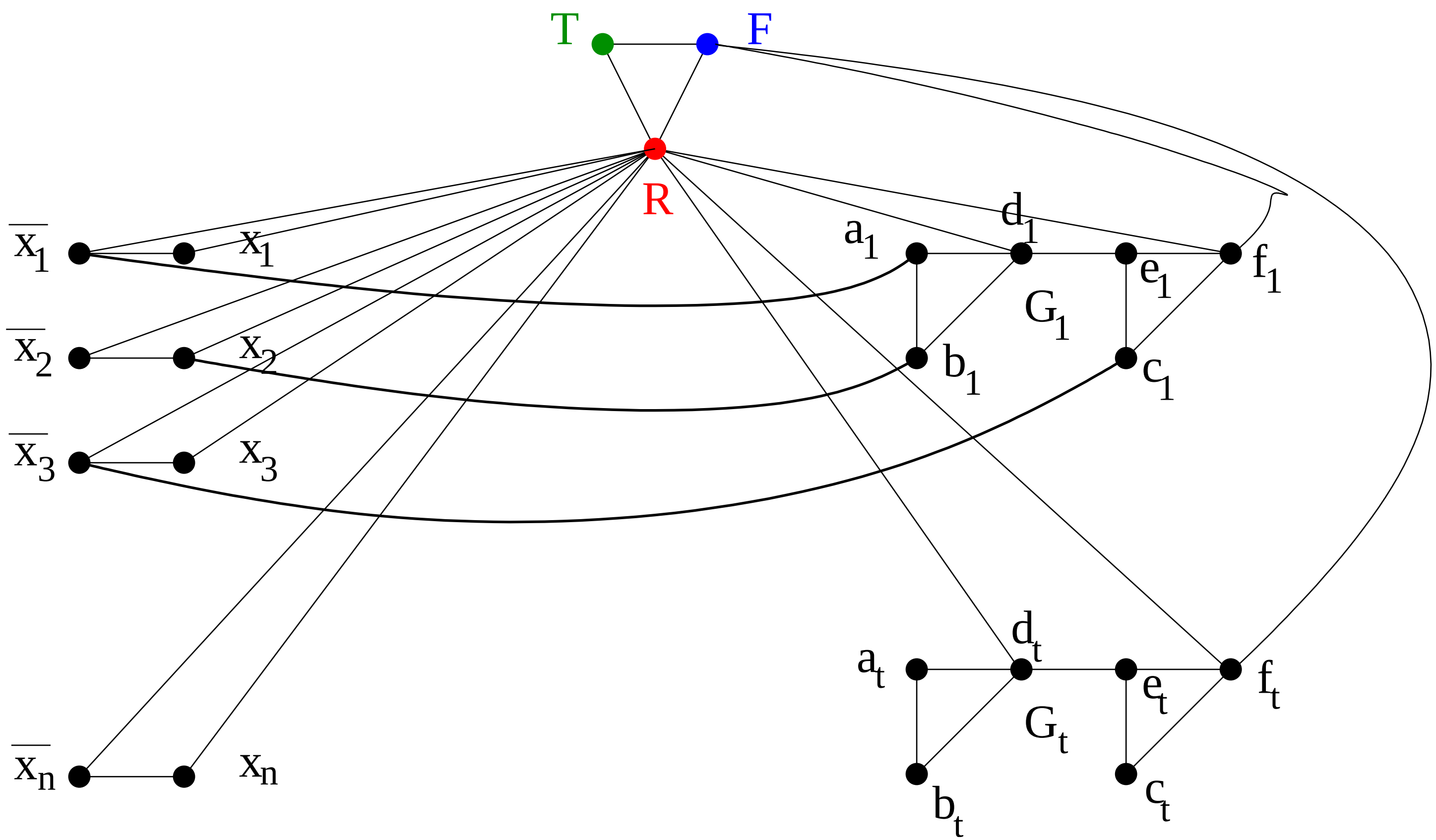
Show 3-SAT $\leq$ 3-COLOR:

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

$$\textbf{VAR}[\varphi] \quad = \quad \left\{x_1, x_2, \ldots, x_n\right\}$$

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

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



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".

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

Prop: CLIQUE is NP-complete.

Proof: Show SAT $\leq$ CLIQUE.

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

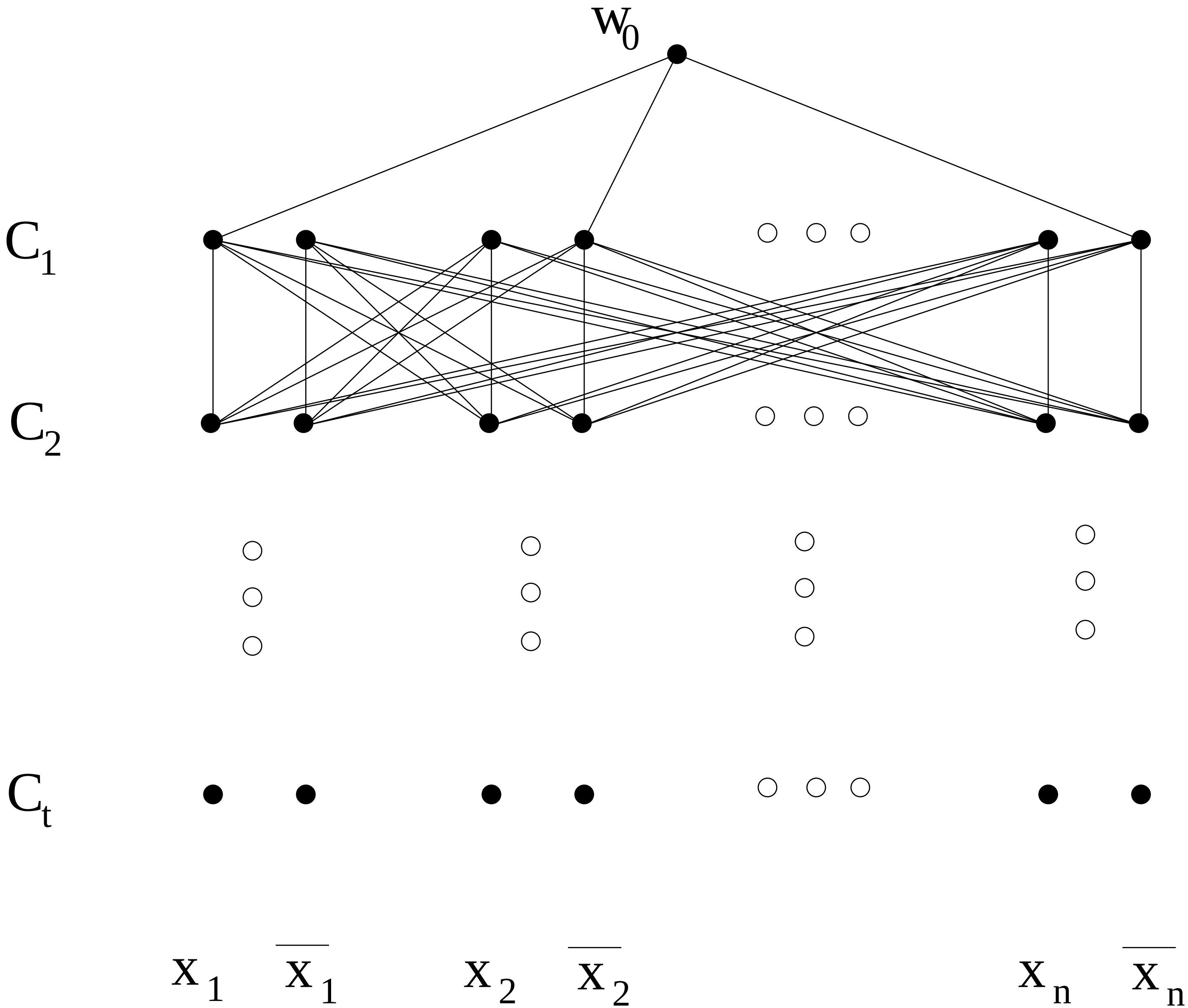
$$\textbf{VAR}[\varphi] \quad = \quad \left\{x_1, x_2, \ldots, x_n\right\}$$

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

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

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

$$g(\varphi) \quad = \quad (V^{g(\varphi)}, E^{g(\varphi)}, k^{g(\varphi)})$$



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

$$V^{g(\varphi)} = (C \times L) \cup \{w_0\}$$

$$E^{g(\varphi)} = \{(\langle c_1, \ell_1\rangle, \langle c_2, \ell_2\rangle) \mid c_1 \neq c_2 \text{ and } \overline{\ell}_1 \neq \ell_2\} \cup$$

$$\{(w_0, \langle c, \ell\rangle), (\langle c, \ell\rangle, w_0) \mid \ell \text{ occurs in } c\}$$

$$k^{g(\varphi)} = t + 1$$

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

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

Prop: Subset Sum is NP-Complete.

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

Show 3-SAT $\leq$ Subset Sum.

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

$$\mathbf{VAR}[\varphi] = \left\{ x_1, x_2, \ldots, x_n \right\}$$

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

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

	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
$\overline{x_1}$	1	0	$\cdots$	0	0	1	$\cdots$	0
x_2	0	1	$\cdots$	0	0	1	$\cdots$	1
$\overline{x_2}$	0	1	$\cdots$	0	1	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
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

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

$$C_2 = (\overline{x_1} \vee x_2 \vee x_n)$$

$$C_t = (x_1 \vee x_2 \vee \overline{x_n})$$

Knapsack

n objects; W = max weight I can carry in my knapsack.

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	

Optimization Problem: choose $S \subseteq \{1, \ldots, n\}$ to

maximize $\sum_{i \in S} v_i$ such that $\sum_{i \in S} w_i \leq W$

Decision Problem: Given $\bar{w}, \bar{v}, W, V$, can I get

total value $\geq V$ while

total weight is $\leq W$?

Prop: Knapsack is NP-Complete.

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

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

$f(I) = \langle m_1, \ldots m_n, m_1, \ldots, m_n, T, T\rangle$ is a Knapsack instance.

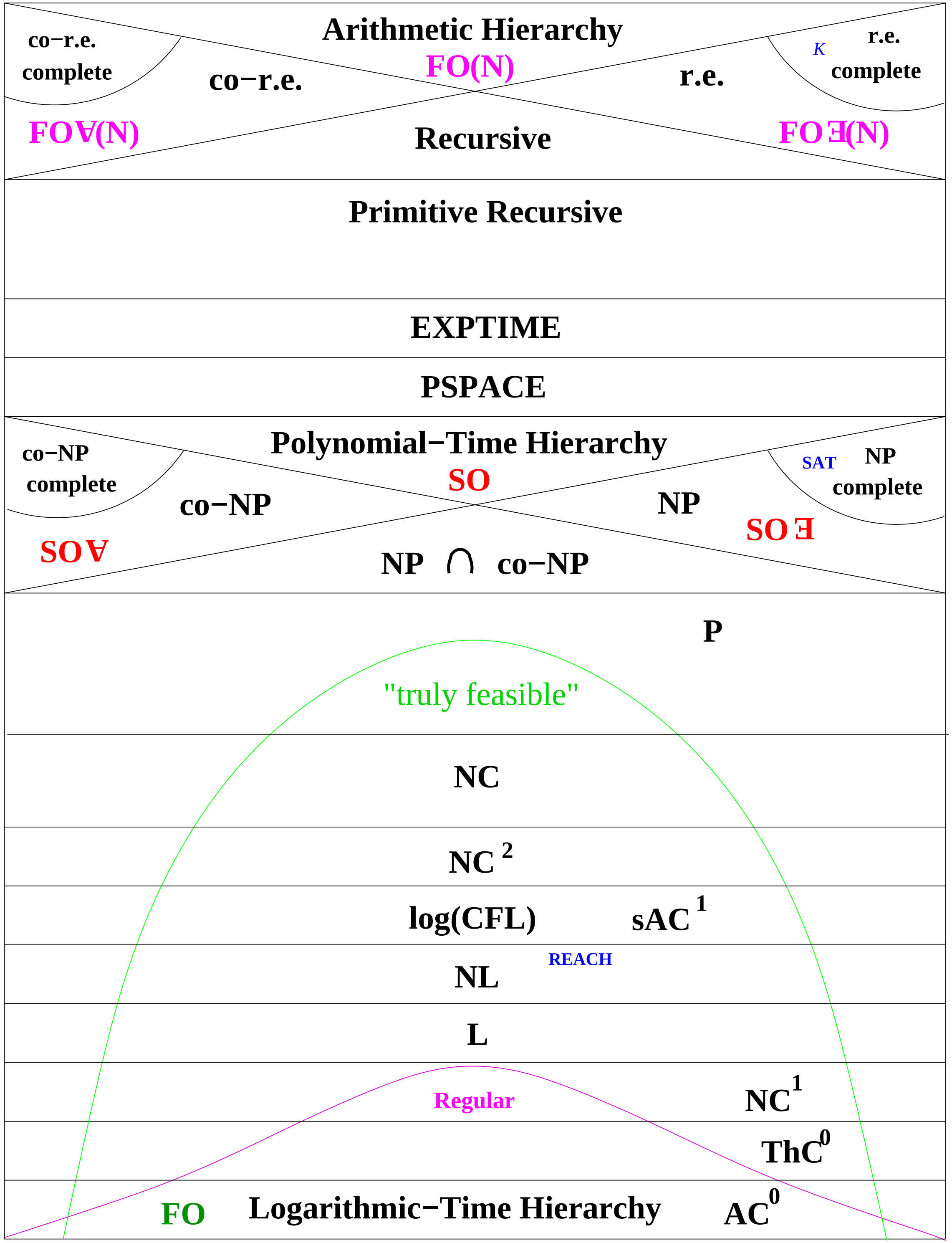
Claim: $I \in$ Subset Sum $\quad \Leftrightarrow \quad f(I) \in$ Knapsack

$$\exists S \subseteq \{1, \ldots, n\} \left(\sum_{i \in S} m_i \ = \ T\right) \qquad \Leftrightarrow$$

$$\exists S \subseteq \{1, \ldots, n\} \left(\textstyle\sum_{i \in S} m_i \ \geq \ T \quad \wedge \quad \sum_{i \in S} m_i \ \leq \ T\right) \qquad \square$$

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

More about that next class . . .


Arithmetic Hierarchy
FO(N)
co-r.e. complete
co-r.e.
r.e.
K
r.e. complete
FO∀(N)
Recursive
FO∃(N)
Primitive Recursive
EXPTIME
PSPACE
Polynomial-Time Hierarchy
SO
co-NP complete
co-NP
NP
SAT
NP complete
SO∃
SO∀
NP ∩ co-NP
P
"truly feasible"
NC
NC 2
log(CFL)
sAC 1
REACH
NL
L
Regular
NC1
ThC0
FO
Logarithmic-Time Hierarchy
AC0