The complexity of
\[ \mathcal{A} \models \phi \]

\[ \text{FOL} \]
\[ \text{SOL} \]
\[ \exists \text{SOL} \]
\[ \text{HornSOL} \]

---

Computing \( \mathcal{M}(\mathcal{A}, z, \phi) \), I

Given
\[ \mathcal{A} \text{, a } \tau \text{-structure with } |A| = m \]
\[ \phi \in \text{FOL}(\tau) \text{ of length } n \]
and quantifier depth \( q \)
\[ z \text{ an assignment } z : \text{Var}_{\text{FOL}} \rightarrow A \]

We want to compute \textbf{inductively} the meaning function
\[ \mathcal{M}(\mathcal{A}, z, \phi) \]
and estimate its computational complexity with respect to time and space denoted by
\[ \text{TIME}(\mathcal{A}, z, \phi), \text{SPACE}(\mathcal{A}, z, \phi) \]

---

Computing \( \mathcal{M}(\mathcal{A}, z, \phi) \), II

Recall that \( \tau \) is purely relational and terms \( t \) are either constants or variables.

Atomic formulas:
\[ R(t_1, t_2, \ldots, t_\tau) \text{ with } R \in \tau \text{ and } t_1 \approx t_2. \]

Takes one step in a random access look-up table.
Takes \( m^\tau \), resp. \( m^{2} \) steps for searching the table.
One bit space for the result.

Boolean operations:
\[ \phi = (\phi_1 \land \phi_2), \phi = (\phi_1 \lor \phi_2), \phi = \neg \phi_1 \]
\[ \text{TIME}(\mathcal{A}, z, \phi) = \text{TIME}(\mathcal{A}, z, \phi_1) + \text{TIME}(\mathcal{A}, z, \phi_2) + 1 \]
\[ \text{SPACE}(\mathcal{A}, z, \phi) = \max(\text{SPACE}(\mathcal{A}, z, \phi_1), \text{SPACE}(\mathcal{A}, z, \phi_2)) \]

---

Computing \( \mathcal{M}(\mathcal{A}, z, \phi) \), III

Quantifiers:
\[ \phi = \exists x \phi_1(x), \phi = \forall x \phi_1(x) \]

We search the structure for an element, hence
\[ \text{TIME}(\mathcal{A}, z, \phi) = m \cdot \text{TIME}(\mathcal{A}, z, \phi_1) \]

We can denote location of search in binary, hence
\[ \text{SPACE}(\mathcal{A}, z, \phi) = \log m \cdot \text{SPACE}(\mathcal{A}, z, \phi_1) \]

Conclusion:
\[ \text{TIME}(\mathcal{A}, z, \phi) = O(n \cdot m^q) \]
\[ \text{SPACE}(\mathcal{A}, z, \phi) = O(q \cdot \log m) \]
Computing $\mathcal{M}(\mathfrak{A}, z, \phi)$, IV

We have considered two problems for $FOL$:

1. The combined complexity of $\mathcal{M}(\mathfrak{A}, z, \phi)$ where both $\mathfrak{A}$ and $\phi$ are the input.
   
   This is in $PSPACE$.

2. The complexity of $\mathcal{M}(\mathfrak{A}, z, \phi)$ for fixed $\phi$, where only $\mathfrak{A}$ is the input.

   This is in $P$ and even in $LOGSPACE \subseteq P$.

---

Computing $\mathcal{M}(\mathfrak{A}, z, \phi)$, V

Now we consider $SOL$.

The only change comes from the quantifiers: Search is over all subsets of $A^r$.

This takes time $2^{m^r}$.

The characteristic function of these sets has size $m^r$.

Conclusion:

$TIME(\mathfrak{A}, z, \phi) = O(n \cdot 2^{m^r})$

$SPACE(\mathfrak{A}, z, \phi) = O(q \cdot \log m^r) = O(q \cdot r \cdot m)$

---

Computing $\mathcal{M}(\mathfrak{A}, z, \phi)$, VI

We consider two problems for $SOL$:

1. The combined complexity of $\mathcal{M}(\mathfrak{A}, z, \phi)$ where both $\mathfrak{A}$ and $\phi$ are the input.

   This is also in $PSPACE$.

2. The complexity of $\mathcal{M}(\mathfrak{A}, z, \phi)$ for fixed $\phi$, where only $\mathfrak{A}$ is the input.

   This is in $PSPACE$.

---

Computing $\mathcal{M}(\mathfrak{A}, z, \phi)$, VII

We want to use now non-deterministic machines.

We denote by $\exists SOL(\tau)$ the set of $SOL(\tau)$-formulas $\psi$ of the form

$\psi = \exists x_1 \exists x_2 \ldots \exists x_k \phi(x_1, x_2, \ldots, x_k)$

with $\phi \in FOL(\tau \cup \{x_1, x_2, \ldots, x_k\})$.

For fixed $\psi \in \exists SOL$ we have that $\mathcal{M}(\mathfrak{A}, z, \psi)$ is in $NP$.

Hence, for fixed $\psi \in SOL$ we have that $\mathcal{M}(\mathfrak{A}, z, \psi)$ is in $PH$,

the Polynomial Hierarchy.
The Polynomial Hierarchy, I

We look at **Oracle Turing Machines** OTM. Let \( X \) be a problem and \( C \) be a class of problems.

We define

\[
\begin{align*}
P^X &= \{ Y : \exists M \text{ accepts } Y \text{ using } X \text{ as oracle } \} \\
P^C &= \{ Y : \exists M \text{ accepts } Y \text{ using } X \in C \text{ as oracle } \}
\end{align*}
\]

Here \( M \) is a deterministic polynomial time OTM.

Similarly,

\[
\begin{align*}
NP^X &= \{ Y : \exists M \text{ accepts } Y \text{ using } X \text{ as oracle } \} \\
NP^C &= \{ Y : \exists M \text{ accepts } Y \text{ using } X \in C \text{ as oracle } \}
\end{align*}
\]

Here \( M \) is a non-deterministic polynomial time OTM.

---

The Polynomial Hierarchy, II

We define inductively:

\[
\begin{align*}
\Delta_0 P &= \Sigma_0 P = \Pi_0 P = P \\
\Delta_{i+1} P &= P_{\Sigma_i P} \\
\Sigma_{i+1} P &= \text{NP}_{\Sigma_i P} \\
\Pi_{i+1} P &= \text{CoNP}_{\Sigma_i P}
\end{align*}
\]

Finally,

\[
\Phi = \bigcup_{i \in \mathbb{N}} \Sigma_i P
\]

Note that \( \Phi \subseteq PSPACE \) and

\( P = \text{NP} \) iff \( P = \Phi \).

---

Horn formulas, I

A **propositional Horn clause** is a formula of the form

\[
\neg p_1 \lor \neg p_2 \lor \ldots \lor \neg p_m \lor q
\]

with at most one variable unnegated.

Equivalently, we can write

\[
(p_1 \land p_2 \land \ldots \land p_m \rightarrow q)
\]

\( m = 0 \) gives \( q \) and the absence of \( q \) gives

\[
(p_1 \land p_2 \land \ldots \land p_m \rightarrow \text{false})
\]

or

\[
\neg (p_1 \land p_2 \land \ldots \land p_m)
\]

A **FOL** Horn clause is obtained by replacing variables by atomic formulas.

---

Horn formulas, II

The **size** \( s(C) \) of a clause \( C \) is the number of variables occurring in \( C \).

The **size** \( s(\Sigma) \) of set of clauses \( \Sigma \) is defined as \( \sum_{C \in \Sigma} s(C) \)

\( SAT \) is the problem of deciding whether a set \( \Sigma \) of clauses with \( n \) variables of size \( m \) is satisfiable.

**Theorem**: \( SAT \) can be solved in \( TIME(2^n \cdot m) \) and is NP-complete.

\( HORNSAT \) is like \( SAT \) but with \( \Sigma \) a set of Horn clauses.

**Theorem**: \( HORNSAT \) is in \( P \).

**Proof**: Use unit resolution.
Horn formulas, III

The formulas of HornSO are of the form
\[ \psi = \exists X_1 \exists X_2 \ldots \exists X_j \forall x_1 \forall x_2 \ldots \forall x_k \bigwedge_{\ell=1}^n \Phi_\ell \]
where each \( \Phi_\ell \) is a \textit{FOL}-Horn clause.

**Theorem:** (Grädel)
For \( \psi \) a fixed HornSO formula \( \mathcal{M}(\mathfrak{A}, z, \phi) \) is in \textsf{P}.

We give a proof.

---

**Horn formulas, IV**

For simplicity let
\[ \psi = \exists X \forall x_1 \forall x_2 \ldots \forall x_k \bigwedge_{\ell=1}^n \Phi_\ell \]
a \( \tau_{\text{graph}} \)-formula with \( X \) \( r \)-ary.

So each \( \Phi_\ell \) consists of atomic or negated atomic formulas \( x_i \approx x_j \), \( E(x_i, x_j) \) or \( X(x_{i_1}, \ldots, x_{i_r}) \).

Let \( \mathfrak{A} \) be a structure with elements \( a_1, \ldots, a_n \).

There are \( n_k \) many assignments for the variables \( x_i \).

Let \( h \) be the length of \( \Phi = \bigwedge_\ell \Phi_\ell \).

---

**Horn formulas, V**

We now form the formula
\[ \bigwedge_z \text{subst}(\Phi, z) \]
This formula has exactly \( h \cdot n^k \) many literals.

In \( \mathfrak{A} \) each atomic formula \( E(a_i, a_j) \) or \( a_i \approx a_j \) is true or false, so we can replace them by true or false respectively.

We replace each \( X(a_{i_1}, \ldots, a_{i_r}) \) by a propositional variable \( p_{a_{i_1} \ldots a_{i_r}} \).

We obtain so a propositional formula \( \Psi \).

---

**Horn formulas, VI**

**Claim 1:**
If \( \mathfrak{A} \models \psi \) then \( \Psi \) is satisfiable.

**Proof:** Assume \( \mathfrak{A} \models \psi \).
Then there is \( U \subset A^r \) such that
\[ \mathfrak{A}, U \models \forall x_1 \forall x_2 \ldots \forall x_k \bigwedge_{\ell=1}^n \Phi_\ell \]

We now define an assignment
\[ z(p_\alpha) = \begin{cases} 1 & \text{if } \alpha \in U \\ 0 & \text{if } \alpha \notin U \end{cases} \]

**Exercise:** Show that this \( z \) makes \( \Psi \) true.
Horn formulas, VII

Claim 2:
If $\Psi$ is satisfiable then $\mathcal{A} \models \Psi$.

Proof: Assume $z$ is an assignment which makes $\Psi$ true.

We define an interpretation $U$ for $X$ by
$$a \in U \text{ iff } z(p_a) = 1$$

Exercise: Show that
$$\mathcal{A}, U \models \forall x_1 \forall x_2 \ldots \forall x_k \bigwedge_{\ell=1}^n \Phi_\ell$$

Proof of Theorem:

- The construction of $\bar{\Psi}$ from $\Psi$ is done in polynomial time
- The size of $\bar{\Psi}$ is polynomial in the size of $\Psi$
- Using the polynomial time algorithm for HORNSAT, we check the satisfiability of $\bar{\Psi}$
- Using the lemma, this settles $\mathcal{A} \models \Psi$

2SAT

Let $\Sigma$ be a set of propositional clauses of at most two literals each.
These are sometimes called Krom clauses.
Both Horn and Krom are names of Logicians

2SAT is the problem of deciding whether such a $\Sigma$ is satisfiable.

$\text{NL}$ denotes the class of problems decidable in non-deterministic logarithmic space.

Theorem: 2SAT is decidable in $\text{NL}$.

$\text{Krom} \exists \text{SOL}$ is like $\text{Horn} \exists \text{SOL}$ but with clauses of size 2 rather than Horn clauses.

It is now easy to prove that

**Theorem:** For fixed $\Psi \in \text{Krom} \exists \text{SOL}$ the problem $M(\mathcal{A}, z \Psi)$ is in $\text{NL}$.
The proof is exactly like for $\text{Horn} \exists \text{SOL}$

Definability and Complexity, I

Let $K$ be a class of finite $\tau$-structures.

Let $\mathcal{L}(\tau) \subseteq \text{SOL}(\tau)$.
Typically $\mathcal{L}(\tau)$ is one of $\text{Krom} \exists \text{SOL}(\tau)$, $\text{Horn} \exists \text{SOL}(\tau)$, $\exists \text{SOL}(\tau)$, $\text{SOL}(\tau)$, $\text{MSOL}(\tau)$.

$K$ is **definable in** $\mathcal{L}(\tau)$ if there exists $\Psi \in \mathcal{L}(\tau)$ such that
$$\mathcal{A} \in K \text{ iff } \mathcal{A} \models \Psi$$

Let $C$ be a complexity class.
typically $\text{LOGSPACE}$, $\text{NL}$, $\text{P}$, $\text{NP}$, $\text{PH}$, $\text{PSPACE}$

$K$ is in $C$ iff the problem $\mathcal{A} \in K$ can be decided with the resources allowed in $C$. 
We have shown:

- If $K$ is definable in $FOL$ then $K \in \text{LOGSPACE}$.
- If $K$ is definable in $\text{Krom}\exists\text{SOL}$ then $K \in \text{NL}$.
- If $K$ is definable in $\text{Horn}\exists\text{SOL}$ then $K \in \text{P}$.
- If $K$ is definable in $\exists\text{SOL}$ then $K \in \text{NP}$.
- If $K$ is definable in $\text{SOL}$ then $K \in \text{PH}$.

We will show in the sequel for ordered structures

- (Grädel) If $K \in \text{NL}$, then $K$ is definable in $\text{Krom}\exists\text{SOL}$.
- (Grädel) If $K \in \text{P}$, then $K$ is definable in $\text{Horn}\exists\text{SOL}$.

For arbitrary structures we have

- (Fagin, Christen) If $K \in \text{NP}$, then $K$ is definable in $\exists\text{SOL}$.
- (Meyer and Stockmeyer) If $K \in \text{PH}$, then $K$ is definable in $\text{SOL}$.

---

**LOGSPACE**

What about $FOL$-definability and $\text{LOGSPACE}$?

**Exercise:**
Show that the set of words of even size is not $FOL$-definable.

**Exercise:**
Show that the set of words of even size is in $\text{LOGSPACE}$.

**Exercise:**
Conclude that $FOL$-definability is weaker than decidability in $\text{LOGSPACE}$.

**Questions:**
Which logic corresponds to $\text{LOGSPACE}$?
$FOL+$ deterministic transitive closure

Which complexity class corresponds to $FOL$?
The circuit complexity class $\text{AC}_0$. 

---

CS 236 331:2001    Lecture 6