

The complexity of

$$\mathcal{A} \models \phi$$

FOL

SOL

ESOL

HornSOL

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

Given

\mathfrak{A} , a τ -structure with $|A| = m$

$\phi \in FOL(\tau)$ of length n
and quantifier depth q

z an assignment $z : Var_{FOL} \rightarrow A$

We want to compute **inductively** the meaning function

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

and estimate its computational complexity with respect to time and space denoted by

$$TIME(\mathfrak{A}, z, \phi), SPACE(\mathfrak{A}, z, \phi)$$

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

Recall that τ is purely relational and terms t are either constants or variables.

Atomic formulas:

$R(t_1, t_2, \dots, t_r)$ with $R \in \tau$ and $t_1 \approx t_2$.

Takes one step in a random access look-up table.

Takes m^r , resp. m^2 steps for searching the table.

One bit space for the result.

Boolean operations:

$\phi = (\phi_1 \wedge \phi_2)$, $\phi = (\phi_1 \vee \phi_2)$, $\phi = \neg\phi_1$

$TIME(\mathfrak{A}, z, \phi) = TIME(\mathfrak{A}, z, \phi_1) + TIME(\mathfrak{A}, z, \phi_2) + 1$

$SPACE(\mathfrak{A}, z, \phi) = \max(SPACE(\mathfrak{A}, z, \phi_1), SPACE(\mathfrak{A}, z, \phi_2))$

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

Quantifiers:

$$\phi = \exists x \phi_1(x), \quad \phi = \forall x \phi_1(x)$$

We search the structure for an element, hence
 $TIME(\mathfrak{A}, z, \phi) = m \cdot TIME(\mathfrak{A}, z, \phi_1)$

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

Conclusion:

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

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

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 ϕ are the input.

This is in *PSPACE*.

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

This is in \mathbf{P} and even in $\text{LOGSPACE} \subseteq \mathbf{P}$.

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

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:

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

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

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 ϕ are the input.

This is also in *PSPACE*.

2. The complexity of $\mathcal{M}(\mathfrak{A}, z, \phi)$ for fixed ϕ , 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 ψ of the form

$$\psi = \exists X_1 \exists X_2 \dots \exists X_k \phi(X_1, X_2, \dots, X_k)$$

with $\phi \in FOL(\tau \cup \{X_1, X_2, \dots, X_k\})$

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

$$\mathbf{P}^X = \{Y : \exists M \text{ accepts } Y \text{ using } X \text{ as oracle} \}$$

$$\mathbf{P}^C = \{Y : \exists M \text{ accepts } Y \text{ using } X \in C \text{ as oracle} \}$$

Here M is a deterministic polynomial time *OTM*.

Similarly,

$$\mathbf{NP}^X = \{Y : \exists M \text{ accepts } Y \text{ using } X \text{ as oracle} \}$$

$$\mathbf{NP}^C = \{Y : \exists M \text{ accepts } Y \text{ using } X \in C \text{ as oracle} \}$$

Here M is a non-deterministic polynomial time *OTM*.

The Polynomial Hierarchy, II

We define inductively:

$$\Delta_0\mathbf{P} = \Sigma_0\mathbf{P} = \Pi_0\mathbf{P} = \mathbf{P}$$

and

$$\begin{aligned}\Delta_{i+1}\mathbf{P} &= \mathbf{P}^{\Sigma_i\mathbf{P}} \\ \Sigma_{i+1}\mathbf{P} &= \mathbf{NP}^{\Sigma_i\mathbf{P}} \\ \Pi_{i+1}\mathbf{P} &= \mathbf{CoNP}^{\Sigma_i\mathbf{P}}\end{aligned}$$

Finally,

$$\mathbf{PH} = \bigcup_{i \in \mathbb{N}} \Sigma_i\mathbf{P}$$

Note that $\mathbf{PH} \subseteq \mathit{PSPACE}$ and
 $\mathbf{P} = \mathbf{NP}$ iff $\mathbf{P} = \mathbf{PH}$.

Horn formulas, I

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

$$\neg p_1 \vee \neg p_2 \vee \dots \vee \neg p_m \vee q$$

with at most one variable unnegated.

Equivalently, we can write

$$(p_1 \wedge p_2 \wedge \dots \wedge p_m \rightarrow q)$$

$m = 0$ gives $\text{true} \rightarrow q$ and the absence of q gives

$$(p_1 \wedge p_2 \wedge \dots \wedge p_m \rightarrow \mathbf{false})$$

or

$$\neg(p_1 \wedge p_2 \wedge \dots \wedge p_m)$$

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 Σ .

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

SAT is the problem of deciding whether a set Σ of clauses with n variables of size m is satisfiable.

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

HORN SAT is like *SAT* but with Σ a set of Horn clauses.

Theorem: *HORN SAT* is in P.

Proof: Use unit resolution.

Horn formulas, III

The formulas of *Horn \exists SOL* are of the form

$$\Psi = \exists X_1 \exists X_2 \dots \exists X_j \forall x_1 \forall x_2 \dots \forall x_k \bigwedge_{\ell=1}^n \Phi_\ell$$

where each Φ_{ell} is a *FOL*-Horn clause.

Theorem:(Grädel)

For Ψ a fixed *Horn \exists SOL* formula
 $\mathcal{M}(\mathcal{A}, z, \phi)$ is in **P**.

We give a proof.

Horn formulas, IV

For simplicity let

$$\Psi = \exists X \forall x_1 \forall x_2 \dots \forall x_k \bigwedge_{\ell=1}^n \Phi_\ell$$

a τ_{graphs} -formula with X r -ary.

So each Φ_ℓ consists of atomic or negated atomic formulas $x_i \approx x_j$, $E(x_i, x_j)$ or $X(x_{i_1}, \dots, x_{i_r})$.

Let \mathfrak{A} be a structure with elements a_1, \dots, 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, \forall

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 $X(a_{i_1}, \dots, a_{i_r})$ by a propositional variable $p_{a_{i_1}, \dots, a_{i_r}}$.

We obtain so a propositional formula $\bar{\Psi}$.

Horn formulas, VI

Claim 1:

If $\mathfrak{A} \models \Psi$ then $\bar{\Psi}$ is satisfiable.

Proof: Assume $\mathfrak{A} \models \Psi$.

Then there is $U \subset A^r$ such that

$$\mathfrak{A}, U \models \forall x_1 \forall x_2 \dots \forall x_k \bigwedge_{\ell=1}^n \Phi_\ell$$

.

We now define an assignment

$$z(p_{\bar{a}}) = \begin{cases} 1 & \text{if } \bar{a} \in U \\ 0 & \text{if } \bar{a} \notin U \end{cases}$$

Exercise: Show that this z makes $\bar{\Psi}$ true.

Horn formulas, VII

Claim 2:

If $\bar{\Psi}$ is satisfiable then $\mathfrak{A} \models \Psi$.

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

We define an interpretation U for X by

$$\bar{a} \in U \text{ iff } z(p_{\bar{a}}) = 1$$

.

Exercise: Show that

$$\mathfrak{A}, U \models \forall x_1 \forall x_2 \dots \forall x_k \bigwedge_{\ell=1}^n \Phi_{\ell}$$

.

Horn formulas, VII

Proof of Theorem:

- The construction of $\bar{\Psi}$ from Ψ is done in polynomial time
- The size of $\bar{\Psi}$ is polynomial in the size of Ψ .
- Using the polynomial time algorithm for *HORN SAT*, we check the satisfiability of $\bar{\Psi}$.
- Using the lemma, this settles $\mathfrak{A} \models \Psi$.

$2SAT$

Let Σ 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 Σ is satisfiable.

NL denotes the class of problems decidable in non-deterministic logarithmic space.

Theorem: $2SAT$ is decidable in NL .

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

It is now easy to prove that

Theorem: For fixed $\Psi \in Krom\exists SOL$ the problem $\mathcal{M}(\mathcal{A}, z\Psi)$ is in NL .

The proof is exactly like for $Horn\exists SOL$.

Definability and Complexity, I

Let K be a class of finite τ -structures.

Let $\mathcal{L}(\tau) \subseteq SOL(\tau)$.

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

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

$$\mathfrak{A} \in K \text{ iff } \mathfrak{A} \models \Psi$$

Let \mathbf{C} be a complexity class.

typically $LOGSPACE$, NL , P , NP , PH , $PSPACE$

K is in \mathbf{C} iff the problem $\mathfrak{A} \in K$ can be decided with the resources allowed in \mathbf{C} .

Definability and Complexity, II

We have shown:

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

Definability and Complexity, III

We will show in the sequel
for **ordered structures**

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

For **arbitrary structures** we have

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

LOGSPACE

What about *FOL*-definability and *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 *LOGSPACE*.

Exercise:

Conclude that *FOL*-definability is weaker than decidability in *LOGSPACE*.

Questions:

Which logic corresponds to *LOGSPACE*?

FOL+deterministic transitive closure

Which complexity class corresponds to *FOL*?

The circuit complexity class AC_0 .