Enter first order logic

- We now establish the connection to first order logic
- We define vocabularies as set of function, relation and constant symbols.
- We define structures and formulas.
- We introduce quantifier elimination.

First order structures $\mathfrak{A}$ over a vocabulary τ (reminder)

- A vocabulary is a set of function symbols, relation symbols and constant symbols of arity $\mathsf{rf}(i), \mathsf{rr}(i) \in \mathbb{N}$.

- Constant symbols: c_i of arity 0
  Relation symbols: $R_{i,\mathsf{rr}(i)}$ of arity $\mathsf{rr}(i) \geq 1$.
  Function symbols: $F_{i,\mathsf{rf}(i)}$ of arity $\mathsf{rf}(i) \geq 1$.
  Vocabulary: $\tau \subseteq \{c_i : i \in \mathbb{N}\} \cup \{R_{i,\mathsf{rr}(i)} :\in \mathbb{N}\} \cup \{F_{i,\mathsf{rr}(i)} :\in \mathbb{N}\}$

- A τ-structure $\mathfrak{A}$ is an interpretation of a vocabulary.
  The universe of $\mathfrak{A}$ is a non-empty set A.
  The interpretation of c_i is an element $c_i^A \in A$.
  The interpretation of $R_{i,\mathsf{rr}(i)}$ is a relation $R^A_{i,\mathsf{rr}(i)} \subseteq A^{\mathsf{rr}(i)}$.
  The interpretation of $F_{i,\mathsf{rf}(i)}$ is a function $F^A_{i,\mathsf{rf}(i)} : A^{\mathsf{rf}(i)} \to A$.

Unless otherwise stated, vocabularies are **finite**.

τ-formulas.

Let τ be a set of function and relation symbols.

- Variables are $v_0, v_1, \ldots, v_i, \ldots$ for $i \in \mathbb{N}$
- Terms are either variables, constant symbols, or of the form $F(t_1, \ldots, t_{rf(F)})$ for terms t_i and $F \in \tau$.
- Atomic formulas are $t_i \approx t_j$ or of the form $R(t_1, \ldots, t_{rf(F)})$ for terms t_i and $R \in \tau$.
- Quantifier-free formulas are boolean combinations of atomic formulas.
- existential formulas are of the form $\exists v_1, \ldots \exists v_\ell \phi$ with ϕ a quantifier-free formula.
- τ-formulas are closed under boolean combinations and quantifcation of free variables.

The size of a τ-formula is the number of connectives and quantifiers occuring in it.

Accepting sets for circuits

Let $\mathbf{M}_\tau$ be a non-deterministic τ-machine on $\mathfrak{A}$ with running time $t(n)$.
For $b \in A^m$ let $\mathrm{Acc}_n(\mathbf{M}, b)$ be the set

$$\mathrm{Acc}_n(\mathbf{M}, b) = \{a \in A^n : \mathbf{M} \text{ accepts input } a \text{ with guess } b \in A^m\}$$

and $\mathrm{NAcc}_n(\mathbf{M})$ be the set

$$\mathrm{NAcc}_n(\mathbf{M}) = \{a \in A^n : \text{ there is a guess } b \in A^m \text{ s.t. } \mathbf{M} \text{ accepts input } a\}$$

Theorem 4

(i) $\mathrm{Acc}_n(\mathbf{M}, b)$ *is definable by a quantifier-free* τ*-formula with equality* $\phi(v_1, \ldots, v_n, u_1, \ldots, u_m)$.

(ii) $\mathrm{NAcc}_n(\mathbf{M})$ *is definable by the existential* τ*-formula* $\exists u_1, \ldots, u_m \phi(v_1, \ldots, v_n, u_1, \ldots, u_m)$.

Both formulas are of size polynomial in $t(n)$.

Proof sketch: Induction using Theorem 2. We replace $\mathbf{M}$ for input of size n by its circuit.

- Single gates: write down the input/output relation.
- Use induction over the depth of the DAG.

□

From τ-formulas to τ-circuits

Let $C = C(a, b)$ be a τ-circuit for input a and guess b.
$\mathrm{Acc}_n(C, b)$ and $\mathrm{NAcc}_n(C)$ are defined as for τ-machines.

Proposition 5
Let $\phi(a, b)$ be a quantifier-free τ-formula of size $s(n)$ in n variables.
There exists a τ-circuit $C(a, b)$ such that

- *The size of $C(a, b)$ is polynomial in the size $s(n)$,*
- $\mathrm{Acc}_n(C, b)$ *is the formula $\phi(a, b)$.*
- $\mathrm{NAcc}_n(C)$ *is the formula $\exists b \phi(a, b)$.*

Proof: We have to implement the gates of the boolean operations of truth values $0, 1$ using explicit truthtables and use relation gates for testing equality and the relations.

Then we follow the tree presentation of ϕ. □

All quantifier-free formulas occur
as descriptions of accepting sets

Lemma 6
Let $\phi(a,b)$ be a quantifier free formula with input a of size n guess b.

There are τ-machines $\mathbf{M}_D$ and $\mathbf{M}_N$ solving $P_D \in \mathbf{P}_{\mathfrak{A}}$ and $P_N \in \mathbf{NP}_{\mathfrak{A}}$ such that

- $\mathsf{Acc}_n(\mathbf{M}_D, b)$ *is the formula* $\phi(a,b)$.
- $\mathsf{NAcc}_n(\mathbf{M}_N)$ *is the formula* $\exists b \phi(a,b)$.

FSAT

$\mathrm{FSAT}_{\mathfrak{A}}$ is the following problem:

Input $w(\phi)a \in A^\star$, with $w(\phi)$ a binary encoding of a quantifier-free formula ϕ.

Problem (q) Is there $b \in A^\star$ such that

$$\langle \mathfrak{A}, a, b \rangle \models \phi(a, b)$$

Problem (e) Is it true that

$$\langle \mathfrak{A}, a \rangle \models \exists b \phi(a, b)$$

Clearly the two problems are equivalent.

Theorem 7
For every τ-structure $\mathfrak{A}$ the problem $\mathrm{FSAT}_{\mathfrak{A}}$ is $\mathbf{NP}_{\mathfrak{A}}$-complete.

Proof of Theorem 7

Clearly $\mathrm{FSAT}_{\mathfrak{A}} \in \mathbf{NP}_{\mathfrak{A}}$.

Conversely, we interpret CSAT in FSAT.
Let $w(C(x,y)), a \in A^\star$ be an instance of CSAT.

For a gate γ in C of in-degree k let $\gamma_1, \ldots, \gamma_k$ the predecessor gates of γ. For each gate γ in C which is not an input gate, a constant gate or an output gate let z_γ be a new variable and z be the sequence of all these.

Consider the quantifier-free formula

$$\phi_C = \left(\bigwedge_\gamma z_\gamma \approx \gamma(\gamma_1, \ldots, \gamma_k) \right) \wedge \gamma_{output} \approx 1$$

ϕ_C is of size polynomial in the size of C.

Now $\exists y(C(a,y) = 1$ iff $\exists y \exists \bar{z} \phi_C(y,z)$ which is of the form of Problem (e). $\square$

Quantifier elimination over $\mathfrak{A}$

Let $\mathfrak{A}$ be a τ-structure and

$$\mathrm{Th}(\mathfrak{A}) = \{\theta \in \mathrm{FOL}(\tau) : \mathfrak{A} \models \theta \text{ where } \theta \text{ has no free variables}\}$$

Let ϕ and ψ vary over τ-formulas with free variables $x_1, \ldots, x_n$. We say that $\mathfrak{A}$ allows

- Elimination of quantifiers (QE($\mathfrak{A}$)) if for every ϕ there is a quantifier-free ψ such that
$$\mathrm{Th}(\mathfrak{A}) \models \forall x(\phi \Leftrightarrow \psi)$$
- Elimination of existential quantifiers (EQE($\mathfrak{A}$)) if for every existential ϕ there is such a quantifier-free ψ.
- Short elimination of quantifiers (SQE($\mathfrak{A}$)) if for every existential ϕ there is such a quantifier-free ψ whose size is polynomial in the size of ϕ.
- Fast elimination of quantifiers (FQE($\mathfrak{A}$)) if there is a polynomial time algorithm giving the result of SQE($\mathfrak{A}$).

Quantifier elimination

- Clearly we have

$$\mathrm{FQE}(\mathfrak{A}) \Rightarrow \mathrm{SQE}(\mathfrak{A}) \Rightarrow \mathrm{EQE}(\mathfrak{A}) \Rightarrow \mathrm{QE}(\mathfrak{A})$$

- There are quite a few τ-strutures $\mathfrak{A}$ with $\mathrm{QE}(\mathfrak{A})$:
  – Infinite set (with $\tau = \emptyset$).
  – The field of complex numbers and the ordered field of real numbers.
  – Dense linear orders with prescribed extreme elements (4 cases)

- Every finite τ-structure $\mathfrak{F}$ has $\mathrm{EQE}(\mathfrak{F})$ but $\mathrm{SQE}(\mathfrak{F})$ is equivalent to $\mathbf{P} = \mathbf{NP}$ in the Turing model of computation.

- M. Prunescu showed that there is an infinite structure $\mathfrak{A}$ with $\mathrm{FQE}(\mathfrak{A})$.

Poizat's fast QE Theorem

Theorem 8 (Poizat)
Let $\mathfrak{A}$ be a τ-structure.

The following are equivalent.

- $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$
- $\mathfrak{A}$ *allows* **fast quantifier elimination (FQE).**
- $\mathfrak{A}$ *allows* **short quantifier elimination (SQE).**

(i): $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$ implies SQE($\mathfrak{A}$)

- Let $\exists b \phi(x, b)$ be an existential formula with n free variables.
  By Lemma 6 there is a polynomial time non-deterministic machine $\mathbf{M}_N$ such that

$$\exists b \phi(x, b) \text{ describes } \mathrm{NAcc}_n(\mathbf{M}_N)$$

- As $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$ there is a polynomial time deterministic machine $\mathbf{M}_D$ which accepts the same inputs as $\mathbf{M}_N$.

- Let $\psi(x)$ be a quantifier-free formula which describes $\mathrm{Acc}_n(\mathbf{M}_D)$.

- $\psi(x)$ is of size polynomial in n and is equivalent to $\exists b \phi(x, b)$, which proves SQE($\mathfrak{A}$).

□

(ii): $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$ implies FQE($\mathfrak{A}$)

- By $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$ we have also $\mathrm{F-SAT}_{\mathfrak{A}} \in \mathbf{P}_{\mathfrak{A}}$.
- We use Poizat's Theorem (Theorem 2) from lecture NEWBSS.
- Let C_n be the polynomial time computable family of circuits which solves $\mathrm{F-SAT}_{\mathfrak{A}}$, and $(\exists b \phi(a,b), a)$ be an instance.
- Let $w = w(\exists b \phi(a,b))$ be the code of $\exists b \phi(a,b)$. $w \in \{0,1\}^\star$, and C_w the circuit obtained from C_n by replacing the input gates for w by the correcponding constants of w. C_w is a circuit wich accepts a iff C_n accepts wa.
- So $a \in \mathrm{Acc}_n(\mathbf{M})$ iff $\langle \mathfrak{A}, a, b \rangle \models \exists b \phi(a,b)$.
- Finally the quantifier-free formula $\psi(a)$ describing $\mathrm{Acc}_n(\mathbf{M})$ is the required formula.

□

(iii): SQE($\mathfrak{A}$) implies $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$

- It suffices to show that $\mathrm{F-SAT}_{\mathfrak{A}} \in \mathbf{P}_{\mathfrak{A}}$.
- Let $(\exists b \phi(x, b), a)$ be an instance of $\mathrm{F-SAT}_{\mathfrak{A}}$.
  By SQE($\mathfrak{A}$) there is a polynomial size quantifier-free formula $\psi(x)$ equivalent to $\exists b \phi(x, b)$.
- Evaluation of quantifier-free can be done in polynomial time.

□

To prove the theorem we need (ii) and (iii) and FQE($\mathfrak{A}$) $\Rightarrow$ SQE($\mathfrak{A}$).

(i) and (iii) give SQE($\mathfrak{A}$) iff $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$

The **true** challenge!

- For τ-structures $\mathfrak{A}$ with QE($\mathfrak{A}$):
  Find **lower** and **upper** bounds for the length of the quantifier eliminating formulas.

- The *Million Dollar Problems*:
  Prove or disprove SQE($\mathfrak{A}$) for $\mathfrak{A}$

  any finite structure
  the field of complex numbers
  the ordered field of real numbers

Note: The field of the real numbers without order has no QE.

Outline of the ESSLLI-course given by

J.A. Makowsky (Haifa) and K. Meer (Cottbus)

LECTURE 1 (JAM): Introduction INTRO (5 slides)
Turing machines over relational structures, NEWBSS, (19 slides)
Short quantifier elimination. SHORTQE (16 slides)

LECTURE 2 (JAM): Introduction to quantifier elimination QE (26 slides)
Fields, rings and other structures TABLE (incomplete, 20 slides)

LECTURE 3 (JAM): Computing with the reals: Removing order or multiplication;
Adding `Fortran`-libraries. FORTRAN (24 slides)
Comparing Poizat's Theorem with descriptive complexity FAGIN (20 slides)

LECTURE 4 (KM): Inside $\mathbf{NP}_{\mathbb{R}}$ and analogues to Ladner's Theorem, Meer-1 (149 slides)

LECTURE 5 (KM): PCP-Theorem over $\mathbb{R}$, Meer-2 (139 slides)

ADDITIONAL MATERIAL see next slide.

Additional material for the ESSLLI-course

LECTURE 6 (JAM): Quantifier elimination in algebraically closed fields, ACF-0 (21 slides)
By JAM after Kreisel and Krivine.

LECTURE 7 (JAM): $\mathbf{P}_G \neq \mathbf{NP}_G$ for all abelian groups. ABELIAN (18 slides)
By JAM After M. Prunescu

LECTURE 8 (JAM): $\mathbf{P}_G \neq \mathbf{NP}_G$ for all boolean algebras. BOOLEAN (23 slides)
By I. Bentov after M. Prunescu

LECTURE 9 (JAM): $\mathbf{P}_G \neq \mathbf{NP}_G$ for all real matrix rings. MATRIX (70 slides)
By N. Labai after A. Rybalov