Two logical approaches to the

classical $\mathbf{P} = \mathbf{NP}$ problem

In this lecture we want to compare

- Poizat's Fast QE Theorem

  dealing with $\mathbf{P}_{\mathfrak{A}} = \mathbf{NP}_{\mathfrak{A}}$ for fixed finite τ-structures, and quantifier elimination in the first order theory of $\mathfrak{A}$.

- Fagin's Theorem

  dealing with classes of finite τ-structures definable in fragments of second order logic SOL(τ).

Poizat's fast QE Theorem

Theorem 8 (Poizat, see lecture 1)

Let $\mathfrak{A}$ be a τ-structure.

$\mathfrak{A}$ can be finite or infinite, ordered or without order.

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)**.

Poizat's Theorem for $\mathfrak{A} = \mathbb{Z}_2$, I

We look at binary words $w \in \{0,1\}^*$.

- Let $w = 0^m 1^m$ of size $n = 2m$. This is accepted or rejected in polynomial time machine M_{0m1m}.

- How does the accepting formula look like ?

  For fixed n it must be a formula $\phi_n(v_1, \ldots, v_n)$ in n free variables.

- For $n = 2m + 1$ odd ϕ_n is unsatisfiable.

- For $n = 2m$ even we can write

$$\phi_n(v_1, \ldots, v_n) = \left(\bigwedge_{i=1}^{m} v_i = 0 \wedge \bigwedge_{i=m+1}^{2m} v_i = 1 \right)$$

Poizat's Theorem for $\mathfrak{A} = \mathbb{Z}_2$, II

- Let $L_1 \subseteq \{0,1\}^*$ be a regular language.

  To construct the accepting formula for input size n we use the Myhill-Nerode Theorem to construct ϕ_n inductively.

  More on the blackboard.

- Let L_2 be the set of words w of size $2m$ with an equal number of 0's and 1's.
  Given $w \in L_2$ we can re-order the letters to get a word $w' = 0^m 1^m$.
  This can be expressed as

$$\phi_n(v_1, \ldots, v_n) = \exists u_1, \ldots, u_n \left(\bigwedge_{i=1}^{n} u_i = v_i \wedge \bigwedge_{i=1}^{m} u_i = 0 \wedge \bigwedge_{i=m+1}^{2m} u_i = 1 \right)$$

  How does the quantifier-free formula look like which is equivalent to ϕ_n and of size polynomial in n?

Words in $\mathfrak{A}$ and words as τ_{word}-structures

In this course, so far, we have looked at words $w \in A^\star$ for a fixed τ-structure $\mathfrak{A}$.

Now we shall look at words $w \in A^\star$ of length n as τ^A_{word}-structures with universe $[n] = \{0, 1, \ldots, n-1\}$.

- τ^A_{word} consists of a binary relation $<$, unary predicates symbols P_a for each $a \in A$, and two constant symbols c_{first} and c_{last}.
- For a word $w \in A^\star$ of length n, $<_w$ is the natural linear order on $[n]$, with c_{first} and c_{last} the first and last element respectively.
- We think of $i \in [n]$ as position of the word w, and put $i \in P_a$ iff the i-th letter of w is a.

In the sequel we write τ_A for τ^A_{word}.

For a word w we write $\mathfrak{A}_w$ for the corresponding τ_A-structure.

Second order logic SOL(τ)

Second order logic SOL(τ) is defined like first order logic FOL(τ), with additional quantification rules:

- First order variables $v_i, i \in \mathbb{N}$, and second order variables $U_{i,r}, i, r \in \mathbb{N}$.
- τ-terms and atomic τ-formulas are as in FOL(τ);
- Additionally, $U_{i,r}(t_1, \ldots, t_r)$ are atomic formulas, where $t_1, \ldots, t_r$ are terms.
- SOL(τ) is closed under boolean operations and first order quantification rules;
- Additionally, we have existential and universal quantification for second order variables.

$$\forall U_{i,r} \text{ and } \exists U_{i,r}$$

The semantics is defined in the natural way, where an assignment z assigns elements of $[n]$ to first order variables, and $z(U_{i,r}) \subseteq [n]^r$.

Examples

- A word $w = aa^*b*b$ is characterized by:

  "there is a position i such that all positions $j < i$ contain the letter a and all positions $j \geq i$ contain the letter b"

  This can be expressed as

$$\exists i((\forall j(j < i \rightarrow P_a(j)) \wedge (\forall j(j \geq i \rightarrow P_b(j)))$$

- A word $w = (ab^+)^+a$ is characterized by

  "there is a set $U \subset [n]$ of positions, including the first and last position, holding the letter a, such that all positions $j \notin U$ hold the letter b"

  This can be expressed as

$$\exists U_{1,1}(U_{1,1}(c_{first}) \wedge U_{1,1}(c_{loast}) \wedge (\forall j U_{1,1}(j) \rightarrow P_a(j)) \wedge (\forall j \neg U_{1,1}(j) \rightarrow P_b(j)))$$

The Büchi-Elgot-Trakhtenbrot (BET) Theorem

Monadic second order logic MSOL(τ) is like SOL(τ) but with second order variables restricted to unary variables $U_{i,1}$.

Existential second order logic ESOL(τ) consists of existential formulas of the form $\exists U_{1,r_1}, \ldots, U_{m,r_m}\phi$
where $\phi \in$ FOL(τ) is a first order formula.

A language L is a set of words, $L \subset A^*$.

A language is regular if it is recognizable by a finite automaton.

Theorem 9 (BET, ca. 1960)

A language $L \subseteq A^*$ is regular iff the set of finite τ_A-structures

$$\{\mathfrak{A}_w : w \in L\}$$

is definable by a monadic second order formula $\Phi_L \in$ MSOL(τ), i.e.

$$w \in L \text{ iff } \mathfrak{A}_w \models \Phi_L$$

Ordered finite structures

For some of the theorems we need that finite τ structures have a predicate expressing that the universe is linearly orderd.

- We therefore assume that the universe is a set $[n] = \{0, 1, \ldots, n-1\}$ and the ordering is the natural order on $[n]$.

- Expressing that $[n]$ is odd over the empty vocabulary is NOT possible in MSOL.

- In the presence of the ordering this is possible by saying that the last element is reachable by jumps of two along the order.

  In fact we need the order to express this, but any other order on $[n]$ would serve the same purpose.

Some classes of graphs definable in SOL(τ_{graph})

- Graphs of even cardinality, of even degree. order is needed !
- regular graphs, and regular graphs of degree d.
- Connected graphs
- Eulerian graphs
- 3-colorable graphs
- Hamiltonian graphs

To be discussed on the blackboard.

Coding finite τ-structures as words

Let $w(\mathfrak{A})$ denote a fixed coding of the structure $\mathfrak{A}$

- Let L_τ be the language which consists of all the words coding some finite τ-structure, i.e.,

$$L_\tau = \{w(\mathfrak{A}) : \mathfrak{A} \text{ is } \text{ is a } \tau - \text{structure}\}$$

- We assume that $L_\tau \in \mathbf{P}$.

- We also assume that given a word $w \in L_\tau$ we can find in polynomial time a τ-structure $\mathfrak{A}$ such that $w(\mathfrak{A}) = w$.

  To make this more precise, we assume that the universe A of $\mathfrak{A}$ is $A = [n]$ for some $n \in \mathbb{N}$. The structure is given if we can determine in polynomial time whether a tuple $k_1, \ldots, k_r$ is in the relation R of arity r.

Defining $\mathbf{P}$ and $\mathbf{NP}$ for classes of structures

Let $w(\mathfrak{A})$ denote the coding of the structure $\mathfrak{A}$ as a word.

- A class of finite τ-structures $\mathcal{C}$ is said to be in $\mathbf{P}$ if the language $L_{\mathcal{C}}$ is in $\mathbf{P}$.
- A class of finite τ-structures $\mathcal{C}$ is said to be in $\mathbf{NP}$ if the language $L_{\mathcal{C}}$ is in $\mathbf{NP}$.

Some classes of graphs and their complexity

In $\mathbf{P}$:

- Graphs of even cardinality, of even degree. order is needed !
- regular graphs, and regular graphs of degree d.
- Connected graphs
- Eulerian graphs

$\mathbf{NP}$-complete:

- 3-colorable graphs
- Hamiltonian graphs

To be discussed on the blackboard.

Fagin's Theorem

Let τ be a relational vocabulary
(possibly without a binary relation for the ordering of the universe).

Theorem 10 (Fagin 1975)

Let $\mathcal{C}$ be a set of finite τ-structures. The following are equivalent:

- $\mathcal{C} \in \mathbf{NP}$;
- *there is a τ-formula $\phi \in \mathrm{ESOL}(\tau)$ such that*

$$\mathfrak{A} \in \mathcal{C} \text{ iff } \mathfrak{A} \models \phi$$

The fragment HornESOL(τ).

- A quantifier-free τ-formula is a Horn clause if it is a disjunction of atomic or negated atomic formulas where at most one is not negated.

$$\neg\alpha_1 \vee \neg\alpha_2 \vee \ldots \vee \neg\alpha_n \vee \beta$$

  where α_i, β are atomic.

- A quantifier-free τ-formula is a Horn formula if it is a conjunction of Horn clauses.

- A formula $\phi \in SOL(\tau)$ is in $HSOL(\tau)$ of it is of the form

$$\exists U_{1,r_1}, U_{2,r_2}, \ldots, U_{k,r_k} \forall v_1, \ldots, v_m H(v_1, \ldots, v_m, U_{1,r_1}, U_{2,r_2}, \ldots, U_{k,r_k})$$

  where H is a Horn formula and v_i are first order variables.

Some classes of graphs definable in HornESOL(τ_{graph})

- Graphs of even cardinality, of even degree. order is needed !
- regular graphs, and regular graphs of degree d.
- Connected graphs
- Eulerian graphs

To be discussed on the blackboard.

The Immerman-Vardi-Grädel Theorem (IVG)

Let τ be a relational vocabulary with a binary relation for the ordering of the universe.

Theorem 11 (Immermann, Vardi, Graedel 1980-4)

Let $\mathcal{C}$ be a set of finite τ-structures. The following are equivalent:

- $\mathcal{C} \in \mathbf{P}$;
- *there is a τ-formula $\phi \in$ HornESOL(τ) such that $\mathfrak{A} \in \mathcal{C}$ iff $\mathfrak{A} \models \phi$.*

Here the presence of the ordering is crucial:

Without it the class of structures for the empty vocabulary of even cardinality is in $\mathbf{P}$, but not definable in HornESOL.

Conclusion: The logical equivalent to $\mathbf{P} = \mathbf{NP}$

Let τ be a relational vocabulary which contains
a binary relation for the ordering of the universe.

The following are equivalent:

- $\mathbf{P} = \mathbf{NP}$ in the classical framework.
- There is a finite τ-structure $\mathfrak{A}$ such that
  the first order theory $\mathrm{Th}(\mathfrak{A})$ has fast (short) QE.
- For every finite τ-structure $\mathfrak{A}$
  the first order theory $\mathrm{Th}(\mathfrak{A})$ has fast (short) QE.
- Every ESOL(τ)-formula is equivalent over finite ordered τ-structures to
  some HornESOL(τ)-formula.

Thank you all !

Homepage: www.cs.technion.ac.il/∼janos

e-mail: `janos@cs.technion.ac.il`

The slides will be posted within a few days on my homepage as

ESSLLI-2014-Lectures

Check regularly.

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