Structures with \( P = \text{NP} \) with Respect to the Uniform Model of Computation

The uniform model of computation:
- The size of an input \((x_1, \ldots, x_n)\): \(n\).
- The cost of the execution of one instruction: 1.
Concepts and Notations

\[ \Sigma = (S; c_1, \ldots, c_k; g_1, \ldots, g_k; r_1, \ldots, r_k, =) \]

\( \Sigma \) is a structure of finite signature, \( k \geq 2 \).

\[ A_1, A_2 \subseteq S^\infty = \bigcup_{n=1}^\infty S^n \]

\( A_1, A_2 \) are problems.

\[ f : S^\infty \mapsto S^\infty \text{ with } f(x) \in A_2 \iff x \in A_1 \]

\( f \) is a function reducing \( A_1 \) to \( A_2 \).

\[ A_1 \leq_{\text{pol}} A_2, \quad A_1 \to_{[\text{pol}]} A_2 \]

\( A_1 \) can be reduced to \( A_2 \) by a function in polynomial time.

\[ A_1 \to_{f \text{ pol}} A_2 \]

The function \( f \) reduces \( A_1 \) to \( A_2 \) in polynomial time.

\[ A \to_R \{c_1\} \]

\( A \) is decidable by means of \( R \) in constant time.

\[ \forall (A \in \text{NP}) (A \leq_{\text{pol}} A_0) \]

\( A_0 \) is NP-hard.

\[ \text{SAT} \]

The Satisfiability Problem.
Conditions for $P = NP$

We have $P = NP$ for a structure if we can show one of the following reductions for each problem $A \in NP$.

\[ (1) \quad A \xrightarrow{f_1} SAT \xrightarrow{f_2} \{c_1\} \]

or

\[ (2) \quad A \subseteq S^\infty \xrightarrow{f_1} SAT \subseteq S^\infty \xrightarrow{f_2} A_0 \subseteq S \xrightarrow{R} \{c_1\} \subseteq S \]

The idea to define a new relation goes back to B. Poizat.

Tuples are encoded by means of single elements.

and so on.
What about a Tree of Computation Paths?

Symbols:

A sequence of instructions without branching
(computations defined by the operations, copy instructions)

A condition of a branching instruction
(a literal defined by a relation)

Input

t steps of the execution

Structures with $P = NP$, Christine Gaßner, Ernst-Moritz-Arndt-Universität, Greifswald
What about the Canonical Path? – An Example (A)

A structure: \((\mathbb{Z}; 0, 1; +, \cdot; \equiv)\).

A program of a machine:

1: Input \((x_1, \ldots, x_n)\);  
\(\text{size: } n \)
2: \(x_2 := 0;\)
3: \(x_2 := x_2 + 1;\)
4: \(x_3 := x_2 \cdot x_2;\)
5: if \(x_1 = x_3\) then goto 6 else goto 3;
6: \(x_1 := 1;\)
7: Output \(x_1\).

The canonical path is traversed by all tuples (vectors of components from \(\mathbb{Z}\)) whose first component is not a square number.
What about the Canonical Path? – An Example (B)

All inputs accepted by the machine only traverse a finite initial segment of the canonical path.

- The $n$-tuples $(1, x_2, \ldots, x_n)$, $(4, x_2, \ldots, x_n)$, \ldots, $((p(n))^2, x_2, \ldots, x_n)$ are accepted by the machine in $4 + 3p(n)$ steps.

For all the other inputs, the machine has the same behaviour during the first $4 + 3p(n)$ steps. These inputs can not be accepted in this time.

There is no more information up to here.
A Structure of Paths Denoted by Strings and the Reduction

\[
\begin{align*}
A \subseteq (S^*)^\infty & \rightarrow SAT \subseteq (S^*)^\infty \\
A_0 \subseteq S^* & \rightarrow R \{c_1\} \\
\end{align*}
\]

Particularly,

\[(x_1, \ldots, x_n) \in S^n \rightarrow (x_1, \ldots, x_n, z_1, \ldots, z_{p(n)}) \rightarrow x'_1 \ldots x'_n z'_1 \ldots z'_{p(n)} c_1 \cdots c_1 \rightarrow \ldots .\]

\(A \in \text{NP}.
\]

Let \((z_1, \ldots, z_{p(n)}) \in \{c_1, c_2\}^{p(n)}\) be the code of a formula which is true for any tuple from \((S^*)^n\) iff this tuple is accepted by the NP-machine recognized \(A\) non-deterministically.

(We use unary codes for the indices.)

A path with nodes labelled by elements of \(S\) or a string over the alphabet \(S\).

\[
\begin{align*}
x'_1 & \ x'_2 \ x'_{n-1} \ x'_n \ z'_1 & \ z'_2 \ z'_{p(n)-1} z'_{p(n)} & \ c_1 & \ c_1 & \ c_1
\end{align*}
\]

SAT described by a tree.
Problems for which We Need a Solution

• The tuples have to be encoded by means of single elements.
  ➢ Extension of the structure.
  We shall use paths (denoted by strings) as elements.

• We need a new operation for the concatenation of paths or strings.
  ➢ Expansion of the structure.

• We need a new relation.
  ➢ Expansion of the structure.
Problems in Defining $R$ and a Solution

• For the new structure: $SAT$ is decidable $\iff R$ is definable.
  
  ➢ The different evaluations of the free variables have to be finitely describable.

  We shall replace arbitrary paths by short paths.
  (As the replacements of numbers in a paper by P. Koiran.)

• The formulae of $SAT$ contain $R$.
  
  ➢ Recursive definition of $R$.

• The formulae of $SAT$ also contain free variables.
  
  ➢ If there is an evaluation of the variables satisfying a formula from $SAT$, then we also want to have an evaluation of these variables by paths which are shorter than the code of the corresponding tuple from $SAT$ (in general).

  We use operations allowing few modifications of paths only such that we can restrict the free variables in the formulae with “=” to paths of polynomial length.
⇒ The new structure:

\[ \Sigma^{\text{string}} = (S^*; c_1, \ldots, c_k, \lambda; g_1', \ldots, g_k', \text{add}, \text{sub}_{\ell}, \text{sub}_r; r_1', \ldots, r_k', R, \text{=}). \]

\[ w \in S^*, \ x \in S, \ \nu \in S^* \setminus S \quad \Rightarrow \quad \text{add}(w, x) = wx, \ \text{add}(w, \nu) = \lambda, \]

\[ \text{sub}_{\ell}(\lambda) = \lambda, \ \text{sub}_{\ell}(wx) = w, \]

\[ \text{sub}_{r}(\lambda) = \lambda, \ \text{sub}_{r}(wx) = x, \]

\[ x_1, x_2, \ldots \in S \quad \Rightarrow \quad g_i'(x_1, \ldots, x_{m_i}) = g_i(x_1, \ldots, x_{m_i}), \quad r_j'(x_1, \ldots, x_{l_j}) = r_j(x_1, \ldots, x_{l_j}), \]

\[ (v_1, \ldots, v_{m_i}) \in (S^*)^{m_i} \setminus S^{m_i} \quad \Rightarrow \quad g_i(v_1, \ldots, v_{m_i}) = \lambda, \]

\[ (v_1, \ldots, v_{l_j}) \in (S^*)^{l_j} \setminus S^{l_j} \quad \Rightarrow \quad r_j'(v_1, \ldots, v_{l_j}) = \text{false}. \]
Example: \( \Sigma_0 = \langle \{0, 1\}; 0,1; ; \rangle \Rightarrow \Sigma_0^{\text{string}} = \langle \{0, 1\}^*; 0,1,\lambda; \text{add}, \text{sub}_l, \text{sub}_r; R,= \rangle \)

The literals in the formulae in \( SAT_0^{\text{string}} (= SAT \text{ for the structure } \Sigma_0^{\text{string}}) \) have the form:

\[
\begin{align*}
\text{add}(U, V) &= W, & R(U), & U = V, \\
\text{add}(U, V) &\neq W, & \neg R(U), & U \neq V.
\end{align*}
\]

1. A conjunction with \textit{add} and \textit{“=”} can describe connections between strings like the following.

\[
\begin{align*}
\text{add}(V_0, a_0) &= V_1 & \cdots & \& \text{add}(V_{j-1}, a_{j-1}) &= V_j & (a_0, ..., a_{j-1} \in \{0,1\}) \\
\Rightarrow & V_j = V_0a_0 \cdots a_{j-1} & \& j < \text{length of the code of the conjunction}.
\end{align*}
\]

The lengths of these segments of the strings are less than the length of the code of the conjunction.

String \( V_0 \) with an arbitrary length.

The characters of \( V_0 \) can not be determined or restricted by means of the given description.
2. Assume that $l$ is greater than $m$ (= the length of the code of the conjunction) and one of the strings satisfies $R$.

Assume that the string $V_00^j$ satisfies $R$.

The behaviour of the given strings with respect to $R$ should be only dependent on the characters of these parts (at most $m$ characters).

We want to have at most one string $w$ with $R(V_0w) = true$. Then, we only have to know at most one number $j < m$ with $R(V_00^j) = true$. The characters of $V_0$ do not play a role.
⇒ For each formula:
   We restrict the free variables to a domain of short paths (or strings).

   The lengths of the paths in such a domain are only dependent on the
   length of the code of this formula and of the lengths of the paths in the
   bounded variables.
⇒ New problem \textit{RES-SAT} (with \textit{RES-SAT} = \textit{SAT}).

⇒ We describe \( R \) by means of a tree being similar to a tree of
   computation paths.

⇒ The behaviour of the paths and their modifications with respect to \( R \)
   has to be similar to the behaviour of the inputs traversing computation
   paths.
The Tree Describing $R$

Each string corresponding to a path from a leaf to the root satisfies $R$.

All nodes of this path are labelled by the same constant $c_1$.

Here we consider an arbitrary structure (→infinite alphabet). For $\Sigma_0^{string}$, we can take a binary tree.

Structures with $P = NP$, Christine Gaßner, Ernst-Moritz-Arndt-Universität, Greifswald
A Path Corresponding to a String Satisfying $R$

This segment without important information has the length $l$.

This segment containing the information has the length $l$.

Each path from the root to a leaf consists of two segments of the same length.
Connections between Different Strings

\[ R(w_2 c_1^l) \Rightarrow R(v_2 c_1^l) \]

\[ w_1 = w_2 c_1^m \]

\[ v_1 = v_2 c_1^m \]

Structures with P = NP, Christine Gaßner, Ernst-Moritz-Arndt-Universität, Greifswald
The Replacement of $w_1$ by $v_1$ and of $w_2$ by $v_2$ for a Given Formula

Only the last characters are important for the truth value of the given formula.

The width of this segment is dependent on the length of the code of the considered formula.
Few Definitions for \( \Sigma_0^{\text{string}} \)

\[
\text{SAT}_0^{\text{string}} = \{(x_1, \ldots, x_n, z_1, \ldots, z_m) \mid n, m \in \mathbb{N}^+ \land (x_1, \ldots, x_n) \in (\{0, 1\}^*)^n \\
\land (z_1, \ldots, z_m) \in \{0, 1\}^m \text{ is the code of a formula } \varphi(X_1, \ldots, X_n, Y_1, \ldots, Y_l) \text{ of the first order logic over } \Sigma_0^{\text{string}} \\
\land \text{ there is a } (y_1, \ldots, y_l) \text{ with } \Sigma_0^{\text{string}} \models \varphi(x_1, \ldots, x_n, y_1, \ldots, y_l) \}\.
\]

\[
M_{j, k} = \{0^i 1^{j-i}s \mid s \in \{0, 1\}^* \land |s| < k \land 1 \leq i \leq j - 4\}
\]

For all \( s \in M = \bigcup_{j, k \in \mathbb{N}} M_{j, k} \), let \( R(s) = \text{true} \) iff \( s = 0^i 1^{j-i}0^j \) for some \( 1 \leq i \leq j - 4 \). \( R(\lambda) = \text{false} \). For \( s \in \{0, 1\}^* \setminus M \) with \( |s| > 0 \), let \( R(s) = \text{true} \) iff

\[
s = \overline{x_1}11\overline{x_2}11\cdots\overline{x_n}110z_10 z_2\cdots0z_m1100 \cdots 00\)
\]

\[
t \text{ characters}
\]

\[
x_i = x_{i,1}x_{i,2}\cdots x_{i,m_i}
\]

\[
\Rightarrow \overline{x_i} = 0x_{i,1}0x_{i,2}\cdots0x_{i,m_i}
\]

and

\[
\Sigma_0^{\text{string},R'} \models \bigvee (y_1, \ldots, y_l) \in (\{y \mid y \in \{0,1\}^* \land |y| < 2t\} \cup M_{t, t+1})^l
\]

for all \( R' \) with \( \forall(w \in \{y \mid y \in \{0,1\}^* \land |y| < 2t\} \cup M_{t, t+1}) \) \( (R'(w) = R(w)) \) where \( \Sigma_0^{\text{string},R'} = (\{0, 1\}^*; 0, 1, \lambda; \text{add}, \text{sub}_l, \text{sub}_r; R', =) \) for any relation \( R' \).
Replacements

Example – Replacements of long paths for free variables if the considered formula $\varphi$ does not describe connections between these variables and bounded variables:

1. $|V_i| > m, R(V_i, 0^j) = true$ for some $j < m$.

$$4 \leq i, m = \text{the length of the code of } \varphi \Rightarrow i < m$$

2. $|V_i| > m$ and $R(V_i, 0^j) = false$ for all $j < m$.

Structures with P = NP, Christine Gaßner, Ernst-Moritz-Arndt-Universität, Greifswald
The Decision of $A \in \text{NP}$ in Polynomial Time

$$(w_1, \ldots, w_n)$$

$f_1$ ⇒ the code of a formula which describes the acceptance of the inputs from $A \cap (\{0,1\}^*)^n$

$$(w_1, \ldots, w_n, z_1, \ldots, z_{p(n)})$$

$\in A \subseteq (\{0,1\}^*)^\infty$

$A$ problem where the free variables are restricted

$\in SAT = RES-SAT \subseteq (\{0,1\}^*)^\infty$

$w_i = w_{i,1} \cdots w_{i,t} + 1 \cdots w_{i,m_i}$

replace $w'_{i} = w_{i,1} \cdots w_{i,t}, w'_{i,t+1} \cdots w'_{i,m_i}$

$$(w'_1, \ldots, w'_n, z_1, \ldots, z_{p(n)})$$

$f_2$ ⇒ $w'_i \Rightarrow w_i = 0w_{i,1}0w_{i,2} \cdots 0w_{i,t}, 0w'_i,t_1+1 \cdots 0w'_i,m_i$

$$(w_1 11 \cdots w_n 11 0z_1 0z_2 \cdots 0z_{p(n)} 11 00 \cdots 0)$$

$\in A_0 \subseteq \{0,1\}^*$

$R$

$0 / 1$

$\in \{1\} \subseteq \{0,1\}$
Thank you!

Christine Gaßner

Ernst-Moritz-Arndt-Universität, Greifswald

I thank Mihai Prunescu, Gunther Mainhardt, and Günther Asser for discussions.

I thank Mihai Prunescu and Pascal Koiran for suggestions.

Authors of references: Lenore Blum, Felipe Cucker, Michael Shub, Steve Smale, Armin Hemmerling, Klaus Meer, Pascal Koiran, Bruno Poizat