Language equivalence is undecidable [Hütte’94].

Strong bisimilarity is decidable [Christensen, Hirshfeld, Moller’93], even in PSPACE [Jančar’03].

We will argue how to show PSPACE-hardness [Srba’02].

**Summary of Results for BPP**

- Language equivalence is undecidable [Hütte’94].
- Strong bisimilarity is decidable [Christensen, Hirshfeld, Moller’93], even in PSPACE [Jančar’03].
- We will argue how to show PSPACE-hardness [Srba’02].

**PSPACE-Hardness of Strong Bisimilarity for BPP**

Jiri Srba, BRICS Aalborg
30.11 – 1.12 2005
PhD Course at FIRST Graduate School, IT University

**BPP - Basic Parallel Processes**

Basic Parallel Processes (BPP)

(1, P)-PRS

Rule type:

\[ X \xrightarrow{a} Y | W | Z \]

- a basic model of purely parallel programs
- fragment of CCS without restriction, relabelling and communication
- equivalent to communication free subclass of Petri nets

**Example**

Let \( C \) be a QSAT formula.

For a QSAT formula \( C \) we construct a BPP system with two processes \( X \) and \( X' \) such that:

\( C \) is true if and only if \( X \sim X' \).

Quantified Satisfiability (QSAT) or also Quantified Boolean formula (QBF) problem is PSPACE-complete.

**Problem:** QSAT

**Instance:** A natural number \( n > 0 \) and a Boolean formula \( \phi \) in conjunctive normal form with Boolean variables \( x_1, \ldots, x_n \) and \( y_1, \ldots, y_m \).

**Question:** Is \( \exists x_1 \forall y_1 \exists x_2 \forall y_2 \ldots \exists x_n \forall y_m \phi \) true?

**Example:**

\[ \exists x_1 \forall y_1 \exists x_2 \forall y_2 \ldots (x_1 \lor y_1 \lor y_2) \land (x_1 \lor y_1 \lor y_2 \lor \neg y_2) \land (x_1 \lor y_1 \lor y_2 \lor \neg y_2) \]

For a QSAT formula \( C \) we construct a BPP system with two processes \( X \) and \( X' \) such that:

\( C \) is true if and only if \( X \sim X' \).
How to Represent Clauses

Let us fix a QSAT formula $C$:

$$C ≡ ∃ x_1∀y_1∃x_2∀y_2 . . . ∃x_n∀y_n. C_1 ∧ C_2 ∧ . . . ∧ C_k$$

New process constants $Q_1, . . . , Q_k$ such that for all $j, 1 ≤ j ≤ k$:

$$Q_j \overset{α_j}{→} Q_j$$

Example:

Satisfied clauses $C_1$, $C_3$ and $C_4$ are represented by $Q_1 | Q_3 | Q_4$.

We present a construction enabling the defender to force the attacker to perform a certain move (Defender’s Choice).

We show a way how to remember (encode) and check satisfied clauses.

Let

$$C ≡ \exists x_1\forall y_1\exists x_2\forall y_2 . . . \exists x_n\forall y_n. C_1 ∧ C_2 ∧ . . . ∧ C_k$$

Let

$$α_j ≡ Q_{i_1} | Q_{i_2} | . . . | Q_{i_ℓ}$$ such that $x_i$ occurs positively in $C_{i_1}, C_{i_2}, . . . , C_{i_ℓ}$

$$β_j ≡ Q_{i_1} | Q_{i_2} | . . . | Q_{i_ℓ}$$ such that $y_i$ occurs positively in $C_{i_1}, C_{i_2}, . . . , C_{i_ℓ}$

$$γ_j ≡ Q_{i_1} | Q_{i_2} | . . . | Q_{i_ℓ}$$ such that $y_i$ occurs negatively in $C_{i_1}, C_{i_2}, . . . , C_{i_ℓ}$

$$δ_j ≡ Q_{i_1} | Q_{i_2} | . . . | Q_{i_ℓ}$$ such that $y_i$ occurs negatively in $C_{i_1}, C_{i_2}, . . . , C_{i_ℓ}$

Reduction from QSAT to $\sim$ of BPP

We show a way how to remember (encode) and check satisfied clauses.
Strong bisimilarity on BPP is PSPACE-Complete

Theorem
Strong bisimilarity on BPP is PSPACE-hard.

Theorem
Strong bisimilarity on BPP is PSPACE-complete.

Generation Phase from X and X'

Checking Clauses

\[ Z | \alpha_1 | \overline{\alpha}_2 | \overline{\alpha}_2 \]

\[ Z' | \alpha_1 | \overline{\alpha}_2 | \overline{\alpha}_2 \]

\[ Z \xrightarrow{a} Q_1 | Q_2 | \cdots | Q_k \]

\[ Z' \xrightarrow{a} \epsilon \]