Seminar Decision Procedures and Applications

Background Informations

Viorica Sofronie-Stokkermans
University Koblenz-Landau

2 June 2015
Topics for the talks

- **Julia Schöenberger**: Automata approach to Presburger arithmetic (in chapter by Hubert Common and Claude Kirchner)

- **David Nagel**: An efficient decision procedure for Unit Two-Variables per Inequality (UTVPI) Constraints (Work by Shuvendu Lahiri and Madanlal Musuvathi)

- **David Friedrich**: Abstract Congruence Closure (work by Leo Bachmair and Ashish Tiwari)

- **Cynthia Engel**: Shostak’s method (Work by R.E. Shostak and Harald Ganzinger)

- **Shuyi Weng**: What’s decidable about arrays (Work by A. Bradley, Z. Manna and H. Sipma)

- **Kilian Laudt**: Decision procedures for recursive data structures with integer constraints (Work by T. Zhang, H.B. Sipma, Z. Manna)

- **Carl Brenk**: Terminological cycles in a description logic with existential restriction (Work by Franz Baader)
Possible Schedule

- 9.06.2015: Short presentations (5 min)
- 30.06.2015: Background TRS (VSS)
- 7.07.2015: Arithmetic
  - Julia Schönberger: Automata approach to Presburger arithmetic
  - David Nagel: An efficient decision procedure for Unit Two-Variables per Inequality (UTVPI) Constraints
- 14.07.2015: Congruence Closure, Combinations
  - David Friedrich: Abstract Congruence Closure
  - Cynthia Engel: Shostak’s method
- 21.07.2015: Applications
  - Shuyi Weng: What’s decidable about arrays
  - Kilian Laudt: Decision procedures for recursive data structures with integer constraints
  - Carl Brenk: Terminological cycles in a description logic with existential restriction
Last time

- **Reasoning in standard theories**
  - A crash course: Decidable logical theories and theory fragments

- **Reasoning in complex theories**
  - Modular reasoning in combinations of theories
    - disjoint signature: the Nelson-Oppen method

- **Applications**
Today

• Reasoning in standard theories

  A crash course: Decidable logical theories and theory fragments

Reasoning in complex theories

  Modular reasoning in combinations of theories
  disjoint signature: the Nelson-Oppen method

• Applications
Reasoning in combinations of theories

We are interested in testing satisfiability of ground formulae
Combination of theories
Combinations of theories and models

Forgetting symbols

Let $\Sigma = (\Omega, \Pi)$ and $\Sigma' = (\Omega', \Pi')$ s.t. $\Sigma \subseteq \Sigma'$, i.e., $\Omega \subseteq \Omega'$ and $\Pi \subseteq \Pi'$

For $A \in \Sigma'$-alg, we denote by $A_{|\Sigma}$ the $\Sigma$-structure for which:

$$U_{A_{|\Sigma}} = U_A, \quad f_{A_{|\Sigma}} = f_A \quad \text{for } f \in \Omega;$$
$$P_{A_{|\Sigma}} = P_A \quad \text{for } P \in \Pi$$

(ignoring functions and predicates associated with symbols in $\Sigma' \setminus \Sigma$)

$A_{|\Sigma}$ is called the restriction (or the reduct) of $A$ to $\Sigma$.

Example: $\Sigma' = (\{\div/2, \times/2, \div0\}, \{\leq /2, \text{even}/1, \text{odd}/1\})$
$\Sigma = (\{\div/2, \div0\}, \{\leq /2\}) \subseteq \Sigma'$
$\mathcal{N} = (\mathbb{N}, +, *, 1, \leq, \text{even, odd}) \quad \mathcal{N}_{|\Sigma} = (\mathbb{N}, +, 1, \leq)$
One possibility of combining theories

**Syntactic view:** $\mathcal{T}_1 + \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X)$

$\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \{ \mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2 \}$

where $\Sigma_1 \cup \Sigma_2 = (\Omega_1, \Pi_1) \cup (\Omega_2, \Pi_2) = (\Omega_1 \cup \Omega_2, \Pi_1 \cup \Pi_2)$
One possibility of combining theories

**Syntactic view:** $\mathcal{T}_1 + \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X)$

$\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \{ \mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2 \}$

**Semantic view:** Let $\mathcal{M}_i = \text{Mod}(\mathcal{T}_i), i = 1, 2$

$\mathcal{M}_1 + \mathcal{M}_2 = \{ \mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A}|_{\Sigma_i} \in \mathcal{M}_i \text{ for } i = 1, 2 \}$
One possibility of combining theories

**Syntactic view:** \( T_1 + T_2 = T_1 \cup T_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X) \)

\[
\text{Mod}(T_1 \cup T_2) = \{ A \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid A \models G, \text{ for all } G \text{ in } T_1 \cup T_2 \}
\]

**Semantic view:** Let \( M_i = \text{Mod}(T_i), i = 1, 2 \)

\[
M_1 + M_2 = \{ A \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid A|_{\Sigma_i} \in M_i \text{ for } i = 1, 2 \}
\]

\[
A \in \text{Mod}(T_1 \cup T_2) \iff A \models G, \text{ for all } G \text{ in } T_1 \cup T_2 \\
\text{iff } A|_{\Sigma_i} \models G, \text{ for all } G \text{ in } T_i, i = 1, 2 \\
\text{iff } A|_{\Sigma_i} \in M_i, i = 1, 2 \\
\text{iff } A \in M_1 + M_2
\]
One possibility of combining theories

**Syntactic view:** $\mathcal{T}_1 + \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \subseteq F_{\Sigma_1 \cup \Sigma_2}(X)$

$\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \{ \mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{T}_1 \cup \mathcal{T}_2 \}$

**Semantic view:** Let $\mathcal{M}_i = \text{Mod}(\mathcal{T}_i), i = 1, 2$

$\mathcal{M}_1 + \mathcal{M}_2 = \{ \mathcal{A} \in (\Sigma_1 \cup \Sigma_2)\text{-alg} \mid \mathcal{A}|_{\Sigma_i} \in \mathcal{M}_i \text{ for } i = 1, 2 \}$

**Remark:** $\mathcal{A} \in \text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2)$ iff $(\mathcal{A}|_{\Sigma_1} \in \text{Mod}(\mathcal{T}_1) \text{ and } \mathcal{A}|_{\Sigma_2} \in \text{Mod}(\mathcal{T}_2))$

**Consequence:** $\text{Th}(\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2)) = \text{Th}(\mathcal{M}_1 + \mathcal{M}_2)$
Example

1. **Presburger arithmetic + UIF**

   \[ \text{Th}(\mathbb{Z}_+) \cup \text{UIF} \quad \Sigma = (\Omega, \Pi) \]

   Models: \((A, 0, s, +, \{f_A\}_{f \in \Omega}, \leq, \{P_A\}_{P \in \Pi})\)

   where \((A, 0, s, +, \leq) \in \text{Mod}(\text{Th}(\mathbb{Z}_+)) \).

2. **The theory of reals + the theory of a monotone function \(f\)**

   \[ \text{Th}(\mathbb{R}) \cup \text{Mon}(f) \quad \text{Mon}(f) : \forall x, y (x \leq y \rightarrow f(x) \leq f(y)) \]

   Models: \((A, +, \ast, f_A, \{\leq\})\), where

   where \((A, +, \ast, \leq) \in \text{Mod}(\text{Th}(\mathbb{R}))\).

   \((A, f_A, \leq) \models \text{Mon}(f)\), i.e. \(f_A : A \rightarrow A\) monotone.

**Note:** The signatures of the two theories share the \(\leq\) predicate symbol
Combinations of theories

**Definition.** A theory is consistent if it has at least one model.

**Question:** Is the union of two consistent theories always consistent?

**Answer:** No. (Not even when the two theories have disjoint signatures)

**Example:**

\[\Sigma_1 = (\Omega_1, \emptyset), \quad \Sigma_2 = (\{c/0, d/0\}, \emptyset), \quad c, d \notin \Omega_1\]

\[T_1 = \{\exists x, y, z (x \not\equiv y \land x \not\equiv z \land y \not\equiv z)\}\]

\[T_2 = \{\forall x (x \equiv c \lor x \equiv d)\}\]

\[A \in \text{Mod}(T_1) \quad \text{iff} \quad |U_A| \geq 3.\]

\[B \in \text{Mod}(T_2) \quad \text{iff} \quad |U_B| \leq 2.\]
The combined \textit{decidability} problem

For $i = 1, 2$  
\begin{itemize}
  \item let $\mathcal{T}_i$ be a first-order theory in signature $\Sigma_i$
  \item assume the $\mathcal{T}_i$ ground satisfiability problem is decidable
\end{itemize}

Let $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ be a combination of $\mathcal{T}_1$ and $\mathcal{T}_2$

\textbf{Question:}
Is the $\mathcal{T}_1 \bigoplus \mathcal{T}_2$ ground satisfiability problem decidable?
Goal: Modularity

Modular Reasoning

\( \mathcal{T}_0 \): \( \Sigma_0 \)-theory.

\( \mathcal{T}_i \): \( \Sigma_i \)-theory; \( \mathcal{T}_0 \subseteq \mathcal{T}_i \) \( \Sigma_0 \subseteq \Sigma_i \).

Example:

\( \text{lists}(\mathbb{R}) \cup \text{arrays}(\mathbb{R}) \)

Can use provers for \( \mathcal{T}_1, \mathcal{T}_2 \) as blackboxes to prove theorems in \( \mathcal{T}_1 \cup \mathcal{T}_2 \)?

Which information needs to be exchanged between the provers?
Combination of theories over disjoint signatures

The Nelson/Oppen procedure

**Given:** $\mathcal{T}_1, \mathcal{T}_2$ stably infinite first-order theories with signatures $\Sigma_1, \Sigma_2$

Assume that $\Sigma_1 \cap \Sigma_2 = \emptyset$ (share only $\approx$)

$P_i$ decision procedures for satisfiability of ground formulae w.r.t. $\mathcal{T}_i$

$\phi$ quantifier-free formula over $\Sigma_1 \cup \Sigma_2$

**Task:** Check whether $\phi$ is satisfiable w.r.t. $\mathcal{T}_1 \cup \mathcal{T}_2$

**Note:** Restrict to conjunctive quantifier-free formulae

$\phi \mapsto \text{DNF}(\phi)$

$\text{DNF}(\phi)$ satisfiable in $\mathcal{T}$ iff one of the disjuncts satisfiable in $\mathcal{T}$
**Example**

[Nelson & Oppen, 1979]

<table>
<thead>
<tr>
<th>Theory</th>
<th>Symbols</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{R}$ theory of rationals</td>
<td>$\Sigma_{\mathcal{R}} = {\leq, +, -, 0, 1}$</td>
<td>$\simeq$</td>
</tr>
<tr>
<td>$\mathcal{L}$ theory of lists</td>
<td>$\Sigma_{\mathcal{L}} = {\text{car, cdr, cons}}$</td>
<td>$\simeq$</td>
</tr>
<tr>
<td>$\mathcal{E}$ theory of equality (UIF)</td>
<td>$\Sigma$: free function and predicate symbols</td>
<td>$\simeq$</td>
</tr>
</tbody>
</table>
Example

[Nelson & Oppen, 1979]

### Theories

<table>
<thead>
<tr>
<th>$\mathcal{R}$</th>
<th>theory of rationals</th>
<th>$\Sigma_{\mathcal{R}} = {\leq, +, -, 0, 1}$</th>
<th>$\simeq$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}$</td>
<td>theory of lists</td>
<td>$\Sigma_{\mathcal{L}} = {\text{car, cdr, cons}}$</td>
<td>$\simeq$</td>
</tr>
<tr>
<td>$\mathcal{E}$</td>
<td>theory of equality (UIF)</td>
<td>$\Sigma$: free function and predicate symbols</td>
<td>$\simeq$</td>
</tr>
</tbody>
</table>

### Problems:

1. $\mathcal{R} \cup \mathcal{L} \cup \mathcal{E} \models \forall x, y (x \leq y \land y \leq x + \text{car(cons}(0, x))) \land P(h(x) - h(y)) \rightarrow P(0))$

2. Is the following conjunction:

   $c \leq d \land d \leq c + \text{car(cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0)$

   satisfiable in $\mathcal{R} \cup \mathcal{L} \cup \mathcal{E}$?
Is the following conjunction:

\[ c \leq d \land d \leq c + \text{car}(\text{cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]

satisfiable in \( \mathcal{R} \cup \mathcal{L} \cup \mathcal{E} \)?
Step 1: Purification

**Given:** \( \phi \) conjunctive quantifier-free formula over \( \Sigma_1 \cup \Sigma_2 \)

**Task:** Find \( \phi_1, \phi_2 \) s.t. \( \phi_i \) is a pure \( \Sigma_i \)-formula and \( \phi_1 \land \phi_2 \) equivalent with \( \phi \)

\[
\begin{align*}
  f(s_1, \ldots, s_n) \approx g(t_1, \ldots, t_m) & \implies u \approx f(s_1, \ldots, s_n) \land u \approx g(t_1, \ldots, t_m) \\
  f(s_1, \ldots, s_n) \not\approx g(t_1, \ldots, t_m) & \implies u \approx f(s_1, \ldots, s_n) \land v \approx g(t_1, \ldots, t_m) \land u \not\approx v \\
  (\neg) P(\ldots, s_i, \ldots) & \implies (\neg) P(\ldots, u, \ldots) \land u \approx s_i \\
  (\neg) P(\ldots, s_i[t], \ldots) & \implies (\neg) P(\ldots, s_i[t \mapsto u], \ldots) \land u \approx t
\end{align*}
\]

where \( t \approx f(t_1, \ldots, t_n) \)

**Termination:** Obvious

**Correctness:** \( \phi_1 \land \phi_2 \) and \( \phi \) equisatisfiable.
Step 1: Purification

\[ c \leq d \land d \leq c + \text{car(cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]
Step 1: Purification

\[
c \leq d \land d \leq c + \text{car} \left( \text{cons}(0, c) \right) \land P(h(c) - h(d)) \land \neg P(0)
\]
Step 1: Purification

\[ c \leq d \land d \leq c + \text{car}(\text{cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]
Step 1: Purification

\[ c \leq d \land d \leq c + \text{car(cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]
Step 1: Purification

\[ c \leq d \wedge d \leq c + \text{car} (\text{cons}(0, c)) \wedge P(h(c) - h(d)) \wedge \neg P(0) \]

<table>
<thead>
<tr>
<th>( R )</th>
<th>( L )</th>
<th>( E )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c \leq d )</td>
<td>( c_1 \approx \text{car} (\text{cons}(c_5, c)) )</td>
<td>( P(c_2) )</td>
</tr>
<tr>
<td>( d \leq c + c_1 )</td>
<td></td>
<td>( \neg P(c_5) )</td>
</tr>
<tr>
<td>( c_2 \approx c_3 - c_4 )</td>
<td></td>
<td>( c_3 \approx h(c) )</td>
</tr>
<tr>
<td>( c_5 \approx 0 )</td>
<td></td>
<td>( c_4 \approx h(d) )</td>
</tr>
</tbody>
</table>
Step 1: Purification

\[ c \leq d \land d \leq c + \text{car}(\text{cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]

<table>
<thead>
<tr>
<th>( \mathcal{R} )</th>
<th>( \mathcal{L} )</th>
<th>( \mathcal{E} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c \leq d )</td>
<td>( c_1 \approx \text{car}(\text{cons}(c_5, c)) )</td>
<td>( P(c_2) )</td>
</tr>
<tr>
<td>( d \leq c + c_1 )</td>
<td>( c_2 \approx c_3 - c_4 )</td>
<td>( \neg P(c_5) )</td>
</tr>
<tr>
<td>( c_2 \approx c_3 - c_4 )</td>
<td>( c_5 \approx 0 )</td>
<td>( c_3 \approx h(c) )</td>
</tr>
<tr>
<td>satisfiable</td>
<td>satisfiable</td>
<td>satisfiable</td>
</tr>
</tbody>
</table>

satisfiable
Step 2: Propagation

$$c \leq d \land d \leq c + \text{car}(\text{cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0)$$

<table>
<thead>
<tr>
<th>R</th>
<th>L</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c \leq d$</td>
<td>$c_1 \approx \text{car}(\text{cons}(c_5, c))$</td>
<td>$P(c_2)$</td>
</tr>
<tr>
<td>$d \leq c + c_1$</td>
<td></td>
<td>$\neg P(c_5)$</td>
</tr>
<tr>
<td>$c_2 \approx c_3 - c_4$</td>
<td></td>
<td>$c_3 \approx h(c)$</td>
</tr>
<tr>
<td>$c_5 \approx 0$</td>
<td></td>
<td>$c_4 \approx h(d)$</td>
</tr>
</tbody>
</table>

deduce and propagate equalities between constants entailed by components
Step 2: Propagation

\[ c \leq d \land d \leq c + \text{car}(\text{cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]

<table>
<thead>
<tr>
<th>( \mathcal{R} )</th>
<th>( \mathcal{L} )</th>
<th>( \mathcal{E} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c \leq d )</td>
<td>( c_1 \approx \text{car}(\text{cons}(c_5, c)) )</td>
<td>( P(c_2) )</td>
</tr>
<tr>
<td>( d \leq c + c_1 )</td>
<td>( c_2 \approx c_3 - c_4 )</td>
<td>( \neg P(c_5) )</td>
</tr>
<tr>
<td>( c_2 \approx c_3 - c_4 )</td>
<td>( c_5 \approx 0 )</td>
<td>( c_3 \approx h(c) )</td>
</tr>
<tr>
<td>( c_5 \approx 0 )</td>
<td>( c_1 \approx c_5 )</td>
<td>( c_4 \approx h(d) )</td>
</tr>
</tbody>
</table>
Step 2: Propagation

\[
c \leq d \land d \leq c + \text{car(cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0)
\]

<table>
<thead>
<tr>
<th>(R)</th>
<th>(L)</th>
<th>(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c \leq d)</td>
<td>(c_1 \approx \text{car(cons}(c_5, c)))</td>
<td>(P(c_2))</td>
</tr>
<tr>
<td>(d \leq c + c_1)</td>
<td></td>
<td>(\neg P(c_5))</td>
</tr>
<tr>
<td>(c_2 \approx c_3 - c_4)</td>
<td></td>
<td>(c_3 \approx h(c))</td>
</tr>
<tr>
<td>(c_5 \approx 0)</td>
<td></td>
<td>(c_4 \approx h(d))</td>
</tr>
<tr>
<td>(c_1 \approx c_5)</td>
<td>(c_1 \approx c_5)</td>
<td></td>
</tr>
<tr>
<td>(c \approx d)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 2: Propagation

\[ c \leq d \land d \leq c + \text{car}(\text{cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]

<table>
<thead>
<tr>
<th>( \mathcal{R} )</th>
<th>( \mathcal{L} )</th>
<th>( \mathcal{E} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c \leq d )</td>
<td>( c_1 \approx \text{car}(\text{cons}(c_5, c)) )</td>
<td>( P(c_2) )</td>
</tr>
<tr>
<td>( d \leq c + c_1 )</td>
<td></td>
<td>( \neg P(c_5) )</td>
</tr>
<tr>
<td>( c_2 \approx c_3 - c_4 )</td>
<td></td>
<td>( c_3 \approx h(c) )</td>
</tr>
<tr>
<td>( c_5 \approx 0 )</td>
<td></td>
<td>( c_4 \approx h(d) )</td>
</tr>
<tr>
<td>( c_1 \approx c_5 )</td>
<td>( c_1 \approx c_5 )</td>
<td>( c \approx d )</td>
</tr>
<tr>
<td>( c \approx d )</td>
<td></td>
<td>( c_3 \approx c_4 )</td>
</tr>
</tbody>
</table>
Step 2: Propagation

\[ c \leq d \land d \leq c + \text{car}(\text{cons}(0, c)) \land P(h(c) - h(d)) \land \neg P(0) \]

<table>
<thead>
<tr>
<th>( \mathcal{R} )</th>
<th>( \mathcal{L} )</th>
<th>( \mathcal{E} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c \leq d )</td>
<td>( c_1 \approx \text{car}(\text{cons}(c_5, c)) )</td>
<td>( P(c_2) )</td>
</tr>
<tr>
<td>( d \leq c + c_1 )</td>
<td></td>
<td>( \neg P(c_5) )</td>
</tr>
<tr>
<td>( c_2 \approx c_3 - c_4 )</td>
<td></td>
<td>( c_3 \approx h(c) )</td>
</tr>
<tr>
<td>( c_5 \approx 0 )</td>
<td></td>
<td>( c_4 \approx h(d) )</td>
</tr>
<tr>
<td>( c_1 \approx c_5 )</td>
<td>( c_1 \approx c_5 )</td>
<td>( c \approx d )</td>
</tr>
<tr>
<td>( c \approx d )</td>
<td></td>
<td>( c_3 \approx c_4 )</td>
</tr>
<tr>
<td>( c_2 \approx c_5 )</td>
<td></td>
<td>( \bot )</td>
</tr>
</tbody>
</table>
The Nelson-Oppen algorithm

\(\phi\) conjunction of literals

**Step 1.** Purification \(\mathcal{T}_1 \cup \mathcal{T}_2 \cup \phi \mapsto (\mathcal{T}_1 \cup \phi_1) \cup (\mathcal{T}_2 \cup \phi_2)\):

where \(\phi_i\) is a pure \(\Sigma_i\)-formula and \(\phi_1 \land \phi_2\) is equisatisfiable with \(\phi\).

**Step 2.** Propagation.

The decision procedure for ground satisfiability for \(\mathcal{T}_1\) and \(\mathcal{T}_2\) fairly exchange information concerning entailed unsatisfiability of constraints in the shared signature i.e. clauses over the shared variables.

until an inconsistency is detected or a saturation state is reached.
The Nelson-Oppen algorithm

φ conjunction of literals

Step 1. Purification \( T_1 \cup T_2 \cup \phi \mapsto (T_1 \cup \phi_1) \cup (T_2 \cup \phi_2) \):
where \( \phi_i \) is a pure \( \Sigma_i \)-formula and \( \phi_1 \land \phi_2 \) is equisatisfiable with \( \phi \).

Step 2. Propagation.
The decision procedure for ground satisfiability for \( T_1 \) and \( T_2 \) fairly exchange information concerning entailed unsatisfiability of constraints in the shared signature i.e. clauses over the shared variables.
until an inconsistency is detected or a saturation state is reached.

Sound: if inconsistency detected input unsatisfiable
Complete: under additional assumptions
Implementation

\( \varphi \) conjunction of literals

**Step 1. Purification:** \( \mathcal{T}_1 \cup \mathcal{T}_2 \cup \varphi \mapsto (\mathcal{T}_1 \cup \varphi_1) \cup (\mathcal{T}_2 \cup \varphi_2) \),

where \( \varphi_i \) is a pure \( \Sigma_i \)-formula and \( \varphi_1 \land \varphi_2 \) is equisatisfiable with \( \varphi \).

**Step 2. Propagation:** The decision procedure for ground satisfiability for \( \mathcal{T}_1 \) and \( \mathcal{T}_2 \) fairly exchange information concerning entailed unsatisfiability of constraints in the shared signature i.e. clauses over the shared variables.

until an inconsistency is detected or a saturation state is reached.

**How to implement Propagation?**

**Guessing:** guess a maximal set of literals containing the shared variables; check it for \( \mathcal{T}_i \cup \varphi_i \) consistency.

**Backtracking:** identify disjunction of equalities between shared variables entailed by \( \mathcal{T}_i \cup \varphi_i \); make case split by adding some of these equalities to \( \varphi_1, \varphi_2 \). Repeat as long as possible.
The Nelson-Oppen algorithm

**Termination:** only finitely many shared variables to be identified

**Soundness:** If procedure answers “unsatisfiable” then $\phi$ is unsatisfiable

**Completeness:** Under additional hypotheses
## Completeness

**Example:**

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(g(x), g(y)) \approx x$</td>
<td></td>
<td>$k(x) \approx k(x)$</td>
</tr>
<tr>
<td>$f(g(x), h(y)) \approx y$</td>
<td></td>
<td>non-trivial</td>
</tr>
</tbody>
</table>

$g(c) \approx h(c) \land k(c) \not\approx c$

- $g(c) \approx h(c)$
- $k(c) \not\approx c$

satisfiable in $E_1$

satisfiable in $E_2$

no equations between shared variables; **Nelson-Oppen answers “satisfiable”**
Completeness

Example:

\[
\begin{array}{c|c}
E_1 & E_2 \\
\hline
f(g(x), g(y)) \approx x & k(x) \approx k(x) \\
f(g(x), h(y)) \approx y & \\
\text{non-trivial} & \text{non-trivial} \\
g(c) \approx h(c) \land k(c) \not\approx c \\
g(c) \approx h(c) & k(c) \not\approx c \\
\text{satisfiable in } E_1 & \text{satisfiable in } E_2 \\
\end{array}
\]

no equations between shared variables; Nelson-Oppen answers “satisfiable”

A model of \( E_1 \) satisfies \( g(c) \approx h(c) \) iff \( \exists e \in A \text{ s.t. } g(e) = h(e) \).

Then, for all \( a \in A \): \( a = f_A(g(a), g(e)) = f_A(g(a), h(e)) = e \)

\( g(c) \approx h(c) \land k(c) \not\approx c \) unsatisfiable
Completeness

Another example

$\mathcal{T}_1$ theory admitting models of cardinality at most 2

$\mathcal{T}_2$ theory admitting models of any cardinality

such that $\mathcal{T}_i \models \forall x, y \ f_i(x) = f_i(y)$.

$$\phi = f_1(c_1) \not\approx f_1(c_2) \land f_2(c_1) \not\approx f_2(c_3) \land f_2(c_2) \not\approx f_2(c_3)$$

$\phi_1 = f_1(c_1) \not\approx f_1(c_2)$ \quad $\phi_2 = f_2(c_1) \not\approx f_2(c_3) \land f_2(c_2) \not\approx f_2(c_3)$

The Nelson-Oppen procedure returns “satisfiable”

$\mathcal{T}_1 \cup \mathcal{T}_2 \models \forall x, y, z (f_1(x) \not\approx f_1(y) \land f_2(x) \not\approx f_2(z) \land f_2(y) \not\approx f_2(z)$

$\rightarrow (x \not\approx y \land x \not\approx z \land y \not\approx z))$

$f_1(c_1) \not\approx f_1(c_2) \land f_2(c_1) \not\approx f_2(c_3) \land f_2(c_2) \not\approx f_2(c_3)$ unsatisfiable
Completeness

Cause of incompleteness

There exist formulae satisfiable in finite models of bounded cardinality

**Solution:** Consider stably infinite theories.

\[ \mathcal{T} \text{ is stably infinite iff for every quantifier-free formula } \phi \]

\[ \phi \text{ satisfiable in } \mathcal{T} \text{ iff } \phi \text{ satisfiable in an infinite model of } \mathcal{T}. \]

**Note:** This restriction is not mentioned in [Nelson Oppen 1979]; introduced by Oppen in 1980.
Completeness

Guessing version: $C$ set of constants shared by $\phi_1, \phi_2$

$R$ equiv. relation assoc. with partition of $C$ \[ ar(C, R) = \bigwedge_{R(c,d)} c \approx d \land \bigwedge_{\neg R(c,d)} c \not\approx d \]

**Lemma.** Assume that there exists a partition of $C$ s.t. $\phi_i \land ar(C, R)$ is $\mathcal{T}_i$-satisfiable. Then $\phi_1 \land \phi_2$ is $\mathcal{T}_1 \cup \mathcal{T}_2$-satisfiable.

Idea of proof: Let $A_i \in \text{Mod}(\mathcal{T}_i)$ s.t. $A_i \models \phi_i \land ar(C, R)$. Then $c_{A_1} = d_{A_1}$ iff $c_{A_2} = d_{A_2}$.

Let $i : \{c_{A_1} \mid c \in C\} \to \{c_{A_2} \mid c \in C\}$, $i(c_{A_1}) = c_{A_2}$ well-defined; bijection.

Stable infinity: can assume w.l.o.g. that $A_1, A_2$ have the same cardinality

Let $h : A_1 \to A_2$ bijection s.t. $h(c_{A_1}) = c_{A_2}$

Use $h$ to transfer the $\Sigma_1$-structure on $A_2$.

**Theorem.** If $\mathcal{T}_1, \mathcal{T}_2$ are both stably infinite and the shared signature is empty then the Nelson-Oppen procedure is sound, complete and terminating.

Thus, it transfers decidability of ground satisfiability from $\mathcal{T}_1, \mathcal{T}_2$ to $\mathcal{T}_1 \cup \mathcal{T}_2$. 
Complexity

Main sources of complexity:

(i) transformation of the formula in DNF
(ii) propagation
   (a) decide whether there is a disjunction of equalities between variables
   (b) investigate different branches corresponding to disjunctions
Complexity

Main sources of complexity:

(i) transformation of the formula in DNF

(ii) propagation

\[ T \text{ is convex} \iff \text{ for every quantifier-free formula } \phi, \]
\[ \phi \models \bigvee_i x_i \approx y_i \implies \phi \models x_j \approx y_j \text{ for some } j. \]

\[ \rightarrow \text{ No branching} \]
Complexity

Main sources of complexity:

(i) transformation of the formula in DNF
(ii) propagation

\[ \mathcal{T} \text{ is convex} \iff \text{for every quantifier-free formula } \phi, \]
\[ \phi \models \bigvee_i x_i \approx y_i \Rightarrow \phi \models x_j \approx y_j \text{ for some } j. \]

\[ \mapsto \text{No branching} \]

**Theorem.** Let \( \mathcal{T}_1 \) and \( \mathcal{T}_2 \) be convex and stably infinite; \( \Sigma_1 \cap \Sigma_2 = \emptyset \)
If satisfiability of conjunctions of literals in \( \mathcal{T}_i \) is in PTIME
Then satisfiability of conjunctions of literals in \( \mathcal{T}_1 \cup \mathcal{T}_2 \) is in PTIME
In general: non-deterministic procedure

**Theorem.** Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be convex and stably infinite; $\Sigma_1 \cap \Sigma_2 = \emptyset$

If satisfiability of conjunctions of literals in $\mathcal{T}_i$ is in NP

Then satisfiability of conjunctions of literals in $\mathcal{T}_1 \cup \mathcal{T}_2$ is in NP
Extensions of the Nelson-Oppen procedure

• relax the stable infiniteness requirement

• relax the requirement that the theories have disjoint signatures
Extensions of the Nelson-Oppen procedure

- relax the stable infiniteness requirement
  
  \[\text{[Tinelli,Zarba'03] One theory “shiny” (for each satisf. constraint we can compute a finite } k \text{ s.t. the theory has models of every cardinality } \lambda \geq k\]}

- relax the requirement that the theories have disjoint signatures
  
  \[\text{[Tinelli,Ringeissen'03] Theories sharing absolutely free constructors} \]
  \[\text{[Ghilardi'04] Model theoretical conditions.} \]

**Main idea:**

Find situations in which \( \mathcal{T}_i \) models of \( \phi_i, i = 1, 2 \) can be “amalgamated” to a \( \mathcal{T}_1 \cup \mathcal{T}_2 \) model of \( \phi_1 \land \phi_2 \).
Limitations of the Nelson-Oppen procedure

1. Does not fully exploit the properties of the component theories

No simplification mechanisms built-in

**Improvements:** Cynthia Engel: Shostak’s method

- Reasoning in a combinations of a theory with UIF
- use a canonizer and a solver
Limitations of the Nelson-Oppen procedure

**Problem:** The conditions which guarantee completeness of the Nelson-Oppen procedure are quite restrictive.

- Need to consider data structures with elements in a finite domain
- Sometimes the theories are tighter interrelated.

**Example 1:** Theories of arrays with elements of a certain type:

- **Sorts:** i (Indices), e (Elements), a (Arrays)
- **Theories:** $T_i$, $T_e$, $T_a$
- **Axioms for** $T_a$

\[
\forall a, i, e : \quad \text{read(write}(a, i, e), i) = e \\
\forall a, i, j, e : \quad i \neq j \rightarrow \text{read(write}(a, i, e), j) = \text{read}(a, j)
\]

**Example 2:** lists with elements of a certain type; length functions, ...
Theories of arrays

- **Array property fragment** [Bradley, Manna, Sipma’06]
  
  \( \exists \)-closed Bool. comb. of array property formulae & QF formulae (\( \exists \forall i \))

  **Array property** \((\forall i)(\varphi_I(i) \rightarrow \varphi_V(i))\)  
  \(\varphi_I: \) positive Boolean combination of \( t \leq u \) or \( t = u \), \( \varphi_V: \) any universally quantified \( i \) occurs in an array read; no nestings

  Not: \( x < y \rightarrow a(x) < a(y) \)  
  Not: \( x + 3 \leq 2 \ast y \)  
  Not: \( x + 1 \leq c \)

  \( t, u \) ground index terms or variables

  **Main idea:** regard arrays as “functions”

  \( \text{read}(a, i) \leftrightarrow a(i) \)  
  \( \text{write}(a, i_0, e) \leftrightarrow b = \text{write}(a, i_0, e) \)  
  \( + \) Updates

  \( \forall j(j = i_0 \rightarrow b(j) = e) \)  
  \( i_1 = i_0 - 1, i_2 = i_0 + 1 \)  
  \( \forall j(j \leq i_1 \lor j \geq i_2 \rightarrow b(j) = a(j)) \)
Theories of arrays

- **Array property fragment** [Bradley,Manna,Sipma’06]

  $\exists$-closed Bool. comb. of array property formulae & QF formulae ($\exists\forall_i$)

  **Array property** $\left(\forall i\right)\left(\varphi_I(i) \rightarrow \varphi_V(i)\right)$

  Not: $x < y \rightarrow a(x) < a(y)$

  $\varphi_I$: positive Boolean combination of $t \leq u$ or $t = u$, Not: $x + 3 \leq 2 \times y$

  where $t, u$ ground index terms or variables

  $x + 1 \leq c$

  $\varphi_V$: any universally quantified $i$ occurs in a direct array read; no nestings

  Shuyi Weng: What’s decidable about arrays
Theories of recursive data structures with size

Theories of constructors/selectors

Lists (cons/car/cdr)

Binary trees (tree/left/right)

Size functions:

Lists:

\[
\text{size}(\text{nil}) = 0
\]

\[
\text{size}(\text{cons}(a, l)) = 1 + \text{size}(l)
\]

Trees

\[
\text{size}(\text{nil}) = 0
\]

\[
\text{size}(\text{tree}(t_1, t_2)) = 1 + \text{size}(t_1) + \text{size}(t_2)
\]

Kilian Laudt: Decision procedures
Overview

• **Reasoning in standard theories**
  
  A *crash course*: Decidable logical theories and theory fragments

• **Reasoning in complex theories**
  
  Modular reasoning in combinations of theories
  
  disjoint signature

• **Applications**