Combining Theories

Emina Torlak
emina@cs.washington.edu
Today
Today

Last lecture

• A survey of theory solvers and deciding $T_=$ with congruence closure
Today

Last lecture
  • A survey of theory solvers and deciding $T_\equiv$ with congruence closure

Today
  • Deciding a combination of theories
Today

Last lecture
  • A survey of theory solvers and deciding $T_\approx$ with congruence closure

Today
  • Deciding a combination of theories

Reminders
  • Email us your project topic and brief abstract by 11pm today
  • Homework 2 posted
    • Start early
    • Submit self-contained runnable code
Satisfiability Modulo Theories (SMT)

\[ x = g(y) \]
\[ 2x + y > 5 \]
\[ (b \gg 2) = c \]
\[ \vdots \]
\[ a[i] = x \]

First-Order Logic

Theories

SMT solver

Core solver

DPLL(T)

\begin{align*}
\text{Theory solver} & \quad \cdots \quad \text{Theory solver}
\end{align*}
Combining T-solvers with Nelson-Oppen

\[ \Sigma_1\text{-theory } T_1 \text{ with axioms } A_1 \]

Theory solver

\[ \ldots \]

\[ \Sigma_n\text{-theory } T_n \text{ with axioms } A_n \]

Theory solver

\[ \text{Combination solver} \]

Theory \( T_1 \cup \ldots \cup T_n \) with signature \( \Sigma_1 \cup \ldots \cup \Sigma_n \) and axioms \( A_1 \cup \ldots \cup A_n \)
Combining T-solvers with Nelson-Oppen

\[ \Sigma_1 \text{-theory } T_1 \text{ with axioms } A_1 \]

\[ \Sigma_2 \text{-theory } T_2 \text{ with axioms } A_2 \]

\[ \Sigma_1 \cup \Sigma_2 \text{ with signature } \Sigma_1 \cup \Sigma_2 \text{ and axioms } A_1 \cup A_2 \]

We’ll see how to combine two theories. Easy to generalize to n.
Combining $T$-solvers with Nelson-Oppen

$\Sigma_1$-theory $T_1$ with axioms $A_1$

Theory solver

$\Sigma_2$-theory $T_2$ with axioms $A_2$

Theory solver

Combination solver

Theory $T_1 \cup T_2$ with signature $\Sigma_1 \cup \Sigma_2$ and axioms $A_1 \cup A_2$

We’ll see how to combine two theories. Easy to generalize to $n$.

The combination problem is undecidable for arbitrary (decidable) theories. It becomes decidable under Nelson-Oppen restrictions.
Nelson-Oppen restrictions

$T_1$ and $T_2$ can be combined when

- Both are quantifier-free (conjunctive) fragments
- Equality ($=$) is the only symbol in the intersection of their signatures
- Both are stably infinite
Nelson-Oppen restrictions

$T_1$ and $T_2$ can be combined when

- Both are quantifier-free (conjunctive) fragments
- Equality (=) is the only symbol in the intersection of their signatures
- Both are stably infinite

A theory $T$ is stably infinite iff for every satisfiable $\Sigma_T$-formula $F$, there is a $T$-model that satisfies $F$ and that has a universe of infinite cardinality.
Examples of (non-)stably infinite theories

$\Sigma_T: \{ a, b, = \}$

$A_T: \forall x . x = a \lor x = b$
Examples of (non-)stably infinite theories

\[ \Sigma_T: \{ a, b, = \} \]
\[ A_T: \forall x. x = a \lor x = b \]
Examples of (non-)stably infinite theories

\[ \Sigma_T: \{ a, b, = \} \]
\[ A_T: \forall x . x = a \lor x = b \]

Fixed width bit vectors \((T_{bv})\)
Examples of (non-)stably infinite theories

$\Sigma_T$: \{ a, b, = \}

$A_T$: $\forall x \cdot x = a \lor x = b$

Fixed width bit vectors ($T_{bv}$)
Examples of (non-)stably infinite theories

\[ \Sigma_T: \{ a, b, = \} \]
\[ A_T: \forall x . x = a \lor x = b \]

Equality and uninterpreted functions (\(T_\approx\))

Fixed width bit vectors (\(T_{bv}\))
Examples of (non-)stably infinite theories

\[ \Sigma_T: \{ a, b, = \} \]
\[ A_T: \forall x . x = a \lor x = b \]

Equality and uninterpreted functions (\( T= \))

Fixed width bit vectors (\( T_{bv} \))
Examples of (non-)stably infinite theories

\[ \Sigma_T: \{ a, b, = \} \]
\[ A_T: \forall x . x = a \lor x = b \]

Equality and uninterpreted functions (\(T_e\))

- Fixed width bit vectors (\(T_{bv}\))
- Arrays (\(T_A\))
- Linear integer arithmetic (\(T_{IR}\))
- Linear real arithmetic (\(T_{R}\))

- ✓
- ✓
- ✓
- ✓
Overview of Nelson-Oppen

\[(\Sigma_1 \cup \Sigma_2)\text{-formula } F\]

\[\Sigma_1\text{-formula } F_1\]
\[\Sigma_2\text{-formula } F_2\]

Equality Propagation

\[T_1 \text{ solver} \quad = \quad \quad T_2 \text{ solver}\]
Overview of purification

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)
Overview of purification

Transforms a $(\Sigma_1 \cup \Sigma_2)$-formula $F$ into an equisatisfiable formula $F_1 \land F_2$ with $F_1$ in $T_1$ and $F_2$ in $T_2$

Repeat until fix point:

- If $f$ is in $T_i$ and $t$ is not, and $u$ is fresh:
  $F[f(\ldots, t, \ldots)] \leftrightarrow F[f(\ldots, u, \ldots)] \land u = t$

- If $p$ is in $T_i$ and $t$ is not, and $v$ is fresh:
  $F[p(\ldots, t, \ldots)] \leftrightarrow F[p(\ldots, v, \ldots)] \land v = t$
Overview of purification

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \(F[f(\ldots, t, \ldots)] \rightsquigarrow F[f(\ldots, u, \ldots)] \land u = t\)

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \(F[p(\ldots, t, \ldots)] \rightsquigarrow F[p(\ldots, v, \ldots)] \land v = t\)
Overview of purification

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \(F[f(\ldots, t, \ldots)] \rightsquigarrow F[f(\ldots, u, \ldots)] \land u = t\)

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \(F[p(\ldots, t, \ldots)] \rightsquigarrow F[p(\ldots, v, \ldots)] \land v = t\)
Another purification example

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \(F[f(\ldots, t, \ldots)] \iff F[f(\ldots, u, \ldots)] \land u = t\)

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \(F[p(\ldots, t, \ldots)] \iff F[p(\ldots, v, \ldots)] \land v = t\)

\[f(x + g(y)) \leq g(a) + f(b)\]
Another purification example

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \[ F[f(\ldots, t, \ldots)] \rightsquigarrow F[f(\ldots, u, \ldots)] \land u = t \]

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \[ F[p(\ldots, t, \ldots)] \rightsquigarrow F[p(\ldots, v, \ldots)] \land v = t \]
Another purification example

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \(F[f(\ldots, t, \ldots)] \rightsquigarrow F[f(\ldots, u, \ldots)] \land u = t\)

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \(F[p(\ldots, t, \ldots)] \rightsquigarrow F[p(\ldots, v, \ldots)] \land v = t\)

\[f(x + u_1) \leq u_2 + u_3\]

\[u_1 = g(y)\]
\[u_2 = g(a)\]
\[u_3 = f(b)\]
Another purification example

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \[F[f(\ldots, t, \ldots)] \rightsquigarrow F[f(\ldots, u, \ldots)] \land u = t\]

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \[F[p(\ldots, t, \ldots)] \rightsquigarrow F[p(\ldots, v, \ldots)] \land v = t\]

\[f(x + u_1) \leq u_2 + u_3\]

\(u_1 = g(y)\)
\(u_2 = g(a)\)
\(u_3 = f(b)\)
Another purification example

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \[ F[f(\ldots, t, \ldots)] \rightsquigarrow F[f(\ldots, u, \ldots)] \land u = t \]

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \[ F[p(\ldots, t, \ldots)] \rightsquigarrow F[p(\ldots, v, \ldots)] \land v = t \]
Another purification example

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \[F[f(\ldots, t, \ldots)] \iff F[f(\ldots, u, \ldots)] \land u = t\]

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \[F[p(\ldots, t, \ldots)] \iff F[p(\ldots, v, \ldots)] \land v = t\]

\[f(u_4) \leq u_2 + u_3\]
Another purification example

Transforms a \((\Sigma_1 \cup \Sigma_2)\)-formula \(F\) into an equisatisfiable formula \(F_1 \land F_2\) with \(F_1\) in \(T_1\) and \(F_2\) in \(T_2\)

Repeat until fix point:

- If \(f\) is in \(T_i\) and \(t\) is not, and \(u\) is fresh:
  \[F[f(\ldots, t, \ldots)] \iff F[f(\ldots, u, \ldots)] \land u = t\]

- If \(p\) is in \(T_i\) and \(t\) is not, and \(v\) is fresh:
  \[F[p(\ldots, t, \ldots)] \iff F[p(\ldots, v, \ldots)] \land v = t\]

\[\Sigma_R\]  \[\Sigma_=\]
\[u_4 = x + u_1\]  \[u_1 = g(y)\]
\[u_5 \leq u_2 + u_3\]  \[u_2 = g(a)\]
\[u_3 = f(b)\]  \[u_5 = f(u_4)\]
Shared and local variables

A variable is *shared* if it occurs in both $F_1$ and $F_2$, and it is *local* otherwise.
A variable is *shared* if it occurs in both \( F_1 \) and \( F_2 \), and it is *local* otherwise.

**Shared:** \{\( u_1, u_2, u_3, u_4, u_5 \)\}

**Local:** \{\( x, y, a, b \)\}

**Purification**

\[
\begin{align*}
\Sigma_R & \quad \Sigma_E \\
\Sigma_R & \quad \Sigma_E \\
\sum & \quad \sum \\
\end{align*}
\]

\[
\begin{align*}
\Sigma_R & \quad \Sigma_E \\
\sum & \quad \sum \\
\end{align*}
\]

\[
\begin{align*}
\Sigma_R & \quad \Sigma_E \\
\sum & \quad \sum \\
\end{align*}
\]

\[
\begin{align*}
\Sigma_R & \quad \Sigma_E \\
\sum & \quad \sum \\
\end{align*}
\]
Overview of Nelson-Oppen

\[(\Sigma_1 \cup \Sigma_2)\text{-formula } F\]

\[\Sigma_1\text{-formula } F_1\]
\[\Sigma_2\text{-formula } F_2\]

**Purification**

**Equality Propagation**

\[T_1\text{ solver} = T_2\text{ solver}\]
Overview of Nelson-Oppen

\((\Sigma_1 \cup \Sigma_2)\)-formula \(F\)

\(\Sigma_1\)-formula \(F_1\)

\(\Sigma_2\)-formula \(F_2\)

**Purification**

**Equality Propagation**
- Convex theories
- Non-convex theories
Convex theories

A theory $T$ is *convex* if for every conjunctive formula $F$, the following holds:

If $F \Rightarrow x_1 = y_1 \lor \ldots \lor x_n = y_n$ for a finite $n > 1$,
then $F \Rightarrow x_i = y_i$ for some $i \in \{1, \ldots, n\}$. 

Convex theories

A theory \( T \) is **convex** if for every conjunctive formula \( F \), the following holds:

If \( F \Rightarrow x_1 = y_1 \lor \ldots \lor x_n = y_n \) for a finite \( n > 1 \), then \( F \Rightarrow x_i = y_i \) for some \( i \in \{1, \ldots, n\} \).

If \( F \) implies a disjunction of equalities, then it also implies at least one of the equalities.
Examples of (non-)convex theories

Linear arithmetic over integers ($T_Z$)
Examples of (non-)convex theories

Linear arithmetic over integers ($T_Z$)

$1 \leq x \land x \leq 2 \Rightarrow x = 1 \lor x = 2$ but
not $1 \leq x \land x \leq 2 \Rightarrow x = 1$
not $1 \leq x \land x \leq 2 \Rightarrow x = 2$
Examples of (non-)convex theories

Linear arithmetic over integers ($T_\mathbb{Z}$)

$1 \leq x \land x \leq 2 \Rightarrow x = 1 \lor x = 2$ but

not $1 \leq x \land x \leq 2 \Rightarrow x = 1$

not $1 \leq x \land x \leq 2 \Rightarrow x = 2$

Equality and uninterpreted functions ($T_=$)

$\checkmark$

Linear real arithmetic ($T_\mathbb{R}$)

$\checkmark$
Nelson-Oppen method for convex theories

Nelson-Oppen-Convex(F)
Nelson-Oppen method for convex theories

**NELSON-OPPEN-CONVEX**(*F*)

1. Purify *F* into *F*₁ ∧ *F*₂
Nelson-Oppen method for convex theories

**Nelson-Oppen-Convex(F)**

1. Purify $F$ into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \models x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT
Nelson-Oppen method for convex theories

**NELSON-OPPEN-CONVEX**(F)

1. Purify F into \( F_1 \land F_2 \)
2. Run \( T_1 \)-solver on \( F_1 \) and \( T_2 \)-solver on \( F_2 \) and return UNSAT if either is unsatisfiable

Is \( F \) satisfiable if both \( F_1 \) and \( F_2 \) are satisfiable?
Nelson-Oppen method for convex theories

NELSON-OPPEN-CONVEX(F)
1. Purify F into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable

Is F satisfiable if both $F_1$ and $F_2$ are satisfiable? No:
$x = 1 \land 2 = x + y \land f(x) \neq f(y)$
Nelson-Oppen method for convex theories

\textsc{Nelson-Oppen-Convex}(F)

1. Purify F into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables x and y such that $F_i \implies x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT
Nelson-Oppen method for convex theories

\textbf{NELSON-OPPEN-CONVEX}(F)

1. Purify F into $F_1 \land F_2$

2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable

3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.

4. Return SAT
**Nelson-Oppen for convex theories: example**

**NELSON-OPPEN-CONVEX(F)**

1. Purify $F$ into $F_1 \land F_2$

2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable

3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.

4. Return SAT

$$f(f(x) - f(y)) \neq f(z) \land x \leq y \land y + z \leq x \land 0 \leq z$$
Nelson-Oppen for convex theories: example

**Nelson-Oppen-Convex(F)**

1. Purify $F$ into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT

\[
\begin{align*}
  f(f(x) - f(y)) \neq f(z) \land x \leq y \\
  \land y + z \leq x \land 0 \leq z
\end{align*}
\]

\[
\begin{align*}
  x \leq y & \land f(w) \neq f(z) \\
  y + z \leq x & \land u = f(x) \\
  0 \leq z & \land v = f(y) \\
  w = u - v
\end{align*}
\]

\[
\Sigma_R \quad \Sigma_e
\]
**Nelson-Oppen for convex theories: example**

**NELSON-OPPEN-CONVEX(F)**

1. Purify F into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT

\[
\begin{align*}
f(f(x) - f(y)) &\neq f(z) \land x \leq y \\
&\land y + z \leq x \land 0 \leq z
\end{align*}
\]

<table>
<thead>
<tr>
<th>$x \leq y \land$</th>
<th>$f(w) \neq f(z) \land$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y + z \leq x \land$</td>
<td>$u = f(x) \land$</td>
</tr>
<tr>
<td>$0 \leq z \land$</td>
<td>$v = f(y)$</td>
</tr>
</tbody>
</table>
\[
w = u - v
\]
\[
x = y \land
\]
\[
\Sigma_R
\]
\[
\Sigma =
\]
Nelson-Oppen for convex theories: example

**NELSON-OPPEN-CONVEX(F)**

1. Purify F into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT

<table>
<thead>
<tr>
<th>$f(f(x) - f(y)) \neq f(z) \land x \leq y$</th>
<th>$\land y + z \leq x \land 0 \leq z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \leq y \land$</td>
<td>$f(w) \neq f(z) \land$</td>
</tr>
<tr>
<td>$y + z \leq x \land$</td>
<td>$u = f(x) \land$</td>
</tr>
<tr>
<td>$0 \leq z \land$</td>
<td>$v = f(y)$</td>
</tr>
<tr>
<td>$w = u - v$</td>
<td></td>
</tr>
</tbody>
</table>

```
| $x = y \land$                      | $x = y \land$
| $u = v \land$                      | $u = v \land$
```

\[ \Sigma_R \]

\[ \Sigma = \]
**Nelson-Oppen for convex theories: example**

**NELSON-OPPEN-CONVEX(F)**

1. Purify $F$ into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT

$$f(f(x) - f(y)) \neq f(z) \land x \leq y \land y + z \leq x \land 0 \leq z$$

| $x \leq y$ | $f(w) \neq f(z)$ |
| $y + z \leq x$ | $u = f(x)$ |
| $0 \leq z$ | $v = f(y)$ |
| $w = u - v$ | $x = y$ |
| $u = v$ | $u = v$ |
| $w = z$ | $w = z$ |

$$\Sigma_R$$

$$\Sigma =$$
Nelson-Oppen for convex theories: example

**Nelson-Oppen-Convex(F)**

1. Purify F into $F_1 \land F_2$

2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable

3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.

4. Return SAT

\[
\begin{align*}
\text{f(f(x) - f(y))} &\neq \text{f(z)} \land x \leq y \\
\land y + z &\leq x \land 0 \leq z
\end{align*}
\]

\[
\begin{array}{l|l}
\text{x} \leq y & \text{f(w)} \neq \text{f(z)} \land \text{u} = \text{f(x)} \\
y + z &\leq x \land \text{v} = \text{f(y)} \\
0 &\leq z \\
w = u - v
\end{array}
\]

\[
\begin{array}{l|l}
x = y & x = y \\
u = v & u = v \\
w = z & w = z \land \text{UNSAT}
\end{array}
\]

$\Sigma_R \qquad \Sigma_L$
This doesn’t work for non-convex theories ...

\[ \text{NELSON-OPPEN-CONVEX}(F) \]

1. Purify \( F \) into \( F_1 \land F_2 \)

2. Run \( T_1 \)-solver on \( F_1 \) and \( T_2 \)-solver on \( F_2 \) and return UNSAT if either is unsatisfiable

3. If there are shared variables \( x \) and \( y \) such that \( F_i \Rightarrow x = y \) but \( F_j \) does not
   
   1. \( F_j \leftarrow F_j \land x = y \)
   2. Go to step 2.

4. Return SAT

\[ 1 \leq x \land x \leq 2 \land 
   \text{f}(x) \neq \text{f}(1) \land \text{f}(x) \neq \text{f}(2) \]
This doesn’t work for non-convex theories ...

**NELSON-OPPEN-CONVEX(F)**

1. Purify $F$ into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT

| 1 ≤ $x \land x ≤ 2 \land$
| $f(x) \neq f(1) \land f(x) \neq f(2)$
| 1 ≤ $x \land$
| $x ≤ 2 \land$
| $z_1 = 1 \land$
| $z_2 = 2$

$\Sigma_z \Sigma = \Sigma$
This doesn’t work for non-convex theories ...

**NELSON-OPPEN-CONVEX(F)**

1. Purify F into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT

<table>
<thead>
<tr>
<th>$1 \leq x \land x \leq 2 \land$</th>
<th>$f(x) \neq f(1) \land f(x) \neq f(2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 \leq x \land f(x) \neq f(z_1) \land$</td>
<td></td>
</tr>
<tr>
<td>$x \leq 2 \land f(x) \neq f(z_2) \land$</td>
<td></td>
</tr>
<tr>
<td>$z_1 = 1 \land$</td>
<td></td>
</tr>
<tr>
<td>$z_2 = 2$</td>
<td></td>
</tr>
</tbody>
</table>

SAT

$\Sigma_Z$ $\Sigma=$ SAT
This doesn’t work for non-convex theories ...

\textbf{NELSON-OPPEN-CONVEX}(F)

1. Purify \( F \) into \( F_1 \land F_2 \)
2. Run \( T_1 \)-solver on \( F_1 \) and \( T_2 \)-solver on \( F_2 \) and return UNSAT if either is unsatisfiable
3. If there are shared variables \( x \) and \( y \) such that \( F_i \Rightarrow x = y \) but \( F_j \) does not
   1. \( F_j \leftarrow F_j \land x = y \)
   2. Go to step 2.
4. Return SAT
This doesn’t work for non-convex theories …

**NELSON-OPPEN-CONVEX**(F)

1. Purify $F$ into $F_1 \land F_2$
2. Run $T_1$-solver on $F_1$ and $T_2$-solver on $F_2$ and return UNSAT if either is unsatisfiable
3. If there are shared variables $x$ and $y$ such that $F_i \Rightarrow x = y$ but $F_j$ does not
   1. $F_j \leftarrow F_j \land x = y$
   2. Go to step 2.
4. Return SAT

If $T$ is non-convex, it may imply a disjunction of equalities without implying any single equality.

We have to propagate disjunctions as well as individual equalities. Why is this possible? How do we propagate disjunctions to theory solvers which only reason about conjunctions?
Nelson-Oppen method for non-convex theories

NELSON-OPPEN(F)

1. Purify F into F₁ ∧ F₂

2. Run T₁-solver on F₁ and T₂-solver on F₂ and return UNSAT if either is unsatisfiable

3. If there are shared variables x and y such that Fᵢ ⇒ x = y but Fⱼ does not
   1. Fⱼ ← Fⱼ ∧ x = y
   2. Go to step 2.

4. If Fᵢ ⇒ x₁ = y₁ ∨ ... ∨ xₙ = yₙ but Fⱼ does not, then if NELSON-OPPEN(Fᵢ ∧ Fⱼ ∧ xₖ = yₖ) outputs SAT for any k, return SAT. Otherwise, return UNSAT.

5. Return SAT
Nelson-Oppen method for non-convex theories

NELSON-OPPEN(F)
1. Purify F into F₁ ∧ F₂
2. Run T₁-solver on F₁ and T₂-solver on F₂ and return UNSAT if either is unsatisfiable
3. If there are shared variables x and y such that Fᵢ \implies x = y but Fⱼ does not
   1. \( Fⱼ \leftarrow Fⱼ ∧ x = y \)
   2. Go to step 2.
4. If \( Fᵢ \implies x₁ = y₁ \lor \ldots \lor xₙ = yₙ \) but Fⱼ does not, then if NELSON-OPPEN(\( Fᵢ \land Fⱼ \land xₖ = yₖ \)) outputs SAT for any k, return SAT. Otherwise, return UNSAT.
5. Return SAT

Propagate a minimal disjunction.
Nelson-Oppen for non-convex theories: example

\[ 1 \leq x \land x \leq 2 \land \]
\[ f(x) \neq f(1) \land f(x) \neq f(2) \]
Nelson-Oppen for non-convex theories: example

<table>
<thead>
<tr>
<th>1 ≤ x ∧ x ≤ 2 ∧ f(x) ≠ f(1) ∧ f(x) ≠ f(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ≤ x ∧</td>
</tr>
<tr>
<td>x ≤ 2 ∧</td>
</tr>
<tr>
<td>z₁ = 1 ∧</td>
</tr>
<tr>
<td>z₂ = 2</td>
</tr>
<tr>
<td>Σ z</td>
</tr>
<tr>
<td>Σ =</td>
</tr>
</tbody>
</table>
Nelson-Oppen for non-convex theories: example

\[ 1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \]

<table>
<thead>
<tr>
<th>( 1 \leq x \land )</th>
<th>( f(x) \neq f(z_1) \land )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \leq 2 \land )</td>
<td>( f(x) \neq f(z_2) )</td>
</tr>
<tr>
<td>( z_1 = 1 \land )</td>
<td></td>
</tr>
<tr>
<td>( z_2 = 2 )</td>
<td></td>
</tr>
<tr>
<td>( (x=z_1 \lor x=z_2) \land )</td>
<td></td>
</tr>
<tr>
<td>( \Sigma_Z )</td>
<td>( \Sigma = )</td>
</tr>
</tbody>
</table>
Nelson-Oppen for non-convex theories: example

\[
\begin{align*}
1 & \leq x \land x \leq 2 \land \\
& \text{f(x) \neq f(1) \land f(x) \neq f(2)} \\
\hline
1 & \leq x \land \\
x & \leq 2 \land \\
z_1 & = 1 \land \\
z_2 & = 2 \\
(x=z_1 \lor x=z_2) & \land \\
\Sigma_Z & \quad \Sigma= \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 & \leq x \land \\
x & \leq 2 \land \\
z_1 & = 1 \land \\
z_2 & = 2 \\
x & = z_1 \land \\
x & = z_1 \land \\
\text{UNSAT} & \quad \text{f(x) \neq f(z_1) \land} \\
& \quad \text{f(x) \neq f(z_2)}
\end{align*}
\]
Nelson-Oppen for non-convex theories: example

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
(x = z_1 \lor x = z_2) & \\
\Sigma_Z & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_1 & \\
\Sigma = & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_2 & \\
\Sigma = & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_2 & \\
\Sigma = & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_1 & \\
\Sigma = & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_2 & \\
\Sigma = & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_1 & \\
\Sigma = & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_2 & \\
\Sigma = & \\
\hline
\end{align*}
\]

\[
\begin{align*}
1 \leq x \land x & \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2) \\
\hline
1 \leq x & f(x) \neq f(z_1) \\
x \leq 2 & f(x) \neq f(z_2) \\
z_1 = 1 & \\
z_2 = 2 & \\
x = z_1 & \\
\Sigma = & \\
\hline
\end{align*}
\]
Complexity of Nelson-Oppen

If decision procedures for convex theories $T_1$ and $T_2$ have polynomial time complexity, so does their Nelson-Oppen combination.

If decision procedures for convex theories $T_1$ and $T_2$ have NP time complexity, so does their Nelson-Oppen combination.
Summary

Today

• Sound and complete procedure for a combination of restricted theories

• Stably infinite, conjunctive, quantifier-free, signatures disjoint except for =

Next lecture

• Deciding satisfiability of arbitrary boolean combinations of quantifier-free first-order formulas