ECE750T-28:
Computer-aided Reasoning for Software Engineering

Lecture 16: Decision Procedures for Combination Theories

Vijay Ganesh
(Original notes from Isil Dillig)
Motivation

▶ So far, learned about decision procedures for useful theories
Motivation

- So far, learned about decision procedures for useful theories

- **Examples**: Theory of equality with uninterpreted functions, theory of rationals, theory of integers
Motivation

- So far, learned about decision procedures for useful theories

- **Examples:** Theory of equality with uninterpreted functions, theory of rationals, theory of integers

- But in many cases, we need to decide satisfiability of formulas involving multiple theories
Motivation

- So far, learned about decision procedures for useful theories

- **Examples:** Theory of equality with uninterpreted functions, theory of rationals, theory of integers

- But in many cases, we need to decide satisfiability of formulas involving multiple theories

- **Example:** $1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$
Motivation

- So far, learned about decision procedures for useful theories

- **Examples:** Theory of equality with uninterpreted functions, theory of rationals, theory of integers

- But in many cases, we need to decide satisfiability of formulas involving multiple theories

- **Example:** $1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$

- This formula does not belong to any individual theory
Motivation

- So far, learned about decision procedures for useful theories

- **Examples:** Theory of equality with uninterpreted functions, theory of rationals, theory of integers

- But in many cases, we need to decide satisfiability of formulas involving multiple theories

- **Example:** $1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$

- This formula does not belong to any individual theory

- But it does belong, for instance, to combination of $T_-$ and $T_Z$
Overview

► **Recall:** Given two theories $T_1$ and $T_2$ that have the $=$ predicate, we define a combined theory $T_1 \cup T_2$
Overview

- **Recall**: Given two theories $T_1$ and $T_2$ that have the $=$ predicate, we define a combined theory $T_1 \cup T_2$

- Signature of $T_1 \cup T_2$: $\Sigma_1 \cup \Sigma_2$
Overview

- **Recall:** Given two theories $T_1$ and $T_2$ that have the $=$ predicate, we define a combined theory $T_1 \cup T_2$

- **Signature of** $T_1 \cup T_2$: $\Sigma_1 \cup \Sigma_2$

- **Axioms of** $T_1 \cup T_2$: $A_1 \cup A_2$
Overview

- **Recall:** Given two theories $T_1$ and $T_2$ that have the $=$ predicate, we define a combined theory $T_1 \cup T_2$

- **Signature of** $T_1 \cup T_2$: $\Sigma_1 \cup \Sigma_2$

- **Axioms of** $T_1 \cup T_2$: $A_1 \cup A_2$

- Given decision procedures for $T_1$ and $T_2$, we want a decision procedure to decide satisfiability of formulas in $T_1 \cup T_2$
Overview

- **Recall**: Given two theories $T_1$ and $T_2$ that have the $=$ predicate, we define a combined theory $T_1 \cup T_2$

- **Signature of** $T_1 \cup T_2$: $\Sigma_1 \cup \Sigma_2$

- **Axioms of** $T_1 \cup T_2$: $A_1 \cup A_2$

- Given decision procedures for $T_1$ and $T_2$, we want a decision procedure to decide satisfiability of formulas in $T_1 \cup T_2$

- **Today's lecture**: Learn about Nelson-Oppen method for constructing decision procedure for combined theory $T_1 \cup T_2$ from individual decision procedures for $T_1$ and $T_2$
Nelson-Oppen Overview

For instance, to combine $T_1$, $T_2$, $T_3$, first combine $T_1$, $T_2$.

Then, combine $T_1 \cup T_2$ and $T_3$ again using Nelson-Oppen.

However, Nelson-Oppen imposes some restrictions on theories that can be combined.
Nelson-Oppen Overview

This method also allows combining arbitrary number of theories

Σ₁-theory $T_1$

$P_1$ for $T_1$-satisfiability

Σ₂-theory $T_2$

$P_2$ for $T_2$-satisfiability

$P$ for $(T_1 \cup T_2)$-satisfiability

Nelson-Oppen

- This method also allows combining arbitrary number of theories
Nelson-Oppen Overview

This method also allows combining arbitrary number of theories

For instance, to combine $T_1, T_2, T_3$, first combine $T_1, T_2$
This method also allows combining arbitrary number of theories

For instance, to combine $T_1, T_2, T_3$, first combine $T_1, T_2$

Then, combine $T_1 \cup T_2$ and $T_3$ again using Nelson-Oppen
This method also allows combining arbitrary number of theories

For instance, to combine $T_1, T_2, T_3$, first combine $T_1, T_2$

Then, combine $T_1 \cup T_2$ and $T_3$ again using Nelson-Oppen

However, Nelson-Oppen imposes some restrictions on theories that can be combined
Restrictions of Nelson-Oppen

- Nelson-Oppen method imposes the following restrictions:
  1. Only allows combining quantifier-free fragments
  2. Only allows combining formulas without disjunctions, but not a major limitation because can convert to DNF
  3. Signatures can only share equality: $\Sigma_1 \cap \Sigma_2 = \{=\}$
  4. Theories $T_1$ and $T_2$ must be stably infinite

- Theory $T$ is stably infinite iff every satisfiable qff formula is satisfiable in a universe of discourse with infinite cardinality.
- In other words, if qff $F$ is satisfiable, then there exists $T$-model that satisfies $F$ and has infinite cardinality.
- Thus, theories with only finite models are not stably infinite.
Restrictions of Nelson-Oppen

- Nelson-Oppen method imposes the following restrictions:
  1. Only allows combining quantifier-free fragments
Restrictions of Nelson-Oppen

- Nelson-Oppen method imposes the following restrictions:
  
  1. Only allows combining quantifier-free fragments
  
  2. Only allows combining formulas without disjunctions, but not a major limitation because can convert to DNF
Restrictions of Nelson-Oppen

- Nelson-Oppen method imposes the following restrictions:
  
  1. Only allows combining quantifier-free fragments
  
  2. Only allows combining formulas without disjunctions, but not a major limitation because can convert to DNF
  
  3. Signatures can only share equality: $\Sigma_1 \cap \Sigma_2 = \{=\}$
Restrictions of Nelson-Oppen

- Nelson-Oppen method imposes the following restrictions:
  1. Only allows combining quantifier-free fragments
  2. Only allows combining formulas without disjunctions, but not a major limitation because can convert to DNF
  3. Signatures can only share equality: $\Sigma_1 \cap \Sigma_2 = \{=\}$
  4. Theories $T_1$ and $T_2$ must be stably infinite
Restrictions of Nelson-Oppen

N Nelson-Oppen method imposes the following restrictions:

1. Only allows combining quantifier-free fragments

2. Only allows combining formulas without disjunctions, but not a major limitation because can convert to DNF

3. Signatures can only share equality: $\Sigma_1 \cap \Sigma_2 = \{=\}$

4. Theories $T_1$ and $T_2$ must be stably infinite

Theory $T$ is stably infinite if every satisfiable qff formula is satisfiable in a universe of discourse with infinite cardinality.
Restrictions of Nelson-Oppen

- Nelson-Oppen method imposes the following restrictions:

  1. Only allows combining quantifier-free fragments

  2. Only allows combining formulas without disjunctions, but not a major limitation because can convert to DNF

  3. Signatures can only share equality: \( \Sigma_1 \cap \Sigma_2 = \{=\} \)

  4. Theories \( T_1 \) and \( T_2 \) must be stably infinite

- Theory \( T \) is stably infinite iff every satisfiable qff formula is satisfiable in a universe of discourse with infinite cardinality

- In other words, if qff \( F \) is satisfiable, then there exists \( T \)-model that satisfies \( F \) and has infinite cardinality.
Restrictions of Nelson-Oppen

- Nelson-Oppen method imposes the following restrictions:

1. Only allows combining quantifier-free fragments

2. Only allows combining formulas without disjunctions, but not a major limitation because can convert to DNF

3. Signatures can only share equality: $\Sigma_1 \cap \Sigma_2 = \{=\}$

4. Theories $T_1$ and $T_2$ must be stably infinite

- Theory $T$ is stably infinite iff every satisfiable qff formula is satisfiable in a universe of discourse with infinite cardinality

- In other words, if qff $F$ is satisfiable, then there exists $T$-model that satisfies $F$ and has infinite cardinality.

- Thus, theories with only finite models are not stably infinite.
Example of Non-Stably Infinite Theory

Signature: \{a, b, =\}
Axiom: \( \forall x. x = a \lor x = b \)
Example of Non-Stably Infinite Theory

Signature: \{a, b, =\}
Axiom: \forall x. x = a \lor x = b

- Axiom says that any object in the universe of discourse must be equal to either \(a\) or \(b\)
Example of Non-Stably Infinite Theory

Signature : \{ a, b, = \}
Axiom : \forall x. x = a \lor x = b

- Axiom says that any object in the universe of discourse must be equal to either \( a \) or \( b \)

- Now consider \( U \) containing more than 2 elements

Hence, theory only has finite models, and is not stably infinite.
Example of Non-Stably Infinite Theory

Signature: \{a, b, =\}  
Axiom: \(\forall x. x = a \lor x = b\)

- Axiom says that any object in the universe of discourse must be equal to either \(a\) or \(b\)

- Now consider \(U\) containing more than 2 elements

- Then, there is at least one element distinct from both \(a\) and \(b\)
Example of Non-Stably Infinite Theory

Signature: \( \{a, b, =\} \)

Axiom: \( \forall x. x = a \lor x = b \)

- Axiom says that any object in the universe of discourse must be equal to either \( a \) or \( b \)

- Now consider \( U \) containing more than 2 elements

- Then, there is at least one element distinct from both \( a \) and \( b \)

- Thus, any \( U \) with more than 2 elements violates axiom
Example of Non-Stably Infinite Theory

Signature: \{a, b, =\}
Axiom: \forall x. x = a \lor x = b

- Axiom says that any object in the universe of discourse must be equal to either a or b

- Now consider \( U \) containing more than 2 elements

- Then, there is at least one element distinct from both a and b

- Thus, any \( U \) with more than 2 elements violates axiom

- Hence, theory only has finite models, and is not stably infinite
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite
- All theories we discussed, $T$, $T_Q$, $T_Z$, $T_A$, are stably infinite
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite.
- All theories we discussed, $T_\equiv$, $T_\mathbb{Q}$, $T_\mathbb{Z}$, $T_A$, are stably infinite.
- Which of these theories can we combine using Nelson-Oppen?
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite
- All theories we discussed, \( T =, T_Q, T_Z, T_A \), are stably infinite
- Which of these theories can we combine using Nelson-Oppen?
  1. \( T = \) and \( T_Q \)?
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite.

- All theories we discussed, $T_\equiv$, $T_\mathbb{Q}$, $T_\mathbb{Z}$, $T_\mathcal{A}$, are stably infinite.

- Which of these theories can we combine using Nelson-Oppen?
  1. $T_\equiv$ and $T_\mathbb{Q}$? yes
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite

- All theories we discussed, $T_\mathbb{N}$, $T_\mathbb{Q}$, $T_\mathbb{Z}$, $T_A$, are stably infinite

- Which of these theories can we combine using Nelson-Oppen?
  1. $T_\mathbb{N}$ and $T_\mathbb{Q}$? yes
  2. $T_\mathbb{N}$ and $T_\mathbb{Z}$?
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite

- All theories we discussed, $T_\equiv$, $T_Q$, $T_Z$, $T_A$, are stably infinite

- Which of these theories can we combine using Nelson-Oppen?
  1. $T_\equiv$ and $T_Q$? yes
  2. $T_\equiv$ and $T_Z$? yes
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite

- All theories we discussed, $T_\mathbb{R}$, $T_\mathbb{Q}$, $T_\mathbb{Z}$, $T_\mathbb{A}$, are stably infinite

- Which of these theories can we combine using Nelson-Oppen?
  1. $T_\mathbb{R}$ and $T_\mathbb{Q}$? yes
  2. $T_\mathbb{R}$ and $T_\mathbb{Z}$? yes
  3. $T_\mathbb{A}$ and $T_\mathbb{Z}$?
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite

- All theories we discussed, $T_\equiv$, $T_Q$, $T_Z$, $T_A$, are stably infinite

- Which of these theories can we combine using Nelson-Oppen?
  1. $T_\equiv$ and $T_Q$? yes
  2. $T_\equiv$ and $T_Z$? yes
  3. $T_A$ and $T_Z$? yes
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite.

- All theories we discussed, $T_{=}$, $T_{\mathbb{Q}}$, $T_{\mathbb{Z}}$, $T_{A}$, are stably infinite.

- Which of these theories can we combine using Nelson-Oppen?
  1. $T_{=}$ and $T_{\mathbb{Q}}$?  yes
  2. $T_{=}$ and $T_{\mathbb{Z}}$?  yes
  3. $T_{A}$ and $T_{\mathbb{Z}}$?  yes

- In general, almost any theory we care about can be combined using Nelson-Oppen.
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite
- All theories we discussed, $T_{\equiv}$, $T_\mathbb{Q}$, $T_\mathbb{Z}$, $T_A$, are stably infinite
- Which of these theories can we combine using Nelson-Oppen?
  1. $T_{\equiv}$ and $T_\mathbb{Q}$? yes
  2. $T_{\equiv}$ and $T_\mathbb{Z}$? yes
  3. $T_A$ and $T_\mathbb{Z}$? yes
- In general, almost any theory we care about can be combined using Nelson-Oppen
Examples of Stably Infinite Theories

- Fortunately, almost any theory of interest is stably infinite.
- All theories we discussed, $T_\equiv$, $T_Q$, $T_Z$, $T_A$, are stably infinite.
- Which of these theories can we combine using Nelson-Oppen?
  1. $T_\equiv$ and $T_Q$? yes
  2. $T_\equiv$ and $T_Z$? yes
  3. $T_A$ and $T_Z$? yes
- In general, almost any theory we care about can be combined using Nelson-Oppen.
- More recent work has also extended Nelson-Oppen to non-stably-infinite theories.
Nelson-Oppen Overview

- Nelson-Oppen method has conceptually two different phases:
  1. Purification: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$
  2. Equality propagation: Propagate all relevant equalities between theories

- Purification step is always the same for any arbitrary theory
- But equality propagation is different between convex and non-convex theories
Nelson-Oppen Overview

- Nelson-Oppen method has conceptually two-different phases:

  1. Purification: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$
Nelson-Oppen Overview

- Nelson-Oppen method has conceptually two-different phases:

1. **Purification**: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$

2. **Equality propagation**: Propagate all relevant equalities between theories
Nelson-Oppen Overview

- Nelson-Oppen method has conceptually two different phases:

  1. **Purification**: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$

  2. **Equality propagation**: Propagate all relevant equalities between theories

- Purification step is always the same for any arbitrary theory
Nelson-Oppen Overview

- Nelson-Oppen method has conceptually two-different phases:

  1. **Purification**: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$

  2. **Equality propagation**: Propagate all relevant equalities between theories

- Purification step is always the same for any arbitrary theory

- But equality propagation is different between **convex** and **non-convex** theories
Purification Overview

- Input to Nelson-Oppen is formula $F$ in $T_1 \cup T_2$
Purification Overview

- Input to Nelson-Oppen is formula $F$ in $T_1 \cup T_2$
- Goal of purification is to separate $F$ into formulas $F_1$ and $F_2$ such that:
  1. $F_1$ belongs only to $T_1$ (is "pure")
  2. $F_2$ belongs only to $T_2$ (is "pure")
  3. $F_1 \land F_2$ is equisatisfiable as $F$
Purification Overview

- Input to Nelson-Oppen is formula $F$ in $T_1 \cup T_2$

- Goal of purification is to separate $F$ into formulas $F_1$ and $F_2$ such that:
  1. $F_1$ belongs only to $T_1$ (is "pure")
Purification Overview

- Input to Nelson-Oppen is formula $F$ in $T_1 \cup T_2$

- Goal of purification is to separate $F$ into formulas $F_1$ and $F_2$ such that:
  
  1. $F_1$ belongs only to $T_1$ (is "pure")
  2. $F_2$ belong only to $T_2$ (is "pure")
Purification Overview

- Input to Nelson-Oppen is formula $F$ in $T_1 \cup T_2$

- Goal of purification is to separate $F$ into formulas $F_1$ and $F_2$ such that:
  1. $F_1$ belongs only to $T_1$ (is "pure")
  2. $F_2$ belong only to $T_2$ (is "pure")
  3. $F_1 \land F_2$ is equisatisfiable as $F$

Resulting formula after purification is not equivalent.
But since goal is to decide satisfiability, this is good enough.
Purification Overview

- Input to Nelson-Oppen is formula $F$ in $T_1 \cup T_2$

- Goal of purification is to separate $F$ into formulas $F_1$ and $F_2$ such that:

  1. $F_1$ belongs only to $T_1$ (is "pure")
  2. $F_2$ belong only to $T_2$ (is "pure")
  3. $F_1 \land F_2$ is equisatisfiable as $F$

- Resulting formula after purification is not equivalent
Purification Overview

- Input to Nelson-Oppen is formula $F$ in $T_1 \cup T_2$

- Goal of purification is to separate $F$ into formulas $F_1$ and $F_2$ such that:
  
  1. $F_1$ belongs only to $T_1$ (is "pure")
  
  2. $F_2$ belong only to $T_2$ (is "pure")
  
  3. $F_1 \land F_2$ is equisatisfiable as $F$

- Resulting formula after purification is not equivalent

- But since goal is to decide satisfiability, this is good enough
How To Purify

- To purify formula $F$, exhaustively apply the following:

1. Consider term $f(\ldots, t_i, \ldots)$. If $f \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $z$ and conjoin $z = t_i$.

2. Consider predicate $p(\ldots, t_i, \ldots)$. If $p \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $w$ and conjoin $w = t_i$.

Literals in resulting formula belong to either only $T_1$ or $T_2$.

Thus, we can write $F$ as a conjunction of formulas $F_1$ in $T_1$ and $F_2$ in $T_2$. 
How To Purify

> To purify formula $F$, exhaustively apply the following:

1. Consider term $f(\ldots, t_i, \ldots)$. If $f \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $z$ and conjoin $z = t_i$
How To Purify

To purify formula $F$, exhaustively apply the following:

1. Consider term $f(\ldots, t_i, \ldots)$. If $f \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $z$ and conjoin $z = t_i$

2. Consider predicate $p(\ldots, t_i, \ldots)$. If $p \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $w$ and conjoin $w = t_i$
How To Purify

▸ To purify formula $F$, exhaustively apply the following:

1. Consider term $f(\ldots, t_i, \ldots)$. If $f \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $z$ and conjoin $z = t_i$

2. Consider predicate $p(\ldots, t_i, \ldots)$. If $p \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $w$ and conjoin $w = t_i$

▸ Literals in resulting formula belong to either only $T_1$ or $T_2$. 
How To Purify

To purify formula $F$, exhaustively apply the following:

1. Consider term $f(\ldots, t_i, \ldots)$. If $f \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $z$ and conjoin $z = t_i$

2. Consider predicate $p(\ldots, t_i, \ldots)$. If $p \in \Sigma_i$ but $t_i$ is not a term in $T_i$, replace $t_i$ with fresh variable $w$ and conjoin $w = t_i$

- Literals in resulting formula belong to either only $T_1$ or $T_2$.

- Thus, we can write $F$ as a conjunction of formulas $F_1$ in $T_1$ and $F_2$ in $T_2$
Purification Example 1

- Consider $T = \cup T_Q$ formula $x \leq f(x) + 1$
Purification Example 1

- Consider $T_{\equiv} \cup T_{\mathbb{Q}}$ formula $x \leq f(x) + 1$

- Is this formula already pure?
Purification Example 1

- Consider $T \cup T_Q$ formula $x \leq f(x) + 1$

- Is this formula already pure? No
Purification Example 1

- Consider $T_{=} \cup T_Q$ formula $x \leq f(x) + 1$

- Is this formula already pure? No

- Since $f(x)$ is not in $T_Q$, replace with new variable $y$ and add equality constraint $y = f(x)$
Purification Example 1

- Consider $T_\leq \cup T_Q$ formula $x \leq f(x) + 1$

- Is this formula already pure? No

- Since $f(x)$ is not in $T_Q$, replace with new variable $y$ and add equality constraint $y = f(x)$

- Thus, formula after purification:

$$x \leq y + 1 \land y = f(x)$$
Purification Example II

Consider following $\Sigma = \Sigma_{=} \cup \Sigma_{\mathbb{Z}}$ formula:

$$f(x + g(y)) \leq g(a) + f(b)$$
Purification Example II

- Consider following \( \Sigma_\text{=} \cup \Sigma_\mathbb{Z} \) formula:

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

- Easiest to purify "inside out"
Consider following $\Sigma_{=} \cup \Sigma_{\mathbb{Z}}$ formula:

$$f(x + g(y)) \leq g(a) + f(b)$$

Easiest to purify "inside out"

Is the term $x + g(y)$ pure?
Consider following $\Sigma_\equiv \cup \Sigma_\mathbb{Z}$ formula:

$$f(x + g(y)) \leq g(a) + f(b)$$

Easiest to purify "inside out"

Is the term $x + g(y)$ pure? no
Purification Example II

- Consider following $\Sigma_\equiv \cup \Sigma_\mathbb{Z}$ formula:

$$f(x + g(y)) \leq g(a) + f(b)$$

- Easiest to purify "inside out"

- Is the term $x + g(y)$ pure? no

- How do we purify it?
Consider following $\Sigma_E \cup \Sigma_Z$ formula:

$$f(x + g(y)) \leq g(a) + f(b)$$

Easiest to purify "inside out"

Is the term $x + g(y)$ pure? no

How do we purify it? replace $g(y)$ with $z_1$, add constraint $z_1 = g(y)$
Consider following \( \Sigma = \Sigma_{= \cup} \Sigma_{\mathbb{Z}} \) formula:

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

Easiest to purify "inside out"

Is the term \( x + g(y) \) pure? no

How do we purify it? replace \( g(y) \) with \( z_1 \), add constraint \( z_1 = g(y) \)

Resulting formula:

\[
f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y)
\]
Purification Example II, cont

\[ f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y) \]

- Is \( f(x + z_1) \) pure?
Purification Example II, cont

\[ f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y) \]

- Is \( f(x + z_1) \) pure? no
Purification Example II, cont

\[ f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y) \]

- Is \( f(x + z_1) \) pure? no

- How do we purify?
Purification Example II, cont

\[ f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y) \]

- Is \( f(x + z_1) \) pure? no

- How do we purify? replace \( x + z_1 \) with \( z_2 \), add constraint \( z_2 = x + z_1 \)
Purification Example II, cont

\[ f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y) \]

- Is \( f(x + z_1) \) pure? no

- How do we purify? replace \( x + z_1 \) with \( z_2 \), add constraint \( z_2 = x + z_1 \)

- Resulting formula:
  \[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]
Purification Example II, cont

\[ f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y) \]

- Is \( f(x + z_1) \) pure? no

- How do we purify? replace \( x + z_1 \) with \( z_2 \), add constraint \( z_2 = x + z_1 \)

- Resulting formula:

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Is formula purified now?
Purification Example II, cont

\[ f(x + z_1) \leq g(a) + f(b) \land z_1 = g(y) \]

- Is \( f(x + z_1) \) pure? no

- How do we purify? replace \( x + z_1 \) with \( z_2 \), add constraint \( z_2 = x + z_1 \)

- Resulting formula:

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Is formula purified now? no
Purification Example II, cont

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Which terms/predicate is impure?
Purification Example II, cont

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Which terms/predicate is impure? \( g(a) + f(b) \)
Purification Example II, cont

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Which terms/predicate is impure? \( g(a) + f(b) \)

- How do we purify?

\[ \text{Resulting formula:} \quad f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \]

- Is formula purified now? No
Purification Example II, cont

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Which terms/predicate is impure? \( g(a) + f(b) \)

- How do we purify? replace \( g(a) \) with \( z_3 \) and \( f(b) \) with \( z_4 \), add constraint \( z_3 = g(a) \land z_4 = f(b) \)
Purification Example II, cont

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Which terms/predicate is impure? \( g(a) + f(b) \)

- How do we purify? replace \( g(a) \) with \( z_3 \) and \( f(b) \) with \( z_4 \), add constraint \( z_3 = g(a) \land z_4 = f(b) \)

- Resulting formula:

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]
Purification Example II, cont

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Which terms/predicate is impure? \( g(a) + f(b) \)

- How do we purify? replace \( g(a) \) with \( z_3 \) and \( f(b) \) with \( z_4 \), add constraint \( z_3 = g(a) \land z_4 = f(b) \)

- Resulting formula:

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Is formula purified now?

\[ \text{no} \]
Purification Example II, cont

\[ f(z_2) \leq g(a) + f(b) \land z_1 = g(y) \land z_2 = x + z_1 \]

- Which terms/predicate is impure? \( g(a) + f(b) \)

- How do we purify? replace \( g(a) \) with \( z_3 \) and \( f(b) \) with \( z_4 \), add constraint \( z_3 = g(a) \land z_4 = f(b) \)

- Resulting formula:
  \[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Is formula purified now? no
Purification Example II, cont

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Which terms/predicate is impure?
Purification Example II, cont

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Which terms/predicate is impure? \( f(z_2) \leq z_3 + z_4 \)
Purification Example II, cont

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Which terms/predicate is impure? \( f(z_2) \leq z_3 + z_4 \)

- How do we purify?

\[ \text{Resulting formula:} \]

\[ z_5 \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \land z_5 = f(z_2) \]

- Is formula purified now? Yes, finally!
Purification Example II, cont

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Which terms/predicate is impure? \( f(z_2) \leq z_3 + z_4 \)
- How do we purify? replace \( f(z_2) \) with \( z_5 \), add constraint \( z_5 = f(z_2) \)
Purification Example II, cont

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Which terms/predicate is impure? \( f(z_2) \leq z_3 + z_4 \)

- How do we purify? replace \( f(z_2) \) with \( z_5 \), add constraint \( z_5 = f(z_2) \)

- Resulting formula:

\[
\begin{align*}
z_5 & \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land \\
z_3 & = g(a) \land z_4 = f(b) \land z_5 = f(z_2)
\end{align*}
\]
Purification Example II, cont

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Which terms/predicate is impure? \( f(z_2) \leq z_3 + z_4 \)

- How do we purify? replace \( f(z_2) \) with \( z_5 \), add constraint \( z_5 = f(z_2) \)

- Resulting formula:
  
  \[ z_5 \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land \]
  
  \[ z_3 = g(a) \land z_4 = f(b) \land z_5 = f(z_2) \]

- Is formula purified now?

\[ \text{Yes, finally!} \]
Purification Example II, cont

\[ f(z_2) \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \]

- Which terms/predicate is impure? \( f(z_2) \leq z_3 + z_4 \)

- How do we purify? replace \( f(z_2) \) with \( z_5 \), add constraint \( z_5 = f(z_2) \)

- Resulting formula:

\[ z_5 \leq z_3 + z_4 \land z_1 = g(y) \land z_2 = x + z_1 \land z_3 = g(a) \land z_4 = f(b) \land z_5 = f(z_2) \]

- Is formula purified now? Yes, finally!
Shared vs. Unshared Variables

- After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$.
Shared vs. Unshared Variables

- After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$

- If $x$ occurs in both $F_1$ and $F_2$, $x$ is called shared variable
Shared vs. Unshared Variables

- After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$.

- If $x$ occurs in both $F_1$ and $F_2$, $x$ is called shared variable.

- If $y$ occurs only in $F_1$ or only in $F_2$, it is called unshared variable.
Shared vs. Unshared Variables

- After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$

- If $x$ occurs in both $F_1$ and $F_2$, $x$ is called shared variable

- If $y$ occurs only in $F_1$ or only in $F_2$, it is called unshared variable

- Consider the following purified formula:

  \[
  w_1 = x + y \land y = 1 \land w_2 = 2 \land w_1 = f(x) \land f(x) \neq f(w_2) \quad T_Z \land T_=
  \]
Shared vs. Unshared Variables

- After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$.

- If $x$ occurs in both $F_1$ and $F_2$, $x$ is called shared variable.

- If $y$ occurs only in $F_1$ or only in $F_2$, it is called unshared variable.

- Consider the following purified formula:

\[
\begin{align*}
\text{T}_{\mathbb{Z}} & : w_1 = x + y \land y = 1 \land w_2 = 2 \\
\text{T}_{=} & : w_1 = f(x) \land f(x) \neq f(w_2)
\end{align*}
\]

- Which variables are shared?
Shared vs. Unshared Variables

- After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$

- If $x$ occurs in both $F_1$ and $F_2$, $x$ is called shared variable

- If $y$ occurs only in $F_1$ or only in $F_2$, it is called unshared variable

- Consider the following purified formula:
  
  \[ w_1 = x + y \land y = 1 \land w_2 = 2 \land w_1 = f(x) \land f(x) \neq f(w_2) \]

  \[ T_Z \land T_\leq \]

- Which variables are shared? $w_1, x, w_2$
Shared vs. Unshared Variables

- After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$

- If $x$ occurs in both $F_1$ and $F_2$, $x$ is called shared variable

- If $y$ occurs only in $F_1$ or only in $F_2$, it is called unshared variable

- Consider the following purified formula:

$$w_1 = x + y \land y = 1 \land w_2 = 2 \land w_1 = f(x) \land f(x) \neq f(w_2)$$

- Which variables are shared? $w_1, x, w_2$

- Which variables are unshared?
After purification, we have decomposed a formula $F$ into two pure formulas $F_1$ and $F_2$.

- If $x$ occurs in both $F_1$ and $F_2$, $x$ is called **shared variable**

- If $y$ occurs only in $F_1$ or only in $F_2$, it is called **unshared variable**

Consider the following purified formula:

$$w_1 = x + y \land y \equiv 1 \land w_2 = 2 \land w_1 = f(x) \land f(x) \neq f(w_2)$$

- Which variables are shared? $w_1, x, w_2$

- Which variables are unshared? $y$
Two Phases of Nelson-Oppen

- **Recall**: Nelson-Oppen method has two different phases:

  1. **Purification**: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$

  2. **Equality propagation**: Propagate all relevant equalities between theories
Two Phases of Nelson-Oppen

- **Recall**: Nelson-Oppen method has two different phases:

  1. **Purification**: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$

  2. **Equality propagation**: Propagate all relevant equalities between theories

- Talk about second phase next
Two Phases of Nelson-Oppen

- Recall: Nelson-Oppen method has two different phases:

  1. **Purification**: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$

  2. **Equality propagation**: Propagate all relevant equalities between theories

- Talk about second phase next

- But this phase is different for convex vs. non-convex theories
Two Phases of Nelson-Oppen

- **Recall**: Nelson-Oppen method has two different phases:

  1. **Purification**: Separate formula $F$ in $T_1 \cup T_2$ into two formulas $F_1$ in $T_1$ and $F_2$ in $T_2$.

  2. **Equality propagation**: Propagate all relevant equalities between theories.

- Talk about second phase next.

- But this phase is different for convex vs. non-convex theories.

- So, need to talk about convex and non-convex theories.
Convex Theories

- Theory $T$ is called convex if for every conjunctive formula $F$:

  - If $F \Rightarrow \bigvee_{i=1}^{n} x_i = y_i$ for finite $n$
  - Then, $F \Rightarrow x_i = y_i$ for some $i \in [1, n]$

  Thus, in convex theory, if $F$ implies disjunction of equalities, $F$ also implies at least one of these equalities on its own.

  If a theory does not satisfy this condition, it is called non-convex.
Convex Theories

- Theory $T$ is called convex if for every conjunctive formula $F$:
  - If $F \Rightarrow \bigvee_{i=1}^{n} x_i = y_i$ for finite $n$
Theory $T$ is called convex if for every conjunctive formula $F$:

- If $F \Rightarrow \bigvee_{i=1}^{n} x_i = y_i$ for finite $n$

- Then, $F \Rightarrow x_i = y_i$ for some $i \in [1, n]$
Convex Theories

- Theory $T$ is called **convex** if for every conjunctive formula $F$:
  - If $F \Rightarrow \bigvee_{i=1}^{n} x_i = y_i$ for finite $n$
  - Then, $F \Rightarrow x_i = y_i$ for some $i \in [1, n]$

- Thus, in convex theory, if $F$ implies disjunction of equalities, $F$ also implies at least one of these equalities on its own
Theory $T$ is called convex if for every conjunctive formula $F$:

- If $F \Rightarrow \bigvee_{i=1}^{n} x_i = y_i$ for finite $n$

- Then, $F \Rightarrow x_i = y_i$ for some $i \in [1, n]$ 

Thus, in convex theory, if $F$ implies disjunction of equalities, $F$ also implies at least one of these equalities on its own.

- If a theory does not satisfy this condition, it is called non-convex
Examples of Convex and Non-Convex Theories

- **Example:** Consider formula $1 \leq x \land x \leq 2$ in $T_Z$.
Examples of Convex and Non-Convex Theories

- **Example:** Consider formula $1 \leq x \wedge x \leq 2$ in $T_Z$

- Does it imply $x = 1 \lor x = 2$?

- No
Examples of Convex and Non-Convex Theories

▶ Example: Consider formula $1 \leq x \land x \leq 2$ in $T_Z$

▶ Does it imply $x = 1 \lor x = 2$? yes
Examples of Convex and Non-Convex Theories

- **Example**: Consider formula $1 \leq x \land x \leq 2$ in $T_Z$

- Does it imply $x = 1 \lor x = 2$? *yes*

- Does it imply $x = 1$?

- Is $T_Z$ convex? *no*
Examples of Convex and Non-Convex Theories

- **Example**: Consider formula $1 \leq x \land x \leq 2$ in $T_\mathbb{Z}$

- Does it imply $x = 1 \lor x = 2$? **yes**

- Does it imply $x = 1$? **no**
Examples of Convex and Non-Convex Theories

▶ Example: Consider formula $1 \leq x \land x \leq 2$ in $T_Z$

▶ Does it imply $x = 1 \lor x = 2$? yes

▶ Does it imply $x = 1$? no

▶ Does it imply $x = 2$?

▶ Is $T_Z$ convex? no
Examples of Convex and Non-Convex Theories

- **Example**: Consider formula $1 \leq x \land x \leq 2$ in $T_Z$

- Does it imply $x = 1 \lor x = 2$? **yes**

- Does it imply $x = 1$? **no**

- Does it imply $x = 2$? **no**
Examples of Convex and Non-Convex Theories

- **Example**: Consider formula $1 \leq x \land x \leq 2$ in $T_Z$

- Does it imply $x = 1 \lor x = 2$? **yes**

- Does it imply $x = 1$? **no**

- Does it imply $x = 2$? **no**

- Is $T_Z$ convex?
Examples of Convex and Non-Convex Theories

▶ Example: Consider formula $1 \leq x \land x \leq 2$ in $T_Z$

▶ Does it imply $x = 1 \lor x = 2$? yes

▶ Does it imply $x = 1$? no

▶ Does it imply $x = 2$? no

▶ Is $T_Z$ convex? no
Examples of Convex and Non-Convex Theories

▶ Example: Consider formula $1 \leq x \land x \leq 2$ in $T_Z$

▶ Does it imply $x = 1 \lor x = 2$? yes

▶ Does it imply $x = 1$? no

▶ Does it imply $x = 2$? no

▶ Is $T_Z$ convex? no
Examples of Convex and Non-Convex Theories

- **Example:** Consider formula $1 \leq x \land x \leq 2$ in $T_\mathbb{Z}$
  - Does it imply $x = 1 \lor x = 2$? yes
  - Does it imply $x = 1$? no
  - Does it imply $x = 2$? no
  - Is $T_\mathbb{Z}$ convex? no
  - Theory of equality $T_=$ is convex
Nelson-Oppen for Convex vs Non-Convex Theories

- Combining decision procedures for two convex theories is easier and more efficient.
Combining decision procedures for two convex theories is easier and more efficient.

Intuition: When we have convexity, there are fewer facts that need to be communicated between theories.
Combining decision procedures for two convex theories is easier and more efficient.

Intuition: When we have convexity, there are fewer facts that need to be communicated between theories.

Unfortunately, some theories of interest such as $T_Z$ and theory of arrays are non-convex.
Combining decision procedures for two convex theories is easier and more efficient.

Intuition: When we have convexity, there are fewer facts that need to be communicated between theories.

Unfortunately, some theories of interest such as $T_Z$ and theory of arrays are non-convex.

If one of the theories we want to combine is non-convex, decision procedure for combination theory is much less efficient.
Nelson-Oppen for Convex vs Non-Convex Theories

- Combining decision procedures for two convex theories is easier and more efficient.

- Intuition: When we have convexity, there are fewer facts that need to be communicated between theories.

- Unfortunately, some theories of interest such as $T_\mathbb{Z}$ and theory of arrays are non-convex.

- If one of the theories we want to combine is non-convex, decision procedure for combination theory is much less efficient.

- We’ll first talk about Nelson-Oppen method for convex theories, then for non-convex theories.
Nelson-Oppen Method for Convex Theories

- Given formula $F$ in $T_1 \cup T_2$ ($T_1$, $T_2$ convex), want to decide if $F$ is satisfiable

- First, purify $F$ into $F_1$ and $F_2$.
- Run decision procedures for $T_1$, $T_2$ to decide sat. of $F_1$, $F_2$.
- If either is unsat, $F$ is unsatisfiable. Why?
  - Because $F$ is equisatisfiable to $F_1 \land F_2$, which is unsat.
Given formula $F$ in $T_1 \cup T_2$ ($T_1$, $T_2$ convex), want to decide if $F$ is satisfiable

First, purify $F$ into $F_1$ and $F_2$
Nelson-Oppen Method for Convex Theories

- Given formula $F$ in $T_1 \cup T_2$ ($T_1$, $T_2$ convex), want to decide if $F$ is satisfiable

- First, purify $F$ into $F_1$ and $F_2$

- Run decision procedures for $T_1$, $T_2$ to decide sat. of $F_1$, $F_2$
Nelson-Oppen Method for Convex Theories

- Given formula $F$ in $T_1 \cup T_2$ ($T_1, T_2$ convex), want to decide if $F$ is satisfiable
- First, purify $F$ into $F_1$ and $F_2$
- Run decision procedures for $T_1, T_2$ to decide sat. of $F_1, F_2$
- If either is unsat, $F$ is unsatisfiable. Why?
Nelson-Oppen Method for Convex Theories

- Given formula $F$ in $T_1 \cup T_2$ ($T_1$, $T_2$ convex), want to decide if $F$ is satisfiable

- First, purify $F$ into $F_1$ and $F_2$

- Run decision procedures for $T_1$, $T_2$ to decide sat. of $F_1$, $F_2$

- If either is unsat, $F$ is unsatisfiable. Why?

- Because $F$ is equisatisfiable to $F_1 \land F_2$, which is unsat
Nelson-Oppen Method for Convex Theories

- If both are SAT, does this mean $F$ is sat?

Example:

\[ x + y = 2 \land x = 1 \]

Here, $F_1$ and $F_2$ are individually sat, but their combination is unsat b/c $TZ$ implies $x = y$.

In the case where $F_1$ and $F_2$ are sat, theories have to exchange all implied equalities.

Why only equalities? b/c it is the only shared symbol.
Nelson-Oppen Method for Convex Theories

- If both are SAT, does this mean $F$ is sat?

- No because if $F_1$ and $F_2$ are individually satisfiable, $F_1 \land F_2$ does not have to be satisfiable.
Nelson-Oppen Method for Convex Theories

- If both are SAT, does this mean $F$ is sat?

- No because if $F_1$ and $F_2$ are individually satisfiable, $F_1 \land F_2$ does not have to be satisfiable

- Example: $x + y = 2 \land x = 1 \land f(x) \neq f(y)$

Here, $F_1$ and $F_2$ are individually sat, but their combination is unsat b/c $T_Z$ implies $x = y$.

In the case where $F_1$ and $F_2$ are sat, theories have to exchange all implied equalities.

Why only equalities? b/c it is the only shared symbol.
Nelson-Oppen Method for Convex Theories

- If both are SAT, does this mean $F$ is sat?
- No because if $F_1$ and $F_2$ are individually satisfiable, $F_1 \land F_2$ does not have to be satisfiable

Example:

$$x + y = 2 \land x = 1 \land f(x) \neq f(y)$$

Here, $F_1$ and $F_2$ are individually sat, but their combination is unsat b/c $T_Z$ implies $x = y$
Nelson-Oppen Method for Convex Theories

- If both are SAT, does this mean $F$ is sat?

- No because if $F_1$ and $F_2$ are individually satisfiable, $F_1 \land F_2$ does not have to be satisfiable

- Example: $x + y = 2 \land x = 1 \land f(x) \neq f(y)$

- Here, $F_1$ and $F_2$ are individually sat, but their combination is unsat b/c $T_Z$ implies $x = y$

- In the case where $F_1$ and $F_2$ are sat, theories have to exchange all implied equalities
Nelson-Oppen Method for Convex Theories

- If both are SAT, does this mean $F$ is sat?

- No because if $F_1$ and $F_2$ are individually satisfiable, $F_1 \land F_2$ does not have to be satisfiable

- Example: $x + y = 2 \land x = 1 \land f(x) \neq f(y)$

- Here, $F_1$ and $F_2$ are individually sat, but their combination is unsat b/c $T_Z$ implies $x = y$

- In the case where $F_1$ and $F_2$ are sat, theories have to exchange all implied equalities

- Why only equalities?
Nelson-Oppen Method for Convex Theories

- If both are SAT, does this mean $F$ is sat?

- No because if $F_1$ and $F_2$ are individually satisfiable, $F_1 \land F_2$ does not have to be satisfiable

- Example: $x + y = 2 \land x = 1 \land f(x) \neq f(y)$

- Here, $F_1$ and $F_2$ are individually sat, but their combination is unsat b/c $T_Z$ implies $x = y$

- In the case where $F_1$ and $F_2$ are sat, theories have to exchange all implied equalities

- Why only equalities? b/c it is the only shared symbol
Nelson-Oppen Method for Convex Theories

- For each pair of shared variables $x, y$, determine if:

  1. $F_1 \Rightarrow x = y$
  2. $F_2 \Rightarrow x = y$

  - If (1) holds but not (2), conjoin $x = y$ with $F_2$
  - If (2) holds but not (1), conjoin $x = y$ with $F_1$

  - Let $F'_1$ and $F'_2$ denote new formulas
  - Check satisfiability of $F'_1$ and $F'_2$
  - Repeat until either formula becomes unsat or no new equalities can be inferred
Nelson-Oppen Method for Convex Theories

- For each pair of shared variables $x, y$, determine if:
  1. $F_1 \Rightarrow x = y$
Nelson-Oppen Method for Convex Theories

- For each pair of shared variables \( x, y \), determine if:
  1. \( F_1 \Rightarrow x = y \)
  2. \( F_2 \Rightarrow x = y \)

- If (1) holds but not (2), conjoin \( x = y \) with \( F_2 \)
- If (2) holds but not (1), conjoin \( x = y \) with \( F_1 \)
- Let \( F'_1 \) and \( F'_2 \) denote new formulas
- Check satisfiability of \( F'_1 \) and \( F'_2 \)
- Repeat until either formula becomes unsat or no new equalities can be inferred
For each pair of shared variables $x, y$, determine if:

1. $F_1 \Rightarrow x = y$
2. $F_2 \Rightarrow x = y$

If (1) holds but not (2), conjoin $x = y$ with $F_2$
Nelson-Oppen Method for Convex Theories

- For each pair of shared variables $x, y$, determine if:
  
  1. $F_1 \Rightarrow x = y$
  2. $F_2 \Rightarrow x = y$

- If (1) holds but not (2), conjoin $x = y$ with $F_2$

- If (2) holds but not (1), conjoin $x = y$ with $F_1$
Nelson-Oppen Method for Convex Theories

- For each pair of shared variables $x, y$, determine if:
  
  1. $F_1 \Rightarrow x = y$
  
  2. $F_2 \Rightarrow x = y$

- If (1) holds but not (2), conjoin $x = y$ with $F_2$

- If (2) holds but not (1), conjoin $x = y$ with $F_1$

- Let $F'_1$ and $F'_2$ denote new formulas
Nelson-Oppen Method for Convex Theories

- For each pair of shared variables $x, y$, determine if:
  1. $F_1 \Rightarrow x = y$
  2. $F_2 \Rightarrow x = y$

- If (1) holds but not (2), conjoin $x = y$ with $F_2$
- If (2) holds but not (1), conjoin $x = y$ with $F_1$
- Let $F_1'$ and $F_2'$ denote new formulas
- Check satisfiability of $F_1'$ and $F_2'$
Nelson-Oppen Method for Convex Theories

- For each pair of shared variables \( x, y \), determine if:
  1. \( F_1 \Rightarrow x = y \)
  2. \( F_2 \Rightarrow x = y \)

- If (1) holds but not (2), conjoin \( x = y \) with \( F_2 \)

- If (2) holds but not (1), conjoin \( x = y \) with \( F_1 \)

- Let \( F'_1 \) and \( F'_2 \) denote new formulas

- Check satisfiability of \( F'_1 \) and \( F'_2 \)

- Repeat until either formula becomes unsat or no new equalities can be inferred
Example

- Use Nelson-Oppen to decide sat of following $T_{\leq} \cup T_{\mathbb{Q}}$ formula:

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

- Use Nelson-Oppen to decide sat of following $T_\leq \cup T_\mathbb{Q}$ formula:

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

- First, we need to purify:
Example

- Use Nelson-Oppen to decide sat of following $\mathcal{T}_= \cup \mathcal{T}_\mathbb{Q}$ formula:

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

- First, we need to purify:
  - Replace $f(x)$ with new variable $w_1$
Use Nelson-Oppen to decide sat of following $T_\leq \cup T_Q$ formula:

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

First, we need to purify:

- Replace $f(x)$ with new variable $w_1$
- Replace $f(y)$ with new variable $w_2$
Example

- Use Nelson-Oppen to decide sat of following $T_\leq \cup T_\mathbb{Q}$ formula:
  
  $$f(f(x) - f(y)) \neq f(z) \land x \leq y \land y + z \leq x \land 0 \leq z$$

- First, we need to purify:
  
  - Replace $f(x)$ with new variable $w_1$
  - Replace $f(y)$ with new variable $w_2$
  - $f(x) - f(y)$ is now replaced with $w_1 - w_2$ and we conjoin
    
    $$w_1 = f(x) \land w_2 = f(y)$$
Example

- Use Nelson-Oppen to decide sat of following $T_\leq \cup T_Q$ formula:

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

- First, we need to purify:
  - Replace $f(x)$ with new variable $w_1$
  - Replace $f(y)$ with new variable $w_2$
  - $f(x) - f(y)$ is now replaced with $w_1 - w_2$ and we conjoin
    $$w_1 = f(x) \land w_2 = f(y)$$
  - First literal is now $f(w_1 - w_2) \neq f(z)$; still not pure!
Example

- Use Nelson-Oppen to decide sat of following \( T_\leq \cup T_\mathbb{Q} \) formula:

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

- First, we need to purify:
  - Replace \( f(x) \) with new variable \( w_1 \)
  - Replace \( f(y) \) with new variable \( w_2 \)
  - \( f(x) - f(y) \) is now replaced with \( w_1 - w_2 \) and we conjoin

\[
w_1 = f(x) \land w_2 = f(y)
\]

- First literal is now \( f(w_1 - w_2) \neq f(z) \); still not pure!

- Replace \( w_1 - w_2 \) with \( w_3 \) and add equality \( w_3 = w_1 - w_2 \)
Example, cont

- Purified formula is $F_1 \land F_2$ where:

  $F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$

  $F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$
Example, cont

- Purified formula is $F_1 \land F_2$ where:

  $F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$
  $F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$

- Which variables are shared?
Example, cont

- Purified formula is $F_1 \land F_2$ where:

  $F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$
  
  $F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$

- Which variables are shared? all
Example, cont

- Purified formula is $F_1 \land F_2$ where:

  $F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$
  $F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$

- Which variables are shared? all

- Check sat of $F_1$. Is it SAT?
Example, cont

- Purified formula is $F_1 \land F_2$ where:
  
  $F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$
  
  $F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$

- Which variables are shared? all

- Check sat of $F_1$. Is it SAT? yes
Example, cont

- Purified formula is $F_1 \land F_2$ where:

  $F_1: \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$

  $F_2: \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$

- Which variables are shared? all

- Check sat of $F_1$. Is it SAT? yes

- Check sat of $F_2$. Is it SAT?
Example, cont

- Purified formula is $F_1 \land F_2$ where:

  $F_1:\ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$
  $F_2:\ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$

- Which variables are shared? all

- Check sat of $F_1$. Is it SAT? yes

- Check sat of $F_2$. Is it SAT? yes
Example, cont

- Purified formula is $F_1 \land F_2$ where:

  $F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z)$
  $F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z$

- Which variables are shared? all

- Check sat of $F_1$. Is it SAT? yes

- Check sat of $F_2$. Is it SAT? yes

- Now, for each pair of shared variable $x_i, x_j$, we query whether $F_1$ or $F_2$
imply $x_i = x_j$
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

▶ Consider the query \( x = y \) – is it implied by either \( F_1 \) or \( F_2 \)?
Example, cont

\[ F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \]
\[ F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

Consider the query \( x = y \) – is it implied by either \( F_1 \) or \( F_2 \)? implied by \( F_2 \)
Consider the query $x = y$ – is it implied by either $F_1$ or $F_2$? **implied by $F_2$**

- $y + z \leq x \land 0 \leq z$ imply $0 \leq z \leq x - y$, i.e., $y \leq x$
Consider the query \( x = y \) – is it implied by either \( F_1 \) or \( F_2 \)? implied by \( F_2 \)

\[ y + z \leq x \land 0 \leq z \implies 0 \leq z \leq x - y, \text{ i.e., } y \leq x \]

\[ \text{Since we also have } x \leq y, T_Q \text{ implies } x = y \]
Example, cont

\[ F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \]
\[ F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

- Consider the query \( x = y \) – is it implied by either \( F_1 \) or \( F_2 \)? \( \text{implied by } F_2 \)

- \( y + z \leq x \land 0 \leq z \) imply \( 0 \leq z \leq x - y \), i.e., \( y \leq x \)

- Since we also have \( x \leq y \), \( T_Q \) implies \( x = y \)

- Now, propagate this to \( T_\equiv \), so \( F'_1 \) becomes:

\[ F'_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
Example, cont

\[
\begin{align*}
F_1 : & \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \\
F_2 : & \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z
\end{align*}
\]

- Consider the query \( x = y \) – is it implied by either \( F_1 \) or \( F_2 \)? implied by \( F_2 \)

- \( y + z \leq x \land 0 \leq z \) imply \( 0 \leq z \leq x - y \), i.e., \( y \leq x \)

- Since we also have \( x \leq y \), \( T_Q \) implies \( x = y \)

- Now, propagate this to \( T_- \), so \( F_1' \) becomes:

\[
F_1' : w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y
\]

- Check sat of \( F_1' \). Is it SAT?
Example, cont

\[ F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \]
\[ F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

- Consider the query \( x = y \) — is it implied by either \( F_1 \) or \( F_2 \)? implied by \( F_2 \)

- \( y + z \leq x \land 0 \leq z \) imply \( 0 \leq z \leq x - y \), i.e., \( y \leq x \)

- Since we also have \( x \leq y \), \( T_Q \) implies \( x = y \)

- Now, propagate this to \( T_\leq \), so \( F'_1 \) becomes:

\[ F'_1 : w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]

- Check sat of \( F'_1 \). Is it SAT? yes
Example, cont

\begin{align*}
F_1 : & \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \\
F_2 : & \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z
\end{align*}

- Consider the query \( x = y \) – is it implied by either \( F_1 \) or \( F_2 \)? implied by \( F_2 \)
- \( y + z \leq x \land 0 \leq z \) imply \( 0 \leq z \leq x - y \), i.e., \( y \leq x \)
- Since we also have \( x \leq y \), \( T_Q \) implies \( x = y \)
- Now, propagate this to \( T_{=} \), so \( F'_1 \) becomes:

\begin{align*}
F'_1 : & \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y
\end{align*}

- Check sat of \( F'_1 \). Is it SAT? yes
- Are we done?
Consider the query $x = y$ – is it implied by either $F_1$ or $F_2$? \textit{implied by $F_2$}

$y + z \leq x \land 0 \leq z$ imply $0 \leq z \leq x - y$, i.e., $y \leq x$

Since we also have $x \leq y$, $T_Q$ implies $x = y$

Now, propagate this to $T_\leq$, so $F_1'$ becomes:

$F_1': w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y$

Check sat of $F_1'$. Is it SAT? \textit{yes}

Are we done? \textit{no}
Example, cont

\( F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \)

\( F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \)

- Since \( F_1 \) changed, need to check if it implies any new equality
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

- Since \( F_1 \) changed, need to check if it implies any new equality

- Does it imply a new equality?
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

- Since \( F_1 \) changed, need to check if it implies any new equality

- Does it imply a new equality? yes, \( w_1 = w_2 \)
Example, cont

\[ F_1 : \ w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

- Since \( F_1 \) changed, need to check if it implies any new equality

- Does it imply a new equality? yes, \( w_1 = w_2 \)

- Now, we add \( w_1 = w_2 \) to \( F_2 \):

\[ F_2 : \ w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

- Since \( F_1 \) changed, need to check if it implies any new equality

- Does it imply a new equality? yes, \( w_1 = w_2 \)

- Now, we add \( w_1 = w_2 \) to \( F_2 \):

\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]

- We recheck sat of \( F_2 \). Is it SAT?
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

- Since \( F_1 \) changed, need to check if it implies any new equality

- Does it imply a new equality? \text{yes, } w_1 = w_2

- Now, we add \( w_1 = w_2 \) to \( F_2 \):

\[ F_2 : w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]

- We recheck sat of \( F_2 \). Is it SAT? \text{yes}
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \]

▶ Since \( F_1 \) changed, need to check if it implies any new equality

▶ Does it imply a new equality? yes, \( w_1 = w_2 \)

▶ Now, we add \( w_1 = w_2 \) to \( F_2 \):

\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]

▶ We recheck sat of \( F_2 \). Is it SAT? yes

▶ Still not done b/c need to check if \( F_2 \) implies any new equalities
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]

▶ Consider the query \( w_3 = z \)?
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]

- Consider the query \( w_3 = z \)?

- \( w_3 = w_1 - w_2 \) and \( w_1 = w_2 \) imply \( w_3 = 0 \)
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]

- Consider the query \( w_3 = z? \)

- \( w_3 = w_1 - w_2 \) and \( w_1 = w_2 \) imply \( w_3 = 0 \)

- Since \( x = y \), \( y + z \leq x \) implies \( z \leq 0 \)
Example, cont

\[ F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \]
\[ F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2 \]

- Consider the query \( w_3 = z \)?

- \( w_3 = w_1 - w_2 \) and \( w_1 = w_2 \) imply \( w_3 = 0 \)

- Since \( x = y \), \( y + z \leq x \) implies \( z \leq 0 \)

- Since \( z \leq 0 \) and \( 0 \leq z \), we have \( z = 0 \)
Example, cont

\[
F_1 : \quad w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \\
F_2 : \quad w_3 = w_1 - w_2 \land x \leq y \land y + z \leq x \land 0 \leq z \land w_1 = w_2
\]

- Consider the query \( w_3 = z \)?
  - \( w_3 = w_1 - w_2 \) and \( w_1 = w_2 \) imply \( w_3 = 0 \)
  - Since \( x = y \), \( y + z \leq x \) implies \( z \leq 0 \)
  - Since \( z \leq 0 \) and \( 0 \leq z \), we have \( z = 0 \)
  - Thus, \( T_Q \) answer "yes" for query \( w_3 = z \)
Example, cont

- Now, propagate $w_3 = z$ to $F_1$:

$$F_1 : w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \land w_3 = z$$

Is this sat?

No, because $w_3 = z$ implies $f(w_3) = f(z)$

This contradicts $f(w_3) \neq f(z)$

Thus, original formula is UNSAT
Example, cont

- Now, propagate $w_3 = z$ to $F_1$:

  \[
  F_1 : w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \land w_3 = z
  \]

- Is this sat?
Example, cont

- Now, propagate $w_3 = z$ to $F_1$:

$$F_1 : w_1 = f(x) \land w_2 = f(y) \land f(w_3) \not= f(z) \land x = y \land w_3 = z$$

- Is this sat?

- No, because $w_3 = z$ implies $f(w_3) = f(z)$
Example, cont

- Now, propagate $w_3 = z$ to $F_1$:

  \[ F_1 : w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \land w_3 = z \]

- Is this sat?

- No, because $w_3 = z$ implies $f(w_3) = f(z)$

- This contradicts $f(w_3) \neq f(z)$
Example, cont

- Now, propagate $w_3 = z$ to $F_1$:

  \[ F_1 : w_1 = f(x) \land w_2 = f(y) \land f(w_3) \neq f(z) \land x = y \land w_3 = z \]

- Is this sat?

- No, because $w_3 = z$ implies $f(w_3) = f(z)$

- This contradicts $f(w_3) \neq f(z)$

- Thus, original formula is UNSAT
Non-Convex Theories

- Unfortunately, technique discussed so far does not work for non-convex theories
Non-Convex Theories

- Unfortunately, technique discussed so far does not work for non-convex theories

- Consider the following $T\mathbb{Z} \cup T_=$ formula:

$$1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$$
Non-Convex Theories

- Unfortunately, technique discussed so far does not work for non-convex theories

- Consider the following $T_{\mathbb{Z}} \cup T_{=}$ formula:

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

- Is this formula SAT?
Non-Convex Theories

- Unfortunately, technique discussed so far does not work for non-convex theories

- Consider the following $T_Z \cup T_E$ formula:

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

- Is this formula SAT? no
Non-Convex Theories

- Unfortunately, technique discussed so far does not work for non-convex theories

- Consider the following $T_\mathbb{Z} \cup T_\mathbb{R}$ formula:

  $$1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$$

- Is this formula SAT? no

- Let’s see what happens if we use technique described so far
Non-Convex Theories

- Unfortunately, technique discussed so far does not work for non-convex theories

- Consider the following $T_Z \cup T_\leq$ formula:

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

- Is this formula SAT? no

- Let’s see what happens if we use technique described so far

- If we purify, we get the following formulas:

  \[
  F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \\
  F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2
  \]
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is \( F_1 \) SAT?

- Is \( F_2 \) SAT?

- Does \( F_1 \) imply a new equality by itself?

- Does \( F_2 \) imply a new equality by itself?

Thus technique discussed so far returns SAT, although formula in unsat
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

▶ Is \( F_1 \) SAT? yes
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is \( F_1 \) SAT? yes

- Is \( F_2 \) SAT?
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is \( F_1 \) SAT? yes
- Is \( F_2 \) SAT? yes
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is \( F_1 \) SAT? yes
- Is \( F_2 \) SAT? yes
- Does \( F_1 \) imply a new equality by itself? no
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is \( F_1 \) SAT? yes
- Is \( F_2 \) SAT? yes
- Does \( F_1 \) imply a new equality by itself? no
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is \( F_1 \) SAT? yes
- Is \( F_2 \) SAT? yes
- Does \( F_1 \) imply a new equality by itself? no
- Does \( F_2 \) imply a new equality by itself?
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is \( F_1 \) SAT? yes
- Is \( F_2 \) SAT? yes
- Does \( F_1 \) imply a new equality by itself? no
- Does \( F_2 \) imply a new equality by itself? no
Example, cont

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is $F_1$ SAT? yes
- Is $F_2$ SAT? yes
- Does $F_1$ imply a new equality by itself? no
- Does $F_2$ imply a new equality by itself? no
- Thus technique discussed so far returns sat, although formula in unsat
Problem is that in non-convex theories, a formula might imply a disjunction of equalities
Problem is that in non-convex theories, a formula might imply a disjunction of equalities

But it doesn’t have to imply any single equality on its own
Nelson-Oppen with Non-Convex Theories

- Problem is that in non-convex theories, a formula might imply a disjunction of equalities

- But it doesn’t have to imply any single equality on its own

- Thus, it is not enough to query individual equality relations between variables
Problem is that in non-convex theories, a formula might imply a disjunction of equalities

But it doesn’t have to imply any single equality on its own

Thus, it is not enough to query individual equality relations between variables

We also have to query and propagate disjunctions of equalities
Problem is that in non-convex theories, a formula might imply a disjunction of equalities

But it doesn’t have to imply any single equality on its own

Thus, it is not enough to query individual equality relations between variables

We also have to query and propagate disjunctions of equalities

Two questions:
Nelson-Oppen with Non-Convex Theories

- Problem is that in non-convex theories, a formula might imply a disjunction of equalities

- But it doesn’t have to imply any single equality on its own

- Thus, it is not enough to query individual equality relations between variables

- We also have to query and propagate disjunctions of equalities

- Two questions:
  1. Which disjunctions do we query?
Nelson-Oppen with Non-Convex Theories

- Problem is that in non-convex theories, a formula might imply a disjunction of equalities

- But it doesn’t have to imply any single equality on its own

- Thus, it is not enough to query individual equality relations between variables

- We also have to query and propagate disjunctions of equalities

- Two questions:
  1. Which disjunctions do we query?
  2. How do we propagate disjunctions since we are considering disjunction-free formulas?
What Disjunctions to Query?

- **Recall:** We only have a finite set of shared variables
What Disjunctions to Query?

- Recall: We only have a finite set of shared variables

- From these, we can only generate a finite number of disjunctions of equalities
What Disjunctions to Query?

- **Recall**: We only have a finite set of shared variables

- From these, we can only generate a finite number of disjunctions of equalities

- Thus, for each possible disjunction, we need to issue a query
What Disjunctions to Query?

- **Recall:** We only have a finite set of shared variables

- From these, we can only generate a finite number of disjunctions of equalities

- Thus, for each possible disjunction, we need to issue a query

- **Example:** If we have shared variables $x, y, z$, which queries do we need to issue?
What Disjunctions to Query?

- **Recall**: We only have a finite set of shared variables

- From these, we can only generate a finite number of disjunctions of equalities

- Thus, for each possible disjunction, we need to issue a query

- **Example**: If we have shared variables $x, y, z$, which queries do we need to issue?

  
  \[
  \begin{align*}
  x &= y \\
  x &= z \\
  y &= z \\
  x &= y \lor x = z
  \end{align*}
  \]
Propagating Disjunctions

- Suppose answer to some disjunctive query \( \bigvee_{i=1}^{n} x_i = y_i \) is yes
Propagating Disjunctions

- Suppose answer to some disjunctive query $\bigvee_{i=1}^{n} x_i = y_i$ is yes

- In this case, we need to branch and consider all $n$ possibilities
Propagating Disjunctions

- Suppose answer to some disjunctive query $\bigvee_{i=1}^{n} x_i = y_i$ is yes

- In this case, we need to branch and consider all $n$ possibilities

- Thus, create $n$ subproblems where we propagate $x_i = y_i$ in $i$’th subproblem
Propagating Disjunctions

- Suppose answer to some disjunctive query $\bigvee_{i=1}^{n} x_i = y_i$ is yes

- In this case, we need to branch and consider all $n$ possibilities

- Thus, create $n$ subproblems where we propagate $x_i = y_i$ in $i$’th subproblem

- If there is any subproblem that is satisfiable, original formula is satisfiable
Propagating Disjunctions

- Suppose answer to some disjunctive query $\bigvee_{i=1}^{n} x_i = y_i$ is yes

- In this case, we need to branch and consider all $n$ possibilities

- Thus, create $n$ subproblems where we propagate $x_i = y_i$ in $i$’th subproblem

- If there is any subproblem that is satisfiable, original formula is satisfiable

- If every subproblem is unsatisfiable, then original formula is unsatisfiable
Consider $\mathcal{T}_\leq \cup \mathcal{T}_\mathbb{Z}$ formula:

$$1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$$
Example

- Consider $T_{=} \cup T_{\mathbb{Z}}$ formula:

$$1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$$

- After purification, we get:

$$F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2)$$
$$F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2$$
Example

- Consider $T_{\subseteq} \cup T_{\mathbb{Z}}$ formula:

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

- After purification, we get:

\[
F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \\
F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2
\]

- Which queries do we need to issue?
Example

- Consider $T_{\leq} \cup T_{\geq}$ formula:

$$1 \leq x \land x \leq 2 \land f(x) \neq f(1) \land f(x) \neq f(2)$$

- After purification, we get:

$$F_1 : \ f(x) \neq f(w_1) \land f(x) \neq f(w_2)$$

$$F_2 : \ 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2$$

- Which queries do we need to issue?

(1) $x = w_1$

(2) $x = w_2$

(3) $x = w_1 \lor x = w_2$
Example

- Consider $T_{\leq} \cup T_{\mathbb{Z}}$ formula:

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

- After purification, we get:

  \[ F_1 : f(x) \neq f(w_1) \land f(x) \neq f(w_2) \]
  \[ F_2 : 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Which queries do we need to issue?

  \( (1) \ x = w_1 \)
  \( (2) \ x = w_2 \)
  \( (3) \ x = w_1 \lor x = w_2 \)

- Answer to queries (1) and (2) are no, but $F_2$ implies query (3)
Example, cont

Now, we create two subproblems, one where we propagate $x = w_1$ and $x = w_2$. Is this satisfiable? No because $x = w_1$ implies $f(x) = f(w_1)$. 
Example, cont

Now, we create two subproblems, one where we propagate $x = w_1$ and $x = w_2$.

First subproblem:

$F_1 : f(x) \neq f(w_1) \land f(x) \neq f(w_2) \land x = w_1$

$F_2 : 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2$

Is this satisfiable?

No because $x = w_1$ implies $f(x) = f(w_1)$. 
Now, we create two subproblems, one where we propagate $x = w_1$ and $x = w_2$.

First subproblem:

- $F_1$:\quad $f(x) \neq f(w_1) \land f(x) \neq f(w_2) \land x = w_1$
- $F_2$:\quad $1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2$

Is this satisfiable?
Example, cont

Now, we create two subproblems, one where we propagate $x = w_1$ and $x = w_2$

First subproblem:

\[ F_1 : f(x) \neq f(w_1) \land f(x) \neq f(w_2) \land x = w_1 \]
\[ F_2 : 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

Is this satisfiable?

No because $x = w_1$ implies $f(x) = f(w_1)$
Example, cont

- Second subproblem:

  \[ F_1 : \quad f(x) \neq f(w_1) \wedge f(x) \neq f(w_2) \wedge x = w_2 \]

  \[ F_2 : \quad 1 \leq x \wedge x \leq 2 \wedge w_1 = 1 \wedge w_2 = 2 \]
Example, cont

- Second subproblem:

  \[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \land x = w_2 \]

  \[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is this satisfiable?
Example, cont

- Second subproblem:

\[
F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \land x = w_2 \\
F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2
\]

- Is this satisfiable?

- **No** because \(x = w_2\) implies \(f(x) = f(w_2)\)
Example, cont

- Second subproblem:

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_2) \land x = w_2 \]
\[ F_2 : \quad 1 \leq x \land x \leq 2 \land w_1 = 1 \land w_2 = 2 \]

- Is this satisfiable?

- No because \( x = w_2 \) implies \( f(x) = f(w_2) \)

- Since neither subproblem is satisfiable, Nelson-Oppen returns \textbf{unsat} for original formula
Example II

Consider the following $T_\approx \cup T_\mathbb{Z}$ formula:

$$1 \leq x \land x \leq 3 \land f(x) \neq f(1) \land f(x) \neq f(3) \land f(1) \neq f(2)$$
Consider the following $T_\leq \cup T_\mathbb{Z}$ formula:

$$1 \leq x \land x \leq 3 \land f(x) \neq f(1) \land f(x) \neq f(3) \land f(1) \neq f(2)$$

Formulas after purification:

$$F_1 : f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2)$$

$$F_2 : 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3$$
Example II

- Consider the following $T_\leq \cup T_\mathbb{Z}$ formula:

$$1 \leq x \wedge x \leq 3 \wedge f(x) \neq f(1) \wedge f(x) \neq f(3) \wedge f(1) \neq f(2)$$

- Formulas after purification:

$$F_1 : f(x) \neq f(w_1) \wedge f(x) \neq f(w_3) \wedge f(w_1) \neq f(w_2)$$

$$F_2 : 1 \leq x \wedge x \leq 3 \wedge w_1 = 1 \wedge w_2 = 2 \wedge w_3 = 3$$

- Consider the query $x = w_1 \lor x = w_2 \lor x = w_3$
Example II

Consider the following $T_\leq \cup T\mathbb{Z}$ formula:

$$1 \leq x \land x \leq 3 \land f(x) \neq f(1) \land f(x) \neq f(3) \land f(1) \neq f(2)$$

Formulas after purification:

$F_1: \ f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2)$

$F_2: \ 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3$

Consider the query $x = w_1 \lor x = w_2 \lor x = w_3$

Does either formula imply this query?
Example II

Consider the following $T_\leq \cup T_\mathbb{Z}$ formula:

$$1 \leq x \land x \leq 3 \land f(x) \neq f(1) \land f(x) \neq f(3) \land f(1) \neq f(2)$$

Formulas after purification:

$$F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2)$$
$$F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3$$

Consider the query $x = w_1 \lor x = w_2 \lor x = w_3$

Does either formula imply this query? Yes
Example II, cont

▶ First subproblem:

\[
F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_1
\]
\[
F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3
\]
Example II, cont

First subproblem:

$F_1 : f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_1$
$F_2 : 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3$

Is this satisfiable?
Example II, cont

- First subproblem:

  \[ F_1 : f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_1 \]
  \[ F_2 : 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- Is this satisfiable? no
Example II, cont

- First subproblem:

  \( F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_1 \)

  \( F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \)

- Is this satisfiable? **no**

- Second subproblem:

  \( F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \)

  \( F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \)
Example II, cont

- First subproblem:

\[
F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_1
\]
\[
F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3
\]

- Is this satisfiable? no

- Second subproblem:

\[
F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2
\]
\[
F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3
\]

- Is this satisfiable?
Example II, cont

- First subproblem:

  \[ F_1 : \ f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_1 \]
  \[ F_2 : \ 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- Is this satisfiable? no

- Second subproblem:

  \[ F_1 : \ f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]
  \[ F_2 : \ 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- Is this satisfiable? Yes
Example II, cont

Second subproblem:

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]
\[ F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- So it’s satisfiable, are we done?
Example II, cont

Second subproblem:

\[
F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \\
F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3
\]

- So it’s satisfiable, are we done? No, need to check for new equalities
Example II, cont

Second subproblem:

\[ F_1 : \ f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]
\[ F_2 : \ 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

▶ So it’s satisfiable, are we done? No, need to check for new equalities

▶ Thus, we now issue new queries such as \( x = w_1, x = w_2 \), etc
Example II, cont

Second subproblem:

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]
\[ F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- So it’s satisfiable, are we done? No, need to check for new equalities
- Thus, we now issue new queries such as \( x = w_1, x = w_2 \), etc
- Are there any new implied equalities or disjunctions of equalities?
Example II, cont

Second subproblem:

\[ F_1 : f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]
\[ F_2 : 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- So it’s satisfiable, are we done? No, need to check for new equalities

- Thus, we now issue new queries such as \( x = w_1, x = w_2 \), etc

- Are there any new implied equalities or disjunctions of equalities? No
Example II, cont

Second subproblem:

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]
\[ F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- So it’s satisfiable, are we done? **No, need to check for new equalities**

- Thus, we now issue new queries such as \( x = w_1, x = w_2, \) etc

- Are there any new implied equalities or disjunctions of equalities? **No**

- Thus, second subproblem is satisfiable
Second subproblem:

\[
F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2
\]

\[
F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3
\]

▶ So it’s satisfiable, are we done? No, need to check for new equalities

▶ Thus, we now issue new queries such as \( x = w_1, x = w_2, \) etc

▶ Are there any new implied equalities or disjunctions of equalities? No

▶ Thus, second subproblem is satisfiable

▶ Do we need to check third subproblem?
Example II, cont

Second subproblem:

\[ F_1 : \ f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]

\[ F_2 : \ 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

- So it’s satisfiable, are we done? No, need to check for new equalities
- Thus, we now issue new queries such as \( x = w_1, x = w_2 \), etc
- Are there any new implied equalities or disjunctions of equalities? No
- Thus, second subproblem is satisfiable
- Do we need to check third subproblem? No
Example II, cont

Second subproblem:

\[ F_1 : \quad f(x) \neq f(w_1) \land f(x) \neq f(w_3) \land f(w_1) \neq f(w_2) \land x = w_2 \]

\[ F_2 : \quad 1 \leq x \land x \leq 3 \land w_1 = 1 \land w_2 = 2 \land w_3 = 3 \]

▶ So it’s satisfiable, are we done? **No, need to check for new equalities**

▶ Thus, we now issue new queries such as \( x = w_1, x = w_2 \), etc

▶ Are there any new implied equalities or disjunctions of equalities? **No**

▶ Thus, second subproblem is satisfiable

▶ Do we need to check third subproblem? **No**

▶ Thus, original formula is **satisfiable**
Optimization

- In presentation so far, we issued some disjunctive queries

But really, we want to find a minimal query that is implied. Minimal query is one where dropping any disjunct causes query to no longer be implied.

Why do we want minimal query?

1. Since $x = y \lor y = z$ already implies $x = y \lor y = z \lor z = w$, no need to consider latter to decide satisfiability.

2. When we propagate the query, using minimal query creates fewer subproblems.
Optimization

- In presentation so far, we issued some disjunctive queries
- As soon as answer was yes to some query, we propagated it by performing case split

Why do we want minimal query?
1. Since $x = y \lor y = z$ already implies $x = y \lor y = z \lor z = w$, no need to consider latter to decide satisfiability
2. When we propagate the query, using minimal query creates fewer subproblems
Optimization

- In presentation so far, we issued some disjunctive queries.

- As soon as answer was **yes** to some query, we propagated it by performing case split.

- But really, we want to find a **minimal** query that is implied.
Optimization

- In presentation so far, we issued some disjunctive queries.

- As soon as answer was yes to some query, we propagated it by performing case split.

- But really, we want to find a minimal query that is implied.

- Minimal query is one where dropping any disjunct causes query to no longer be implied.
Optimization

- In presentation so far, we issued some disjunctive queries

- As soon as answer was yes to some query, we propagated it by performing case split

- But really, we want to find a minimal query that is implied.

- Minimal query is one where dropping any disjunct causes query to no longer be implied

- Why do we want minimal query?
Optimization

- In presentation so far, we issued some disjunctive queries

- As soon as answer was yes to some query, we propagated it by performing case split

- But really, we want to find a minimal query that is implied.

- Minimal query is one where dropping any disjunct causes query to no longer be implied

- Why do we want minimal query?

  1. Since $x = y \vee y = z$ already implies $x = y \vee y = z \vee z = w$, no need to consider latter to decide satisfiability
Optimization

- In presentation so far, we issued some disjunctive queries
- As soon as answer was yes to some query, we propagated it by performing case split
- But really, we want to find a minimal query that is implied.
- Minimal query is one where dropping any disjunct causes query to no longer be implied
- Why do we want minimal query?
  1. Since \( x = y \lor y = z \) already implies \( x = y \lor y = z \lor z = w \), no need to consider latter to decide satisfiability
  2. When we propagate the query, using minimal query creates fewer subproblems
To find minimal query, start with disjunction of all possible equalities
Optimization, cont.

- To find minimal query, start with disjunction of all possible equalities
- If this isn’t implied, no subset will be implied, so we are done
Optimization, cont.

- To find minimal query, start with disjunction of all possible equalities
- If this isn’t implied, no subset will be implied, so we are done
- If it is implied, drop one equality
To find minimal query, start with disjunction of all possible equalities

- If this isn’t implied, no subset will be implied, so we are done
- If it is implied, drop one equality
- If it is still implied, continue with smaller disjunction
Optimization, cont.

- To find minimal query, start with disjunction of all possible equalities
- If this isn't implied, no subset will be implied, so we are done
- If it is implied, drop one equality
- If it is still implied, continue with smaller disjunction
- Otherwise, restore equality and continue with next one
Optimization, cont.

- To find minimal query, start with disjunction of all possible equalities
- If this isn’t implied, no subset will be implied, so we are done
- If it is implied, drop one equality
- If it is still implied, continue with smaller disjunction
- Otherwise, restore equality and continue with next one
- This ensures we find a minimal disjunction that is implied
To find minimal query, start with disjunction of all possible equalities

If this isn’t implied, no subset will be implied, so we are done

If it is implied, drop one equality

If it is still implied, continue with smaller disjunction

Otherwise, restore equality and continue with next one

This ensures we find a minimal disjunction that is implied

This strategy much better than using any disjunction that is implied
Nelson-Oppen for Convex vs. Non-Convex Theories

- Nelson-Oppen method is much more efficient for convex theories than for non-convex theories.
  
  In convex theories:
  1. need to issue one query for each pair of shared variables
  2. If decision procedures for $T_1$ and $T_2$ have polynomial time complexity, combination using Nelson-Oppen also has polynomial complexity.

In non-convex theories:
  1. need to consider disjunctions of equalities between each pair of shared variables
  2. If decision procedures for $T_1$ and $T_2$ have $NP$ time complexity, combination using Nelson-Oppen also has $NP$ time complexity.
Nelson-Oppen for Convex vs. Non-Convex Theories

- Nelson-Oppen method is much more efficient for convex theories than for non-convex theories

- In convex theories:
  1. need to issue one query for each pair of shared variables
Nelson-Oppen for Convex vs. Non-Convex Theories

- Nelson-Oppen method is much more efficient for convex theories than for non-convex theories

- In convex theories:
  1. need to issue one query for each pair of shared variables
  2. If decision procedures for $T_1$ and $T_2$ have polynomial time complexity, combination using Nelson-Oppen also has polynomial complexity
Nelson-Oppen for Convex vs. Non-Convex Theories

- Nelson-Oppen method is much more efficient for convex theories than for non-convex theories.

  - **In convex theories:**
    1. need to issue one query for each pair of shared variables
    2. If decision procedures for $T_1$ and $T_2$ have polynomial time complexity, combination using Nelson-Oppen also has polynomial complexity

  - **In non-convex theories:**
    1. need to consider disjunctions of equalities between each pair of shared variables
Nelson-Oppen for Convex vs. Non-Convex Theories

- Nelson-Oppen method is much more efficient for convex theories than for non-convex theories

- In convex theories:
  1. need to issue one query for each pair of shared variables
  2. If decision procedures for $T_1$ and $T_2$ have polynomial time complexity, combination using Nelson-Oppen also has polynomial complexity

- In non-convex theories:
  1. need to consider disjunctions of equalities between each pair of shared variables
  2. If decision procedures for $T_1$ and $T_2$ have $NP$ time complexity, combination using Nelson-Oppen also has $NP$ time complexity
Summary

- Nelson-Oppen method gives a sound and complete decision procedure for combination theories
Summary

- Nelson-Oppen method gives a **sound and complete** decision procedure for combination theories

- However, it only works for quantifier-free theories that are infinitely stable
Summary

- Nelson-Oppen method gives a sound and complete decision procedure for combination theories

- However, it only works for quantifier-free theories that are infinitely stable

- Not a severe restriction because most theories of interest are infinitely stable
Summary

- Nelson-Oppen method gives a sound and complete decision procedure for combination theories.
- However, it only works for quantifier-free theories that are infinitely stable.
- Not a severe restriction because most theories of interest are infinitely stable.
- Next lecture: How to decide satisfiability in first-order theories without converting to DNF.
Summary

- Nelson-Oppen method gives a **sound and complete** decision procedure for combination theories

- However, it only works for quantifier-free theories that are infinitely stable

- Not a severe restriction because most theories of interest are infinitely stable

- **Next lecture:** How to decide satisfiability in first-order theories without converting to DNF

- **Reminder:** homework due next lecture