SAT and SMT Solving

Sarah Winkler

SS 2018

Department of Computer Science
University of Innsbruck

Outline

- Summary of Last Week
- Correctness of DPLL(T)
- Congruence Closure
- Some More Practical SMT

Summary of Last Week

Definitions

for formulas F and G and list of literals M:

- theory T is set of first-order logic formulas without free variables
- F is T-consistent (or T-satisfiable) if $F \wedge T$ is satisfiable in first-order sense
- F is T-inconsistent (or T-unsatisfiable) if not T-consistent
- $M = l_1, \ldots, l_k$ is T-consistent if $l_1 \wedge \cdots \wedge l_k$ is
- M is T-model of F if $M \vDash F$ and M is T-consistent
- F entails G in T (denoted $F \vDash_T G$) if $F \wedge \neg G$ is T-inconsistent
- F and G are T-equivalent (denoted $F \equiv_T G$) if $F \vDash_T G$ and $G \vDash_T F$

Definition (Theory of Equality)

theory of equality (EQ) uses binary predicate $\approx$ and consists of axioms

$$\forall x\, (x \approx x) \qquad \forall x\, y\, (x \approx y \rightarrow y \approx x) \qquad \forall x\, y\, z\, (x \approx y \wedge y \approx z \rightarrow x \approx z)$$

Definition (Theory of Equality With Uninterpreted Functions)

EUF over set of function symbols $\mathcal{F}$ consists of equality axioms:

$$\forall x\, (x \approx x) \qquad \forall x\, y\, (x \approx y \rightarrow y \approx x) \qquad \forall x\, y\, z\, (x \approx y \wedge y \approx z \rightarrow x \approx z)$$

plus for all $f/n \in \mathcal{F}$ {the functional consistency axiom:

$$\forall x_1 y_1 \, \dots \, x_n y_n \, (x_1 \approx y_1 \wedge \cdots \wedge x_n \approx y_n \rightarrow f(x_1, \dots, x_n) \approx f(y_1, \dots, y_n))$$

Definition (DPLL(T) Transition Rules)

DPLL(T) consists of DPLL rules unit propagate, decide, fail, and restart plus

- T-backjump $\quad M\ l^d\ N \parallel F, C \quad \Longrightarrow \quad M\ l' \parallel F, C$

  if $M\ l^d\ N \models \neg C$ and $\exists$ clause $C' \vee l'$ such that
  - $F, C \models_T C' \vee l'$
  - $M \models \neg C'$ and l' is undefined in M, and l' or l'^c occurs in F or in $M\ l^d\ N$
- T-learn $\quad M \parallel F \quad \Longrightarrow \quad M \parallel F,\ C$

  if $F \models_T C$ and all atoms of C occur in M or F
- T-forget $\quad M \parallel F,\ C \quad \Longrightarrow \quad M \parallel F$

  if $F \models_T C$
- T-propagate $\quad M \parallel F \quad \Longrightarrow \quad M\ l \parallel F$

  if $M \models_T l$, literal l or l^c occurs in F, and l is undefined in M

Naive Lazy Approach in DPLL(T)

- whenever state $M \parallel F$ is final wrt unit propagate, decide, fail, T-backjump: check T-consistency of M with T-solver
- if M is T-consistent then satisfiability is proven
- otherwise $\exists l_1, \ldots, l_k$ subset of M such that $\vDash_T \neg(l_1 \wedge \cdots \wedge l_k)$
- use T-learn to add $\neg l_1 \vee \cdots \vee \neg l_k$
- apply restart

Improvement 1: Incremental T-Solver

- T-solver checks T-consistency of model M whenever literal is added to M

Improvement 2: On-Line SAT solver

- after T-learn added clause, apply fail or T-backjump instead of restart

Improvement 3: Eager Theory Propagation

- apply T-propagate before decide

Remark

all three improvements can be combined

Correctness of DPLL(T)

Definition (Basic DPLL(T))

system $\mathcal{B}$ consists of unit propagate, decide, fail, T-backjump, and T-propagate

Definition (Basic DPLL(T))

system $\mathcal{B}$ consists of unit propagate, decide, fail, T-backjump, and T-propagate

Definition (Full DPLL(T))

system $\mathcal{F}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Definition (Basic DPLL(T))

system $\mathcal{B}$ consists of unit propagate, decide, fail, T-backjump, and T-propagate

Definition (Full DPLL(T))

system $\mathcal{F}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Lemma

if $\parallel F \Longrightarrow_{\mathsf{F}}^{*} M \parallel G$ *then*

- *all atoms in M and G are atoms in F*

Definition (Basic DPLL(T))

system $\mathcal{B}$ consists of unit propagate, decide, fail, T-backjump, and T-propagate

Definition (Full DPLL(T))

system $\mathcal{F}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Lemma

if $\parallel F \Longrightarrow^*_{\mathsf{F}} M \parallel G$ *then*

- *all atoms in* M *and* G *are atoms in* F
- M *does not contain complementary literals, and every literal at most once*

Definition (Basic DPLL(T))

system $\mathcal{B}$ consists of unit propagate, decide, fail, T-backjump, and T-propagate

Definition (Full DPLL(T))

system $\mathcal{F}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Lemma

if $\| \ F \Longrightarrow_{\mathsf{F}}^{*} M \ \| \ G$ *then*

- *all atoms in* M *and* G *are atoms in* F
- M *does not contain complementary literals, and every literal at most once*
- G *is* T*-equivalent to* F ($F \equiv_T G$)

Definition (Basic DPLL(T))

system $\mathcal{B}$ consists of unit propagate, decide, fail, T-backjump, and T-propagate

Definition (Full DPLL(T))

system $\mathcal{F}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Lemma

if $\parallel F \Longrightarrow_{\mathsf{F}}^{*} M \parallel G$ *then*

- *all atoms in M and G are atoms in F*
- *M does not contain complementary literals, and every literal at most once*
- *G is T-equivalent to F ($F \equiv_T G$)*
- *if $M = M_0\ l_1^d\ M_1\ l_2^d\ M_2 \dots\ l_k^d\ M_k$ with $l_1, \dots, l_k$ all the decision literals then $F, l_1, \dots, l_i \vDash_T M_i$ for all $0 \leqslant i \leqslant k$*

Consider derivation with final state S_n:

$$\| F \implies_{\mathcal{F}} S_1 \implies_{\mathcal{F}} S_2 \implies_{\mathcal{F}} \dots \implies_{\mathcal{F}} S_n$$

Consider derivation with final state S_n:

$$\| F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \text{FailState}$ *then* F *is* T*-unsatisfiable*

Consider derivation with final state S_n:

$$\| \, F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \text{FailState}$ then F is T-unsatisfiable

Proof.

- must have $\| \, F \Longrightarrow^*_{\mathcal{F}} M \, \| \, F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{F}} \text{FailState}$
  such that M contains no decision literals and $M \models \neg C$ for some C in F'

Consider derivation with final state S_n:

$$\| F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \text{FailState}$ then F is T-unsatisfiable

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{F}} M \| F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{F}} \text{FailState}$
  such that M contains no decision literals and $M \models \neg C$ for some C in F'
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$

Consider derivation with final state S_n:

$$\| F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \mathsf{FailState}$ then F is T-unsatisfiable

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{F}} M \| F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{F}} \mathsf{FailState}$
  such that M contains no decision literals and $M \models \neg C$ for some C in F'
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$
- also have $F' \models_T C$ because C is in F', so $F \equiv_T F'$ is unsatisfiable ■

Consider derivation with final state S_n:

$$\| F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \text{FailState}$ *then* F *is* T*-unsatisfiable*

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{F}} M \| F' \stackrel{\text{fail}}{\Longrightarrow}_{\mathcal{F}} \text{FailState}$
  such that M contains no decision literals and $M \vDash \neg C$ for some C in F'
- by Lemma before, $F' \vDash_T M$, so $F' \vDash_T \neg C$
- also have $F' \vDash_T C$ because C is in F', so $F \equiv_T F'$ is unsatisfiable ∎

Theorem

if $S_n = M \| F'$ *and* M *is* T*-consistent then* F *is* T*-satisfiable and* $M \vDash_T F$

Consider derivation with final state S_n:

$$\| F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \text{FailState}$ then F is T-unsatisfiable

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{F}} M \| F' \stackrel{\text{fail}}{\Longrightarrow}_{\mathcal{F}} \text{FailState}$
  such that M contains no decision literals and $M \models \neg C$ for some C in F'
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$
- also have $F' \models_T C$ because C is in F', so $F \equiv_T F'$ is unsatisfiable ■

Theorem

if $S_n = M \| F'$ and M is T-consistent then F is T-satisfiable and $M \models_T F$

Proof.

- S_n is final, so all literals of F' are defined in M (otherwise decide applicable)

Consider derivation with final state S_n:

$$\| F \implies_{\mathcal{F}} S_1 \implies_{\mathcal{F}} S_2 \implies_{\mathcal{F}} \dots \implies_{\mathcal{F}} S_n$$

Theorem

if $S_n =$ FailState *then* F *is* T*-unsatisfiable*

Proof.

- must have $\| F \implies^*_{\mathcal{F}} M \| F' \stackrel{\text{fail}}{\implies}_{\mathcal{F}}$ FailState
  such that M contains no decision literals and $M \models \neg C$ for some C in F'
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$
- also have $F' \models_T C$ because C is in F', so $F \equiv_T F'$ is unsatisfiable ■

Theorem

if $S_n = M \| F'$ *and* M *is* T*-consistent then* F *is* T*-satisfiable and* $M \models_T F$

Proof.

- S_n is final, so all literals of F' are defined in M (otherwise decide applicable)
- $\nexists$ clause C in F' such that $M \models_T \neg C$ (otherwise backjump or fail applicable)

Consider derivation with final state S_n:

$$\| F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \text{FailState}$ *then* F *is* *T-unsatisfiable*

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{F}} M \| F' \stackrel{\text{fail}}{\Longrightarrow}_{\mathcal{F}} \text{FailState}$
  such that M contains no decision literals and $M \vDash \neg C$ for some C in F'
- by Lemma before, $F' \vDash_T M$, so $F' \vDash_T \neg C$
- also have $F' \vDash_T C$ because C is in F', so $F \equiv_T F'$ is unsatisfiable ∎

Theorem

if $S_n = M \| F'$ *and* M *is* T*-consistent then* F *is* T*-satisfiable and* $M \vDash_T F$

Proof.

- S_n is final, so all literals of F' are defined in M (otherwise decide applicable)
- $\nexists$ clause C in F' such that $M \vDash_T \neg C$ (otherwise backjump or fail applicable)
- by T-consistency $M \vDash_T F'$

Consider derivation with final state S_n:

$$\| F \quad \Longrightarrow_{\mathcal{F}} \quad S_1 \quad \Longrightarrow_{\mathcal{F}} \quad S_2 \quad \Longrightarrow_{\mathcal{F}} \quad \dots \quad \Longrightarrow_{\mathcal{F}} \quad S_n$$

Theorem

if $S_n = \text{FailState}$ then F is T-unsatisfiable

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{F}} M \| F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{F}} \text{FailState}$
  such that M contains no decision literals and $M \vDash \neg C$ for some C in F'
- by Lemma before, $F' \vDash_T M$, so $F' \vDash_T \neg C$
- also have $F' \vDash_T C$ because C is in F', so $F \equiv_T F'$ is unsatisfiable ∎

Theorem

if $S_n = M \| F'$ and M is T-consistent then F is T-satisfiable and $M \vDash_T F$

Proof.

- S_n is final, so all literals of F' are defined in M (otherwise decide applicable)
- $\nexists$ clause C in F' such that $M \vDash_T \neg C$ (otherwise backjump or fail applicable)
- by T-consistency $M \vDash_T F'$
- have $F \equiv_T F'$ so M does also T-satisfy F ∎

Theorem (Termination)

$$\Gamma : \quad \| \, F \Longrightarrow^*_{\mathcal{F}} S_0 \Longrightarrow^*_{\mathcal{F}} S_1 \Longrightarrow^*_{\mathcal{F}} \ldots$$

is finite if

Theorem (Termination)

$$\Gamma: \quad \| \, F \Longrightarrow^*_{\mathcal{F}} S_0 \Longrightarrow^*_{\mathcal{F}} S_1 \Longrightarrow^*_{\mathcal{F}} \ldots$$

is finite if

- *there is no infinite sub-derivation of only T-learn and T-forget steps, and*

Theorem (Termination)

$$\Gamma: \quad \| \, F \Longrightarrow_{\mathcal{F}}^{*} S_0 \Longrightarrow_{\mathcal{F}}^{*} S_1 \Longrightarrow_{\mathcal{F}}^{*} \ldots$$

is finite if

- *there is no infinite sub-derivation of only T-learn and T-forget steps, and*
- *for every sub-derivation*

$$S_i \overset{\textit{restart}}{\Longrightarrow}_{\mathcal{F}} S_{i+1} \Longrightarrow_{\mathcal{F}}^{*} S_j \overset{\textit{restart}}{\Longrightarrow}_{\mathcal{F}} S_{j+1} \Longrightarrow_{\mathcal{F}}^{*} S_k$$

with no restart steps in $S_{i+1} \Longrightarrow_{\mathcal{F}}^{*} S_j$ *and* $S_{j+1} \Longrightarrow_{\mathcal{F}}^{*} S_k$*:*

Theorem (Termination)

$$\Gamma: \quad \| \; F \Longrightarrow^*_{\mathcal{F}} S_0 \Longrightarrow^*_{\mathcal{F}} S_1 \Longrightarrow^*_{\mathcal{F}} \dots$$

is finite if

- *there is no infinite sub-derivation of only T-learn and T-forget steps, and*
- *for every sub-derivation*

$$S_i \overset{restart}{\Longrightarrow}_{\mathcal{F}} S_{i+1} \Longrightarrow^*_{\mathcal{F}} S_j \overset{restart}{\Longrightarrow}_{\mathcal{F}} S_{j+1} \Longrightarrow^*_{\mathcal{F}} S_k$$

  with no restart steps in $S_{i+1} \Longrightarrow^*_{\mathcal{F}} S_j$ *and* $S_{j+1} \Longrightarrow^*_{\mathcal{F}} S_k$*:*
  - *there are more* $\mathcal{B}$*-steps in* $S_j \Longrightarrow^*_{\mathcal{F}} S_k$ *than in* $S_i \Longrightarrow^*_{\mathcal{F}} S_j$*, or*

Theorem (Termination)

$$\Gamma: \quad \| \, F \Longrightarrow_{\mathcal{F}}^{*} S_0 \Longrightarrow_{\mathcal{F}}^{*} S_1 \Longrightarrow_{\mathcal{F}}^{*} \ldots$$

is finite if

- *there is no infinite sub-derivation of only T-learn and T-forget steps, and*
- *for every sub-derivation*

$$S_i \overset{restart}{\Longrightarrow}_{\mathcal{F}} S_{i+1} \Longrightarrow_{\mathcal{F}}^{*} S_j \overset{restart}{\Longrightarrow}_{\mathcal{F}} S_{j+1} \Longrightarrow_{\mathcal{F}}^{*} S_k$$

  with no restart steps in $S_{i+1} \Longrightarrow_{\mathcal{F}}^{*} S_j$ *and* $S_{j+1} \Longrightarrow_{\mathcal{F}}^{*} S_k$*:*
  - *there are more* $\mathcal{B}$*-steps in* $S_j \Longrightarrow_{\mathcal{F}}^{*} S_k$ *than in* $S_i \Longrightarrow_{\mathcal{F}}^{*} S_j$*, or*
  - *a clause is learned in* $S_j \Longrightarrow_{\mathcal{F}}^{*} S_k$ *that is never forgotten in* Γ

Theorem (Termination)

$$\Gamma: \quad \| \, F \Longrightarrow^*_{\mathcal{F}} S_0 \Longrightarrow^*_{\mathcal{F}} S_1 \Longrightarrow^*_{\mathcal{F}} \dots$$

is finite if

- *there is no infinite sub-derivation of only T-learn and T-forget steps, and*
- *for every sub-derivation*

$$S_i \overset{\text{restart}}{\Longrightarrow}_{\mathcal{F}} S_{i+1} \Longrightarrow^*_{\mathcal{F}} S_j \overset{\text{restart}}{\Longrightarrow}_{\mathcal{F}} S_{j+1} \Longrightarrow^*_{\mathcal{F}} S_k$$

  with no restart steps in $S_{i+1} \Longrightarrow^*_{\mathcal{F}} S_j$ *and* $S_{j+1} \Longrightarrow^*_{\mathcal{F}} S_k$*:*
  - *there are more* $\mathcal{B}$*-steps in* $S_j \Longrightarrow^*_{\mathcal{F}} S_k$ *than in* $S_i \Longrightarrow^*_{\mathcal{F}} S_j$*, or*
  - *a clause is learned in* $S_j \Longrightarrow^*_{\mathcal{F}} S_k$ *that is never forgotten in* Γ

Proof.

similar as for DPLL:

- restart is applied with increasing periodicity, or
- otherwise clause is learned (and there are only finitely many clauses) ∎

Congruence Closure

Aim

build T-solver for EUF using congruence closure

Aim

build T-solver for EUF using congruence closure

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ function symbols with fixed arity

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ function symbols with fixed arity
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ function symbols with fixed arity
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ function symbols with fixed arity
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $f \in \mathcal{F}$ has arity n and $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ — function symbols with fixed arity
- variables $\mathcal{V}$ — $\mathcal{F} \cap \mathcal{V} = \varnothing$
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ — smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $f \in \mathcal{F}$ has arity n and $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$

Example

- for $\mathcal{F} = \{\mathsf{f}/1, \mathsf{g}/2, \mathsf{a}/0\}$ and $x, y \in \mathcal{V}$ have terms a, $\mathsf{f}(x)$, $\mathsf{f}(\mathsf{a})$, $\mathsf{g}(x, \mathsf{f}(y))$, ...

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ function symbols with fixed arity
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $f \in \mathcal{F}$ has arity n and $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ then $f(t_1, \dots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- subterms

$$\mathcal{S}ub(t) = \begin{cases} \{t\} & \text{if } t \in \mathcal{V} \\ & \end{cases}$$

Example

- for $\mathcal{F} = \{\mathsf{f}/1, \mathsf{g}/2, \mathsf{a}/0\}$ and $x, y \in \mathcal{V}$ have terms a, $\mathsf{f}(x)$, $\mathsf{f}(\mathsf{a})$, $\mathsf{g}(x, \mathsf{f}(y))$, ...

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ function symbols with fixed arity
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $f \in \mathcal{F}$ has arity n and $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ then $f(t_1, \dots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- subterms

$$\mathcal{S}ub(t) = \begin{cases} \{t\} & \text{if } t \in \mathcal{V} \\ \{t\} \cup \bigcup_i \mathcal{S}ub(t_i) & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

Example

- for $\mathcal{F} = \{\mathsf{f}/1, \mathsf{g}/2, \mathsf{a}/0\}$ and $x, y \in \mathcal{V}$ have terms a, $\mathsf{f}(x)$, $\mathsf{f}(\mathsf{a})$, $\mathsf{g}(x, \mathsf{f}(y))$, ...

Aim

build T-solver for EUF using congruence closure

Definitions

- signature $\mathcal{F}$ function symbols with fixed arity
- variables $\mathcal{V}$ $\mathcal{F} \cap \mathcal{V} = \varnothing$
- terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ smallest set such that
  - $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$
  - if $f \in \mathcal{F}$ has arity n and $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$
- subterms

$$\mathcal{S}ub(t) = \begin{cases} \{t\} & \text{if } t \in \mathcal{V} \\ \{t\} \cup \bigcup_i \mathcal{S}ub(t_i) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Example

- for $\mathcal{F} = \{\mathsf{f}/1,\ \mathsf{g}/2,\ \mathsf{a}/0\}$ and $x, y \in \mathcal{V}$ have terms a, $\mathsf{f}(x)$, $\mathsf{f}(\mathsf{a})$, $\mathsf{g}(x, \mathsf{f}(y))$, ...
- for $t = \mathsf{g}(\mathsf{g}(x,x), \mathsf{f}(\mathsf{f}(\mathsf{a})))$ have $\mathcal{S}ub(t) = \{t,\ \mathsf{g}(x,x),\ x,\ \mathsf{f}(\mathsf{f}(\mathsf{a})),\ \mathsf{f}(\mathsf{a}),\ \mathsf{a}\}$

Congruence Closure

Input: set of equations E and equation $s \approx t$, both without variables

Output: *valid* ($E \models_T s \approx t$) or *invalid* ($E \models_T s \not\approx t$)

Congruence Closure

Input: set of equations E and equation $s \approx t$, both without variables

Output: *valid* ($E \vDash_T s \approx t$) or *invalid* ($E \vDash_T s \not\approx t$)

1. build congruence classes
   (a) put different subterms of terms in $E \cup \{s \approx t\}$ in separate sets

Congruence Closure

Input: set of equations E and equation $s \approx t$, both without variables

Output: *valid* ($E \vDash_T s \approx t$) or *invalid* ($E \vDash_T s \not\approx t$)

1. build congruence classes
   (a) put different subterms of terms in $E \cup \{s \approx t\}$ in separate sets
   (b) merge sets $\{\ldots, t_1, \ldots\}$ and $\{\ldots, t_2, \ldots\}$ for all $t_1 \approx t_2$ in E

Congruence Closure

Input: set of equations E and equation $s \approx t$, both without variables

Output: *valid* ($E \vDash_T s \approx t$) or *invalid* ($E \vDash_T s \not\approx t$)

1. build congruence classes
   (a) put different subterms of terms in $E \cup \{s \approx t\}$ in separate sets

   (b) merge sets $\{\dots, t_1, \dots\}$ and $\{\dots, t_2, \dots\}$ for all $t_1 \approx t_2$ in E

   (c) merge sets $\{\dots, f(t_1, \dots, t_n), \dots\}$ and $\{\dots, f(u_1, \dots, u_n), \dots\}$
   if t_i and u_i belong to same set for all $1 \leqslant i \leqslant n$, repeatedly

Congruence Closure

Input: set of equations E and equation $s \approx t$, both without variables

Output: *valid* ($E \vDash_T s \approx t$) or *invalid* ($E \vDash_T s \not\approx t$)

1. build congruence classes
   (a) put different subterms of terms in $E \cup \{s \approx t\}$ in separate sets
   (b) merge sets $\{\ldots, t_1, \ldots\}$ and $\{\ldots, t_2, \ldots\}$ for all $t_1 \approx t_2$ in E
   (c) merge sets $\{\ldots, f(t_1, \ldots, t_n), \ldots\}$ and $\{\ldots, f(u_1, \ldots, u_n), \ldots\}$ if t_i and u_i belong to same set for all $1 \leqslant i \leqslant n$, repeatedly
1. if s and t belong to same set then return *valid* else return *invalid*

Example (1)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \quad \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \approx \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{g}(\mathsf{f}(\mathsf{a})) \quad \mathsf{g}(\mathsf{a}) \approx \mathsf{b}$$

  and equation $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a})$

Example (1)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \quad \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \approx \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{g}(\mathsf{f}(\mathsf{a})) \quad \mathsf{g}(\mathsf{a}) \approx \mathsf{b}$$

  and equation $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a})$

- sets

1. $\{\,\mathsf{a}\,\}$
2. $\{\,\mathsf{f}(\mathsf{a})\,\}$
3. $\{\,\mathsf{b}\,\}$
4. $\{\,\mathsf{g}(\mathsf{b})\,\}$
5. $\{\,\mathsf{f}(\mathsf{f}(\mathsf{a}))\,\}$
6. $\{\,\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))\,\}$
7. $\{\,\mathsf{f}(\mathsf{b})\,\}$
8. $\{\,\mathsf{g}(\mathsf{f}(\mathsf{b}))\,\}$
9. $\{\,\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))\,\}$
10. $\{\,\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))))\,\}$
11. $\{\,\mathsf{g}(\mathsf{g}(\mathsf{b}))\,\}$
12. $\{\,\mathsf{g}(\mathsf{f}(\mathsf{a}))\,\}$
13. $\{\,\mathsf{g}(\mathsf{a})\,\}$

Example (1)

- given set of equations E

$$f(f(f(a))) \approx g(f(g(f(b)))) \quad f(g(f(b))) \approx f(a) \quad g(g(b)) \approx g(f(a)) \quad g(a) \approx b$$

  and equation $f(a) \approx g(a)$

- sets

1. $\{ a \}$
2. $\{ f(a) \}$
3. $\{ b \}$
4. $\{ g(b) \}$
5. $\{ f(f(a)) \}$
6. $\{ f(f(f(a))) \}$
7. $\{ f(b) \}$
8. $\{ g(f(b)) \}$
9. $\{ f(g(f(b))) \}$
10. $\{ g(f(g(f(b)))) \}$
11. $\{ g(g(b)) \}$
12. $\{ g(f(a)) \}$
13. $\{ g(a) \}$

Example (1)

- given set of equations E

  $$\mathsf{f(f(f(a)))} \approx \mathsf{g(f(g(f(b))))} \quad \mathsf{f(g(f(b)))} \approx \mathsf{f(a)} \quad \mathsf{g(g(b))} \approx \mathsf{g(f(a))} \quad \mathsf{g(a)} \approx \mathsf{b}$$

  and equation $\mathsf{f(a)} \approx \mathsf{g(a)}$

- sets

  1. $\{\, \mathsf{a} \,\}$
  2. $\{\, \mathsf{f(a)} \,\}$
  3. $\{\, \mathsf{b} \,\}$
  4. $\{\, \mathsf{g(b)} \,\}$
  5. $\{\, \mathsf{f(f(a))} \,\}$
  6. $\{\, \mathsf{f(f(f(a))), g(f(g(f(b))))} \,\}$
  7. $\{\, \mathsf{f(b)} \,\}$
  8. $\{\, \mathsf{g(f(b))} \,\}$
  9. $\{\, \mathsf{f(g(f(b)))} \,\}$
  11. $\{\, \mathsf{g(g(b))} \,\}$
  12. $\{\, \mathsf{g(f(a))} \,\}$
  13. $\{\, \mathsf{g(a)} \,\}$

Example (1)

- given set of equations E

  $$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \quad \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \approx \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{g}(\mathsf{f}(\mathsf{a})) \quad \mathsf{g}(\mathsf{a}) \approx \mathsf{b}$$

  and equation $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a})$
- sets

1. $\{\, \mathsf{a} \,\}$
2. $\{\, \mathsf{f}(\mathsf{a}) \,\}$
3. $\{\, \mathsf{b} \,\}$
4. $\{\, \mathsf{g}(\mathsf{b}) \,\}$
5. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$
6. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \,\}$
7. $\{\, \mathsf{f}(\mathsf{b}) \,\}$
8. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{b})) \,\}$
9. $\{\, \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \,\}$
11. $\{\, \mathsf{g}(\mathsf{g}(\mathsf{b})) \,\}$
12. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{a})) \,\}$
13. $\{\, \mathsf{g}(\mathsf{a}) \,\}$

Example (1)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \quad \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \approx \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{g}(\mathsf{f}(\mathsf{a})) \quad \mathsf{g}(\mathsf{a}) \approx \mathsf{b}$$

  and equation $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a})$

- sets

1. $\{\, \mathsf{a} \,\}$
2. $\{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \,\}$
3. $\{\, \mathsf{b} \,\}$
4. $\{\, \mathsf{g}(\mathsf{b}) \,\}$
5. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$
6. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \,\}$
7. $\{\, \mathsf{f}(\mathsf{b}) \,\}$
8. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{b})) \,\}$

11. $\{\, \mathsf{g}(\mathsf{g}(\mathsf{b})) \,\}$
12. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{a})) \,\}$
13. $\{\, \mathsf{g}(\mathsf{a}) \,\}$

Example (1)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \quad \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \approx \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{g}(\mathsf{f}(\mathsf{a})) \quad \mathsf{g}(\mathsf{a}) \approx \mathsf{b}$$

  and equation $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a})$

- sets

1. $\{\, \mathsf{a} \,\}$
2. $\{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \,\}$
3. $\{\, \mathsf{b} \,\}$
4. $\{\, \mathsf{g}(\mathsf{b}) \,\}$
5. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$
6. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \,\}$
7. $\{\, \mathsf{f}(\mathsf{b}) \,\}$
8. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{b})) \,\}$
11. $\{\, \mathsf{g}(\mathsf{g}(\mathsf{b})) \,\}$
12. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{a})) \,\}$
13. $\{\, \mathsf{g}(\mathsf{a}) \,\}$

Example (1)

- given set of equations E

$$f(f(f(a))) \approx g(f(g(f(b)))) \quad f(g(f(b))) \approx f(a) \quad g(g(b)) \approx g(f(a)) \quad g(a) \approx b$$

  and equation $f(a) \approx g(a)$

- sets

1. $\{ a \}$
2. $\{ f(a), f(g(f(b))) \}$
3. $\{ b \}$
4. $\{ g(b) \}$
5. $\{ f(f(a)) \}$
6. $\{ f(f(f(a))), g(f(g(f(b)))) \}$
7. $\{ f(b) \}$
8. $\{ g(f(b)) \}$

11. $\{ g(g(b)), g(f(a)) \}$

13. $\{ g(a) \}$

Example (1)

- given set of equations E

$$f(f(f(a))) \approx g(f(g(f(b)))) \quad f(g(f(b))) \approx f(a) \quad g(g(b)) \approx g(f(a)) \quad g(a) \approx b$$

  and equation $f(a) \approx g(a)$

- sets

1. $\{ a \}$
2. $\{ f(a), f(g(f(b))) \}$
3. $\{ b \}$
4. $\{ g(b) \}$
5. $\{ f(f(a)) \}$
6. $\{ f(f(f(a))), g(f(g(f(b)))) \}$
7. $\{ f(b) \}$
8. $\{ g(f(b)) \}$
11. $\{ g(g(b)), g(f(a)) \}$
13. $\{ g(a) \}$

Example (1)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \quad \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \approx \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{g}(\mathsf{f}(\mathsf{a})) \quad \mathsf{g}(\mathsf{a}) \approx \mathsf{b}$$

  and equation $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a})$

- sets

1. $\{\, \mathsf{a} \,\}$
2. $\{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \,\}$
3. $\{\, \mathsf{b}, \mathsf{g}(\mathsf{a}) \,\}$
4. $\{\, \mathsf{g}(\mathsf{b}) \,\}$
5. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$
6. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \,\}$
7. $\{\, \mathsf{f}(\mathsf{b}) \,\}$
8. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{b})) \,\}$

11. $\{\, \mathsf{g}(\mathsf{g}(\mathsf{b})), \mathsf{g}(\mathsf{f}(\mathsf{a})) \,\}$

Example (1)

- given set of equations E

$$\mathsf{f(f(f(a)))} \approx \mathsf{g(f(g(f(b))))} \quad \mathsf{f(g(f(b)))} \approx \mathsf{f(a)} \quad \mathsf{g(g(b))} \approx \mathsf{g(f(a))} \quad \mathsf{g(a)} \approx \mathsf{b}$$

  and equation $\mathsf{f(a)} \approx \mathsf{g(a)}$

- sets

  1. $\{\,\mathsf{a}\,\}$
  2. $\{\,\mathsf{f(a)}, \mathsf{f(g(f(b)))}\,\}$
  3. $\{\,\mathsf{b}, \mathsf{g(a)}\,\}$
  4. $\{\,\mathsf{g(b)}\,\}$
  5. $\{\,\mathsf{f(f(a))}\,\}$
  6. $\{\,\mathsf{f(f(f(a)))}, \mathsf{g(f(g(f(b))))}\,\}$
  7. $\{\,\mathsf{f(b)}\,\}$
  8. $\{\,\mathsf{g(f(b))}\,\}$
  11. $\{\,\mathsf{g(g(b))}, \mathsf{g(f(a))}\,\}$

Example (1)

- given set of equations E

$$\mathsf{f(f(f(a)))} \approx \mathsf{g(f(g(f(b))))} \quad \mathsf{f(g(f(b)))} \approx \mathsf{f(a)} \quad \mathsf{g(g(b))} \approx \mathsf{g(f(a))} \quad \mathsf{g(a)} \approx \mathsf{b}$$

  and equation $\mathsf{f(a)} \approx \mathsf{g(a)}$

- sets

  1. $\{\,\mathsf{a}\,\}$
  2. $\{\,\mathsf{f(a)}, \mathsf{f(g(f(b)))}\,\}$
  3. $\{\,\mathsf{b}, \mathsf{g(a)}\,\}$
  4. $\{\,\mathsf{g(b)}\,\}$
  5. $\{\,\mathsf{f(f(a))}\,\}$
  6. $\{\,\mathsf{f(f(f(a)))}, \mathsf{g(f(g(f(b))))}, \mathsf{g(g(b))}, \mathsf{g(f(a))}\,\}$
  7. $\{\,\mathsf{f(b)}\,\}$
  8. $\{\,\mathsf{g(f(b))}\,\}$

Example (1)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))) \quad \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \approx \mathsf{f}(\mathsf{a}) \quad \mathsf{g}(\mathsf{g}(\mathsf{b})) \approx \mathsf{g}(\mathsf{f}(\mathsf{a})) \quad \mathsf{g}(\mathsf{a}) \approx \mathsf{b}$$

  and equation $\mathsf{f}(\mathsf{a}) \approx \mathsf{g}(\mathsf{a})$

- sets

  1. $\{\, \mathsf{a} \,\}$
  2. $\{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b}))) \,\}$
  3. $\{\, \mathsf{b}, \mathsf{g}(\mathsf{a}) \,\}$
  4. $\{\, \mathsf{g}(\mathsf{b}) \,\}$
  5. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$
  6. $\{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{b})))), \mathsf{g}(\mathsf{g}(\mathsf{b})), \mathsf{g}(\mathsf{f}(\mathsf{a})) \,\}$
  7. $\{\, \mathsf{f}(\mathsf{b}) \,\}$
  8. $\{\, \mathsf{g}(\mathsf{f}(\mathsf{b})) \,\}$

- conclusion: $E \vDash s \not\approx t$

Example (2)

- given set of equations E

$$\mathsf{f(f(f(a)))} \approx \mathsf{a} \qquad \mathsf{f(f(f(f(f(a)))))} \approx \mathsf{a}$$

and equaton $\mathsf{f(a)} \approx \mathsf{a}$

Example (2)

- given set of equations E

$$\mathsf{f(f(f(a)))} \approx \mathsf{a} \qquad \mathsf{f(f(f(f(f(a)))))} \approx \mathsf{a}$$

  and equaton $\mathsf{f(a)} \approx \mathsf{a}$

- $$\{\,\mathsf{a}\,\} \quad \{\,\mathsf{f(a)}\,\} \quad \{\,\mathsf{f(f(a))}\,\} \quad \{\,\mathsf{f(f(f(a)))}\,\} \quad \{\,\mathsf{f(f(f(f(a))))}\,\} \quad \{\,\mathsf{f(f(f(f(f(a)))))}\,\}$$

Example (2)

- given set of equations E

$$f(f(f(a))) \approx a \qquad f(f(f(f(f(a))))) \approx a$$

  and equaton $f(a) \approx a$

- $$\{\, a \,\} \quad \{\, f(a) \,\} \quad \{\, f(f(a)) \,\} \quad \{\, f(f(f(a))) \,\} \quad \{\, f(f(f(f(a)))) \,\} \quad \{\, f(f(f(f(f(a))))) \,\}$$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \,\} \quad \{\, \mathsf{f}(\mathsf{a}) \,\} \quad \{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\} \quad \{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))) \,\} \quad \{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \,\}$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\,\mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))\,\}$ $\{\,\mathsf{f}(\mathsf{a})\,\}$ $\{\,\mathsf{f}(\mathsf{f}(\mathsf{a}))\,\}$ $\{\,\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))\,\}$ $\{\,\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))))\,\}$

Example (2)

- given set of equations E

$$\mathsf{f(f(f(a)))} \approx \mathsf{a} \qquad \mathsf{f(f(f(f(f(a)))))} \approx \mathsf{a}$$

  and equaton $\mathsf{f(a)} \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f(f(f(a)))}, \mathsf{f(f(f(f(f(a)))))} \,\} \quad \{\, \mathsf{f(a)} \,\} \quad \{\, \mathsf{f(f(a))} \,\} \quad \{\, \mathsf{f(f(f(f(a))))} \,\}$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \,\}$ $\{\, \mathsf{f}(\mathsf{a}) \,\}$ $\{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$ $\{\, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))) \,\}$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \,\} \quad \{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))) \,\} \quad \{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \,\}$ $\{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))) \,\}$ $\{\, \mathsf{f}(\mathsf{f}(\mathsf{a})) \,\}$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{a})), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \,\} \quad \{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))) \,\}$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{a})), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \,\} \quad \{\, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))) \,\}$

Example (2)

- given set of equations E

$$\mathsf{f(f(f(a)))} \approx \mathsf{a} \qquad\qquad \mathsf{f(f(f(f(f(a)))))} \approx \mathsf{a}$$

  and equaton $\mathsf{f(a)} \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f(a)}, \mathsf{f(f(a))}, \mathsf{f(f(f(a)))}, \mathsf{f(f(f(f(a))))}, \mathsf{f(f(f(f(f(a)))))} \,\}$

Example (2)

- given set of equations E

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \approx \mathsf{a}$$

  and equaton $\mathsf{f}(\mathsf{a}) \approx \mathsf{a}$

- $\{\, \mathsf{a}, \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{f}(\mathsf{a})), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a})))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))) \,\}$

- conclusion: $E \vDash s \approx t$

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \ldots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \ \ldots, x_n \mapsto c_n]$ where $c_1, \ldots, c_n$ are fresh constants

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction φ
with free variables $x_1, \dots, x_n$

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction $\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$, split into positive and negative literals with free variables $x_1, \dots, x_n$

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1,\ \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction $\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$, split into positive and negative literals with free variables $x_1, \dots, x_n$

$\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable
$\iff \exists x_1 \dots x_n.(\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1,\ \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction $\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$, split into positive and negative literals with free variables $x_1, \dots, x_n$

$\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$ unsatisfiable

$\iff \exists x_1 \dots x_n . (\bigwedge P) \wedge (\bigwedge \neg N)$ unsatisfiable

$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N})$ unsatisfiable — skolemization

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction $\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$, split into positive and negative literals with free variables $x_1, \dots, x_n$

$\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff \exists x_1 \dots x_n . (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N})$	unsatisfiable	skolemization
$\iff \neg \left((\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N}) \right)$	valid	φ unsat iff $\neg\varphi$ valid

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction $\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$, split into positive and negative literals with free variables $x_1, \dots, x_n$

$\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff \exists x_1 \dots x_n . (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N})$	unsatisfiable	skolemization
$\iff \neg \Big((\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N}) \Big)$	valid	φ unsat iff $\neg\varphi$ valid
$\iff \bigwedge \widehat{P} \rightarrow \bigvee \widehat{N}$	valid	deMorgan

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction $\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$, split into positive and negative literals with free variables $x_1, \dots, x_n$

$\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff \exists x_1 \dots x_n . (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N})$	unsatisfiable	skolemization
$\iff \neg \left((\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N}) \right)$	valid	φ unsat iff $\neg\varphi$ valid
$\iff \bigwedge \widehat{P} \to \bigvee \widehat{N}$	valid	deMorgan
$\iff \exists\, s \approx t$ in $\widehat{N}$ such that $\bigwedge \widehat{P} \to s \approx t$ valid		semantics of $\vee$

Ok, But How About a Solver for EUF?

Definition (Skolemization)

given formula φ with free variables $x_1, \dots, x_n$,
$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are fresh constants

Deciding Satisfiability of EUF Conjunctions

for EUF conjunction $\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$, split into positive and negative literals with free variables $x_1, \dots, x_n$

$\varphi = (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff \exists x_1 \dots x_n . (\bigwedge P) \wedge (\bigwedge \neg N)$	unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N})$	unsatisfiable	skolemization
$\iff \neg \Big((\bigwedge \widehat{P}) \wedge (\bigwedge \neg \widehat{N}) \Big)$	valid	φ unsat iff $\neg\varphi$ valid
$\iff \bigwedge \widehat{P} \to \bigvee \widehat{N}$	valid	deMorgan
$\iff \exists\, s \approx t$ in $\widehat{N}$ such that $\bigwedge \widehat{P} \to s \approx t$ valid		semantics of $\vee$
$\iff \exists\, s \approx t$ in $\widehat{N}$ such that $\bigwedge \widehat{P} \models_T s \approx t$		semantics of $\models$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \quad \Longleftrightarrow \quad \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \not\approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \not\approx \mathsf{d}$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \quad \Longleftrightarrow \quad \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \vDash_T s \approx t$$

Example

1. $\mathrm{g}(\mathrm{a}) \approx \mathrm{c} \wedge \mathrm{f}(\mathrm{g}(\mathrm{a})) \not\approx \mathrm{f}(\mathrm{c}) \wedge \mathrm{c} \not\approx \mathrm{d}$
   - split into $P = \{\mathrm{g}(\mathrm{a}) \approx \mathrm{c}\}$ and $N = \{\mathrm{f}(\mathrm{g}(\mathrm{a})) \approx \mathrm{f}(\mathrm{c}),\ \mathrm{c} \approx \mathrm{d}\}$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \iff \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g(a)} \approx \mathsf{c} \wedge \mathsf{f(g(a))} \not\approx \mathsf{f(c)} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g(a)} \approx \mathsf{c}\}$ and $N = \{\mathsf{f(g(a))} \approx \mathsf{f(c)},\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g(a)} \approx \mathsf{c} \models_T \mathsf{f(g(a))} \approx \mathsf{f(c)}$, so unsatisfiable

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \quad \Longleftrightarrow \quad \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \not\approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}),\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \models_T \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{g}(\mathsf{a}) \approx \mathsf{d} \wedge \mathsf{c} \not\approx \mathsf{d}$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \iff \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g(a)} \approx \mathsf{c} \wedge \mathsf{f(g(a))} \not\approx \mathsf{f(c)} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g(a)} \approx \mathsf{c}\}$ and $N = \{\mathsf{f(g(a))} \approx \mathsf{f(c)},\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g(a)} \approx \mathsf{c} \models_T \mathsf{f(g(a))} \approx \mathsf{f(c)}$, so unsatisfiable

2. $\mathsf{g(a)} \approx \mathsf{c} \wedge \mathsf{f(g(a))} \approx \mathsf{f(c)} \wedge \mathsf{g(a)} \approx \mathsf{d} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g(a)} \approx \mathsf{c},\ \mathsf{f(g(a))} \approx \mathsf{f(c)},\ \mathsf{g(a)} \approx \mathsf{d}\}$ and $N = \{\mathsf{c} \approx \mathsf{d}\}$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \iff \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g(a)} \approx \mathsf{c} \wedge \mathsf{f(g(a))} \not\approx \mathsf{f(c)} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g(a)} \approx \mathsf{c}\}$ and $N = \{\mathsf{f(g(a))} \approx \mathsf{f(c)},\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g(a)} \approx \mathsf{c} \models_T \mathsf{f(g(a))} \approx \mathsf{f(c)}$, so unsatisfiable

2. $\mathsf{g(a)} \approx \mathsf{c} \wedge \mathsf{f(g(a))} \approx \mathsf{f(c)} \wedge \mathsf{g(a)} \approx \mathsf{d} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g(a)} \approx \mathsf{c},\ \mathsf{f(g(a))} \approx \mathsf{f(c)},\ \mathsf{g(a)} \approx \mathsf{d}\}$ and $N = \{\mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g(a)} \approx \mathsf{c}, \mathsf{f(g(a))} \approx \mathsf{f(c)}, \mathsf{g(a)} \approx \mathsf{d} \models_T \mathsf{c} \approx \mathsf{d}$, so unsatisfiable

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \iff \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \not\approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}),\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \models_T \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{g}(\mathsf{a}) \approx \mathsf{d} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}),\ \mathsf{g}(\mathsf{a}) \approx \mathsf{d}\}$ and $N = \{\mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) \approx \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}), \mathsf{g}(\mathsf{a}) \approx \mathsf{d} \models_T \mathsf{c} \approx \mathsf{d}$, so unsatisfiable

3. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{c} \approx \mathsf{d} \wedge \mathsf{f}(x) \approx x \wedge \mathsf{d} \not\approx \mathsf{g}(x) \wedge \mathsf{f}(x) \not\approx \mathsf{d}$
   - $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f}(x) \approx x\}$ and $N = \{\mathsf{d} \approx \mathsf{g}(x),\ \mathsf{f}(x) \approx \mathsf{d}\}$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \iff \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \not\approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}),\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \models_T \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{g}(\mathsf{a}) \approx \mathsf{d} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}),\ \mathsf{g}(\mathsf{a}) \approx \mathsf{d}\}$ and $N = \{\mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) \approx \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}), \mathsf{g}(\mathsf{a}) \approx \mathsf{d} \models_T \mathsf{c} \approx \mathsf{d}$, so unsatisfiable

3. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{c} \approx \mathsf{d} \wedge \mathsf{f}(x) \approx x \wedge \mathsf{d} \not\approx \mathsf{g}(x) \wedge \mathsf{f}(x) \not\approx \mathsf{d}$
   - $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f}(x) \approx x\}$ and $N = \{\mathsf{d} \approx \mathsf{g}(x),\ \mathsf{f}(x) \approx \mathsf{d}\}$
   - skolemize $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f}(\mathsf{e}) \approx \mathsf{e}\}$, $N = \{\mathsf{d} \approx \mathsf{g}(\mathsf{e}),\ \mathsf{f}(\mathsf{e}) \approx \mathsf{d}\}$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge \neg N) \text{ unsatisfiable} \iff \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \not\approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}),\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \models_T \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}) \wedge \mathsf{g}(\mathsf{a}) \approx \mathsf{d} \wedge \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}),\ \mathsf{g}(\mathsf{a}) \approx \mathsf{d}\}$ and $N = \{\mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) \approx \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{c}), \mathsf{g}(\mathsf{a}) \approx \mathsf{d} \models_T \mathsf{c} \approx \mathsf{d}$, so unsatisfiable

3. $\mathsf{g}(\mathsf{a}) \approx \mathsf{c} \wedge \mathsf{c} \approx \mathsf{d} \wedge \mathsf{f}(x) \approx x \wedge \mathsf{d} \not\approx \mathsf{g}(x) \wedge \mathsf{f}(x) \not\approx \mathsf{d}$
   - $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f}(x) \approx x\}$ and $N = \{\mathsf{d} \approx \mathsf{g}(x),\ \mathsf{f}(x) \approx \mathsf{d}\}$
   - skolemize $P = \{\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f}(\mathsf{e}) \approx \mathsf{e}\}$, $N = \{\mathsf{d} \approx \mathsf{g}(\mathsf{e}),\ \mathsf{f}(\mathsf{e}) \approx \mathsf{d}\}$
     - $\mathsf{g}(\mathsf{a}) \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f}(\mathsf{e}) \approx \mathsf{e} \not\models_T \mathsf{d} \approx \mathsf{g}(\mathsf{e})$

Obtained Satisfiability Check

$$(\bigwedge P) \land (\bigwedge \neg N) \text{ unsatisfiable} \quad \Longleftrightarrow \quad \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $\mathsf{g(a)} \approx \mathsf{c} \land \mathsf{f(g(a))} \not\approx \mathsf{f(c)} \land \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g(a)} \approx \mathsf{c}\}$ and $N = \{\mathsf{f(g(a))} \approx \mathsf{f(c)},\ \mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g(a)} \approx \mathsf{c} \models_T \mathsf{f(g(a))} \approx \mathsf{f(c)}$, so unsatisfiable

2. $\mathsf{g(a)} \approx \mathsf{c} \land \mathsf{f(g(a))} \approx \mathsf{f(c)} \land \mathsf{g(a)} \approx \mathsf{d} \land \mathsf{c} \not\approx \mathsf{d}$
   - split into $P = \{\mathsf{g(a)} \approx \mathsf{c},\ \mathsf{f(g(a))} \approx \mathsf{f(c)},\ \mathsf{g(a)} \approx \mathsf{d}\}$ and $N = \{\mathsf{c} \approx \mathsf{d}\}$
   - have $\mathsf{g(a)} \approx \mathsf{c}, \mathsf{f(g(a))} \approx \mathsf{f(c)}, \mathsf{g(a)} \approx \mathsf{d} \models_T \mathsf{c} \approx \mathsf{d}$, so unsatisfiable

3. $\mathsf{g(a)} \approx \mathsf{c} \land \mathsf{c} \approx \mathsf{d} \land \mathsf{f}(x) \approx x \land \mathsf{d} \not\approx \mathsf{g}(x) \land \mathsf{f}(x) \not\approx \mathsf{d}$
   - $P = \{\mathsf{g(a)} \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f}(x) \approx x\}$ and $N = \{\mathsf{d} \approx \mathsf{g}(x),\ \mathsf{f}(x) \approx \mathsf{d}\}$
   - skolemize $P = \{\mathsf{g(a)} \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f(e)} \approx \mathsf{e}\}$, $N = \{\mathsf{d} \approx \mathsf{g(e)},\ \mathsf{f(e)} \approx \mathsf{d}\}$
     - $\mathsf{g(a)} \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f(e)} \approx \mathsf{e} \not\models_T \mathsf{d} \approx \mathsf{g(e)}$
     - $\mathsf{g(a)} \approx \mathsf{c},\ \mathsf{c} \approx \mathsf{d},\ \mathsf{f(e)} \approx \mathsf{e} \not\models_T \mathsf{f(e)} \approx \mathsf{d}$

Obtained Satisfiability Check

$$(\bigwedge P) \land (\bigwedge \neg N) \text{ unsatisfiable} \quad \Longleftrightarrow \quad \exists\, s \approx t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \models_T s \approx t$$

Example

1. $g(a) \approx c \land f(g(a)) \not\approx f(c) \land c \not\approx d$
   - split into $P = \{g(a) \approx c\}$ and $N = \{f(g(a)) \approx f(c),\ c \approx d\}$
   - have $g(a) \approx c \models_T f(g(a)) \approx f(c)$, so unsatisfiable

2. $g(a) \approx c \land f(g(a)) \approx f(c) \land g(a) \approx d \land c \not\approx d$
   - split into $P = \{g(a) \approx c,\ f(g(a)) \approx f(c),\ g(a) \approx d\}$ and $N = \{c \approx d\}$
   - have $g(a) \approx c, f(g(a)) \approx f(c), g(a) \approx d \models_T c \approx d$, so unsatisfiable

3. $g(a) \approx c \land c \approx d \land f(x) \approx x \land d \not\approx g(x) \land f(x) \not\approx d$
   - $P = \{g(a) \approx c,\ c \approx d,\ f(x) \approx x\}$ and $N = \{d \approx g(x),\ f(x) \approx d\}$
   - skolemize $P = \{g(a) \approx c,\ c \approx d,\ f(e) \approx e\}$, $N = \{d \approx g(e),\ f(e) \approx d\}$
     - $g(a) \approx c,\ c \approx d,\ f(e) \approx e \not\models_T d \approx g(e)$
     - $g(a) \approx c,\ c \approx d,\ f(e) \approx e \not\models_T f(e) \approx d$

   so satisfiable

Some More Practical SMT

Integer Arithmetic in python/z3

```
from z3 import *
a = Int('a') # create integer variables
b = Int('b')
c = Int('c')

phi = And(c > 0, b >= 0, a < -1)
psi = (a == If (b == c, b - 2, c - 4))
print(phi)
solver = Solver()
solver.add(phi, psi) # assert constraints
solver.add(a + b < 2 * c)

result = solver.check() # check for satisfiability
if result == z3.sat:
  model = solver.model() # get valuation
  print model[a], model[b], model[c] # -3 0 1
```