universität innsbruck

SAT and SMT Solving

Sarah Winkler

KRDB
Department of Computer Science
Free University of Bozen-Bolzano

lecture 6
WS 2022

Outline

- Summary of Last Week
- Deciding EQ: Equality Graphs
- Deciding EUF: Congruence Closure
- Correctness of DPLL(T)
- Some More Practical SMT

First-Order Logic: Syntax

Definitions

- signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$ consists of
  - set of function symbols $\mathcal{F}$
  - set of predicate symbols $\mathcal{P}$

  where each symbol is associated with fixed arity
- Σ-terms t are built according to grammar
$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \dots, t}_{n})$$
- Σ-formulas φ are built according to grammar
$$\varphi \quad ::= \quad Q \mid P(\underbrace{t, \dots, t}_{n}) \mid \bot \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \forall x.\, \varphi \mid \exists x.\, \varphi$$
- variable occurrence is free in φ if it is not bound by quantifier above
- formulas without free variables are sentences

First-Order Logic: Semantics

Definition

model $\mathcal{M}$ for signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$ consists of

1. non-empty set A (universe of concrete values)
2. function $f^{\mathcal{M}} \colon A^n \to A$ for every n-ary $f \in \mathcal{F}$
3. set of n-tuples $P^{\mathcal{M}} \subseteq A^n$ for every n-ary $P \in \mathcal{P}$

First-Order Logic: Semantics

Definition

model $\mathcal{M}$ for signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$ consists of

1. non-empty set A (universe of concrete values)
2. function $f^{\mathcal{M}} \colon A^n \to A$ for every n-ary $f \in \mathcal{F}$
3. set of n-tuples $P^{\mathcal{M}} \subseteq A^n$ for every n-ary $P \in \mathcal{P}$

Definitions

- environment for model $\mathcal{M}$ with universe A is mapping $l \colon X \to A$

First-Order Logic: Semantics

Definition

model $\mathcal{M}$ for signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$ consists of

1. non-empty set A (universe of concrete values)
2. function $f^{\mathcal{M}}\colon A^n \to A$ for every n-ary $f \in \mathcal{F}$
3. set of n-tuples $P^{\mathcal{M}} \subseteq A^n$ for every n-ary $P \in \mathcal{P}$

Definitions

- environment for model $\mathcal{M}$ with universe A is mapping $l\colon X \to A$
- value $t^{\mathcal{M},l}$ of term t in model $\mathcal{M}$ wrt environment l:
  $t^{\mathcal{M},l} = l(t)$ if t is a variable, and $t^{\mathcal{M},l} = f^{\mathcal{M}}(t_n^{\mathcal{M},l}, \dots, t_n^{\mathcal{M},l})$ otherwise

First-Order Logic: Semantics

Definition

model $\mathcal{M}$ for signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$ consists of

1. non-empty set A (universe of concrete values)
2. function $f^{\mathcal{M}}\colon A^n \to A$ for every n-ary $f \in \mathcal{F}$
3. set of n-tuples $P^{\mathcal{M}} \subseteq A^n$ for every n-ary $P \in \mathcal{P}$

Definitions

- environment for model $\mathcal{M}$ with universe A is mapping $I\colon X \to A$
- value $t^{\mathcal{M},I}$ of term t in model $\mathcal{M}$ wrt environment I:
  $t^{\mathcal{M},I} = I(t)$ if t is a variable, and $t^{\mathcal{M},I} = f^{\mathcal{M}}(t_n^{\mathcal{M},I}, \dots, t_n^{\mathcal{M},I})$ otherwise
- $$\mathcal{M} \models_I \varphi \iff \begin{cases} (t_n^{\mathcal{M},I}, \dots, t_n^{\mathcal{M},I}) \in P^{\mathcal{M}} & \text{if } \varphi = P(t_1, \dots, t_n) \\ \mathcal{M} \not\models_I \psi & \text{if } \varphi = \neg\psi \\ \mathcal{M} \models_I \varphi_1 \text{ and } \mathcal{M} \models_I \varphi_2 & \text{if } \varphi = \varphi_1 \wedge \varphi_2 \\ \mathcal{M} \models_I \varphi_1 \text{ or } \mathcal{M} \models_I \varphi_2 & \text{if } \varphi = \varphi_1 \vee \varphi_2 \\ \mathcal{M} \models_{I[x \mapsto a]} \psi \text{ for all } a \in A & \text{if } \varphi = \forall x.\, \psi \\ \mathcal{M} \models_{I[x \mapsto a]} \psi \text{ for some } a \in A & \text{if } \varphi = \exists x.\, \psi \end{cases}$$

Definition

- formula φ is satisfiable if $\mathcal{M} \models_I \varphi$ for some $\mathcal{M}$ and I
- set of formulas T is satisfiable if $\mathcal{M} \models_I \bigwedge_{\varphi \in T} \varphi$ for some $\mathcal{M}$ and I

Remark

if φ is sentence, $\mathcal{M} \models_I \varphi$ is independent of I

Definition

- formula φ is satisfiable if $\mathcal{M} \models_I \varphi$ for some $\mathcal{M}$ and I
- set of formulas T is satisfiable if $\mathcal{M} \models_I \bigwedge_{\varphi \in T} \varphi$ for some $\mathcal{M}$ and I

Remark

if φ is sentence, $\mathcal{M} \models_I \varphi$ is independent of I

Definition (Theory)

Σ-theory T is set of Σ-sentences that is satisfiable

Definition

- formula φ is satisfiable if $\mathcal{M} \models_I \varphi$ for some $\mathcal{M}$ and I
- set of formulas T is satisfiable if $\mathcal{M} \models_I \bigwedge_{\varphi \in T} \varphi$ for some $\mathcal{M}$ and I

Remark

if φ is sentence, $\mathcal{M} \models_I \varphi$ is independent of I

Definition (Theory)

Σ-theory T is set of Σ-sentences that is satisfiable

Definitions

for theory T, formulas F and G and list of literals M:

- F is T-consistent (or T-satisfiable) if $\{F\} \cup T$ is satisfiable
- F is T-inconsistent (or T-unsatisfiable) if not 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 EQ)

- signature: no function symbols, binary predicate $=$
- axioms:

$$\forall x.\ (x = x) \quad \forall x\, y.\ (x = y \ \rightarrow\ y = x) \quad \forall x\, y\, z.\ (x = y \wedge y = z \ \rightarrow\ x = z)$$

Definition (Theory of Equality EQ)

- signature: no function symbols, binary predicate $=$
- axioms:

$$\forall x.\,(x = x) \quad \forall x\, y.\,(x = y \;\rightarrow\; y = x) \quad \forall x\, y\, z.\,(x = y \wedge y = z \;\rightarrow\; x = z)$$

Definition (Theory of Equality With Uninterpreted Functions EUF)

- signature: function symbols $\mathcal{F}$, predicate symbols $\mathcal{P}$ including binary $=$
- axioms:

$$\forall x.\,(x = x) \quad \forall x\, y.\,(x = y \;\rightarrow\; y = x) \quad \forall x\, y\, z.\,(x = y \wedge y = z \;\rightarrow\; x = z)$$

plus for all $f/n \in \mathcal{F}$ and $P/n \in \mathcal{P}$ functional consistency axioms:

$$\forall x_1 y_1 \,\ldots\, x_n y_n.\,(x_1 = y_1 \wedge \cdots \wedge x_n = y_n \;\rightarrow\; f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n))$$

$$\forall x_1\, y_1 \,\ldots\, x_n\, y_n.\,(x_1 = y_1 \wedge \cdots \wedge x_n = y_n \;\rightarrow\; (P(x_1, \ldots, x_n) \rightarrow P(y_1, \ldots, y_n)))$$

Definition

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

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

  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 $\quad M \parallel F \implies M \parallel F,\ C$

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

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

  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 $\exists l_1, \dots, l_k$ 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

Outline

- Summary of Last Week
- Deciding EQ: Equality Graphs
- Deciding EUF: Congruence Closure
- Correctness of DPLL(T)
- Some More Practical SMT

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

equality graph for φ_{EQ} is undirected graph $(V, E_=, E_{\neq})$ with two kinds of edges

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

equality graph for φ_{EQ} is undirected graph $(V, E_{=}, E_{\neq})$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\mathsf{EQ}})$

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

equality graph for φ_{EQ} is undirected graph $(V, E_=, E_{\neq})$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\mathsf{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} equality edge

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

equality graph for φ_{EQ} is undirected graph $(V, E_=, E_{\neq})$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\mathsf{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} — equality edge
- $(x, y) \in E_{\neq}$ iff $x \neq y$ in φ_{EQ} — inequality edge

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

equality graph for φ_{EQ} is undirected graph $(V, E_=, E_{\neq})$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\mathsf{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} — equality edge
- $(x, y) \in E_{\neq}$ iff $x \neq y$ in φ_{EQ} — inequality edge

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$

Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\text{EQ}})$ for set of variables occurring in φ_{EQ}

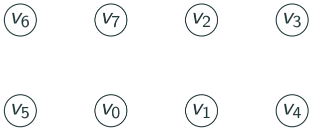
Definition

equality graph for φ_{EQ} is undirected graph $(V, E_=, E_\neq)$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\text{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} equality edge
- $(x, y) \in E_\neq$ iff $x \neq y$ in φ_{EQ} inequality edge

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

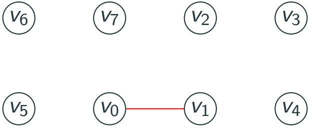
equality graph for φ_{EQ} is undirected graph $(V, E_=, E_{\neq})$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\mathsf{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} — equality edge
- $(x, y) \in E_{\neq}$ iff $x \neq y$ in φ_{EQ} — e

edges $E_=$ are drawn dashed,
$E_{\neq}$ are drawn solid

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

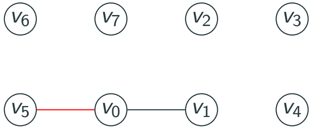
equality graph for φ_{EQ} is undirected graph $(V, E_=, E_\neq)$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\mathsf{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} equality edge
- $(x, y) \in E_\neq$ iff $x \neq y$ in φ_{EQ} e

edges $E_=$ are drawn dashed,
$E_\neq$ are drawn solid

Example

$v_0 \neq v_1$ $\quad v_0 \neq v_5$ $\quad v_1 = v_2$ $\quad v_1 \neq v_4$ $\quad v_1 \neq v_3$ $\quad v_2 = v_3$ $\quad v_5 = v_6$ $\quad v_6 = v_7$ $\quad v_7 = v_0$



Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\text{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

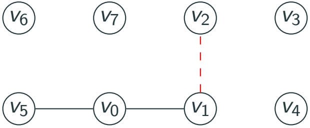
equality graph for φ_{EQ} is undirected graph $(V, E_=, E_{\neq})$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\text{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} — equality edge
- $(x, y) \in E_{\neq}$ iff $x \neq y$ in φ_{EQ}

edges $E_=$ are drawn dashed, $E_{\neq}$ are drawn solid

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\text{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

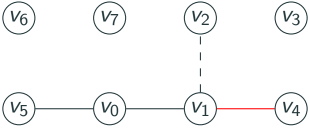
equality graph for φ_{EQ} is undirected graph $(V, E_=, E_{\neq})$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\text{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} equality edge
- $(x, y) \in E_{\neq}$ iff $x \neq y$ in φ_{EQ}

edges $E_=$ are drawn dashed, $E_{\neq}$ are drawn solid

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Equality Graph

Aim

build theory solver for theory of equality (EQ)

Definition

- equality logic formula φ_{EQ} is set of equations and inequalities between variables
- write $\mathcal{V}ar(\varphi_{\mathsf{EQ}})$ for set of variables occurring in φ_{EQ}

Definition

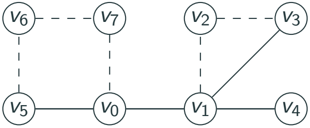
equality graph for φ_{EQ} is undirected graph $(V, E_=, E_\neq)$ with two kinds of edges

- nodes $V = \mathcal{V}ar(\varphi_{\mathsf{EQ}})$
- $(x, y) \in E_=$ iff $x = y$ in φ_{EQ} — equality edge
- $(x, y) \in E_\neq$ iff $x \neq y$ in φ_{EQ} — e

edges $E_=$ are drawn dashed, $E_\neq$ are drawn solid

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Definition (Contradictory cycle)

contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Definition (Contradictory cycle)

contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Theorem

φ_{EQ} is satisfiable iff its equality graph has no contradictory cycle

Definition (Contradictory cycle)

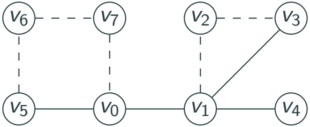
contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Theorem

φ_{EQ} is satisfiable iff its equality graph has no contradictory cycle

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Definition (Contradictory cycle)

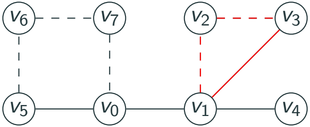
contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Theorem

φ_{EQ} is satisfiable iff its equality graph has no contradictory cycle

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Definition (Contradictory cycle)

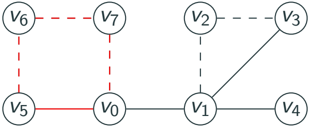
contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Theorem

φ_{EQ} is satisfiable iff its equality graph has no contradictory cycle

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Definition (Contradictory cycle)

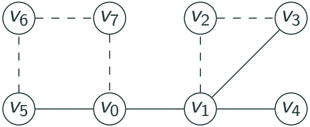
contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Theorem

φ_{EQ} is satisfiable iff its equality graph has no contradictory cycle

Example

$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



Definition (Contradictory cycle)

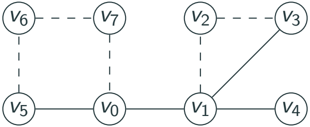
contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Theorem

φ_{EQ} is satisfiable iff its equality graph has no contradictory cycle

Example

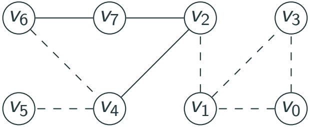
$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



unsatisfiable

Example

$v_0 = v_1 \quad v_0 = v_2 \quad v_1 = v_2 \quad v_1 = v_3 \quad v_2 \neq v_4 \quad v_4 = v_5 \quad v_4 = v_6 \quad v_6 \neq v_7 \quad v_7 \neq v_2$



Definition (Contradictory cycle)

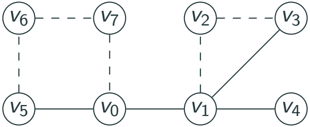
contradictory cycle is simple cycle in equality graph with one $E_{\neq}$ edge and all others $E_{=}$ edges

Theorem

φ_{EQ} is satisfiable iff its equality graph has no contradictory cycle

Example

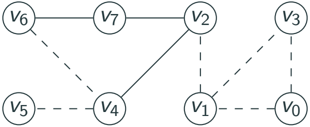
$v_0 \neq v_1 \quad v_0 \neq v_5 \quad v_1 = v_2 \quad v_1 \neq v_4 \quad v_1 \neq v_3 \quad v_2 = v_3 \quad v_5 = v_6 \quad v_6 = v_7 \quad v_7 = v_0$



unsatisfiable

Example

$v_0 = v_1 \quad v_0 = v_2 \quad v_1 = v_2 \quad v_1 = v_3 \quad v_2 \neq v_4 \quad v_4 = v_5 \quad v_4 = v_6 \quad v_6 \neq v_7 \quad v_7 \neq v_2$



satisfiable

Outline

- Summary of Last Week
- Deciding EQ: Equality Graphs
- Deciding EUF: Congruence Closure
- Correctness of DPLL(T)
- Some More Practical SMT

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

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

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

number of arguments

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

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V
- terms $\mathcal{T}(\mathcal{F}, V)$ are built according to grammar

$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \ldots, t}_{n})$$

if $x \in V$, c is constant, and $f \in \mathsf{F}$ has arity n

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V
- terms $\mathcal{T}(\mathcal{F}, V)$ are built according to grammar

$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \dots, t}_{n})$$

if $x \in V$, c is constant, and $f \in \mathsf{F}$ has arity n

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V
- terms $\mathcal{T}(\mathcal{F}, V)$ are built according to grammar

$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \dots, t}_{n})$$

if $x \in V$, c is constant, and $f \in \mathsf{F}$ has arity n

Example

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

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V
- terms $\mathcal{T}(\mathcal{F}, V)$ are built according to grammar

$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \dots, t}_{n})$$

  if $x \in V$, c is constant, and $f \in \mathrm{F}$ has arity n

- subterms

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

Example

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

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V
- terms $\mathcal{T}(\mathcal{F}, V)$ are built according to grammar

$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \dots, t}_{n})$$

  if $x \in V$, c is constant, and $f \in \mathrm{F}$ has arity n

- subterms

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

Example

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

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V
- terms $\mathcal{T}(\mathcal{F}, V)$ are built according to grammar

$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \dots, t}_{n})$$

  if $x \in V$, c is constant, and $f \in \mathsf{F}$ has arity n

- subterms

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

Example

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

Aim

build theory solver for theory of equality with uninterpreted functions (EUF)

Definitions (Terms)

- set of function symbols $\mathcal{F}$ with fixed arity
- set of variables V
- terms $\mathcal{T}(\mathcal{F}, V)$ are built according to grammar

$$t \quad ::= \quad x \mid c \mid f(\underbrace{t, \dots, t}_{n})$$

  if $x \in V$, c is constant, and $f \in \mathsf{F}$ has arity n

- subterms

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

Example

- for $\mathcal{F} = \{\mathsf{f}/1, \mathsf{g}/2, \mathsf{a}/0\}$ and $x, y \in V$ 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 $s = t$ (without variables, only constants)

Output: $s = t$ is *implied* ($E \vDash_{EUF} s = t$) or *not implied* ($E \nvDash_{EUF} s = t$)

Congruence Closure

Input: set of equations E and equation $s = t$ (without variables, only constants)

Output: $s = t$ is *implied* ($E \vDash_{EUF} s = t$) or *not implied* ($E \nvDash_{EUF} s = t$)

1. build congruence classes
   (a) collect all subterms of terms in $E \cup \{s = t\}$

   (b) put different subterms of $E \cup \{s = t\}$ in separate sets

Congruence Closure

Input: set of equations E and equation $s = t$ (without variables, only constants)

Output: $s = t$ is *implied* ($E \vDash_{EUF} s = t$) or *not implied* ($E \nvDash_{EUF} s = t$)

1. build congruence classes
   (a) collect all subterms of terms in $E \cup \{s = t\}$

   (b) put different subterms of $E \cup \{s = t\}$ in separate sets

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

Congruence Closure

Input: set of equations E and equation $s = t$ (without variables, only constants)

Output: $s = t$ is *implied* ($E \vDash_{EUF} s = t$) or *not implied* ($E \nvDash_{EUF} s = t$)

1. build congruence classes
   (a) collect all subterms of terms in $E \cup \{s = t\}$
   (b) put different subterms of $E \cup \{s = t\}$ in separate sets
   (c) merge sets $\{\dots, t_1, \dots\}$ and $\{\dots, t_2, \dots\}$ for all $t_1 = t_2$ in E
   (d) 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 $1 \leqslant i \leqslant n$

Congruence Closure

Input: set of equations E and equation $s = t$ (without variables, only constants)

Output: $s = t$ is *implied* ($E \vDash_{EUF} s = t$) or *not implied* ($E \nvDash_{EUF} s = t$)

1. build congruence classes
   (a) collect all subterms of terms in $E \cup \{s = t\}$
   (b) put different subterms of $E \cup \{s = t\}$ in separate sets
   (c) merge sets $\{\ldots, t_1, \ldots\}$ and $\{\ldots, t_2, \ldots\}$ for all $t_1 = t_2$ in E
   (d) 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 $1 \leqslant i \leqslant n$
   (e) repeat (d) until no change

Congruence Closure

Input: set of equations E and equation $s = t$ (without variables, only constants)

Output: $s = t$ is *implied* ($E \vDash_{EUF} s = t$) or *not implied* ($E \nvDash_{EUF} s = t$)

1. build congruence classes
   (a) collect all subterms of terms in $E \cup \{s = t\}$

   (b) put different subterms of $E \cup \{s = t\}$ in separate sets

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

   (d) 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 $1 \leqslant i \leqslant n$

   (e) repeat (d) until no change

1. if s and t belong to same set then return *implied* else return *not implied*

Example (1)

- given set of equations E

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

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

Example (1)

- given set of equations E

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

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

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

Example (1)

- given set of equations E

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

  and test equation $\mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$
- 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

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

  and test equation $\mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$
- 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

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

  and test equation $\mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$
- 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

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

  and test equation $\mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$
- 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

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

  and test equation $\mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$
- 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

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

  and test equation $f(a) = g(a)$
- 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

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

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

1. $\{\, \mathsf{a} \,\}$
2. $\{\, \mathsf{f(a)}, \mathsf{f(g(f(b)))} \,\}$
3. $\{\, \mathsf{b} \,\}$
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. $\{\, \mathsf{g(g(b))}, \mathsf{g(f(a))} \,\}$
13. $\{\, \mathsf{g(a)} \,\}$

Example (1)

- given set of equations E

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

  and test equation $\mathsf{f(a) = g(a)}$
- 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

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

  and test equation $\mathsf{f(a) = g(a)}$
- 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

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

  and test equation $\mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$
- 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. $\{\,\mathsf{g}(\mathsf{f}(\mathsf{b}))\,\}$

Example (1)

- given set of equations E

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

  and test equation $\mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$
- 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: $E \not\models_{EUF} \mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$

Example (2)

- given set of equations E

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

and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

$$f(f(f(a))) = a \qquad f(f(f(f(f(a))))) = a$$

  and test equaton $f(a) = a$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f(a)} = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f}(\mathsf{a}) = \mathsf{a}$

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

Example (2)

- given set of equations E

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

  and test equaton $\mathsf{f(a)} = \mathsf{a}$

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

- conclusion: $E \vDash_{EUF} \mathsf{f(a)} = \mathsf{a}$

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \ldots, x_n$.

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \ldots, x_n$.

Definition (Skolemization)

$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \ \ldots, x_n \mapsto c_n]$ where $c_1, \ldots, c_n$ are distinct fresh constants

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \ldots, x_n$.

Definition (Skolemization)

$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \ldots, x_n \mapsto c_n]$ where $c_1, \ldots, c_n$ are distinct fresh constants

Lemma

φ is EUF-satisfiable iff $\widehat{\varphi}$ is EUF-satisfiable

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \dots, x_n$.

Definition (Skolemization)

$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are distinct fresh constants

Lemma

φ *is EUF-satisfiable iff* $\widehat{\varphi}$ *is EUF-satisfiable*

Assumption

assume that $=$ is the only predicate in φ

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \dots, x_n$.

Definition (Skolemization)

$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are distinct fresh constants

Lemma

φ is EUF-satisfiable iff $\widehat{\varphi}$ is EUF-satisfiable

Assumption

assume that $=$ is the only predicate in φ

Remark

if φ contains n-ary predicate P different from equality:

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \dots, x_n$.

Definition (Skolemization)

$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are distinct fresh constants

Lemma

φ is EUF-satisfiable iff $\widehat{\varphi}$ is EUF-satisfiable

Assumption

assume that $=$ is the only predicate in φ

Remark

if φ contains n-ary predicate P different from equality:

- add new constant *true* and n-ary function f_P

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \dots, x_n$.

Definition (Skolemization)

$\widehat{\varphi} = \varphi[x_1 \mapsto c_1,\ \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are distinct fresh constants

Lemma

φ is EUF-satisfiable iff $\widehat{\varphi}$ is EUF-satisfiable

Assumption

assume that $=$ is the only predicate in φ

Remark

if φ contains n-ary predicate P different from equality:

- add new constant *true* and n-ary function f_P
- replace $P(t_1, \dots, t_n)$ by $f_P(t_1, \dots, t_n) = \mathit{true}$

Ok, But How About a Solver for EUF?

Assume conjunction of EUF literals φ with free variables $x_1, \dots, x_n$.

Definition (Skolemization)

$\widehat{\varphi} = \varphi[x_1 \mapsto c_1, \dots, x_n \mapsto c_n]$ where $c_1, \dots, c_n$ are distinct fresh constants

Lemma

φ is EUF-satisfiable iff $\widehat{\varphi}$ is EUF-satisfiable

Assumption

assume that $=$ is the only predicate in φ

Remark

if φ contains n-ary predicate P different from equality:

- add new constant *true* and n-ary function f_P
- replace $P(t_1, \dots, t_n)$ by $f_P(t_1, \dots, t_n) = \mathit{true}$
- replace $P(t_1, \dots, t_n)$ by $f_P(t_1, \dots, t_n) \neq \mathit{true}$

Assume conjunction of equations and inequalities φ with free variables $x_1, \ldots, x_n$.

Assume conjunction of equations and inequalities φ with free variables $x_1, \ldots, x_n$.

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

Assume conjunction of equations and inequalities φ with free variables $x_1, \dots, x_n$.

P is set of equations,
N is set of inequalities

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

Assume conjunction of equations and inequalities φ with free variables $x_1, \ldots, x_n$.

P is set of equations,
N is set of inequalities

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

$\varphi = (\bigwedge P) \wedge (\bigwedge N)$ EUF-unsatisfiable

Assume conjunction of equations and inequalities φ with free variables $x_1, \dots, x_n$.

P is set of equations,
N is set of inequalities

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

$\varphi = (\bigwedge P) \wedge (\bigwedge N)$	EUF-unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})$	EUF-unsatisfiable	skolemization

Assume conjunction of equations and inequalities φ with free variables $x_1, \dots, x_n$.

P is set of equations,
N is set of inequalities

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

$\varphi = (\bigwedge P) \wedge (\bigwedge N)$	EUF-unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})$	EUF-unsatisfiable	skolemization
$\iff \neg \Big((\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})\Big)$	EUF-valid	φ unsat iff $\neg\varphi$ valid

Assume conjunction of equations and inequalities φ with free variables $x_1, \dots, x_n$.

P is set of equations,
N is set of inequalities

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

$\varphi = (\bigwedge P) \wedge (\bigwedge N)$	EUF-unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})$	EUF-unsatisfiable	skolemization
$\iff \neg\Big((\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})\Big)$	EUF-valid	φ unsat iff $\neg\varphi$ valid
$\iff \bigwedge \widehat{P} \to \bigvee_{l \in \widehat{N}} \neg l$	EUF-valid	

Assume conjunction of equations and inequalities φ with free variables $x_1, \dots, x_n$.

P is set of equations,
N is set of inequalities

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

$\varphi = (\bigwedge P) \wedge (\bigwedge N)$	EUF-unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})$	EUF-unsatisfiable	skolemization
$\iff \neg\left((\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})\right)$	EUF-valid	φ unsat iff $\neg\varphi$ valid
$\iff \bigwedge \widehat{P} \to \bigvee_{l \in \widehat{N}} \neg l$	EUF-valid	
$\iff \exists\, s \neq t$ in $\widehat{N}$ such that $\bigwedge \widehat{P} \to s = t$ is EUF-valid		semantics of $\vee$

Assume conjunction of equations and inequalities φ with free variables $x_1, \dots, x_n$.

P is set of equations,
N is set of inequalities

Deciding satisfiability of set of EUF literals

split $\varphi = (\bigwedge P) \wedge (\bigwedge N)$ into positive literals P and negative literals N

$\varphi = (\bigwedge P) \wedge (\bigwedge N)$	EUF-unsatisfiable	
$\iff (\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N})$	EUF-unsatisfiable	skolemization
$\iff \neg \Big((\bigwedge \widehat{P}) \wedge (\bigwedge \widehat{N}) \Big)$	EUF-valid	φ unsat iff $\neg\varphi$ valid
$\iff \bigwedge \widehat{P} \to \bigvee_{l \in \widehat{N}} \neg l$	EUF-valid	
$\iff \exists\, s \neq t$ in $\widehat{N}$ such that $\bigwedge \widehat{P} \to s = t$ is EUF-valid		semantics of $\vee$
$\iff \exists\, s \neq t$ in $\widehat{N}$ such that $\bigwedge \widehat{P} \vDash_{EUF} s = t$		semantics of $\vDash_{EUF}$

Obtained Satisfiability Check

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

Example

1. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \land \mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}) \land \mathsf{c} \neq \mathsf{d}$

Obtained Satisfiability Check

$$(\bigwedge P) \wedge (\bigwedge N) \text{ unsatisfiable} \quad \iff \quad \exists\, s \neq t \text{ in } \widehat{N} \text{ such that} \bigwedge \widehat{P} \vDash_T s = t$$

Example

1. $\mathrm{g}(\mathrm{a}) = \mathrm{c} \wedge \mathrm{f}(\mathrm{g}(\mathrm{a})) \neq \mathrm{f}(\mathrm{c}) \wedge \mathrm{c} \neq \mathrm{d}$
   - split into $P = \{\mathrm{g}(\mathrm{a}) = \mathrm{c}\}$ and $N = \{\mathrm{f}(\mathrm{g}(\mathrm{a})) \neq \mathrm{f}(\mathrm{c}),\ \mathrm{c} \neq \mathrm{d}\}$

Obtained Satisfiability Check

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

Example

1. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}),\ \mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c} \vDash_T \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c})$, so unsatisfiable

Obtained Satisfiability Check

$$(\bigwedge P) \land (\bigwedge N) \text{ unsatisfiable} \quad \Longleftrightarrow \quad \exists\, s \neq t \text{ in } \widehat{N} \text{ such that } \bigwedge \widehat{P} \vDash_T s = t$$

Example

1. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \land \mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}) \land \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}),\ \mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c} \vDash_T \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \land \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}) \land \mathsf{g}(\mathsf{a}) = \mathsf{d} \land \mathsf{c} \neq \mathsf{d}$

Obtained Satisfiability Check

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

Example

1. $\mathrm{g(a) = c \wedge f(g(a)) \neq f(c) \wedge c \neq d}$
   - split into $P = \{\mathrm{g(a) = c}\}$ and $N = \{\mathrm{f(g(a)) \neq f(c),\ c \neq d}\}$
   - have $\mathrm{g(a) = c} \vDash_T \mathrm{f(g(a)) = f(c)}$, so unsatisfiable

2. $\mathrm{g(a) = c \wedge f(g(a)) = f(c) \wedge g(a) = d \wedge c \neq d}$
   - split into $P = \{\mathrm{g(a) = c,\ f(g(a)) = f(c),\ g(a) = d}\}$ and $N = \{\mathrm{c \neq d}\}$

Obtained Satisfiability Check

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

Example

1. $\mathsf{g(a) = c \wedge f(g(a)) \neq f(c) \wedge c \neq d}$
   - split into $P = \{\mathsf{g(a) = c}\}$ and $N = \{\mathsf{f(g(a)) \neq f(c),\ c \neq d}\}$
   - have $\mathsf{g(a) = c} \vDash_T \mathsf{f(g(a)) = f(c)}$, so unsatisfiable

2. $\mathsf{g(a) = c \wedge f(g(a)) = f(c) \wedge g(a) = d \wedge c \neq d}$
   - split into $P = \{\mathsf{g(a) = c,\ f(g(a)) = f(c),\ g(a) = d}\}$ and $N = \{\mathsf{c \neq d}\}$
   - have $\mathsf{g(a) = c, f(g(a)) = f(c), g(a) = d} \vDash_T \mathsf{c = d}$, so unsatisfiable

Obtained Satisfiability Check

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

Example

1. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}),\ \mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c} \vDash_T \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}) \wedge \mathsf{g}(\mathsf{a}) = \mathsf{d} \wedge \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}),\ \mathsf{g}(\mathsf{a}) = \mathsf{d}\}$ and $N = \{\mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}), \mathsf{g}(\mathsf{a}) = \mathsf{d} \vDash_T \mathsf{c} = \mathsf{d}$, so unsatisfiable

3. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{c} = \mathsf{d} \wedge \mathsf{f}(x) = x \wedge \mathsf{d} \neq \mathsf{g}(x) \wedge \mathsf{f}(x) \neq \mathsf{d}$
   - $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(x) = x\}$ and $N = \{\mathsf{d} \neq \mathsf{g}(x),\ \mathsf{f}(x) \neq \mathsf{d}\}$

Obtained Satisfiability Check

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

Example

1. $\mathrm{g}(\mathrm{a}) = \mathrm{c} \wedge \mathrm{f}(\mathrm{g}(\mathrm{a})) \neq \mathrm{f}(\mathrm{c}) \wedge \mathrm{c} \neq \mathrm{d}$
   - split into $P = \{\mathrm{g}(\mathrm{a}) = \mathrm{c}\}$ and $N = \{\mathrm{f}(\mathrm{g}(\mathrm{a})) \neq \mathrm{f}(\mathrm{c}),\ \mathrm{c} \neq \mathrm{d}\}$
   - have $\mathrm{g}(\mathrm{a}) = \mathrm{c} \vDash_T \mathrm{f}(\mathrm{g}(\mathrm{a})) = \mathrm{f}(\mathrm{c})$, so unsatisfiable

2. $\mathrm{g}(\mathrm{a}) = \mathrm{c} \wedge \mathrm{f}(\mathrm{g}(\mathrm{a})) = \mathrm{f}(\mathrm{c}) \wedge \mathrm{g}(\mathrm{a}) = \mathrm{d} \wedge \mathrm{c} \neq \mathrm{d}$
   - split into $P = \{\mathrm{g}(\mathrm{a}) = \mathrm{c},\ \mathrm{f}(\mathrm{g}(\mathrm{a})) = \mathrm{f}(\mathrm{c}),\ \mathrm{g}(\mathrm{a}) = \mathrm{d}\}$ and $N = \{\mathrm{c} \neq \mathrm{d}\}$
   - have $\mathrm{g}(\mathrm{a}) = \mathrm{c}, \mathrm{f}(\mathrm{g}(\mathrm{a})) = \mathrm{f}(\mathrm{c}), \mathrm{g}(\mathrm{a}) = \mathrm{d} \vDash_T \mathrm{c} = \mathrm{d}$, so unsatisfiable

3. $\mathrm{g}(\mathrm{a}) = \mathrm{c} \wedge \mathrm{c} = \mathrm{d} \wedge \mathrm{f}(x) = x \wedge \mathrm{d} \neq \mathrm{g}(x) \wedge \mathrm{f}(x) \neq \mathrm{d}$
   - $P = \{\mathrm{g}(\mathrm{a}) = \mathrm{c},\ \mathrm{c} = \mathrm{d},\ \mathrm{f}(x) = x\}$ and $N = \{\mathrm{d} \neq \mathrm{g}(x),\ \mathrm{f}(x) \neq \mathrm{d}\}$
   - skolemize $P = \{\mathrm{g}(\mathrm{a}) = \mathrm{c},\ \mathrm{c} = \mathrm{d},\ \mathrm{f}(\mathrm{e}) = \mathrm{e}\}$, $N = \{\mathrm{d} \neq \mathrm{g}(\mathrm{e}),\ \mathrm{f}(\mathrm{e}) \neq \mathrm{d}\}$

Obtained Satisfiability Check

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

Example

1. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}),\ \mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c} \vDash_T \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}) \wedge \mathsf{g}(\mathsf{a}) = \mathsf{d} \wedge \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}),\ \mathsf{g}(\mathsf{a}) = \mathsf{d}\}$ and $N = \{\mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}), \mathsf{g}(\mathsf{a}) = \mathsf{d} \vDash_T \mathsf{c} = \mathsf{d}$, so unsatisfiable

3. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{c} = \mathsf{d} \wedge \mathsf{f}(x) = x \wedge \mathsf{d} \neq \mathsf{g}(x) \wedge \mathsf{f}(x) \neq \mathsf{d}$
   - $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(x) = x\}$ and $N = \{\mathsf{d} \neq \mathsf{g}(x),\ \mathsf{f}(x) \neq \mathsf{d}\}$
   - skolemize $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e}\}$, $N = \{\mathsf{d} \neq \mathsf{g}(\mathsf{e}),\ \mathsf{f}(\mathsf{e}) \neq \mathsf{d}\}$
     - $\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e} \nvDash_T \mathsf{d} = \mathsf{g}(\mathsf{e})$

Obtained Satisfiability Check

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

Example

1. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}) \wedge \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}),\ \mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c} \vDash_T \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c})$, so unsatisfiable
2. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}) \wedge \mathsf{g}(\mathsf{a}) = \mathsf{d} \wedge \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}),\ \mathsf{g}(\mathsf{a}) = \mathsf{d}\}$ and $N = \{\mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}), \mathsf{g}(\mathsf{a}) = \mathsf{d} \vDash_T \mathsf{c} = \mathsf{d}$, so unsatisfiable
3. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \wedge \mathsf{c} = \mathsf{d} \wedge \mathsf{f}(x) = x \wedge \mathsf{d} \neq \mathsf{g}(x) \wedge \mathsf{f}(x) \neq \mathsf{d}$
   - $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(x) = x\}$ and $N = \{\mathsf{d} \neq \mathsf{g}(x),\ \mathsf{f}(x) \neq \mathsf{d}\}$
   - skolemize $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e}\}$, $N = \{\mathsf{d} \neq \mathsf{g}(\mathsf{e}),\ \mathsf{f}(\mathsf{e}) \neq \mathsf{d}\}$
     - $\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e} \nvDash_T \mathsf{d} = \mathsf{g}(\mathsf{e})$
     - $\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e} \nvDash_T \mathsf{f}(\mathsf{e}) = \mathsf{d}$

Obtained Satisfiability Check

$$(\bigwedge P) \land (\bigwedge N) \text{ unsatisfiable} \quad \Longleftrightarrow \quad \exists\, s \neq t \text{ in } \widehat{N} \text{ such that} \bigwedge \widehat{P} \vDash_T s = t$$

Example

1. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \land \mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}) \land \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c}\}$ and $N = \{\mathsf{f}(\mathsf{g}(\mathsf{a})) \neq \mathsf{f}(\mathsf{c}),\ \mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c} \vDash_T \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c})$, so unsatisfiable

2. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \land \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}) \land \mathsf{g}(\mathsf{a}) = \mathsf{d} \land \mathsf{c} \neq \mathsf{d}$
   - split into $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}),\ \mathsf{g}(\mathsf{a}) = \mathsf{d}\}$ and $N = \{\mathsf{c} \neq \mathsf{d}\}$
   - have $\mathsf{g}(\mathsf{a}) = \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{a})) = \mathsf{f}(\mathsf{c}), \mathsf{g}(\mathsf{a}) = \mathsf{d} \vDash_T \mathsf{c} = \mathsf{d}$, so unsatisfiable

3. $\mathsf{g}(\mathsf{a}) = \mathsf{c} \land \mathsf{c} = \mathsf{d} \land \mathsf{f}(x) = x \land \mathsf{d} \neq \mathsf{g}(x) \land \mathsf{f}(x) \neq \mathsf{d}$
   - $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(x) = x\}$ and $N = \{\mathsf{d} \neq \mathsf{g}(x),\ \mathsf{f}(x) \neq \mathsf{d}\}$
   - skolemize $P = \{\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e}\}$, $N = \{\mathsf{d} \neq \mathsf{g}(\mathsf{e}),\ \mathsf{f}(\mathsf{e}) \neq \mathsf{d}\}$
     - $\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e} \nvDash_T \mathsf{d} = \mathsf{g}(\mathsf{e})$
     - $\mathsf{g}(\mathsf{a}) = \mathsf{c},\ \mathsf{c} = \mathsf{d},\ \mathsf{f}(\mathsf{e}) = \mathsf{e} \nvDash_T \mathsf{f}(\mathsf{e}) = \mathsf{d}$

   so satisfiable

Outline

- Summary of Last Week
- Deciding EQ: Equality Graphs
- Deciding EUF: Congruence Closure
- Correctness of DPLL(T)
- Some More Practical SMT

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

Lemma

if $\parallel F \Longrightarrow_{\mathcal{D}}^{*} M \parallel G$ *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 $\mathcal{D}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Lemma

if $\parallel F \Longrightarrow_{\mathcal{D}}^{*} M \parallel G$ *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 $\mathcal{D}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Lemma
if $\| F \Longrightarrow^*_{\mathcal{D}} M \| G$ *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 $\mathcal{D}$ extends $\mathcal{B}$ by T-learn, T-forget, and restart

Lemma

if $\parallel F \Longrightarrow^*_{\mathcal{D}} M \parallel G$ *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 $M = M_0\ l_1^d\ M_1\ l_2^d\ M_2 \ldots\ l_k^d\ M_k$ with $l_1, \ldots, l_k$ all the decision literals*

Definition (Basic DPLL(T))

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

Definition (Full DPLL(T))

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

Lemma

if $\parallel F \Longrightarrow^*_{\mathcal{D}} M \parallel G$ *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:

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

Consider derivation with final state S_n:

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

Theorem

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

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{D}} M \| F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{D}} \mathsf{FailState}$, so $M \models \neg C$ for some C in F'

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| \, F \Longrightarrow_{\mathcal{D}}^{*} M \, \| \, F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{D}} \mathsf{FailState}$, so $M \models \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{D}} M \| F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{D}} \mathsf{FailState}$, so $M \models \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| \; F \Longrightarrow^*_{\mathcal{D}} M \; \| \; F' \overset{\text{fail}}{\Longrightarrow}_{\mathcal{D}} \text{FailState}$, so $M \models \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$
- also have $F' \models_T C$ because C is in F' and $F \equiv_T F'$ so T-inconsistent ■

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| F \implies^*_{\mathcal{D}} M \| F' \xRightarrow{\text{fail}}_{\mathcal{D}} \text{FailState}$, so $M \models \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$
- also have $F' \models_T C$ because C is in F' and $F \equiv_T F'$ so T-inconsistent ■

Theorem

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

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| \; F \Longrightarrow^*_{\mathcal{D}} M \; \| \; F' \stackrel{\text{fail}}{\Longrightarrow}_{\mathcal{D}} \text{FailState}$, so $M \models \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)
- by Lemma before, $F' \models_T M$, so $F' \models_T \neg C$
- also have $F' \models_T C$ because C is in F' and $F \equiv_T F'$ so T-inconsistent ■

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:

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

Theorem

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

Proof.

- must have $\| \; F \Longrightarrow_{\mathcal{D}}^{*} M \| F' \stackrel{\text{fail}}{\Longrightarrow}_{\mathcal{D}} \text{FailState}$, so $M \vDash \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)
- by Lemma before, $F' \vDash_T M$, so $F' \vDash_T \neg C$
- also have $F' \vDash_T C$ because C is in F' and $F \equiv_T F'$ so T-inconsistent ∎

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 $M \vDash \neg C$ (otherwise backjump or fail applicable)

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| F \implies^*_{\mathcal{D}} M \| F' \xrightarrow{\text{fail}}_{\mathcal{D}}$ FailState, so $M \vDash \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)
- by Lemma before, $F' \vDash_T M$, so $F' \vDash_T \neg C$
- also have $F' \vDash_T C$ because C is in F' and $F \equiv_T F'$ so T-inconsistent ∎

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 $M \vDash \neg C$ (otherwise backjump or fail applicable)
- so $M \vDash F'$ and by T-consistency $M \vDash_T F'$

Consider derivation with final state S_n:

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

Theorem

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

Proof.

- must have $\| F \Longrightarrow^*_{\mathcal{D}} M \| F' \stackrel{\text{fail}}{\Longrightarrow}_{\mathcal{D}} \text{FailState}$, so $M \vDash \neg C$ for some C in F'
- M cannot contain decision literals (otherwise T-backjump applicable)
- by Lemma before, $F' \vDash_T M$, so $F' \vDash_T \neg C$
- also have $F' \vDash_T C$ because C is in F' and $F \equiv_T F'$ so T-inconsistent ∎

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 $M \vDash \neg C$ (otherwise backjump or fail applicable)
- so $M \vDash F'$ and by T-consistency $M \vDash_T F'$
- have $F \equiv_T F'$ so M also T-satisfies F ∎

Theorem (Termination)

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

is finite if

Theorem (Termination)

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

is finite if

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

Theorem (Termination)

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

is finite if

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

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

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

Theorem (Termination)

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

is finite if

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

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

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

- *there are more* $\mathcal{B}$*-steps in* $S_j \Longrightarrow^*_{\mathcal{D}} S_k$ *than in* $S_i \Longrightarrow^*_{\mathcal{D}} S_j$*, or*

Theorem (Termination)

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

is finite if

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

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

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

Theorem (Termination)

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

is finite if

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

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

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

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) # some comparisons
psi = (a == If (b == c, b - 2, c - 4)) # if-then-else expression
print(phi)
solver = Solver()
solver.add(phi, psi) # assert constraints
solver.add(a + b < 2 * c) # arithmetic

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
```