universität innsbruck

SS 2024 lecture 5

Constraint Solving

René Thiemann and Fabian Mitterwallner

based on a previous course by Aart Middeldorp

Outline

1. **Summary of Previous Lecture**
2. **Equality Logic**
3. **Equality Logic with Uninterpreted Functions**
4. **EUF**
5. **Congruence Closure**
6. **Further Reading**

SMT Problem

decide satisfiability of (quantifier-free) formulas in

propositional logic + domain-specific background theories (axiomatic or concrete model)

Terminology

theory solver for T (T-solver) is procedure for deciding T-satisfiability of conjunction of quantifier-free literals

Remark

- SMT solvers often use DPLL(T) framework
- DPLL(T): combine DPLL-based SAT-solver with T-solver; the latter is used for
  - T-consistency checks – find model w.r.t. theory or generate blocking clause
  - T-propagation – find implied literals
  - basic implementation of T-propagation: $M \models_T l$ if $M \wedge \neg l$ is unsatisfiable

Outline

Theory of Equality

- signature: no function symbols, only one binary symbol $=$
- axioms
  - reflexivity $\forall x.\, x = x$
  - symmetry $\forall x\, y.\, x = y \to y = x$
  - transitivity $\forall x\, y\, z.\, x = y \land y = z \to x = z$

Example

$$y = z \;\land\; x = z \quad \lor \quad x \neq z \;\land\; x = y$$

Remark

assumption: infinite domain; consequence: $\bigwedge_{1 \leq i < j \leq n} x_i \neq x_j$ is satisfiable for all $n \in \mathbb{N}$
small model property: satisfiable formula φ with n variables has model with domain $\{1, \ldots, n\}$

Remark

equality logic can be extended by uninterpreted constants

- extend signature by constants $a, b, \ldots$
- uninterpreted: different constants can be interpreted as equal values or as different values
- no significant extension: constants can easily be removed
  - replace each constant a by new variable x_a
  - obtain equisatisfiable formula without constants
- example: $y = z \ \wedge \ b \neq z \ \vee \ a = b$ becomes $y = z \ \wedge \ x_b \neq z \ \vee \ x_a = x_b$

Remark

equality logic can be extended by constants in concrete domain

- extend signature by constants, e.g., from domain in real numbers
- concrete domain: different constants represent different values
- no significant extension: constants can easily be removed
  - replace each constant c_i $(1 \leq i \leq n)$ by new variable x_i
  - add constraint $x_i \neq x_j$ for all $1 \leq i < j \leq n$
  - obtain equisatisfiable formula without constants
- example: $y = z \ \wedge \ 2 \neq z \ \vee \ \sqrt{2} = 2$ becomes $(y = z \ \wedge \ x_2 \neq z \ \vee \ x_1 = x_2) \wedge x_1 \neq x_2$

Consequence

from now on consider equality logic without constants

Theorem

satisfiability problem for equality logic is NP-complete

Proof

- membership in NP

  guess assignment in $\{1, \ldots, n\}$ (small model property)
  where n is number of variables in formula and check correctness

- NP-hardness

  reduction from SAT

  - propositional formula φ with propositional atoms $p_1, \ldots, p_n$
  - introduce variables $x_1, \ldots, x_n, y_1, \ldots, y_n$
  - equality logic formula ψ is obtained from φ by replacing every p_i with $x_i = y_i$
  - φ is satisfiable $\iff$ ψ is satisfiable

Satisfiability Procedure for Conjunctive Fragment of Equality Logic

easy but important case: conjunction of equalities and disequalities φ

1. define equivalence class for each variable in φ
2. for each equality $x = y$ in φ

   merge equivalence classes that contain x and y
3. for each disequality $x \neq y$ in φ

   if x and y belong to same equivalence class, return unsatisfiable
4. return satisfiable

T-solver for equality logic

conjunction φ of equality logic literals over set of variables V

Definitions

- equality graph is undirected graph $G_=(\varphi) = (V, E_=, E_{\neq})$ with
  - $E_=$ edges corresponding to positive (equality) literals in φ
  - $E_{\neq}$ edges corresponding to negative (inequality) literals in φ
- contradictory cycle is cycle with exactly one $E_{\neq}$ edge
- contradictory cycle is simple if no node appears twice

Lemma

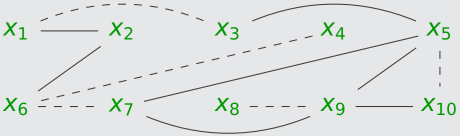
φ is satisfiable $\iff$ $G_=(\varphi)$ contains no simple contradictory cycles

Example

formula φ

$$x_1 = x_2 \;\wedge\; x_1 \neq x_3 \;\wedge\; x_3 = x_5 \;\wedge\; x_4 \neq x_6 \;\wedge\; x_6 \neq x_7 \;\wedge\; x_5 = x_9 \;\wedge$$
$$x_2 = x_6 \;\wedge\; x_5 = x_7 \;\wedge\; x_8 \neq x_9 \;\wedge\; x_9 = x_{10} \;\wedge\; x_7 = x_9 \;\wedge\; x_5 \neq x_{10}$$

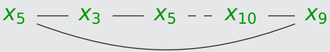
- equality graph $G_=(\varphi)$



- contradictory cycles

simple

simple



- φ is unsatisfiable

Outline

Aim

- further increase expressivity of logic
- one solution: add uninterpreted functions

Theory of Equality with Uninterpreted Symbols

- signature: function and predicate symbols, including binary symbol $=$
- axioms of equality logic, and the following ones
  - function congruence (for every n-ary function symbol f)

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

  - predicate congruence (for every n-ary predicate symbol P)

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

Quiz: Is the formula satisfiable?

- is formula

$$x = g(y,z) \ \wedge \ f(x) \neq f(g(y,z))$$

  satisfiable?

- model $\mathcal{M}$ with $\mathbb{N}$ as carrier:

$$\begin{aligned} f_{\mathcal{M}}(a) &= a+1 && \forall\, a \in \mathbb{N} \\ g_{\mathcal{M}}(a,b) &= 1 && \forall\, a,b \in \mathbb{N} \\ =_{\mathcal{M}} &= \{(a,b) \mid a = b \text{ or } a,b \in \{0,1\}\} \end{aligned}$$

- environment I: $\quad I(x) = I(y) = I(z) = 0$

Congruence axioms are essential!

$=_{\mathcal{M}}$ does not satisfy function congruence axiom $\forall x\, y.\, x = y \rightarrow f(x) = f(y)$

Remark

simplification: predicate symbols can be eliminated

- add fresh constant $\bullet$
- add fresh n-ary function symbol f_P for each predicate symbol P of arity n
- replace every atomic formula $P(t_1, \ldots, t_n)$ by $f_P(t_1, \ldots, t_n) = \bullet$

Example

formula

$$P \land Q(x) \land \neg R(x,y) \land x = z \to R(x,z)$$

is transformed into

$$f_P = \bullet \land f_Q(x) = \bullet \land f_R(x,y) \neq \bullet \land x = z \to f_R(x,z) = \bullet$$

Theorem

satisfiability in theory of equality with uninterpreted functions is undecidable

Proof

reduction from PCP (Post correspondence problem) instance $P \subseteq \Gamma^+ \times \Gamma^+$

- constant e, unary function symbol a for all $a \in \Gamma$, binary predicate symbol Q
- if $\alpha = a_1 a_2 \cdots a_n$ then $\alpha(t)$ denotes $a_n(\cdots(a_2(a_1(t)))\cdots)$
- formula in theory of equality with uninterpreted functions

$$\bigwedge_{(\alpha,\beta)\in P} Q(\alpha(e),\beta(e)) \wedge \left(\forall v\, w. Q(v,w) \rightarrow \bigwedge_{(\alpha,\beta)\in P} Q(\alpha(v),\beta(w))\right) \rightarrow \exists z.\, Q(z,z)$$

is valid $\iff$ P has solution

Outline

1. Summary of Previous Lecture
2. Equality Logic
3. Equality Logic with Uninterpreted Functions
4. **EUF**
5. Congruence Closure
6. Further Reading

Definition

EUF: quantifier-free fragment of equality logic with uninterpreted function symbols

Examples

- $x_1 \neq x_2 \ \vee \ f(x_1) = f(x_2) \ \vee \ f(x_1) \neq f(x_3)$
- $x_1 = x_2 \ \rightarrow \ f(f(g(x_1, x_2))) = f(g(x_2, x_1))$

Examples

- $a \neq b \ \wedge \ f(a) = f(b)$ EUF-consistent
- $a = f(b) \ \wedge \ b = f(a) \ \wedge \ f(b) \neq f(f(f(b)))$ not EUF-consistent
- $a = b \vDash_{\text{EUF}} f(a) = f(b)$
- $a = b \not\vDash_{\text{EUF}} f(a) = f(b)$

Remark

- for satisfiability it does not matter whether one chooses variables or constants
- example: $\mathsf{a} = \mathsf{f}(y)$ is equisatisfiable to $\mathsf{a} = \mathsf{f}(\mathsf{c}_y)$ and to $x_\mathsf{a} = \mathsf{f}(y)$
- consequence: we use EUF restricted to ground terms, i.e., terms without variables

Remark

- SMT solvers are often used to validate certain consequences
- example: $eq_1 \wedge eq_2 \rightarrow eq_3$ (for universally quantified variables)
- therefore prove unsatisfiability of $eq_1 \wedge eq_2 \wedge \neg eq_3$ (for existentially quantified variables)
- consequence: ability of SMT solvers to prove unsat is essential

Example

- two C functions computing $x \mapsto x^3$

```
int power3(int in) {
    int i, out;
    out = in;
    for (i = 0; i < 2; i++)
      out = out * in;
    return out;
}
```

```
int power3_new(int in) {
    int out;
    out = (in * in) * in;
    return out;
}
```

- are these functions equivalent?

$$\varphi_a : \ \mathtt{out}_a^0 = \mathtt{in} \wedge \mathtt{out}_a^1 = g(\mathtt{out}_a^0, \mathtt{in}) \wedge \mathtt{out}_a^2 = g(\mathtt{out}_a^1, \mathtt{in})$$

$$\varphi_b : \ \mathtt{out}_b^0 = g(g(\mathtt{in}, \mathtt{in}), \mathtt{in})$$

$$\varphi_a \wedge \varphi_b \ \rightarrow \ \mathtt{out}_a^2 = \mathtt{out}_b^0$$

- simplify problem by substituting uninterpreted function g for $*$

SMT-LIB 2 Format for EUF

EUF formula $f(f(a)) = a \land f(a) = b \land a \neq b$

```
(declare-sort A)
(declare-const a A)
(declare-const b A)
(declare-fun f (A) A)
(assert (= (f (f a)) a))
(assert (= (f a) b))
(assert (distinct a b))
(check-sat)
(get-model)
```

- terms are sorted
- `declare-const x S`
  creates variable x of sort S
- `declare-fun f (S1 ... Sn) T`
  creates uninterpreted function $f\colon S_1 \times \cdots \times S_n \to T$
- prefix notation for terms and equations
- `(distinct x y)` is equivalent to `not (= x y)`

Outline

Congruence Closure (core algorithm for T-Solver of EUF)

input: set E of ground equations and ground equation $s \approx t$

output: valid ($E \vDash_{EUF} s = t$) or invalid ($E \nvDash_{EUF} s = t$)

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

     $$\{\ldots, f(s_1, \ldots, s_n), \ldots\} \text{ and } \{\ldots, f(t_1, \ldots, t_n), \ldots\}$$

     if s_i and t_i belong to same set for all $1 \leqslant i \leqslant n$
2. if s and t belong to same set then return valid else return invalid

Example (1)

- set of equations E

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

  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: $E \nvDash_{EUF} \mathsf{f}(\mathsf{a}) = \mathsf{g}(\mathsf{a})$

Example (2)

- set of equations E

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

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

- sets

$$1.\ \{\mathsf{a}, \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))), \mathsf{f}(\mathsf{f}(\mathsf{a})), \mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{a}))))\}$$

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

Outline

Kröning and Strichmann

- Chapter 4
- Section 11.3

Bradley and Manna

- Sections 9.1 and 9.2

Important Concepts

- congruence closure
- contradictory cycle
- equality graph
- equality logic
- EUF
- uninterpreted function