AAA528: Computational Logic

Lecture 8 — Decision Procedures for Theory of Equality

Hakjoo Oh
2018 Fall

Goal

Decision procedures for deciding satisfiability in theory of equality.

- quantifier-free fragment (otherwise, undecidable)
- conjunctions of literals (no disjunctions)
- no predicate symbols

Theory of Equality (with Uninterpreted Functions)

The theory of equality T_E is the simplest and most widely-used first-order theory. Its signature

$$\Sigma_E : \{=, a, b, c, \ldots, f, g, h, \ldots, p, q, r, \ldots\}$$

consists of

- $=$ (equality), a binary predicate;
- and all constant, function, and predicate symbols.

Equality $=$ is an **interpreted** predicate symbol; its meaning will be defined via the axioms. The others are **uninterpreted** since functions, predicates, and constants are left unspecified.

Theory of Equality (with Uninterpreted Functions)

The axioms of T_E:

1. Reflexivity: $\forall x.\ x = x$
2. Symmetry: $\forall x, y.\ x = y \implies y = x$
3. Transitivity: $\forall x, y, z.\ x = y \land y = z \implies x = z$
4. Function congruence (consistency): for each positive integer n and n-ary function symbol f,

$$\forall \vec{x}, \vec{y}.\ (\bigwedge_{i=1}^{n} x_i = y_i) \rightarrow f(\vec{x}) = f(\vec{y}).$$

5. Predicate congruence (consistency): for each positive integer n and n-ary predicate symbol p,

$$\forall \vec{x}, \vec{y}.\ (\bigwedge_{i=1}^{n} x_i = y_i) \rightarrow (p(\vec{x}) \leftrightarrow p(\vec{y})).$$

Examples

Decide satisfiability of formulas:

- $f(x) = f(y) \land x \neq y$
- $x = y \land f(x) \neq f(y)$
- $f(f(f(a))) = a \land f(f(f(f(f(a))))) = a \land f(a) \neq a$

Eliminating Predicates

- Simple reduction of formulas with uninterpreted predicates to equisatisfiable formulas without predicates other than $=$.
- For example, the formulas

$$x = y \rightarrow (p(x) \leftrightarrow p(y))$$

  is transformed into

$$x = y \rightarrow ((f_p(x) = \bullet) \leftrightarrow (f_p(y) = \bullet))$$

  where $\bullet$ is a fresh constant and f_p is a fresh function.
- Exercise:

$$p(x) \wedge q(x, y) \wedge q(y, z) \rightarrow \neg q(x, z)$$

Congruence Relations

- A binary relation R over a set S is an equivalence relation if it is
  - reflexive: $\forall s \in S.\ sRs$
  - symmetric: $\forall s_1, s_2 \in S.\ s_1 R s_2 \rightarrow s_2 R s_1$
  - transitive: $\forall s_1, s_2, s_3 \in S.\ s_1 R s_2 \wedge s_2 R s_3 \rightarrow s_1 R s_3$
- A binary relation R over set S equipped with functions $F = \{f_1, \ldots, f_n\}$ is a congruence relation if it equivalence relation and obeys congruence: for every n-ary function $f \in F$,

$$\forall \vec{s}, \vec{t}.\ \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \rightarrow f(\vec{s}) R f(\vec{t})$$

Examples

- Which of these are equivalence relations?
  - $\equiv_2$ over $\mathbb{Z}$
  - $\geq$ over $\mathbb{N}$
  - $R(x, y)$ defined as $|x| = |y|$ over $\mathbb{R}$
- Which of these are congruence relations?
  - $=$ over $\mathbb{N}$ equipped with successor function
  - $\equiv_2$ over $\mathbb{N}$ equipped with successor function
  - $R(x, y)$ defined as $|x| = |y|$ over $\mathbb{R}$ equipped with successor function

Classes and Partitions

- For an equivalence relation R over a set S, the equivalence class of $s \in S$ under R is defined as follows:

$$[s]_R = \{s' \in S \mid sRs'\}$$

- If R is a congruence relation, $[s]_R$ is the congruence class of s.
- What is the equivalence class of 3 under $\equiv_2$?
- A partition P of S is a set of subsets of S such that $\bigcup_{S' \in P} S' = S$ (total) and $\forall S_1, S_2 \in P.\ S_1 \neq S_2 \rightarrow S_1 \cap S_2 = \emptyset$ (disjoint).
- The quotient S/R of S by the equivalence (congruence) relation R is a partition of S: it is a set of equivalence (congruence) classes

$$S/R = \{[s]_R \mid s \in S\}$$

- What is $\mathbb{Z}/\equiv_2$?

Equivalence / Congruence Closure

- The equivalence closure R^E of the binary relation R over S is the equivalence relation such that
  - $R \subseteq R^E$
  - for all other equivalence relation R' such that $R \subseteq R'$, $R^E \subseteq R'$

  That is, R^E is the smallest equivalence relation that includes R.
- What is the equivalence closure of $R = \{(a, b), (b, c), (d, d)\}$ over $S = \{a, b, c, d\}$?
- The congruence closure R^C of the binary relation R over S is the congruence relation such that
  - $R \subseteq R^C$
  - for all other congruence relation R' such that $R \subseteq R'$, $R^C \subseteq R'$
- What is the congruence closure of $R = \{(a, b)\}$ over $S = \{a, b, c\}$ equipped with function f such that $f(a) = b, f(b) = c, f(c) = c$?

Satisfiability in terms of Congruence Closure

- The subterm set S_F of formula F is the set that contains the subterms of F.
- What is S_F for $F : f(a, b) = a \wedge f(f(a, b), b) \neq a$?
- We define satisfiability of F in terms of congruence closure over S_F.
- The formula F

$$F : s_1 = t_1 \wedge \cdots \wedge s_m = t_m \wedge s_{m+1} \neq t_{m+1} \wedge \cdots \wedge s_n \neq t_n$$

  is satisfiable iff the congruence closure $\sim$ of R_F satisfies $s_i \not\sim t_i$ for each $i \in [m + 1, n]$, where $R_F = \{(s_i, t_i) \mid 1 \leq i \leq m\}$.

Congruence Closure Algorithm

To decide the satisfiability of F

$$F : s_1 = t_1 \wedge \cdots \wedge s_m = t_m \wedge s_{m+1} \neq t_{m+1} \wedge \cdots \wedge s_n \neq t_n$$

perform the following steps:

1. Construct the congruence closure $\sim$ of $R_F = \{s_1 = t_1, \ldots, s_m = t_m\}$ over the subterm set S_F.
2. If $s_i \sim t_i$ for any $i \in \{m+1, \ldots, n\}$, F is unsatisfiable.
3. Otherwise, F is satisfiable.

Computing Congruence Closure

Constructing the congruence closure $\sim$ of $R_F = \{s_1 = t_1, \ldots, s_m = t_m\}$ over the subterm set S_F is done as follows:

- Initially, begin with the finest congruence relation $\sim_0$ given by the partition:

  $$\{\{s\} \mid s \in S_F\}$$

  in which each term of S_F is its own congruence class.

- For each $i \in \{1, \ldots, m\}$, impose $s_i = t_i$ by merging the congruence classes

  $$[s_i]_{\sim_{i-1}} \text{ and } [t_i]_{\sim_{i-1}}$$

  to form a new congruence relation $\sim_i$. To accomplish this merging, first form the union of them and then propagate any new congruences that arise within this union.

Examples

- $f(a, b) = a \land f(f(a, b), b) \neq a$
- $f(f(f(a))) = a \land f(f(f(f(f(a))))) = a \land f(a) \neq a$
- $f(x) = f(y) \land x \neq y$