Decision procedures for equality logic with uninterpreted functions

Olga Tveretina

2.12.2009
Outline of the lecture:

• Ackerman reduction

• DPLL for Equality Logic with Uninterpreted Functions

• OBDD for Equality Logic with Uninterpreted Functions
EUF Logic or Equality Logic?

- It is possible to transform an EUF formula to Equality Logic formula
- To enforce the property of functional consistency
- Ackermann's reduction and Bryant's reduction
Ackermann's reduction

**Given an EUF formula F1**

- An equality formula $F2 := FC \implies Flat$
- $FC$ is a conjunction of functional-consistency constraints
- $Flat$ is a flattening of F1
- $F1$ is valid iff $F2$ is valid
Ackermann's reduction: Example

\[(x_1 =/= x_2) \lor F(x_1) = F(x_2) \lor F(x_1) =/= F(x_3)\]

- **Flat** := \( (x_1 =/= x_2) \lor (f_1 = f_2) \lor (f_1 =/= f_3) \)

- **FC** := \( (x_1 = x_2 \Rightarrow f_1 = f_2) \land (x_1 = x_3 \Rightarrow f_1 = f_3) \land (x_2 = x_3 \Rightarrow f_2 = f_3) \)
Ackermann's reduction: example

\[ x_1 = x_2 \rightarrow F(F(G(x_1))) = F(F(G(x_2))) \]

- \( g_1 = G(x_1), g_2 = G(x_2) \)
- \( f_1 = F(G(x_1)), f_2 = F(F(G(x_1))), f_3 = F(G(x_2)), f_4 = F(F(G(x_2))) \)

- **Flat** := \( x_1 = x_2 \rightarrow f_2 = f_4 \)
- **FC** := \( x_1 = x_2 \rightarrow g_1 = g_2 \land \)
  \( g_1 = f_1 \rightarrow f_1 = f_2 \land \)
  \( g_1 = g_2 \rightarrow f_1 = f_3 \land \)
  \( g_1 = f_3 \rightarrow f_1 = f_4 \land \)
  \( f_1 = g_2 \rightarrow f_2 = f_3 \land \)
  \( f_1 = f_3 \rightarrow f_2 = f_4 \land \)
  \( g_2 = f_3 \rightarrow f_3 = f_4 \)
EUF Decision Problem

Task

Determine whether formula $F$ is universally valid

- True for all interpretations of variables and function symbols
- Often expressed as (un)satisfiability problem

- Prove that formula $\neg F$ is not satisfiable

$x = y \rightarrow f(x) = f(y)$ is valid

$x = y \land f(x) = f(y)$ is satisfiable
Inference challenges for EUF

- Want to establish, for example, that $f(f(a,b),b) = a$ follows from $f(a,b) = a$

- Or that $f(f(f(a))) = a$ and $f(f(f(f(f(a))))) = a$ follow from $f(a) = a$

- These kinds of inferences are often required to perform program verification
Axioms of EUF

- Intuition behind decision procedure for EUF: repeatedly apply these axioms to infer new equalities

\[
\begin{align*}
a &= b \\
b &= c \\
\hline
\text{TRANS} \\
a &= c
\end{align*}
\]

\[
\begin{align*}
a_1 &= b_1 \\
a_2 &= b_2 \\
\vdots \\
a_n &= b_n \\
\hline
\text{EQ-PROP} \\
f(a_1, a_2, \ldots, a_n) &= f(b_1, b_2, \ldots, b_n)
\end{align*}
\]
SAT/BDDs and beyond

Propositional Logic

- BDDs
  - Symbolic, Canonical
  - Space intensive
  - Small problems
  - All Solutions

- SAT (resolution, DPLL)
  - Constraint-based
  - Large problems
  - One solution

?: Extend to more expressive systems/logics.

The EUF-logic?
DPLL procedure:

- **Davis, Logemann, Loveland, 1962:** “splitting rule”
  
  - **Input:** a formula in conjunctive normal form (CNF)
  - **Select an atom** $A$
  - **Split into cases** $A$ and $\neg A$
  - **In each case, simplify according new information**
  - **Output:** “satisfiable” or “unsatisfiable”
DPLL for propositional logic:

Is CNF $F$ satisfiable?

Is $F \land a$ satisfiable?

To simplify $F \land a$

Criteria to close branches

Is $F \land \neg a$ satisfiable?

To simplify $F \land \neg a$
Reduction rules for EUF

A unit clause \( s = t \) is not propagated in \( F \) if:

- \( s = t \) is contained in \( F \)
- \( s \) and \( t \) are contained in terms of \( F \{s = t\} \)

Example: \( a = f(b) \land g(a) = f(f(b)) \)
Reduction rules for EUF

• Remove \( t=\neq t \) from all clauses
• Remove clauses containing \( t=t \)
• \( s=t \land F \) to replace with \( s=t \land F[s:=t] \) if \( s \) is not in \( \text{Term}(t) \)
Splitting rule for EUF-DPLL

• For a CNF F, \texttt{Core}(F) is the set of positive clauses of length at least 2

• Choose a literal \( s=t \) contained in \texttt{Core}(F)

• Propagate it in \( s=t \land F \)
Splitting rule for EUF-DPLL

Example:

- $F$: $(x = y \lor y = z) \land f(x) = f(z)$

- Splitting on $f(x) = f(z)$ leads to non-terminating derivation

- Splitting on a literal contained in Core always leads to a terminating derivation
SAT criterion for EUF

Theorem 1:
Let a CNF F contains no purely positive clauses. Then F is satisfiable.

Proof:

• No purely positive clauses, hence, each clause contains at least one negative clause

• Assign different values to all terms in negative clauses
SAT criterion for EUF

Theorem 2 (satisfiability criterion):

Let a CNF $F$ be reduced, does not contain an empty clause and $\text{Core}(F)$ is empty. Then $F$ is satisfiable.

Proof:

- Each clause of length more than one contains at least one negative literal.
- All unit clauses are propagated.
- Vertex represents decision
- Follow green (dashed) line for value 0
- Follow red (solid) line for value 1
- Function value determined by leaf value
Variable ordering

- Assign arbitrary total ordering to variables
  - e.g., $x_1 < x_2 < x_3$
- Variables must appear in ascending order along all paths
Reduction rule: MERGE

Identify and share identical subtrees
Reduction rule: ELIMINATE

Remove nodes whose left and right child are identical
Reduced Ordered BDD

• Canonical representation of Boolean function (for given variable ordering)
• Two functions equivalent if and only if graphs isomorphic: can be tested in linear time
• Tautology checking
BDDs for EUF: deficiencies of approaches based on congruence closure

- *Not* all paths are *consistent*
- *Not canonical* representation
- To check consistency of all paths - > constraint solver can be invoked *exponentially many times* because of the Boolean structure of the formula

?: Construct an ordered EUF-BDD in which *all paths are consistent* by construction
BDDs for EUF: ordering on equalities

- $t < f(t)$
- $s = t$
- $s = t < u = v$

- $s > t$, orientation of equalities
- Total, w.f. order on terms

- Order on equalities:
  - $s < u$ or $s \equiv u$ and $t < v$
Reduction rules: the propositional structure of a formula

1) 
\[
\text{ITE}(e, T, T) \Rightarrow T
\]

2) 
\[
\text{ITE}(e, T_1, \text{ITE}(e, T_2, T_3)) \Rightarrow \text{ITE}(e, T_1, T_3)
\]
Reduction rules: the propositional structure of a formula

ITE(e₁,T₁,ITE(e₂,T₂,T₃)) \Rightarrow ITE(e₂,ITE(e₁,T₁,T₂),ITE(e₁,T₁,T₃))
Reduction rules: rewrite rules

\( t = s \)

\( s > t \)

\[ \text{ITE}(t=s, T_1, T_2) \Rightarrow \text{ITE}(s=t, T_1, T_2) \]

\( s = t \)

\[ \text{ITE}(s=t, T_1, T_2) \Rightarrow \text{ITE}(s=t, T_1[t], T_2) \]
Example:

\[(x=y \land y=z) \rightarrow f(x)=f(z)\]
**EUF-ROBDDs**

- *Nodes* are labeled with *equalities*
- *Rewriting rules* are always *terminating*
- *Tautology* is represented by “1”
- *Contradiction* is represented by “0”
- Checking equivalence of two boolean functions -> comparing their ROBDDs
- *Canonicity* of EUF-BDDs is *lost*
  
  - $\varphi$ and $\psi$ are equivalent if $\varphi \leftrightarrow \psi$ is represented by “1”
Thanks you for attention!