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 => 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

(x1 =/= x2) \/ F(x1)= F(x2) \/ F(x1)=/= F(x3)

- Flat := (x1 =/= x2) \/ (f1 = f2) \/ (f1 =/= f3)
- FC := (x1 = x2 => f1 = f2) /\
  (x1 = x3 => f1 = f3) /\
  (x2 = x3 => f2 = f3)

Ackermann's reduction: example

$x1=x2 \rightarrow F(F(G(x1))) = F(F(G(x2)))$

- $g1=G(x1)$, $g2=G(x2)$
- $f1=F(G(x1))$, $f2=F(F(G(x1)))$, $f3=F(G(x2))$, $f4=F(F(G(x2)))$

- Flat := $x1=x2 \rightarrow f2 = f4$
- FC :=

$$
\begin{aligned}
& x1=x2 \rightarrow g1=g2 \quad \wedge \\
& g1=f1 \rightarrow f1=f2 \quad \wedge \\
& g1=g2 \rightarrow f1=f3 \quad \wedge \\
& g1=f3 \rightarrow f1=f4 \quad \wedge \\
& f1=g2 \rightarrow f2=f3 \quad \wedge \\
& f1=f3 \rightarrow f2=f4 \quad \wedge \\
& g2=f3 \rightarrow f3=f4
\end{aligned}
$$

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

$$x=y \rightarrow f(x) = f(y) \text{ is valid}$$

$$x=y \wedge f(x) = f(y) \text{ is satisfiable}$$

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

$$\frac{a = b \quad b = c}{a = c}\ \textbf{TRANS}$$

$$\frac{a_1 = b_1 \quad a_2 = b_2 \quad \ldots \quad a_n = b_n}{f(a_1, a_2, \ldots, a_n) = f(b_1, b_2, \ldots, b_n)}\ \textbf{EQ-PROP}$$

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

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 $F \wedge a$ satisfiable?

Is $F \wedge \neg a$ satisfiable?

To simplify $F \wedge a$

To simplify $F \wedge \neg a$

Criteria to close branches

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 $F \backslash \{s=t\}$

Example: $a=f(b) \land g(a)=f(f(b))$

Reduction rules for EUF

- Remove $t \neq t$ from all clauses
- Remove clauses containing $t=t$
- $s=t \wedge F$ to replace with $s=t \wedge F[s:=t]$ if s is not in Term(t)

Splitting rule for EUF-DPLL

- For a CNF F, Core(F) is the set of positive clauses of length at least 2
- Choose a literal s=t contained in Core(F)
- Propagate it in s=t /\ F

Splitting rule for EUF-DPLL

Example:

- F: (x=y \/ y=z) /\ 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

Theorem1:

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

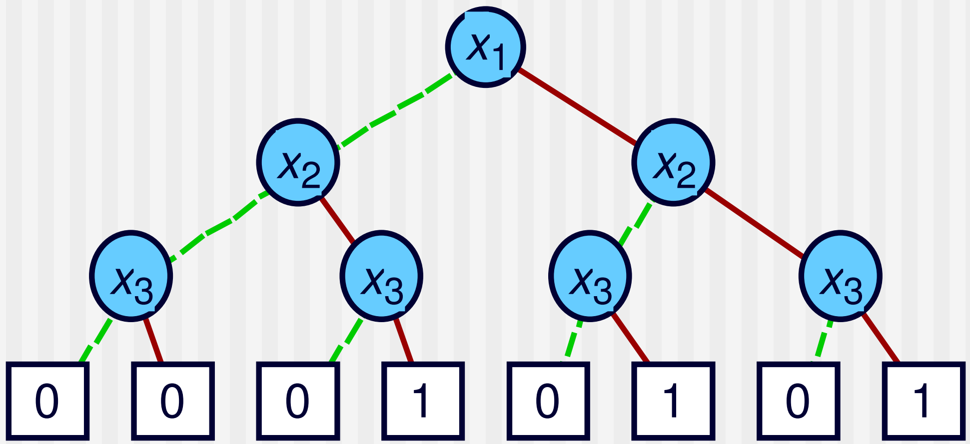
Theorem2 (satisfiability criterion):

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

Proof:

- Each clause of length mote than one contains at least one negative literal.
- All unit clauses are propagated

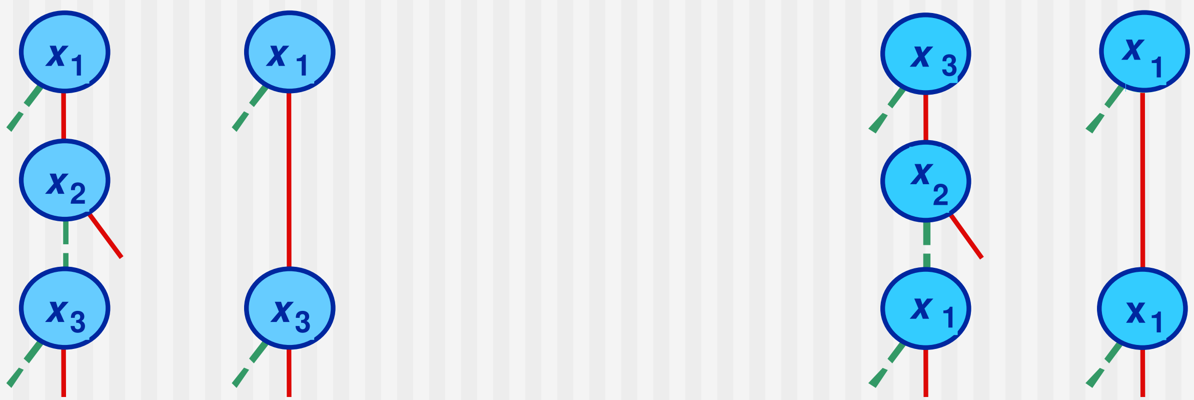
Binary decision diagram (BDD)



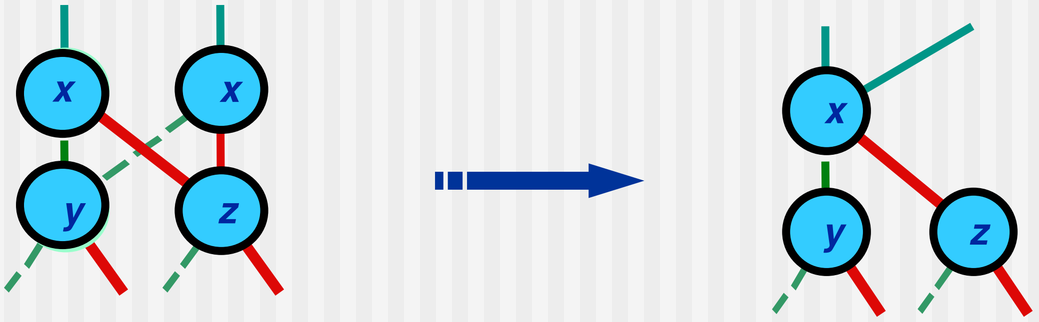
- Vertex represents decision
- Follow green (dashed) line for value 0
- Follow red (solid) line for value 1
- Function value determined by leaf value

Variable ordering

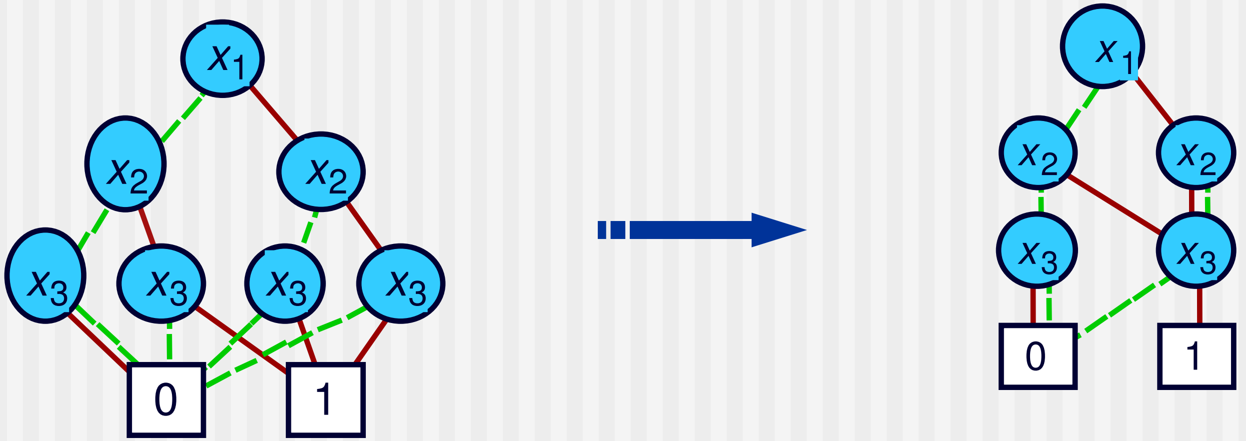
- Assign arbitrary total ordering to variables
  - e.g., $x_1 < x_2 < x_3$
- Variables must appear in ascending order along all paths



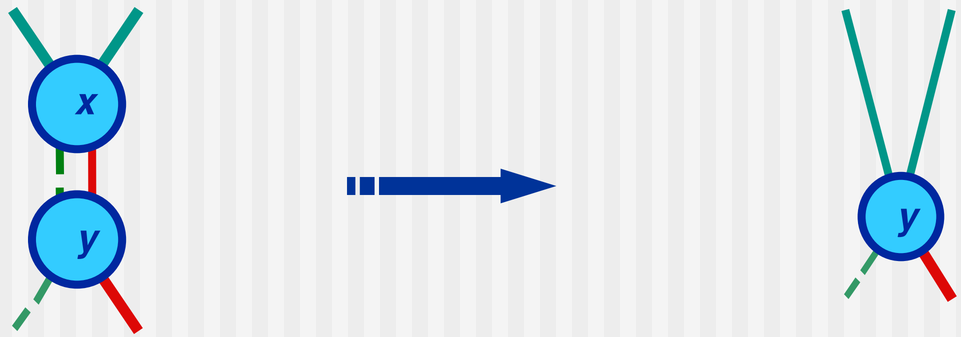
Reduction rule: MERGE



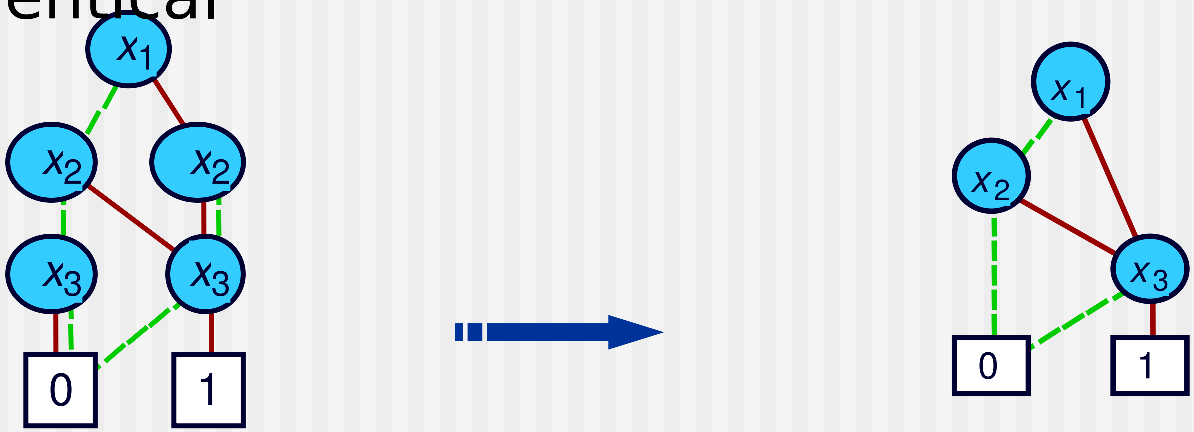
Identify and share identical subtrees



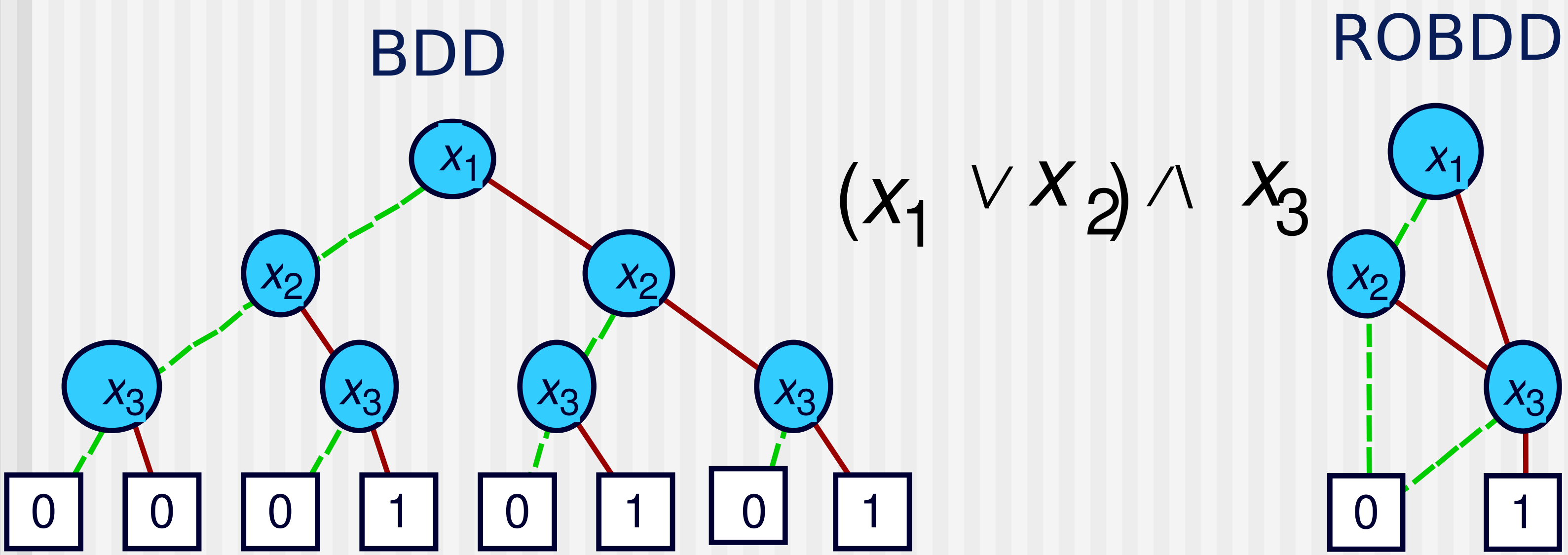
Reduction rule: ELIMINATE



Remove nodes whose left and right child are identical



Reduced Ordered BDD



$$(x_1 \vee x_2) \wedge x_3$$

- 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*

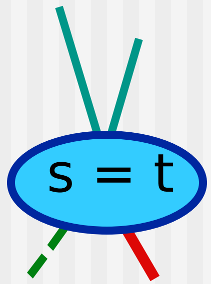
- *Not canonical* representation

- 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) → total, w.f. order on terms



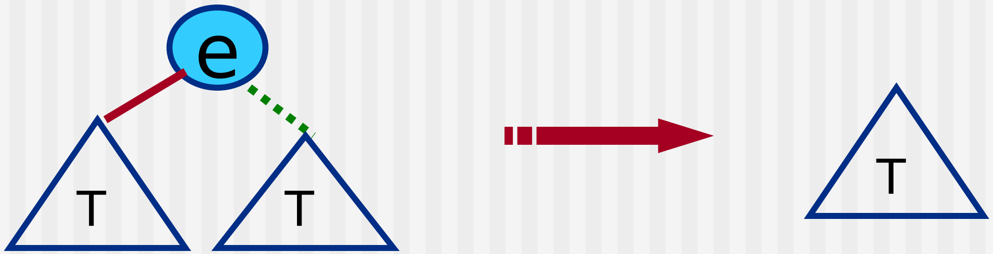
→ s > t, orientation of equalities

s=t < u=v → order on equalities:

s < u or s ≡ u and t < v

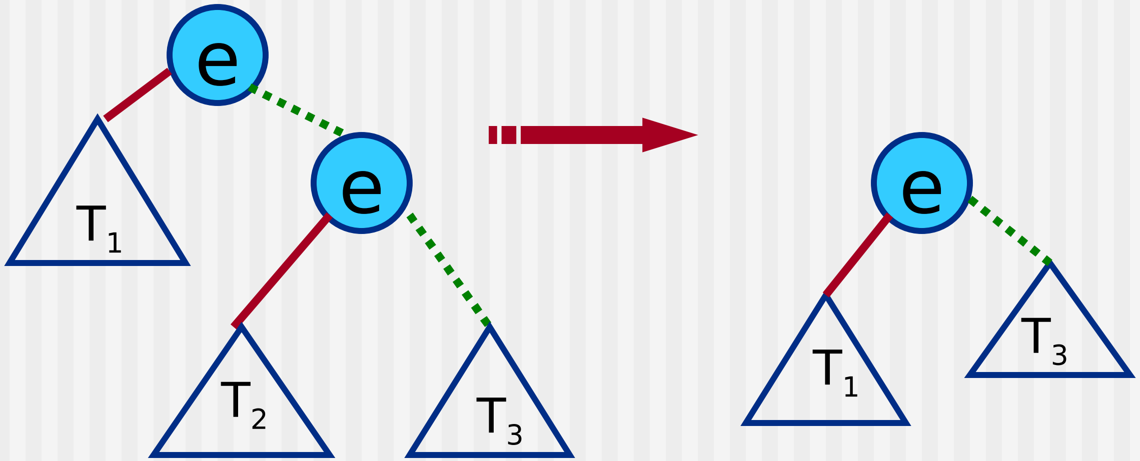
Reduction rules: the propositional structure of a formula

1)



ITE(e,T,T) ⇒ T

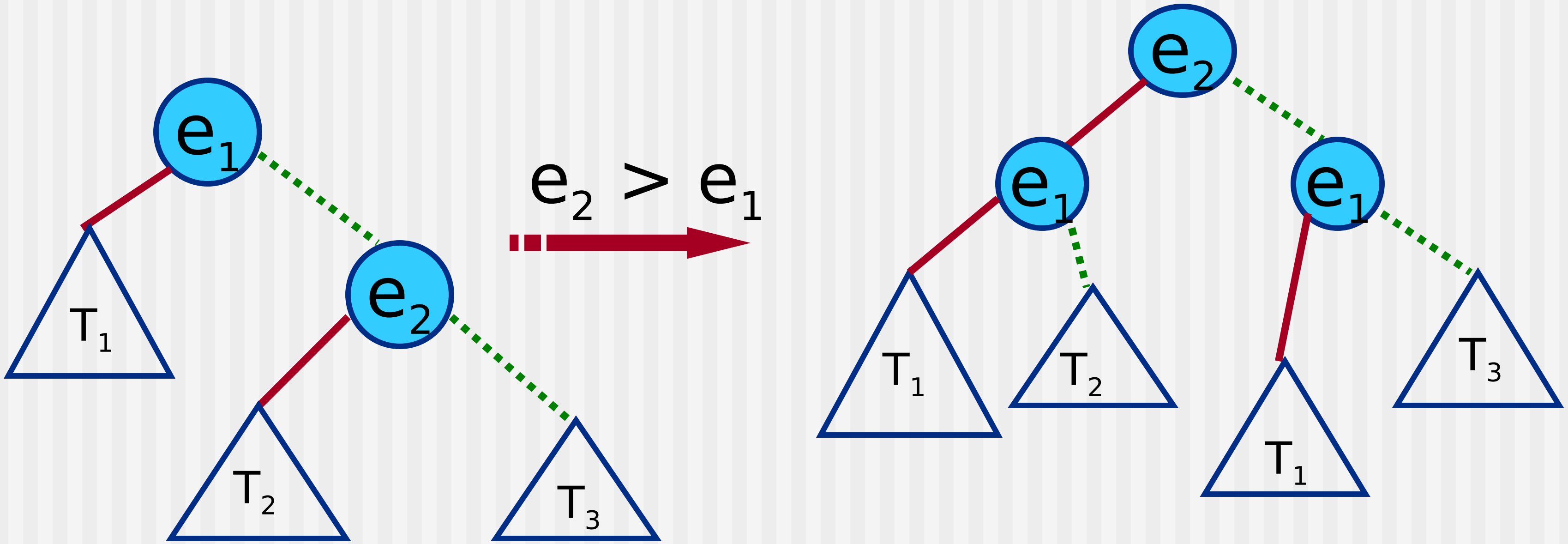
2)



ITE(e,T_1,ITE(e,T_2,T_3)) ⇒
ITE(e,T_1,T_3)

Reduction rules: the propositional structure of a formula

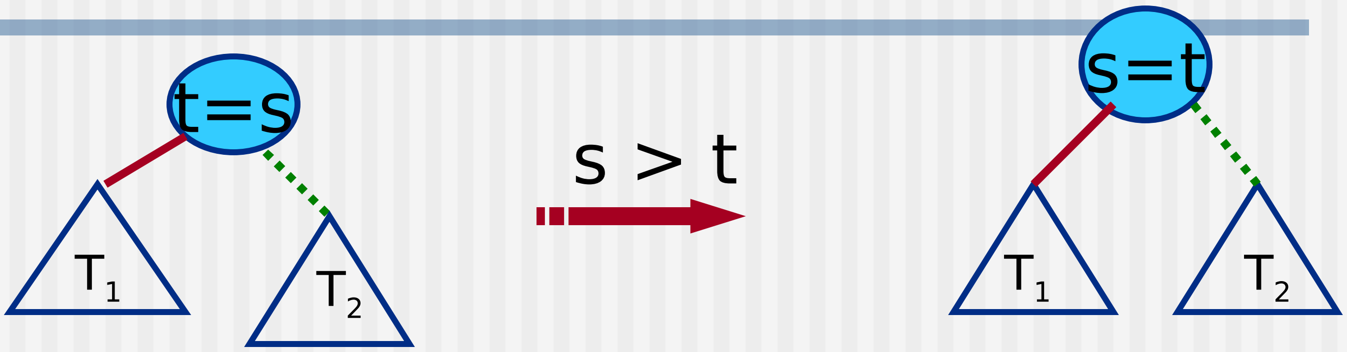
3)



$ITE(e_1,T_1,ITE(e_2,T_2,T_3)) \Rightarrow ITE(e_2,ITE(e_1,T_1,T_2),ITE(e_1,T_1,T_3))$

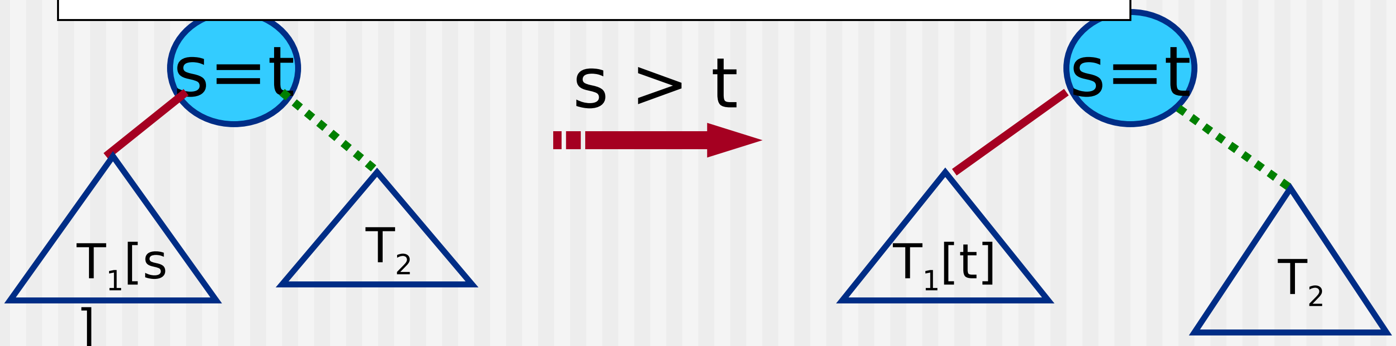
Reduction rules: rewrite rules

4)



$ITE(t=s,T_1,T_2) \Rightarrow ITE(s=t,T_1,T_2)$

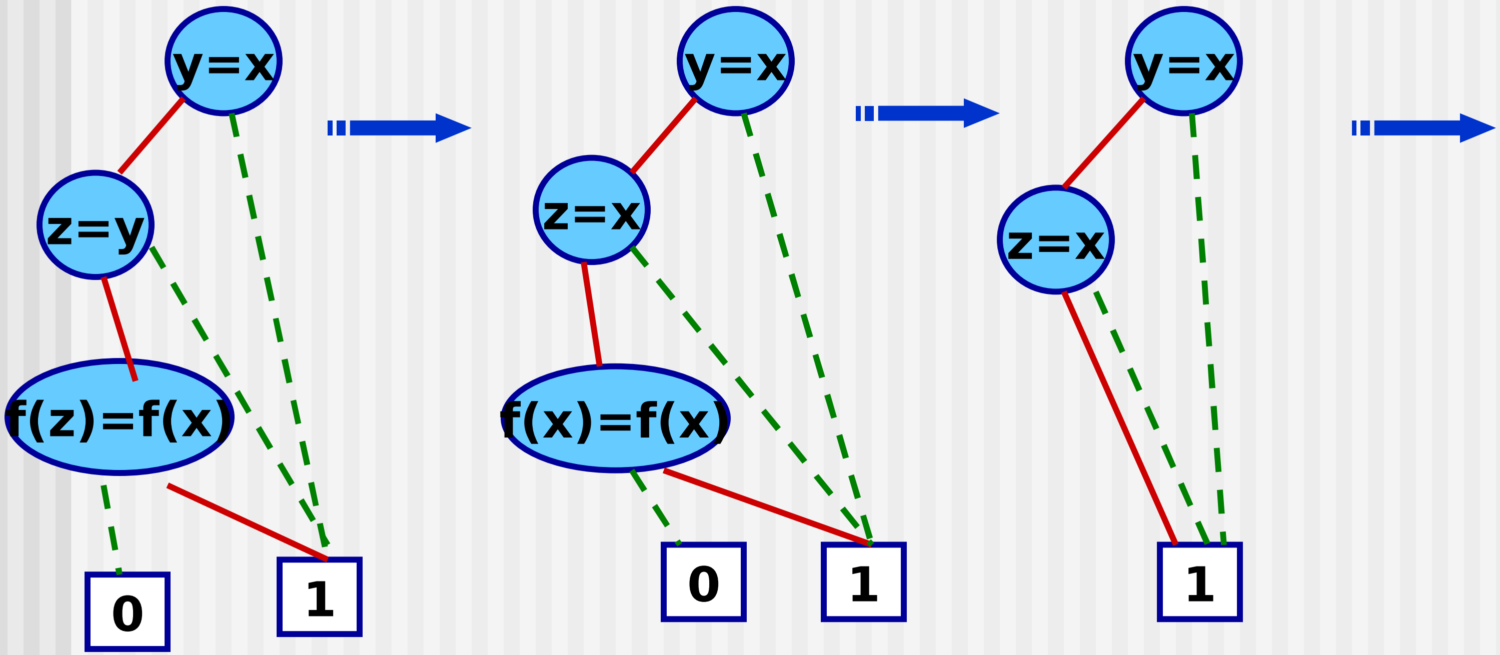
5)



$ITE(s=t,T_1[s],T_2) \Rightarrow ITE(s=t,T_1[t],T_2)$

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*
  - φ and ψ are equivalent if $\varphi \leftrightarrow \psi$ is represented by "1"

Thanks you for attention!