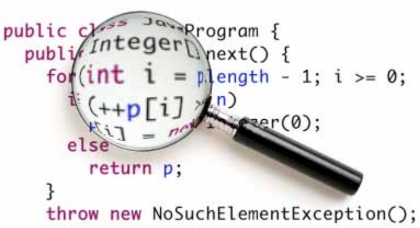
15-819M: Data, Code, Decisions

04: Equality Logic and Uninterpreted Functions

André Platzer

aplatzer@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA



Outline

1. Quantifier-free Equality Logic
   - EUF - QF Equality Logic with Uninterpreted Functions
   - QF Equality Logic without Functions

Outline

1. Quantifier-free Equality Logic
   - EUF - QF Equality Logic with Uninterpreted Functions
   - QF Equality Logic without Functions

QF Equality Logic with Uninterpreted Functions (EUF)

Definition (Quantifier-free Equality Logic)

Quantifier-free fragment of first-order logic with built-in equality.

$$\{\neg, \wedge, \vee, =, f_i/\alpha_i, p_i/\alpha_i\}$$

The semantics of $=$ is object identity.

Unlike $=$, the function symbols f_i of arities α_i are uninterpreted, i.e., have no special meaning or axiomatization.

$$x = g(y, z) \rightarrow f(x) = f(g(y, z))$$

QF Equality Logic with Uninterpreted Functions (EUF)

Definition (Quantifier-free Equality Logic)

Quantifier-free fragment of first-order logic with built-in equality.

$$\{\neg, \wedge, \vee, =, f_i/\alpha_i, p_i/\alpha_i\}$$

The semantics of $=$ is object identity.

Unlike $=$, the function symbols f_i of arities α_i are uninterpreted, i.e., have no special meaning or axiomatization.

$$f(f(f(a))) = a \wedge f(f(f(f(f(a))))) = a \rightarrow f(a) = a$$

QF Equality Logic with Uninterpreted Functions (EUF)

Definition (Quantifier-free Equality Logic)

Quantifier-free fragment of first-order logic with built-in equality.

$$\{\neg, \wedge, \vee, =, f_i/\alpha_i, p_i/\alpha_i\}$$

The semantics of $=$ is object identity.

Unlike $=$, the function symbols f_i of arities α_i are uninterpreted, i.e., have no special meaning or axiomatization.

1. $\forall x\,(x = x)$ reflexive

QF Equality Logic with Uninterpreted Functions (EUF)

Definition (Quantifier-free Equality Logic)

Quantifier-free fragment of first-order logic with built-in equality.

$$\{\neg, \wedge, \vee, =, f_i/\alpha_i, p_i/\alpha_i\}$$

The semantics of $=$ is object identity.

Unlike $=$, the function symbols f_i of arities α_i are uninterpreted, i.e., have no special meaning or axiomatization.

1. $\forall x\,(x = x)$ — reflexive
2. $\forall x\,\forall y\,(x = y \rightarrow y = x)$ — symmetric

QF Equality Logic with Uninterpreted Functions (EUF)

Definition (Quantifier-free Equality Logic)

Quantifier-free fragment of first-order logic with built-in equality.

$$\{\neg, \wedge, \vee, =, f_i/\alpha_i, p_i/\alpha_i\}$$

The semantics of $=$ is object identity.

Unlike $=$, the function symbols f_i of arities α_i are uninterpreted, i.e., have no special meaning or axiomatization.

1. $\forall x\,(x = x)$ — reflexive
2. $\forall x\,\forall y\,(x = y \rightarrow y = x)$ — symmetric
3. $\forall x\,\forall y\,\forall z\,(x = y \wedge y = z \rightarrow x = z)$ — transitive

QF Equality Logic with Uninterpreted Functions (EUF)

Definition (Quantifier-free Equality Logic)

Quantifier-free fragment of first-order logic with built-in equality.

$$\{\neg, \wedge, \vee, =, f_i/\alpha_i, p_i/\alpha_i\}$$

The semantics of $=$ is object identity.

Unlike $=$, the function symbols f_i of arities α_i are uninterpreted, i.e., have no special meaning or axiomatization.

1. $\forall x\,(x = x)$ reflexive
2. $\forall x\,\forall y\,(x = y \rightarrow y = x)$ symmetric
3. $\forall x\,\forall y\,\forall z\,(x = y \wedge y = z \rightarrow x = z)$ transitive
4. $\forall x_1..x_n\,\forall y_1..y_n\,(x_1 = y_1 \wedge .. \wedge x_n = y_n \rightarrow f(x_1, .., x_n) = f(y_1, .., y_n))$ congruence

QF Equality Logic with Uninterpreted Functions (EUF)

Definition (Quantifier-free Equality Logic)

Quantifier-free fragment of first-order logic with built-in equality.

$$\{\neg, \wedge, \vee, =, f_i/\alpha_i, p_i/\alpha_i\}$$

The semantics of $=$ is object identity.

Unlike $=$, the function symbols f_i of arities α_i are uninterpreted, i.e., have no special meaning or axiomatization.

1. $\forall x\, (x = x)$ reflexive
2. $\forall x\, \forall y\, (x = y \rightarrow y = x)$ symmetric
3. $\forall x\, \forall y\, \forall z\, (x = y \wedge y = z \rightarrow x = z)$ transitive
4. $\forall x_1..x_n\, \forall y_1..y_n\, (x_1 = y_1 \wedge .. \wedge x_n = y_n \rightarrow f(x_1, .., x_n) = f(y_1, .., y_n))$ congruence
5. $\forall x_1..x_n\, \forall y_1..y_n\, (x_1 = y_1 \wedge .. \wedge x_n = y_n \rightarrow (p(x_1, .., x_n) \leftrightarrow p(y_1, .., y_n)))$

Equality Logic with Uninterpreted Functions (EUF)

Example (Equality Logic with different functions and meanings)

Interpreted functions $x = y * z + x \rightarrow y = 0 \lor z + 0 = 0$

Equality Logic with Uninterpreted Functions (EUF)

Example (Equality Logic with different functions and meanings)

Interpreted functions	$x = y * z + x \rightarrow y = 0 \lor z + 0 = 0$
Uninterpreted functions	$x = a(m(y,z),x) \rightarrow y = 0 \lor a(z,0) = 0$

Equality Logic with Uninterpreted Functions (EUF)

Example (Equality Logic with different functions and meanings)

Interpreted functions	$x = y * z + x \rightarrow y = 0 \lor z + 0 = 0$
Uninterpreted functions	$x = a(m(y, z), x) \rightarrow y = 0 \lor a(z, 0) = 0$
No functions	$x = c \rightarrow y = 0 \lor b = 0$

Removing Interpretation: A Lossy Transformation

Algorithm: Uninterpreting

Input: formula ϕ in equality logic plus interpreted functions
Output: formula in equality logic plus uninterpreted functions

1. Replace each interpreted function symbol by a new uninterpreted function symbol

Example (Forgetful projection)

Interpreted functions	$x = y * z + x \rightarrow y = 0 \lor z + 0 = 0$
Uninterpreted functions	$x = a(m(y,z),x) \rightarrow y = 0 \lor a(z,0) = 0$

Removing Interpretation: A Lossy Transformation

Algorithm: Uninterpreting

Input: formula ϕ in equality logic plus interpreted functions
Output: formula in equality logic plus uninterpreted functions
of different semantics!

1. Replace each interpreted function symbol by a new uninterpreted function symbol

Example (Forgetful projection)

Interpreted functions	$x = y * z + x \rightarrow y = 0 \lor z + 0 = 0$
Uninterpreted functions	$x = a(m(y, z), x) \rightarrow y = 0 \lor a(z, 0) = 0$

Removing Interpretation: A Lossy Transformation

Algorithm: Uninterpreting

Input: formula ϕ in equality logic plus interpreted functions
Output: formula in equality logic plus uninterpreted functions
of different semantics!

1. Replace each interpreted function symbol by a new uninterpreted function symbol

Example (Forgetful projection)

Interpreted functions	$x = y * z + x \rightarrow y = 0 \lor z + 0 = 0$
Uninterpreted functions	$x = a(m(y,z),x) \rightarrow y = 0 \lor a(z,0) = 0$

If the uninterpreted formula is valid, its interpreted variant is valid too, but not vice versa.

Ackermann's Reduction: Idea

- Goal: remove uninterpreted functions
- Replace uninterpreted function terms with new variables
- Add functional consistency axioms as needed from the following axiom scheme

$$x_1 = y_1 \land \cdots \land x_n = y_n \rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$$

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$f(f(x)) = 1 \ \lor \ f(x) \neq 2$$

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$f(\overbrace{f(x)}^{f_1}) = 1 \lor \overbrace{f(x)}^{f_1} \neq 2$$

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$\underbrace{f(\overbrace{f(x)}^{f_1})}_{f_2} = 1 \;\vee\; \overbrace{f(x)}^{f_1} \neq 2$$

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$\underbrace{f(\overbrace{f(x)}^{f_1})}_{f_2} = 1 \;\lor\; \overbrace{f(x)}^{f_1} \neq 2 \leadsto f_2 = 1 \lor f_1 \neq 2$$

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$\underbrace{f(\overbrace{f(x)}^{f_1})}_{f_2} = 1 \;\lor\; \overbrace{f(x)}^{f_1} \neq 2 \leadsto f_2 = 1 \lor f_1 \neq 2 \quad \left[\begin{array}{l} f_1 = f(x) \\ f_2 = f(f_1) \end{array}\right.$$

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$\underbrace{f(\overbrace{f(x)}^{f_1})}_{f_2} = 1 \ \lor\ \overbrace{f(x)}^{f_1} \neq 2 \rightsquigarrow\ f_2 = 1 \lor f_1 \neq 2 \qquad \left[\begin{array}{l} f_1 = f(x) \\ f_2 = f(f_1) \end{array}\right.$$

3. Add functional consistency axiom for every pair of arguments of f

$$(x = f_1 \rightarrow f_2 = f_1) \quad \rightarrow \quad f_2 = 1 \ \lor\ f_1 \neq 2$$

Ackermann's Reduction

Algorithm: Ackermann's Reduction

Input: quantifier-free ϕ in equality logic plus uninterpreted functions
Output: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$\underbrace{f(\overbrace{f(x)}^{f_1})}_{f_2} = 1 \;\lor\; \overbrace{f(x)}^{f_1} \neq 2 \leadsto \; f_2 = 1 \lor f_1 \neq 2 \qquad \left[\begin{array}{l} f_1 = f(x) \\ f_2 = f(f_1) \end{array}\right.$$

3. Add functional consistency axiom for every pair of arguments of f

$$(x = f_1 \rightarrow f_2 = f_1) \;\;\rightarrow\;\; f_2 = 1 \;\lor\; f_1 \neq 2$$

$\phi^\flat$ valid iff ϕ valid

Ackermann's Reduction: Example

Example

Input: quantifier-free ϕ in equality logic plus uninterpreted functions

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

1. Transform ϕ to negation normal form by pushing negations in

Ackermann's Reduction: Example

Example

Input: quantifier-free ϕ in equality logic plus uninterpreted functions

$$x_1 = x_2 \rightarrow f(x_1) \neq f(x_2) \lor f(x_1) \neq f(x_3)$$

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$x_1 = x_2 \rightarrow f_1 \neq f_2 \lor f_1 \neq f_3$$

Ackermann's Reduction: Example

Example

Input: quantifier-free ϕ in equality logic plus uninterpreted functions

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

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$x_1 = x_2 \rightarrow f_1 \neq f_2 \vee f_1 \neq f_3 \quad \left[\begin{array}{l} f_1 = f(x_1) \\ f_2 = f(x_2) \\ f_3 = f(x_3) \end{array} \right.$$

Ackermann's Reduction: Example

Example

Input: quantifier-free ϕ in equality logic plus uninterpreted functions

$$x_1 = x_2 \rightarrow f(x_1) \neq f(x_2) \lor f(x_1) \neq f(x_3)$$

1. Transform ϕ to negation normal form by pushing negations in
2. Replace function terms by unique identifiers from inside out

$$x_1 = x_2 \rightarrow f_1 \neq f_2 \lor f_1 \neq f_3 \quad \left[\begin{array}{l} f_1 = f(x_1) \\ f_2 = f(x_2) \\ f_3 = f(x_3) \end{array}\right.$$

3. Add functional consistency axiom for every pair of arguments of f

$$\begin{array}{rl} & \big((x_1 = x_2 \rightarrow f_1 = f_2) \\ & \land(x_1 = x_3 \rightarrow f_1 = f_3) \\ & \land(x_2 = x_3 \rightarrow f_2 = f_3)\big) \\ \rightarrow & (x_1 = x_2 \rightarrow f_1 \neq f_2 \lor f_1 \neq f_3) \end{array}$$

QF Equality Logic without Functions

Definition (Quantifier-free Equality Logic without Functions)

Quantifier-free fragment of first-order logic with built-in equality as only predicate and no functions.

$$\{\neg, \land, \lor, =\}$$

The semantics of $=$ is object identity.

Example

$$x = c \rightarrow y = 0 \lor b = 0$$

QF Equality Logic without Functions

Algorithm: Satisfiability of QF Equality Logic without Functions

Input: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions
Output: satisfiable / unsatisfiable

1. Transform $\phi^\flat$ into DNF (can be optimized)

QF Equality Logic without Functions

Algorithm: Satisfiability of QF Equality Logic without Functions

Input: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions
Output: satisfiable / unsatisfiable

1. Transform $\phi^\flat$ into DNF (can be optimized)
2. Consider each disjunct F separately

QF Equality Logic without Functions

Algorithm: Satisfiability of QF Equality Logic without Functions

Input: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions
Output: satisfiable / unsatisfiable

1. Transform $\phi^\flat$ into DNF (can be optimized)
2. Consider each disjunct F separately
3. For each variable x, define equivalence class $[x] := \{x\}$

QF Equality Logic without Functions

Algorithm: Satisfiability of QF Equality Logic without Functions

Input: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions
Output: satisfiable / unsatisfiable

1. Transform $\phi^\flat$ into DNF (can be optimized)
2. Consider each disjunct F separately
3. For each variable x, define equivalence class $[x] := \{x\}$
4. For each $(x = y) \in F$, merge equivalence classes $[x]$ and $[y]$ by $[x] := [y] := [x] \cup [y]$.

QF Equality Logic without Functions

Algorithm: Satisfiability of QF Equality Logic without Functions

Input: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions
Output: satisfiable / unsatisfiable

1. Transform $\phi^\flat$ into DNF (can be optimized)
2. Consider each disjunct F separately
3. For each variable x, define equivalence class $[x] := \{x\}$
4. For each $(x = y) \in F$, merge equivalence classes $[x]$ and $[y]$ by $[x] := [y] := [x] \cup [y]$.
5. For each $(x \neq y) \in F$, if $x \in [y]$ then F unsat; consider next disjunct

QF Equality Logic without Functions

Algorithm: Satisfiability of QF Equality Logic without Functions

Input: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions
Output: satisfiable / unsatisfiable

1. Transform $\phi^\flat$ into DNF (can be optimized)
2. Consider each disjunct F separately
3. For each variable x, define equivalence class $[x] := \{x\}$
4. For each $(x = y) \in F$, merge equivalence classes $[x]$ and $[y]$ by $[x] := [y] := [x] \cup [y]$.
5. For each $(x \neq y) \in F$, if $x \in [y]$ then F unsat; consider next disjunct
6. return sat

QF Equality Logic without Functions

Algorithm: Satisfiability of QF Equality Logic without Functions

Input: quantifier-free $\phi^\flat$ in equality logic w/o uninterpreted functions
Output: satisfiable / unsatisfiable

1. Transform $\phi^\flat$ into DNF (can be optimized)
2. Consider each disjunct F separately
3. For each variable x, define equivalence class $[x] := \{x\}$
4. For each $(x = y) \in F$, merge equivalence classes $[x]$ and $[y]$ by $[x] := [y] := [x] \cup [y]$.
5. For each $(x \neq y) \in F$, if $x \in [y]$ then F unsat; consider next disjunct
6. return sat

Much more efficient algorithms exist even with UF

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

Equivalence classes

$$[a] = \{a \qquad\} \ \ [d] = \{d \quad\}$$

$$[a] = \{a \qquad\} \ \ [d] = \{d \quad\} \ \ [f_1] = \{f_1 \quad\}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$
$$\lor\ f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

Equivalence classes

$$[a] = \{a, b \quad\} \quad [d] = \{d \quad\}$$
$$[a] = \{a \quad\} \quad [d] = \{d \quad\} \quad [f_1] = \{f_1 \quad\}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

Equivalence classes

$$[a] = \{a, b \quad\} \quad [d] = \{d, e\}$$
$$[a] = \{a \quad\quad\} \quad [d] = \{d \quad\} \quad [f_1] = \{f_1 \quad\}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

Equivalence classes

$$[a] = \{a, b, c\} \quad [d] = \{d, e\}$$
$$[a] = \{a \quad\quad \} \quad [d] = \{d \quad \} \quad [f_1] = \{f_1 \quad \}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

Equivalence classes

$$[a] = \{a, b, c\} \quad [d] = \{d, e\} \qquad\qquad a \neq c \text{ unsat}$$

$$[a] = \{a \quad\quad\} \quad [d] = \{d \quad\} \quad [f_1] = \{f_1 \quad\}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \to f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

Equivalence classes

$$[a] = \{a, b, c\} \quad [d] = \{d, e\} \qquad\qquad a \neq c \text{ unsat}$$

$$[a] = \{a, b \quad\} \quad [d] = \{d \quad\} \quad [f_1] = \{f_1 \quad\}$$

QF Equality Logic without Functions: Example

Example

$$a = b \wedge d = e \wedge e \neq a \wedge b = c \wedge f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \wedge a = b \wedge d = e \wedge e \neq a \wedge b = c \wedge f_1 \neq f_2$$

DNF

$$\begin{array}{rl} & a \neq c \wedge a = b \wedge d = e \wedge e \neq a \wedge b = c \wedge f_1 \neq f_2 \\ \vee & f_1 = f_2 \wedge a = b \wedge d = e \wedge e \neq a \wedge b = c \wedge f_1 \neq f_2 \end{array}$$

Equivalence classes

$$[a] = \{a, b, c\} \quad [d] = \{d, e\} \qquad\qquad a \neq c \text{ unsat}$$

$$[a] = \{a, b \quad\} \quad [d] = \{d, e\} \quad [f_1] = \{f_1 \quad\}$$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \to f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

Equivalence classes

$[a] = \{a, b, c\}$ $[d] = \{d, e\}$ $a \neq c$ unsat

$[a] = \{a, b, c\}$ $[d] = \{d, e\}$ $[f_1] = \{f_1 \quad \}$

QF Equality Logic without Functions: Example

Example

$$a = b \land d = e \land e \neq a \land b = c \land f(a) \neq f(c) \quad \text{satisfiable?}$$

Ackermann's reduction (for satisfiability!)

$$(a = c \rightarrow f_1 = f_2) \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2$$

DNF

$$\begin{aligned} & a \neq c \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \\ \lor\ & f_1 = f_2 \land a = b \land d = e \land e \neq a \land b = c \land f_1 \neq f_2 \end{aligned}$$

Equivalence classes

$[a] = \{a, b, c\} \quad [d] = \{d, e\}$ $\qquad a \neq c$ unsat

$[a] = \{a, b, c\} \quad [d] = \{d, e\} \quad [f_1] = \{f_1, f_2\}$ $\qquad f_1 \neq f_2$ unsat