Logic-based Program Verification

Decidability of Propositional and First-Order Logic. First-Order Theories. Theory of Equality

Mădălina Erașcu   Tudor Jebelean

Research Institute for Symbolic Computation,
Johannes Kepler University, Linz, Austria

{merascu,tjebelea}@risc.jku.at

November 20, 2013

Outline

Outline

Preliminaries

Decidability of PL and FOL

First-Order Theories

Theory of Equality (T_{EUF}). Congruence Closure Algorithm for T_{QFEUF}

The Decision Problem of Formulas

The decision problem for a given formula ϕ is to determine whether ϕ is valid/satisfiable.

A procedure for the decision problem is sound if when it returns "Valid"/"Satisfiable", the input formula is indeed valid/satisfiable.

A procedure for the decision problem is complete if

1. it always terminates, and
2. it returns "Valid"/"Satisfiable" when the input formula is indeed valid/satisfiable.

A procedure is called a decision procedure for the theory T (e.g. propositional logic, first-order logic, other theories to be discussed later) if it is sound and complete with respect to every formula of T.

A theory is decidable iff there is a decision procedure for it.

The Decision Problem of Formulas

The decision problem for a given formula ϕ is to determine whether ϕ is valid/satisfiable.

A procedure for the decision problem is sound if when it returns "Valid"/"Satisfiable", the input formula is indeed valid/satisfiable.

A procedure for the decision problem is complete if

1. it always terminates, and
2. it returns "Valid"/"Satisfiable" when the input formula is indeed valid/satisfiable.

A procedure is called a decision procedure for the theory T (e.g. propositional logic, first-order logic, other theories to be discussed later) if it is sound and complete with respect to every formula of T.

A theory is decidable iff there is a decision procedure for it.

The Decision Problem of Formulas

The decision problem for a given formula ϕ is to determine whether ϕ is valid/satisfiable.

A procedure for the decision problem is sound if when it returns "Valid"/"Satisfiable", the input formula is indeed valid/satisfiable.

A procedure for the decision problem is complete if

1. it always terminates, and
2. it returns "Valid"/"Satisfiable" when the input formula is indeed valid/satisfiable.

A procedure is called a decision procedure for the theory T (e.g. propositional logic, first-order logic, other theories to be discussed later) if it is sound and complete with respect to every formula of T.

A theory is decidable iff there is a decision procedure for it.

The Decision Problem of Formulas

The decision problem for a given formula ϕ is to determine whether ϕ is valid/satisfiable.

A procedure for the decision problem is sound if when it returns "Valid"/"Satisfiable", the input formula is indeed valid/satisfiable.

A procedure for the decision problem is complete if

1. it always terminates, and
2. it returns "Valid"/"Satisfiable" when the input formula is indeed valid/satisfiable.

A procedure is called a decision procedure for the theory T (e.g. propositional logic, first-order logic, other theories to be discussed later) if it is sound and complete with respect to every formula of T.

A theory is decidable iff there is a decision procedure for it.

The Decision Problem of Formulas

The decision problem for a given formula ϕ is to determine whether ϕ is valid/satisfiable.

A procedure for the decision problem is sound if when it returns "Valid"/"Satisfiable", the input formula is indeed valid/satisfiable.

A procedure for the decision problem is complete if

1. it always terminates, and
2. it returns "Valid"/"Satisfiable" when the input formula is indeed valid/satisfiable.

A procedure is called a decision procedure for the theory T (e.g. propositional logic, first-order logic, other theories to be discussed later) if it is sound and complete with respect to every formula of T.

A theory is decidable iff there is a decision procedure for it.

Outline

Preliminaries

Decidability of PL and FOL

First-Order Theories

Theory of Equality (T_{EUF}). Congruence Closure Algorithm for T_{QFEUF}

Decidability of PL and FOL

Questions

- Is propositional logic (PL) decidable? If so, give example of decision procedures
- Yes! (truth table, resolution, DPLL)
- Is first-order logic (FOL) decidable? If so, give example of decision procedures.
- FOL is undecidable (Church & Turing): there does not exist a decision procedure/algorithm for deciding if a FOL formula F is valid/satisfiable.
- FOL is semi-decidable: there is a procedure that halts and says "yes" if F is indeed valid/satisfiable.

Decidability of PL and FOL

Questions

- Is propositional logic (PL) decidable? If so, give example of decision procedures
- Yes! (truth table, resolution, DPLL)

- Is first-order logic (FOL) decidable? If so, give example of decision procedures.
- FOL is undecidable (Church & Turing): there does not exist a decision procedure/algorithm for deciding if a FOL formula F is valid/satisfiable.
- FOL is semi-decidable: there is a procedure that halts and says "yes" if F is indeed valid/satisfiable.

Decidability of PL and FOL

Questions

- Is propositional logic (PL) decidable? If so, give example of decision procedures
- Yes! (truth table, resolution, DPLL)
- Is first-order logic (FOL) decidable? If so, give example of decision procedures.
- FOL is undecidable (Church & Turing): there does not exist a decision procedure/algorithm for deciding if a FOL formula F is valid/satisfiable.
- FOL is semi-decidable: there is a procedure that halts and says "yes" if F is indeed valid/satisfiable.

Decidability of PL and FOL

Questions

- Is propositional logic (PL) decidable? If so, give example of decision procedures
- Yes! (truth table, resolution, DPLL)
- Is first-order logic (FOL) decidable? If so, give example of decision procedures.
- FOL is undecidable (Church & Turing): there does not exist a decision procedure/algorithm for deciding if a FOL formula F is valid/satisfiable.
- FOL is semi-decidable: there is a procedure that halts and says "yes" if F is indeed valid/satisfiable.

Decidability of PL and FOL

Questions

- Is propositional logic (PL) decidable? If so, give example of decision procedures
- Yes! (truth table, resolution, DPLL)

- Is first-order logic (FOL) decidable? If so, give example of decision procedures.
- FOL is undecidable (Church & Turing): there does not exist a decision procedure/algorithm for deciding if a FOL formula F is valid/satisfiable.
- FOL is semi-decidable: there is a procedure that halts and says "yes" if F is indeed valid/satisfiable.

Decidability of PL and FOL

Questions

- Is propositional logic (PL) decidable? If so, give example of decision procedures
- Yes! (truth table, resolution, DPLL)

- Is first-order logic (FOL) decidable? If so, give example of decision procedures.
- FOL is undecidable (Church & Turing): there does not exist a decision procedure/algorithm for deciding if a FOL formula F is valid/satisfiable.
- FOL is semi-decidable: there is a procedure that halts and says "yes" if F is indeed valid/satisfiable.

Outline

Preliminaries

Decidability of PL and FOL

First-Order Theories

Theory of Equality (T_{EUF}). Congruence Closure Algorithm for T_{QFEUF}

First-Order Theories

Motivation:

- Reasoning in applications domains, e.g. software, hardware, necessitates various notions (numbers, lists, arrays, memory, etc.) which can be formalized using FOL.
- While FOL is undecidable, validity in particular theories or fragments of theories interesting for verification is sometimes decidable and even efficiently decidable.

First-Order Theories

Motivation:

- Reasoning in applications domains, e.g. software, hardware, necessitates various notions (numbers, lists, arrays, memory, etc.) which can be formalized using FOL.
- While FOL is undecidable, validity in particular theories or fragments of theories interesting for verification is sometimes decidable and even efficiently decidable.

First-Order Theories

Motivation:

- Reasoning in applications domains, e.g. software, hardware, necessitates various notions (numbers, lists, arrays, memory, etc.) which can be formalized using FOL.
- While FOL is undecidable, validity in particular theories or fragments of theories interesting for verification is sometimes decidable and even efficiently decidable.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies $F : I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies $F : I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies F : $I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies F : $I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies $F : I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies $F : I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies $F : I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies F : $I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies F : $I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies $F : I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies $F : I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

First-Order Theories

A first-order theory T is defined by:

1. signature Σ: set of constant, function, predicate symbols
2. a set of axioms $\mathcal{A}$: closed set of FOL formulas in which only constant, function, and predicate symbols of Σ appear.

A formula F is closed if it does not contain any free variables.

A Σ-formula F is valid in T (T-valid), if every interpretation I that satisfies the axioms of T,

$$I \models A \text{ for every } A \in \mathcal{A}, \tag{1}$$

also satisfies F : $I \models F$. We also write $T \models F$ (F is T-valid).

The theory T consists of all (closed) formulas that are T-valid.

An interpretation satisfying (1) is a T-interpretation.

A Σ-formula F is satisfiable in T (T-satisfiable), if there is a T-interpretation I that satisfies F.

A theory T is complete if for every closed Σ-formula F, $T \models F$ or $T \models \neg F$.

A theory is consistent if there is at least one T-interpretation.

A fragment of a theory is a syntactically-restricted subset of formulas of the theory.

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

a, b, c,... – constants, f, g, h,... – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\forall_x \; x = x$ (reflexivity)
2. $\forall_{x,y} \; x = y \implies y = x$ (symmetry)
3. $\forall_{x,y,z} \; x = y \wedge y = z \implies x = z$ (transitivity)
4. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

a, b, c,... – constants, f, g, h,... – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\forall_x \; x = x$ (reflexivity)
2. $\forall_{x,y} \; x = y \implies y = x$ (symmetry)
3. $\forall_{x,y,z} \; x = y \wedge y = z \implies x = z$ (transitivity)
4. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

a, b, c,... – constants, f, g, h,... – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\forall_x \; x = x$ (reflexivity)
2. $\forall_{x,y} \; x = y \implies y = x$ (symmetry)
3. $\forall_{x,y,z} \; x = y \wedge y = z \implies x = z$ (transitivity)
4. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

a, b, c,... – constants, f, g, h,... – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\forall_{x}\; x = x$ (reflexivity)
2. $\forall_{x,y}\; x = y \implies y = x$ (symmetry)
3. $\forall_{x,y,z}\; x = y \land y = z \implies x = z$ (transitivity)
4. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

$a, b, c, ...$ – constants, $f, g, h, ...$ – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\underset{x}{\forall}\ x = x$ (reflexivity)
2. $\underset{x,y}{\forall}\ x = y \implies y = x$ (symmetry)
3. $\underset{x,y,z}{\forall}\ x = y \wedge y = z \implies x = z$ (transitivity)
4. $\underset{\bar{x},\bar{y}}{\forall} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\underset{\bar{x},\bar{y}}{\forall} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

$a, b, c, ...$ – constants, $f, g, h, ...$ – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\forall_{x}\ x = x$ (reflexivity)
2. $\forall_{x,y}\ x = y \implies y = x$ (symmetry)
3. $\forall_{x,y,z}\ x = y \wedge y = z \implies x = z$ (transitivity)
4. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

$a, b, c, ...$ – constants, $f, g, h, ...$ – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\underset{x}{\forall}\ x = x$ (reflexivity)
2. $\underset{x,y}{\forall}\ x = y \implies y = x$ (symmetry)
3. $\underset{x,y,z}{\forall}\ x = y \wedge y = z \implies x = z$ (transitivity)
4. $\underset{\bar{x},\bar{y}}{\forall}\ \left(\bigwedge_{i=1}^{n} x_i = y_i\right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\underset{\bar{x},\bar{y}}{\forall}\ \left(\bigwedge_{i=1}^{n} x_i = y_i\right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

$a, b, c, ...$ – constants, $f, g, h, ...$ – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\underset{x}{\forall}\ x = x$ (reflexivity)
2. $\underset{x,y}{\forall}\ x = y \implies y = x$ (symmetry)
3. $\underset{x,y,z}{\forall}\ x = y \wedge y = z \implies x = z$ (transitivity)
4. $\underset{\bar{x},\bar{y}}{\forall} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\underset{\bar{x},\bar{y}}{\forall} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

$a, b, c, ...$ – constants, $f, g, h, ...$ – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\underset{x}{\forall}\ x = x$ (reflexivity)
2. $\underset{x,y}{\forall}\ x = y \implies y = x$ (symmetry)
3. $\underset{x,y,z}{\forall}\ x = y \wedge y = z \implies x = z$ (transitivity)
4. $\underset{\bar{x},\bar{y}}{\forall} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\underset{\bar{x},\bar{y}}{\forall} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

$a, b, c, ...$ – constants, $f, g, h, ...$ – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\forall_x \; x = x$ (reflexivity)
2. $\forall_{x,y} \; x = y \implies y = x$ (symmetry)
3. $\forall_{x,y,z} \; x = y \wedge y = z \implies x = z$ (transitivity)
4. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF}

This theory is sometimes referred to as the theory of equality with uninterpreted functions (EUF).

Signature: $\Sigma_E = \{=, a, b, c, ..., f, g, h, ..., P, Q, R, ...\}$

a, b, c,... – constants, f, g, h,... – function symbols, P,Q,R,... – predicate symbols

The predicate $=$ is interpreted via the following axioms:

1. $\forall_{x}\ x = x$ (reflexivity)
2. $\forall_{x,y}\ x = y \implies y = x$ (symmetry)
3. $\forall_{x,y,z}\ x = y \wedge y = z \implies x = z$ (transitivity)
4. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies f(\bar{x}) = f(\bar{y})$ (function congruence), where n is a positive integer and f is an n-ary function symbol
5. $\forall_{\bar{x},\bar{y}} \left(\bigwedge_{i=1}^{n} x_i = y_i \right) \implies P(\bar{x}) = P(\bar{y})$ (function congruence), where n is a positive integer and P is an n-ary predicate symbol

We have

1. $=$ is an equivalence relation
2. $=$ is a congruence relation

T_{EUF} (cont'd)

Is T_E decidable?

Is quantifier-free T_E decidable?

Without quantifiers, free variables and constants play the same role.

Example:
Prove that F is T_E valid where

$$F \quad :\Longleftrightarrow \quad a = b \;\wedge\; b = c \quad \Longrightarrow \quad g[f[a], b] = g[f[c], a]$$

Goal: decision procedure for satisfiability of quantifier - free theory of equality (QFEUF)

T_{EUF} (cont'd)

Is T_E decidable?

Is quantifier-free T_E decidable?

Without quantifiers, free variables and constants play the same role.

Example:
Prove that F is T_E valid where

$$F \quad :\Longleftrightarrow \quad a = b \ \wedge \ b = c \ \Longrightarrow \ g[f[a], b] = g[f[c], a]$$

Goal: decision procedure for satisfiability of quantifier - free theory of equality (QFEUF)

T_{EUF} (cont'd)

Is T_E decidable?

Is quantifier-free T_E decidable?

Without quantifiers, free variables and constants play the same role.

Example:
Prove that F is T_E valid where

$$F \quad :\iff \quad a = b \ \wedge \ b = c \quad \Longrightarrow \quad g[f[a], b] = g[f[c], a]$$

Goal: decision procedure for satisfiability of quantifier - free theory of equality (QFEUF)

T_{EUF} (cont'd)

Is T_E decidable?

Is quantifier-free T_E decidable?

Without quantifiers, free variables and constants play the same role.

Example:

Prove that F is T_E valid where

$$F \quad :\Longleftrightarrow \quad a = b \ \wedge \ b = c \quad \Longrightarrow \quad g[f[a], b] = g[f[c], a]$$

Goal: decision procedure for satisfiability of quantifier - free theory of equality (QFEUF)

T_{EUF} (cont'd)

Is T_E decidable?

Is quantifier-free T_E decidable?

Without quantifiers, free variables and constants play the same role.

Example:
Prove that F is T_E valid where

$$F \quad :\Longleftrightarrow \quad a = b \ \wedge \ b = c \quad \Longrightarrow \quad g[f[a], b] = g[f[c], a]$$

Goal: decision procedure for satisfiability of quantifier - free theory of equality (QFEUF)

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements $s_1, s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\forall_{s \in S} \; sRs$
2. symmetric: $\forall_{s_1, s_2 \in S} \; s_1 R s_2 \implies s_2 R s_1$
3. transitive: $\forall_{s_1, s_2, s_3 \in S} \; s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. $1 - 3$ hold
2. for any n-ary function f,

$$\forall_{\bar{s}, \bar{t}} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements s_1, $s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\forall_{s \in S} \; sRs$
2. symmetric: $\forall_{s_1, s_2 \in S} \; s_1 R s_2 \implies s_2 R s_1$
3. transitive: $\forall_{s_1, s_2, s_3 \in S} \; s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. $1 - 3$ hold
2. for any n-ary function f,

$$\forall_{\bar{s}, \bar{t}} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements s_1, $s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\forall_{s \in S} \; sRs$
2. symmetric: $\forall_{s_1, s_2 \in S} \; s_1 R s_2 \Longrightarrow s_2 R s_1$
3. transitive: $\forall_{s_1, s_2, s_3 \in S} \; s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. $1 - 3$ hold
2. for any n-ary function f,

$$\forall_{\bar{s}, \bar{t}} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements s_1, $s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\underset{s \in S}{\forall} sRs$
2. symmetric: $\underset{s_1, s_2 \in S}{\forall} s_1 R s_2 \Longrightarrow s_2 R s_1$
3. transitive: $\underset{s_1, s_2, s_3 \in S}{\forall} s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. $1 - 3$ hold
2. for any n-ary function f,

$$\underset{\bar{s}, \bar{t}}{\forall} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements s_1, $s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\forall_{s \in S}\ sRs$
2. symmetric: $\forall_{s_1, s_2 \in S}\ s_1 R s_2 \Longrightarrow s_2 R s_1$
3. transitive: $\forall_{s_1, s_2, s_3 \in S}\ s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. $1 - 3$ hold
2. for any n-ary function f,

$$\forall_{\bar{s}, \bar{t}} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements s_1, $s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\forall_{s \in S}\ sRs$
2. symmetric: $\forall_{s_1, s_2 \in S}\ s_1 R s_2 \implies s_2 R s_1$
3. transitive: $\forall_{s_1, s_2, s_3 \in S}\ s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. 1 – 3 hold
2. for any n-ary function f,

$$\forall_{\bar{s}, \bar{t}} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements $s_1, s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\forall_{s \in S} \; sRs$
2. symmetric: $\forall_{s_1, s_2 \in S} \; s_1 R s_2 \implies s_2 R s_1$
3. transitive: $\forall_{s_1, s_2, s_3 \in S} \; s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. $1 - 3$ hold
2. for any n-ary function f,

$$\forall_{\bar{s}, \bar{t}} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements $s_1, s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\underset{s \in S}{\forall}\ sRs$
2. symmetric: $\underset{s_1, s_2 \in S}{\forall}\ s_1 R s_2 \Longrightarrow s_2 R s_1$
3. transitive: $\underset{s_1, s_2, s_3 \in S}{\forall}\ s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. 1 – 3 hold
2. for any n-ary function f,

$$\underset{\bar{s}, \bar{t}}{\forall} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations

Let S be a *set* and R a *binary relation* over S.

For two elements s_1, $s_2 \in S$, either $s_1 R s_2$ or $\neg(s_1 R s_2)$.

The relation R is an equivalence relation if it is

1. reflexive: $\underset{s \in S}{\forall} sRs$
2. symmetric: $\underset{s_1, s_2 \in S}{\forall} s_1 R s_2 \Longrightarrow s_2 R s_1$
3. transitive: $\underset{s_1, s_2, s_3 \in S}{\forall} s_1 R s_2 \wedge s_2 R s_3 \implies s_1 R s_3$

The relation R is a congruence relation if

1. $1 - 3$ hold
2. for any n-ary function f,

$$\underset{\bar{s}, \bar{t}}{\forall} \left(\bigwedge_{i=1}^{n} s_i R t_i \right) \implies f(\bar{s}) R f(\bar{t})$$

Relations (cont'd)

Let R be a *equivalence relation* over the set S.

The equivalence class of $s \in S$ under R is the set

$$[s]_R \stackrel{def}{=} \{s' \in S : sRs'\}$$

If R is a congruence relation over S, then $[s]_R$ is the congruence class of s.

A partition P of S is a set of subsets of S that is

1. total: $\left(\bigcup_{S' \in P} S'\right) = S$
2. disjoint: $\forall_{S_1, S_2 \in P} \; S_1 \neq S_2 \implies S_1 \cap S_2 = \emptyset$

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

Relations (cont'd)

Let R be a *equivalence relation* over the set S.

The equivalence class of $s \in S$ under R is the set

$$[s]_R \stackrel{def}{=} \{s' \in S : sRs'\}$$

If R is a congruence relation over S, then $[s]_R$ is the congruence class of s.

A partition P of S is a set of subsets of S that is

1. total: $\left(\bigcup_{S' \in P} S' \right) = S$
2. disjoint: $\forall_{S_1, S_2 \in P} \; S_1 \neq S_2 \implies S_1 \cap S_2 = \emptyset$

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

Relations (cont'd)

Let R be a *equivalence relation* over the set S.

The equivalence class of $s \in S$ under R is the set

$$[s]_R \stackrel{def}{=} \{s' \in S : sRs'\}$$

If R is a congruence relation over S, then $[s]_R$ is the congruence class of s.

A partition P of S is a set of subsets of S that is

1. total: $\left(\bigcup_{S' \in P} S'\right) = S$
2. disjoint: $\forall_{S_1, S_2 \in P} \; S_1 \neq S_2 \implies S_1 \cap S_2 = \emptyset$

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

Relations (cont'd)

Let R be a *equivalence relation* over the set S.

The equivalence class of $s \in S$ under R is the set

$$[s]_R \stackrel{def}{=} \{s' \in S : sRs'\}$$

If R is a congruence relation over S, then $[s]_R$ is the congruence class of s.
A partition P of S is a set of subsets of S that is

1. total: $\left(\bigcup_{S' \in P} S'\right) = S$
2. disjoint: $\forall_{S_1, S_2 \in P} S_1 \neq S_2 \implies S_1 \cap S_2 = \emptyset$

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

Relations (cont'd)

Let R be a *equivalence relation* over the set S.

The equivalence class of $s \in S$ under R is the set

$$[s]_R \stackrel{def}{=} \{s' \in S : sRs'\}$$

If R is a congruence relation over S, then $[s]_R$ is the congruence class of s.
A partition P of S is a set of subsets of S that is

1. total: $\left(\bigcup_{S' \in P} S'\right) = S$
2. disjoint: $\forall_{S_1, S_2 \in P} S_1 \neq S_2 \implies S_1 \cap S_2 = \emptyset$

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

Relations (cont'd)

Let R be a *equivalence relation* over the set S.

The equivalence class of $s \in S$ under R is the set

$$[s]_R \stackrel{def}{=} \{s' \in S : sRs'\}$$

If R is a congruence relation over S, then $[s]_R$ is the congruence class of s.
A partition P of S is a set of subsets of S that is

1. total: $\left(\bigcup_{S' \in P} S'\right) = S$
2. disjoint: $\forall_{S_1, S_2 \in P} S_1 \neq S_2 \implies S_1 \cap S_2 = \emptyset$

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

Relations (cont'd)

Let R be a *equivalence relation* over the set S.

The equivalence class of $s \in S$ under R is the set

$$[s]_R \stackrel{def}{=} \{s' \in S : sRs'\}$$

If R is a congruence relation over S, then $[s]_R$ is the congruence class of s.
A partition P of S is a set of subsets of S that is

1. total: $\left(\bigcup_{S' \in P} S'\right) = S$
2. disjoint: $\forall_{S_1, S_2 \in P} S_1 \neq S_2 \implies S_1 \cap S_2 = \emptyset$

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

Relations (cont'd)

Let R_1 and R_2 be two binary relations over set S.

R_1 is a refinement of R_2, or $R_1 \prec R_2$, if $\underset{s_1, s_2 \in S}{\forall} s_1 R_1 s_2 \implies s_1 R_2 s_2$.

In other words, R_1 refines R_2.

Viewing the relations as sets of pairs, $R_1 \prec R_2$ iff $R_1 \subseteq R_2$.

Examples

- Let $S = a, b$, $R_1 : aR_1b$, $R_2 : aR_2b, bR_2b$. Then $R_1 \prec R_2$.
- Let S be a set.
  Relation $R_1 : sR_1s : s \in S$ induced by the partition $P_1 : s : s \in S$;
  Relation $R_2 : sR_2t : s, t \in S$ induced by the partition $P_2 : S$.
  Then $R_1 \prec R_2$.

Relations (cont'd)

Let R_1 and R_2 be two binary relations over set S.

R_1 is a refinement of R_2, or $R_1 \prec R_2$, if $\underset{s_1, s_2 \in S}{\forall} s_1 R_1 s_2 \implies s_1 R_2 s_2$.

In other words, R_1 refines R_2.

Viewing the relations as sets of pairs, $R_1 \prec R_2$ iff $R_1 \subseteq R_2$.

Examples

- Let $S = a, b$, $R_1 : aR_1b$, $R_2 : aR_2b, bR_2b$. Then $R_1 \prec R_2$.
- Let S be a set.
  Relation $R_1 : sR_1s : s \in S$ induced by the partition $P_1 : s : s \in S$;
  Relation $R_2 : sR_2t : s, t \in S$ induced by the partition $P_2 : S$.
  Then $R_1 \prec R_2$.

Relations (cont'd)

Let R_1 and R_2 be two binary relations over set S.

R_1 is a refinement of R_2, or $R_1 \prec R_2$, if $\underset{s_1, s_2 \in S}{\forall} s_1 R_1 s_2 \implies s_1 R_2 s_2$.

In other words, R_1 refines R_2.

Viewing the relations as sets of pairs, $R_1 \prec R_2$ iff $R_1 \subseteq R_2$.

Examples

- Let $S = a, b$, $R_1 : aR_1b$, $R_2 : aR_2b, bR_2b$. Then $R_1 \prec R_2$.
- Let S be a set.
  Relation $R_1 : sR_1s : s \in S$ induced by the partition $P_1 : s : s \in S$;
  Relation $R_2 : sR_2t : s, t \in S$ induced by the partition $P_2 : S$.
  Then $R_1 \prec R_2$.

Relations (cont'd)

Let R_1 and R_2 be two binary relations over set S.

R_1 is a refinement of R_2, or $R_1 \prec R_2$, if $\underset{s_1, s_2 \in S}{\forall} s_1 R_1 s_2 \implies s_1 R_2 s_2$.

In other words, R_1 refines R_2.

Viewing the relations as sets of pairs, $R_1 \prec R_2$ iff $R_1 \subseteq R_2$.

Examples

- Let $S = a, b$, $R_1 : aR_1 b$, $R_2 : aR_2 b, bR_2 b$. Then $R_1 \prec R_2$.
- Let S be a set.
  Relation $R_1 : sR_1 s : s \in S$ induced by the partition $P_1 : s : s \in S$;
  Relation $R_2 : sR_2 t : s, t \in S$ induced by the partition $P_2 : S$.
  Then $R_1 \prec R_2$.

Relations (cont'd)

Let R_1 and R_2 be two binary relations over set S.

R_1 is a refinement of R_2, or $R_1 \prec R_2$, if $\underset{s_1, s_2 \in S}{\forall} \; s_1 R_1 s_2 \implies s_1 R_2 s_2$.

In other words, R_1 refines R_2.

Viewing the relations as sets of pairs, $R_1 \prec R_2$ iff $R_1 \subseteq R_2$.

Examples

- Let $S = a, b$, $R_1 : aR_1b$, $R_2 : aR_2b, bR_2b$. Then $R_1 \prec R_2$.
- Let S be a set.
  Relation $R_1 : sR_1s : s \in S$ induced by the partition $P_1 : s : s \in S$;
  Relation $R_2 : sR_2t : s, t \in S$ induced by the partition $P_2 : S$.
  Then $R_1 \prec R_2$.

Relations (cont'd)

Let R_1 and R_2 be two binary relations over set S.

R_1 is a refinement of R_2, or $R_1 \prec R_2$, if $\underset{s_1, s_2 \in S}{\forall} s_1 R_1 s_2 \implies s_1 R_2 s_2$.

In other words, R_1 refines R_2.

Viewing the relations as sets of pairs, $R_1 \prec R_2$ iff $R_1 \subseteq R_2$.

Examples

- Let $S = a, b$, $R_1 : aR_1 b$, $R_2 : aR_2 b, bR_2 b$. Then $R_1 \prec R_2$.
- Let S be a set.
  Relation $R_1 : sR_1 s : s \in S$ induced by the partition $P_1 : s : s \in S$;
  Relation $R_2 : sR_2 t : s, t \in S$ induced by the partition $P_2 : S$.
  Then $R_1 \prec R_2$.

Relations (cont'd)

Let R_1 and R_2 be two binary relations over set S.

R_1 is a refinement of R_2, or $R_1 \prec R_2$, if $\underset{s_1, s_2 \in S}{\forall} s_1 R_1 s_2 \implies s_1 R_2 s_2$.

In other words, R_1 refines R_2.

Viewing the relations as sets of pairs, $R_1 \prec R_2$ iff $R_1 \subseteq R_2$.

Examples

- Let $S = a, b$, $R_1 : aR_1 b$, $R_2 : aR_2 b, bR_2 b$. Then $R_1 \prec R_2$.
- Let S be a set.
  Relation $R_1 : sR_1 s : s \in S$ induced by the partition $P_1 : s : s \in S$;
  Relation $R_2 : sR_2 t : s, t \in S$ induced by the partition $P_2 : S$.
  Then $R_1 \prec R_2$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The equivalence closure R^E of the binary relation R over S is the equivalence relation such that

- R refines R^E: $R \prec R_E$;
- for all other equivalence relations R' such that $R \prec R'$, either $R' = R^E$ or $R^E \prec R'$

In other words, R^E is the "smallest" equivalence relation that "covers" R.

The congruence closure R^C of R is the "smallest" congruence relation that "covers" R.

Examples If $S = \{a, b, c, d\}$ and $R = \{aRb, bRc, dRd\}$, then

- $aRb, bRc, dRd \in R^E$ since $R \subseteq R^E$
- $aRa, bRb, cRc \in R^E$ by reflexivity
- $bRa, cRb \in R^E$ by symmetry;
- $aRc \in R^E$ by transitivity;
- $cRa \in R^E$ by symmetry

Hence, $R^E = \{aRb, bRa, aRa, bRb, bRc, cRb, cRc, aRc, cRa, dRd\}$.

Relations (cont'd)

The subterm set S_F of Σ-formula F is the set that contains precisely the subterms of F.

Example: Let

$$F \quad :\iff \quad f[a, b] = a \wedge f[f[a, b], b] \neq a.$$

Then

$$S_F = \{a, b, f[a, b], f[f[a, b], b]\}.$$

Relations (cont'd)

The subterm set S_F of Σ-formula F is the set that contains precisely the subterms of F.

Example: Let

$$F \;:\iff\; f[a,b] = a \land f[f[a,b],b] \neq a.$$

Then

$$S_F = \{a, b, f[a,b], f[f[a,b],b]\}.$$

Congruence Closure Algorithm for T_{QFEUF}

Given Σ_E - formula F

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

with subterm set S_F. F is T_E - satisfiable iff there exists a congruence relation over S_F such that

- for each $i \in \{1, \ldots, m\}$, $s_i \sim t_i$;
- for each $i \in \{m+1, \ldots, n\}$, $s_i \not\sim t_i$.

Congruence Closure Algorithm (Naive Version)

1. Construct the congruence closure $\sim$ of

$$\{s_1 = t_1, \ldots, s_m = t_m\}$$

over the subterm set S_F. Then

$$\sim \models s_1 = t_1 \wedge \ldots \wedge s_m = t_m$$

2. If $s_i \sim t_i$ for any $i \in \{m+1, \ldots, n\}$, return unsatisfiable.
3. Otherwise, $\sim \models F$, so return satisfiable.

Congruence Closure Algorithm for T_{QFEUF}

Given Σ_E - formula F

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

with subterm set S_F. F is T_E - satisfiable iff there exists a congruence relation over S_F such that

- for each $i \in \{1, \ldots, m\}$, $s_i \sim t_i$;
- for each $i \in \{m+1, \ldots, n\}$, $s_i \not\sim t_i$.

Congruence Closure Algorithm (Naive Version)

1. Construct the congruence closure $\sim$ of

$$\{s_1 = t_1, \ldots, s_m = t_m\}$$

   over the subterm set S_F. Then

$$\sim \models s_1 = t_1 \wedge \ldots \wedge s_m = t_m$$

2. If $s_i \sim t_i$ for any $i \in \{m+1, \ldots, n\}$, return unsatisfiable.
3. Otherwise, $\sim \models F$, so return satisfiable.

Congruence Closure Algorithm for T_{QFEUF}

Given Σ_E - formula F

$$F \ :\Longleftrightarrow \ s_1 = t_1 \land \ldots \land s_m = t_m \land s_{m+1} \neq t_{m+1} \land \ldots \land s_n \neq t_n$$

with subterm set S_F. F is T_E - satisfiable iff there exists a congruence relation over S_F such that

- for each $i \in \{1, \ldots, m\}$, $s_i \sim t_i$;
- for each $i \in \{m+1, \ldots, n\}$, $s_i \not\sim t_i$.

Congruence Closure Algorithm (Naive Version)

1. Construct the congruence closure $\sim$ of

$$\{s_1 = t_1, \ldots, s_m = t_m\}$$

over the subterm set S_F. Then

$$\sim \models s_1 = t_1 \land \ldots \land s_m = t_m$$

2. If $s_i \sim t_i$ for any $i \in \{m+1, \ldots, n\}$, return unsatisfiable.
3. Otherwise, $\sim \models F$, so return satisfiable.

Congruence Closure Algorithm for T_{QFEUF}

Given Σ_E - formula F

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

with subterm set S_F. F is T_E - satisfiable iff there exists a congruence relation over S_F such that

- for each $i \in \{1, \ldots, m\}$, $s_i \sim t_i$;
- for each $i \in \{m+1, \ldots, n\}$, $s_i \not\sim t_i$.

Congruence Closure Algorithm (Naive Version)

1. Construct the congruence closure $\sim$ of

$$\{s_1 = t_1, \ldots, s_m = t_m\}$$

over the subterm set S_F. Then

$$\sim \models s_1 = t_1 \wedge \ldots \wedge s_m = t_m$$

2. If $s_i \sim t_i$ for any $i \in \{m+1, \ldots, n\}$, return unsatisfiable.
3. Otherwise, $\sim \models F$, so return satisfiable.

Congruence Closure Algorithm for T_{QFEUF}

Given Σ_E - formula F

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

with subterm set S_F. F is T_E - satisfiable iff there exists a congruence relation over S_F such that

- for each $i \in \{1, \dots, m\}$, $s_i \sim t_i$;
- for each $i \in \{m+1, \dots, n\}$, $s_i \not\sim t_i$.

Congruence Closure Algorithm (Naive Version)

1. Construct the congruence closure $\sim$ of

$$\{s_1 = t_1, \dots, s_m = t_m\}$$

   over the subterm set S_F. Then

$$\sim \models s_1 = t_1 \wedge \dots \wedge s_m = t_m$$

2. If $s_i \sim t_i$ for any $i \in \{m+1, \dots, n\}$, return unsatisfiable.
3. Otherwise, $\sim \models F$, so return satisfiable.

Congruence Closure Algorithm for T_{QFEUF}

Given Σ_E - formula F

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

with subterm set S_F. F is T_E - satisfiable iff there exists a congruence relation over S_F such that

- for each $i \in \{1, \ldots, m\}$, $s_i \sim t_i$;
- for each $i \in \{m+1, \ldots, n\}$, $s_i \not\sim t_i$.

Congruence Closure Algorithm (Naive Version)

1. Construct the congruence closure $\sim$ of

$$\{s_1 = t_1, \ldots, s_m = t_m\}$$

   over the subterm set S_F. Then

$$\sim \models s_1 = t_1 \wedge \ldots \wedge s_m = t_m$$

2. If $s_i \sim t_i$ for any $i \in \{m+1, \ldots, n\}$, return unsatisfiable.
3. Otherwise, $\sim \models F$, so return satisfiable.

Congruence Closure Algorithm for T_{QFEUF}

Given Σ_E - formula F

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

with subterm set S_F. F is T_E - satisfiable iff there exists a congruence relation over S_F such that

- for each $i \in \{1, \ldots, m\}$, $s_i \sim t_i$;
- for each $i \in \{m+1, \ldots, n\}$, $s_i \not\sim t_i$.

Congruence Closure Algorithm (Naive Version)

1. Construct the congruence closure $\sim$ of

$$\{s_1 = t_1, \ldots, s_m = t_m\}$$

   over the subterm set S_F. Then

$$\sim \models s_1 = t_1 \wedge \ldots \wedge s_m = t_m$$

2. If $s_i \sim t_i$ for any $i \in \{m+1, \ldots, n\}$, return unsatisfiable.
3. Otherwise, $\sim \models F$, so return satisfiable.

Congruence Closure Algorithm for T_{QFEUF} (cont'd)

Examples: Determine if the following formulas are satisfiable or not

1. $F_1 \; :\Longleftrightarrow \quad f[a, b] = a \wedge f[f[a, b], b] \neq a$
2. $F_2 \; :\Longleftrightarrow \quad f[x] = f[y] \wedge x \neq y$