Monadic Logic

Monadic logic - Signature only monadic (unary) predicates.

Monadic Logic

Monadic logic - Signature only monadic (unary) predicates.

Theorem. The satisfiability problem for monadic logic is decidable.

Monadic Logic

Monadic logic - Signature only monadic (unary) predicates.

Theorem. The satisfiability problem for monadic logic is decidable.

Follows from
Lemma (Small Model Property) If a monadic formula ψ with k unary predicates is satisfiable then it is satisfiable in a structure with at most 2^k elements.

Monadic Logic

Monadic logic - Signature only monadic (unary) predicates.

Theorem. The satisfiability problem for monadic logic is decidable.

Follows from

Lemma (Small Model Property) If a monadic formula ψ with k unary predicates is satisfiable then it is satisfiable in a structure with at most 2^k elements.

and

Theorem. There is an algorithm that for a formula ϕ and a finite structure M checks whether ϕ is satisfiable in M.

Proof of small model property lemma

Proof of small model property lemma

Let M be a structure for the signature $\{P_1, \ldots P_k\}$ - all P_i are monadic predicate names.

Proof of small model property lemma

Let M be a structure for the signature $\{P_1, \dots P_k\}$ - all P_i are monadic predicate names.
Define an equivalence relation $\sim$ on the universe $|M|$ of M.

$$a \sim b \text{ iff}$$

$$a \in P_i^M \Leftrightarrow b \in P_i^M \text{ for } i = 1, \dots, k$$

Proof of small model property lemma

Let M be a structure for the signature $\{P_1, \dots P_k\}$ - all P_i are monadic predicate names.
Define an equivalence relation $\sim$ on the universe $|M|$ of M.

$$a \sim b \text{ iff}$$

$$a \in P_i^M \Leftrightarrow b \in P_i^M \text{ for } i = 1, \dots, k$$

The equivalence class of a is the set $\hat{a} = \{b \ : \ b \sim a\}$.

Proof of small model property lemma

Let M be a structure for the signature $\{P_1, \ldots P_k\}$ - all P_i are monadic predicate names.
Define an equivalence relation $\sim$ on the universe $|M|$ of M.

$$a \sim b \text{ iff}$$

$$a \in P_i^M \Leftrightarrow b \in P_i^M \text{ for } i = 1, \ldots, k$$

The equivalence class of a is the set $\hat{a} = \{b \ : \ b \sim a\}$.
The number of the $\sim$ equivalence classes is finite.

How many?

Proof of small model property lemma

Let M be a structure for the signature $\{P_1, \dots P_k\}$ - all P_i are monadic predicate names.
Define an equivalence relation $\sim$ on the universe $|M|$ of M.

$$a \sim b \text{ iff}$$

$$a \in P_i^M \Leftrightarrow b \in P_i^M \text{ for } i = 1, \dots, k$$

The equivalence class of a is the set $\hat{a} = \{b \ : \ b \sim a\}$.
The number of the $\sim$ equivalence classes is finite.

How many? at most 2^k

The structure of equivalence classes

Given a structure M for unary predicates.

We define a structure $\hat{M}$ of $\sim$ equivalence classes of M.

The structure of equivalence classes

Given a structure M for unary predicates.

We define a structure $\hat{M}$ of $\sim$ equivalence classes of M.

The Universe of $\hat{M}$: the set of $\sim$ equivalence classes of M.

The structure of equivalence classes

Given a structure M for unary predicates.

We define a structure $\hat{M}$ of $\sim$ equivalence classes of M.

The Universe of $\hat{M}$: the set of $\sim$ equivalence classes of M.

Interpretation of predicates: $\hat{a} \in P_i^{\hat{M}}$ iff $a \in P_i^M$

The structure of equivalence classes

Given a structure M for unary predicates.

We define a structure $\hat{M}$ of $\sim$ equivalence classes of M.

The Universe of $\hat{M}$: the set of $\sim$ equivalence classes of M.

Interpretation of predicates: $\hat{a} \in P_i^{\hat{M}}$ iff $a \in P_i^M$

Lemma. Let ρ and $\hat{\rho}$ be environments for M and $\hat{M}$ such that $\rho(x) \in \hat{\rho}(x)$ for all variables x. Then for every formula ψ

$$[|\psi|]^M \rho = [|\psi|]^{\hat{M}} \hat{\rho}$$

The structure of equivalence classes

Given a structure M for unary predicates.

We define a structure $\hat{M}$ of $\sim$ equivalence classes of M.

The Universe of $\hat{M}$: the set of $\sim$ equivalence classes of M.

Interpretation of predicates: $\hat{a} \in P_i^{\hat{M}}$ iff $a \in P_i^M$

Lemma. Let ρ and $\hat{\rho}$ be environments for M and $\hat{M}$ such that $\rho(x) \in \hat{\rho}(x)$ for all variables x. Then for every formula ψ

$$[|\psi|]^M \rho = [|\psi|]^{\hat{M}} \hat{\rho}$$

Proof: By structural inductions.

The structure of equivalence classes

Given a structure M for unary predicates.

We define a structure $\hat{M}$ of $\sim$ equivalence classes of M.

The Universe of $\hat{M}$: the set of $\sim$ equivalence classes of M.

Interpretation of predicates: $\hat{a} \in P_i^{\hat{M}}$ iff $a \in P_i^M$

Lemma. Let ρ and $\hat{\rho}$ be environments for M and $\hat{M}$ such that $\rho(x) \in \hat{\rho}(x)$ for all variables x. Then for every formula ψ

$$[|\psi|]^M \rho = [|\psi|]^{\hat{M}} \hat{\rho}$$

Proof: By structural inductions.

Corollary. ψ is satisfiable in M iff ψ is satisfiable in $\hat{M}$.

The structure of equivalence classes

Given a structure M for unary predicates.

We define a structure $\hat{M}$ of $\sim$ equivalence classes of M.

The Universe of $\hat{M}$: the set of $\sim$ equivalence classes of M.

Interpretation of predicates: $\hat{a} \in P_i^{\hat{M}}$ iff $a \in P_i^M$

Lemma. Let ρ and $\hat{\rho}$ be environments for M and $\hat{M}$ such that $\rho(x) \in \hat{\rho}(x)$ for all variables x. Then for every formula ψ

$$[|\psi|]^M \rho = [|\psi|]^{\hat{M}} \hat{\rho}$$

Proof: By structural inductions.

Corollary. ψ is satisfiable in M iff ψ is satisfiable in $\hat{M}$.

Corollary. If ψ is satisfiable then ψ is satisfiable in a structure with at most 2^k elements.

Quantifier Elimination

Quantifier Elimination

Theorem. The satisfiability problem for formulas in the signature $\{<, =\}$ over the rationals is decidable.

Quantifier Elimination

Theorem. The satisfiability problem for formulas in the signature $\{<, =\}$ over the rationals is decidable.
Follows from
Lemma (Quantifier Elimination)

1. Every formula is equivalent over Q to a quantifier free formula.

Quantifier Elimination

Theorem. The satisfiability problem for formulas in the signature $\{<, =\}$ over the rationals is decidable.
Follows from
Lemma (Quantifier Elimination)

1. Every formula is equivalent over Q to a quantifier free formula.
2. There is an algorithm which for every ϕ in the signature $\{<, =\}$ constructs a quantifier free ψ such that ψ is equivalent to ϕ over the rationals,

Quantifier Elimination

Theorem. The satisfiability problem for formulas in the signature $\{<, =\}$ over the rationals is decidable.
Follows from
Lemma (Quantifier Elimination)

1. Every formula is equivalent over Q to a quantifier free formula.
2. There is an algorithm which for every ϕ in the signature $\{<, =\}$ constructs a quantifier free ψ such that ψ is equivalent to ϕ over the rationals,

and

Quantifier Elimination

Theorem. The satisfiability problem for formulas in the signature $\{<, =\}$ over the rationals is decidable.
Follows from
Lemma (Quantifier Elimination)

1. Every formula is equivalent over Q to a quantifier free formula.
2. There is an algorithm which for every ϕ in the signature $\{<, =\}$ constructs a quantifier free ψ such that ψ is equivalent to ϕ over the rationals,

and
Lemma. The satisfiability problem over Q for quantifier free formulas is decidable.

How to check the satisfiability of Quantifier free formulas

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable?

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.

Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable?

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$ satisfiable?

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$ satisfiable? No.

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.

Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.

Is $x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$ satisfiable? No.

Is $x_1 < x_2 \wedge x_2 = x_3 \wedge x_3 < x_1$ satisfiable?

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$ satisfiable? No.
Is $x_1 < x_2 \wedge x_2 = x_3 \wedge x_3 < x_1$ satisfiable? No.

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$ satisfiable? No.
Is $x_1 < x_2 \wedge x_2 = x_3 \wedge x_3 < x_1$ satisfiable? No.

How to check whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$ satisfiable? No.
Is $x_1 < x_2 \wedge x_2 = x_3 \wedge x_3 < x_1$ satisfiable? No.

How to check whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

Algorithm:

1. Construct graph:
   - Nodes - variables
   - Edges - put an edge from x_i to x_j if $x_i < x_j$.

How to check the satisfiability of Quantifier free formulas

Is $x_1 < x_2 \wedge x_3 < x_2$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_3 < x_2 \wedge x_3 < x_1$ satisfiable? Yes.
Is $x_1 < x_2 \wedge x_2 < x_3 \wedge x_3 < x_1$ satisfiable? No.
Is $x_1 < x_2 \wedge x_2 = x_3 \wedge x_3 < x_1$ satisfiable? No.

How to check whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

Algorithm:

1. Construct graph:
   - Nodes - variables
   - Edges - put an edge from x_i to x_j if $x_i < x_j$.
2. Check that the graph is cycle free.

How to check the satisfiability of Quantifier free formulas

How to check whether a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$ is satisfiable?

How to check the satisfiability of Quantifier free formulas

How to check whether a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$ is satisfiable?
Reduce to the verification whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

How to check the satisfiability of Quantifier free formulas

How to check whether a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$ is satisfiable?

Reduce to the verification whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

Lemma. Let ϕ be a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$.

How to check the satisfiability of Quantifier free formulas

How to check whether a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$ is satisfiable?

Reduce to the verification whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

Lemma. Let ϕ be a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$.

1. There is ψ such that ϕ is satisfiable (over Q) iff ψ is, and ψ is a conjunction of formulas of the form $x_i < x_j$.

How to check the satisfiability of Quantifier free formulas

How to check whether a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$ is satisfiable?

Reduce to the verification whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

Lemma. Let ϕ be a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$.

1. There is ψ such that ϕ is satisfiable (over Q) iff ψ is, and ψ is a conjunction of formulas of the form $x_i < x_j$.
2. Moreover, there is an algorithm that constructs ψ from ϕ.

How to check the satisfiability of Quantifier free formulas

How to check whether a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$ is satisfiable?

Reduce to the verification whether a conjunction of formulas of the form $x_i < x_j$ is satisfiable?

Lemma. Let ϕ be a conjunction of formulas of the form $x_i < x_j$ and $x_m = x_l$.

1. There is ψ such that ϕ is satisfiable (over Q) iff ψ is, and ψ is a conjunction of formulas of the form $x_i < x_j$.
2. Moreover, there is an algorithm that constructs ψ from ϕ.

Algorithm. If there is a conjunct $x_m = x_l$ replace all occurrences of x_m by x_l and remove the conjunct $x_m = x_l$.

How to check the satisfiability of Quantifier free formulas

1. Construct negation normal form ($\neg$ appears only before atoms).

How to check the satisfiability of Quantifier free formulas

1. Construct negation normal form ($\neg$ appears only before atoms).
2. Eliminate Negations: (a)replace $\neg x_i < x_j$ by $x_i = x_j \vee x_i > x_j$; (b) replace $\neg x_i = x_j$ by $x_i < x_j \vee x_j < x_i$.

How to check the satisfiability of Quantifier free formulas

1. Construct negation normal form ($\neg$ appears only before atoms).
2. Eliminate Negations: (a)replace $\neg x_i < x_j$ by $x_i = x_j \vee x_i > x_j$; (b) replace $\neg x_i = x_j$ by $x_i < x_j \vee x_j < x_i$.
3. Construct DNF: $\bigvee_i \bigwedge_j c_{i,j}$ where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.

How to check the satisfiability of Quantifier free formulas

1. Construct negation normal form ($\neg$ appears only before atoms).
2. Eliminate Negations: (a)replace $\neg x_i < x_j$ by $x_i = x_j \vee x_i > x_j$; (b) replace $\neg x_i = x_j$ by $x_i < x_j \vee x_j < x_i$.
3. Construct DNF: $\bigvee_i \bigwedge_j c_{i,j}$ where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.
4. Check if there is i such that $\bigwedge_j c_{i,j}$ is satisfiable

How to check the satisfiability of Quantifier free formulas

1. Construct negation normal form ($\neg$ appears only before atoms).
2. Eliminate Negations: (a)replace $\neg x_i < x_j$ by $x_i = x_j \vee x_i > x_j$; (b) replace $\neg x_i = x_j$ by $x_i < x_j \vee x_j < x_i$.
3. Construct DNF: $\bigvee_i \bigwedge_j c_{i,j}$ where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.
4. Check if there is i such that $\bigwedge_j c_{i,j}$ is satisfiable

Hence

Lemma. The satisfiability problem over Q for quantifier free formulas is decidable.

Quantifier Elimination

Lemma. Every quantifier free formula is equivalent over linear orders to a formula of the form

$$\bigvee_i \bigwedge_j c_{i,j},$$

where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.

Quantifier Elimination

Lemma. Every quantifier free formula is equivalent over linear orders to a formula of the form

$$\bigvee_i \bigwedge_j c_{i,j},$$

where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.

Theorem (Quantifier Elimination)

Quantifier Elimination

Lemma. Every quantifier free formula is equivalent over linear orders to a formula of the form

$$\bigvee_i \bigwedge_j c_{i,j},$$

where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.

Theorem (Quantifier Elimination)

1. Every formula is equivalent over Q to a quantifier free formula.

Quantifier Elimination

Lemma. Every quantifier free formula is equivalent over linear orders to a formula of the form

$$\bigvee_i \bigwedge_j c_{i,j},$$

where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.

Theorem (Quantifier Elimination)

1. Every formula is equivalent over Q to a quantifier free formula.
2. There is an algorithm which for every ϕ in the signature $\{<,=\}$ constructs a quantifier free ψ such that ψ is equivalent to ϕ over the rationals,

Quantifier Elimination

Lemma. Every quantifier free formula is equivalent over linear orders to a formula of the form

$$\bigvee_i \bigwedge_j c_{i,j},$$

where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$.

Theorem (Quantifier Elimination)

1. Every formula is equivalent over Q to a quantifier free formula.
2. There is an algorithm which for every ϕ in the signature $\{<, =\}$ constructs a quantifier free ψ such that ψ is equivalent to ϕ over the rationals,

Proof by induction on the number of quantifiers.

Proof of Quantifier Elimination

Basis is trivial

Proof of Quantifier Elimination

Basis is trivial

Inductive step it is enough to show

Lemma. $\exists x_k \bigvee_i \bigwedge_j c_{i,j}$, where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$ is equivalent to a quantifier free formula.

Proof of Quantifier Elimination

Basis is trivial
Inductive step it is enough to show

Lemma. $\exists x_k \bigvee_i \bigwedge_j c_{i,j}$, where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$ is equivalent to a quantifier free formula.

$$\exists x_k \bigvee_i \bigwedge_j c_{i,j} \text{ equivalent to } \bigvee_i \exists x_k \bigwedge_j c_{i,j}$$

Proof of Quantifier Elimination

Basis is trivial
Inductive step it is enough to show

Lemma. $\exists x_k \bigvee_i \bigwedge_j c_{i,j}$, where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$ is equivalent to a quantifier free formula.

$$\exists x_k \bigvee_i \bigwedge_j c_{i,j} \text{ equivalent to } \bigvee_i \exists x_k \bigwedge_j c_{i,j}$$

$\exists x_k \bigwedge_j c_j$ is equivalent to $\bigwedge_{j \notin M} c_j \wedge \exists x_k \bigwedge_{j \in M} c_j$, where $M = \{i \ : \ x_k$ occurs in $c_i\}$.

Proof of Quantifier Elimination

Basis is trivial
Inductive step it is enough to show

Lemma. $\exists x_k \bigvee_i \bigwedge_j c_{i,j}$, where $c_{i,j}$ is of the form $x_m < x_l$ or $x_m = x_l$ is equivalent to a quantifier free formula.

$$\exists x_k \bigvee_i \bigwedge_j c_{i,j} \text{ equivalent to } \bigvee_i \exists x_k \bigwedge_j c_{i,j}$$

$\exists x_k \bigwedge_j c_j$ is equivalent to $\bigwedge_{j \notin M} c_j \wedge \exists x_k \bigwedge_{j \in M} c_j$, where $M = \{i \ : \ x_k \text{ occurs in } c_i\}$.
If $x_k = x_l$ is one of the conjuncts in $\exists x_k \bigwedge_{j \in M} c_j$ then this formula is equivalent to

$$\bigwedge_{j \in M} c_j\{x_l/x_k\}$$

Proof of Quantifier Elimination

The last case to consider:

$$\exists x_k \bigwedge_j c_j$$

where every c_j is of the form $x_l < x_k$ or $x_k < x_l$ or $x_k < x_k$.

Proof of Quantifier Elimination

The last case to consider:

$$\exists x_k \bigwedge_j c_j$$

where every c_j is of the form $x_l < x_k$ or $x_k < x_l$ or $x_k < x_k$.
If $x_k < x_k$ is one of the conjunct replace the formula by
$x_k < x_k$

Proof of Quantifier Elimination

The last case to consider:

$$\exists x_k \bigwedge_j c_j$$

where every c_j is of the form $x_l < x_k$ or $x_k < x_l$ or $x_k < x_k$.
If $x_k < x_k$ is one of the conjunct replace the formula by $x_k < x_k$
Otherwise replace it by $\bigwedge_{l \in L, r \in R} x_l < x_r$ where $l \in L$ iff $x_l < x_k$ is one of the conjuncts and $r \in R$ iff $x_k < x_r$ is one of the conjuncts.

Example:

$\exists x_1 (x_1 < x_2 \wedge x_1 < x_3 \wedge x_4 < x_1 \wedge x_5 < x_1)$ is equivalent to

$$x_4 < x_2 \wedge x_4 < x_3 \wedge x_5 < x_2 \wedge x_5 < x_3$$

Validity Problem

Validity Problem
Instance: a formula ψ
Question: Is ψ valid?

Validity Problem

Validity Problem
Instance: a formula ψ
Question: Is ψ valid?

Theorem (Church) The validity problem is undecidable. (i.e., there is no algorithm for the validity problem).

How to show undecidability

How to show undecidability

1. Deep analysis of the notion of algorithm.

How to show undecidability

1. Deep analysis of the notion of algorithm.

The analysis is done in the course of Automata and Computability

How to show undecidability

1. Deep analysis of the notion of algorithm.

The analysis is done in the course of Automata and Computability

The halting problem is undecidable

Instance: a Turing machine M

Question: Does M stops on all inputs.

How to show undecidability

1. Deep analysis of the notion of algorithm.

The analysis is done in the course of Automata and Computability
The halting problem is undecidable
Instance: a Turing machine M
Question: Does M stops on all inputs.

2. Reduction techniques

Reduction Technique - an Example

Reduction Technique - an Example

Satisfiability Problem

Instance: a sentence ψ

Question: Is ψ satisfiable?

Reduction Technique - an Example

Satisfiability Problem
Instance: a sentence ψ
Question: Is ψ satisfiable?

Reduction from Validity to satisfiability.
ϕ is valid iff $\neg\phi$ is unsatisfiable.

Since Validity problem is undecidable we obtain that satisfiability problem is undecidable.

Reduction Technique

Reduction Technique

Assume that P_1 and P_2 are two decision problems

Def. An algorithm f is a reduction from P_1 to P_2 if it maps every Yes-instance of P_1 to a Yes-instance of P_2 and every No-instance of P_1 to a No-instance of P_2

Reduction Technique

Assume that P_1 and P_2 are two decision problems

Def. An algorithm f is a reduction from P_1 to P_2 if it maps every Yes-instance of P_1 to a Yes-instance of P_2 and every No-instance of P_1 to a No-instance of P_2

Theorem. If P_1 is undecidable and there is a reduction from P_1 to P_2 then P_2 is undecidable.

Post Correspondence Problem

Post Problem
Instance: a sequence $\langle a_1, b_1\rangle\langle a_2, b_2\rangle \ldots \langle a_k, b_k\rangle$ of pairs of strings over $\{0, 1\}$.

Post Correspondence Problem

Post Problem

Instance: a sequence $\langle a_1, b_1 \rangle \langle a_2, b_2 \rangle \ldots \langle a_k, b_k \rangle$ of pairs of strings over$\{0, 1\}$.

A sequence $i_1 i_2 \ldots i_m$ (for $i_l \in \{1, \ldots k\}$) is a solution for this instance if the string $a_{i_1} a_{i_2} \ldots a_{i_m}$ is the same as the string $b_{i_1} b_{i_2} \ldots b_{i_m}$

Post Correspondence Problem

Post Problem
Instance: a sequence $\langle a_1, b_1\rangle\langle a_2, b_2\rangle \ldots \langle a_k, b_k\rangle$ of pairs of strings over$\{0, 1\}$.
A sequence $i_1 i_2 \ldots i_m$ (for $i_l \in \{1, \ldots k\}$) is a solution for this instance if the string $a_{i_1} a_{i_2} \ldots a_{i_m}$ is the same as the string $b_{i_1} b_{i_2} \ldots b_{i_m}$
Decision Question: Does an instance has a solution.

Post Correspondence Problem

Post Problem

Instance: a sequence $\langle a_1, b_1\rangle\langle a_2, b_2\rangle \dots \langle a_k, b_k\rangle$ of pairs of strings over$\{0, 1\}$.

A sequence $i_1 i_2 \dots i_m$ (for $i_l \in \{1, \dots k\}$) is a solution for this instance if the string $a_{i_1} a_{i_2} \dots a_{i_m}$ is the same as the string $b_{i_1} b_{i_2} \dots b_{i_m}$

Decision Question: Does an instance has a solution.

Theorem(Post) The Post problem is undecidable.

Reduction of Post Problem to Validity Problem

Reduction of Post Problem to Validity Problem

Signature a constant c, two unary function f_0, f_1 and a binary predicate R

Reduction of Post Problem to Validity Problem

Signature a constant c, two unary function f_0, f_1 and a binary predicate R

For every instance I of the Post Problem we construct a formula ψ^I such that I has a solution iff ψ^I is valid.

Reduction of Post Problem to Validity Problem

Signature a constant c, two unary function f_0, f_1 and a binary predicate R

For every instance I of the Post Problem we construct a formula ψ^I such that I has a solution iff ψ^I is valid.

Notations For a string $u = \alpha_1, \dots \alpha_m$ over $\{0, 1\}$ we write $f_u(x)$ as a shorthand for $f_{\alpha_1}(f_{\alpha_2}(\dots(f_{\alpha_m}(x)))\dots)$.

Reduction of Post Problem to Validity Problem

Reduction of Post Problem to Validity Problem

Construction of ψ^I for an instance
$I = \langle a_1, b_1\rangle\langle a_2, b_2\rangle \ldots \langle a_k, b_k\rangle$ of Post problem

Let ϕ_1^I be

$$R(f_{a_1}(c), f_{b_1}(c)) \wedge \cdots \wedge R(f_{a_k}(c), f_{b_k}(c))$$

Reduction of Post Problem to Validity Problem

Construction of ψ^I for an instance
$I = \langle a_1, b_1 \rangle \langle a_2, b_2 \rangle \ldots \langle a_k, b_k \rangle$ of Post problem

Let ϕ_1^I be

$$R(f_{a_1}(c), f_{b_1}(c)) \wedge \cdots \wedge R(f_{a_k}(c), f_{b_k}(c))$$

Let ϕ_2^I be

$$\forall x \forall y R(x, y) \to (R(f_{a_1}(x), f_{b_1}(y)) \wedge \cdots \wedge R(f_{a_k}(x), f_{b_k}(y)))$$

Reduction of Post Problem to Validity Problem

Construction of ψ^I for an instance
$I = \langle a_1, b_1\rangle\langle a_2, b_2\rangle \dots \langle a_k, b_k\rangle$ of Post problem

Let ϕ_1^I be

$$R(f_{a_1}(c), f_{b_1}(c)) \wedge \cdots \wedge R(f_{a_k}(c), f_{b_k}(c))$$

Let ϕ_2^I be

$$\forall x \forall y R(x, y) \rightarrow (R(f_{a_1}(x), f_{b_1}(y)) \wedge \cdots \wedge R(f_{a_k}(x), f_{b_k}(y)))$$

Let ψ^I be $\phi_1^I \wedge \phi_2^I \rightarrow \exists x R(x, x)$

Correctness of the Reduction

Correctness of the Reduction

Theorem(Correctness of the reduction) An instance I of the Post problem has a solution if and only if ψ^I is a valid formula.

Correctness of the Reduction

Theorem(Correctness of the reduction) An instance I of the Post problem has a solution if and only if ψ^I is a valid formula.

The $\Rightarrow$ direction of the Theorem follows from
Lemma Let $i_1, \ldots, i_n$ be a sequence over $\{1, \ldots, k\}$. Then

$$\phi_1^I \wedge \phi_2^I \models R(f_{a_{i_1} a_{i_2} \ldots a_{i_n}}(c), f_{b_{i_1} b_{i_2} \ldots b_{i_n}}(c))$$

Correctness of the Reduction

Theorem(Correctness of the reduction) An instance I of the Post problem has a solution if and only if ψ^I is a valid formula.

The $\Rightarrow$ direction of the Theorem follows from
Lemma Let $i_1, \ldots, i_n$ be a sequence over $\{1, \ldots, k\}$. Then

$$\phi_1^I \wedge \phi_2^I \models R(f_{a_{i_1} a_{i_2} \ldots a_{i_n}}(c), f_{b_{i_1} b_{i_2} \ldots b_{i_n}}(c))$$

Proof By induction on n.
Basis $n = 1$ - follows from ϕ_1^I

Correctness of the Reduction

Theorem(Correctness of the reduction) An instance I of the Post problem has a solution if and only if ψ^I is a valid formula.

The $\Rightarrow$ direction of the Theorem follows from
Lemma Let $i_1, \ldots, i_n$ be a sequence over $\{1, \ldots, k\}$. Then

$$\phi_1^I \wedge \phi_2^I \models R(f_{a_{i_1} a_{i_2} \ldots a_{i_n}}(c), f_{b_{i_1} b_{i_2} \ldots b_{i_n}}(c))$$

Proof By induction on n.
Basis $n = 1$ - follows from ϕ_1^I
Inductive Step $n \to n + 1$ - use ϕ_2^I.

The $\Leftarrow$ direction of the Correctness Theorem.

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:
The Universe all strings over $\{0, 1\}$.

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:
The Universe all strings over $\{0, 1\}$.
The interpretation of f_1: $f_1^M(u) = 1u$ (put 1 in the front of the string u).

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:

The Universe all strings over $\{0, 1\}$.

The interpretation of f_1: $f_1^M(u) = 1u$ (put 1 in the front of the string u).

The interpretation of f_0: $f_0^M(u) = 0u$

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:
The Universe all strings over $\{0, 1\}$.
The interpretation of f_1: $f_1^M(u) = 1u$ (put 1 in the front of the string u).
The interpretation of f_0: $f_0^M(u) = 0u$
The interpretation of c: the empty string ϵ.

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:
The Universe all strings over $\{0, 1\}$.
The interpretation of f_1: $f_1^M(u) = 1u$ (put 1 in the front of the string u).
The interpretation of f_0: $f_0^M(u) = 0u$
The interpretation of c: the empty string ϵ.
The interpretation of R: $R^M(u, v)$ iff there is a sequence $i_1, \ldots, i_n$ such that $u \equiv a_{i_1} a_{i_2} \cdots a_{i_n}$ and $v \equiv b_{i_1} b_{i_2} \cdots b_{i_n}$.

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:
The Universe all strings over $\{0, 1\}$.
The interpretation of f_1: $f_1^M(u) = 1u$ (put 1 in the front of the string u).
The interpretation of f_0: $f_0^M(u) = 0u$
The interpretation of c: the empty string ϵ.
The interpretation of R: $R^M(u, v)$ iff there is a sequence $i_1, \ldots, i_n$ such that $u \equiv a_{i_1} a_{i_2} \cdots a_{i_n}$ and $v \equiv b_{i_1} b_{i_2} \cdots b_{i_n}$.
Observation ϕ_1^I and ϕ_2^I hold in M.

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:
The Universe all strings over $\{0, 1\}$.
The interpretation of f_1: $f_1^M(u) = 1u$ (put 1 in the front of the string u).
The interpretation of f_0: $f_0^M(u) = 0u$
The interpretation of c: the empty string ϵ.
The interpretation of R: $R^M(u, v)$ iff there is a sequence $i_1, \ldots, i_n$ such that $u \equiv a_{i_1} a_{i_2} \cdots a_{i_n}$ and $v \equiv b_{i_1} b_{i_2} \cdots b_{i_n}$.
Observation ϕ_1^I and ϕ_2^I hold in M.
Therefore, if ψ^I is valid then there is a string u such that $R^M(u, u)$.

The $\Leftarrow$ direction of the Correctness Theorem.

Let M be the structure defined as follows:

The Universe all strings over $\{0, 1\}$.

The interpretation of f_1: $f_1^M(u) = 1u$ (put 1 in the front of the string u).

The interpretation of f_0: $f_0^M(u) = 0u$

The interpretation of c: the empty string ϵ.

The interpretation of R: $R^M(u, v)$ iff there is a sequence $i_1, \ldots, i_n$ such that $u \equiv a_{i_1}a_{i_2}\cdots a_{i_n}$ and $v \equiv b_{i_1}b_{i_2}\cdots b_{i_n}$.

Observation ϕ_1^I and ϕ_2^I hold in M.

Therefore, if ψ^I is valid then there is a string u such that $R^M(u, u)$.

Hence, (by the definition of R^M) the I instance of the Post problem has a solution.