First-order logic
Second part :
Interpretation of a formula

Stéphane Devismes   Pascal Lafourcade   Michel Lévy
Jean-François Monin (jean-francois.monin@imag.fr)

Université Joseph Fourier, Grenoble I

March 06, 2015

Overview

Overview

Model, validity, consequence, equivalence

Defined as in propositional logic.

An assignment

- **In propositional logic** : $V \to \{0, 1\}$
- **In first-order logic** : (I, e) where
  - I is a symbol interpretation
  - e a variable state.

Model, validity, consequence, equivalence

Defined as in propositional logic.

An assignment

- **In propositional logic** : $V \rightarrow \{0, 1\}$
- **In first-order logic** : (I, e) where
  - I is a symbol interpretation
  - e a variable state.

The truth value of a formula only depends on its free variables and its symbols.

Model, validity, consequence, equivalence

Defined as in propositional logic.

An assignment

- **In propositional logic** : $V \to \{0, 1\}$
- **In first-order logic** : (I, e) where
  - I is a symbol interpretation
  - e a variable state.

The truth value of a formula only depends on its free variables and its symbols.

The state of variables is useless for evaluating a formula with no free variables.

Model, validity, consequence, equivalence

Defined as in propositional logic.

An assignment

- **In propositional logic** : $V \to \{0, 1\}$
- **In first-order logic** : (I, e) where
  - I is a symbol interpretation
  - e a variable state.

The truth value of a formula only depends on its free variables and its symbols.

The state of variables is useless for evaluating a formula with no free variables.

We use an interpretation instead of an assignment.

Instantiation

Definition 4.3.34

Let x a variable, t a term and A a formula.

1. $A < x := t >$ is the formula obtained by replacing in formula A all free occurrence of x with the term t.
2. The term t is free for x in A if the variables of t are not bound in the free occurrences of x.

Instantiation : Example

Example 4.3.35

- The term z is free for x in formula $\exists y\, p(x, y)$.

Instantiation : Example

Example 4.3.35

- The term z is free for x in formula $\exists y\, p(x, y)$.
- On the contrary the term y, as all term containing the variable y, is not free for x in this formula.

Instantiation : Example

Example 4.3.35

- The term z is free for x in formula $\exists y\, p(x, y)$.
- On the contrary the term y, as all term containing the variable y, is not free for x in this formula.
- By definition, the term x is free with respect to itself in all formula.

Instantiation : Example

Example 4.3.35

- The term z is free for x in formula $\exists y\, p(x, y)$.
- On the contrary the term y, as all term containing the variable y, is not free for x in this formula.
- By definition, the term x is free with respect to itself in all formula.
- Let A the formula $(\forall x P(x) \vee Q(\mathbf{x}))$, the formula $A < x := b >$ equals

  $(\forall x P(x) \vee Q(b))$ since only the bold occurrence of x is free.

Properties

Theorem 4.3.36

Let A a formula and t a free term for the variable x in A. Let I an interpretation and e a state of the interpretation. We have $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, where $d = [\![t]\!]_{(I,e)}$.

Properties

Theorem 4.3.36

Let A a formula and t a free term for the variable x in A. Let I an interpretation and e a state of the interpretation. We have $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, where $d = [\![t]\!]_{(I,e)}$.

Corollary 4.3.38

Let A a formula and t a free term for x in A.
The formulae $\forall xA \Rightarrow A < x := t >$ and $A < x := t > \Rightarrow \exists xA$ are valid.

The condition on t is necessary :

The condition "t is a free term" is necessary in **theorem 4.3.36**.

Example 4.3.37

The condition on t is necessary :

The condition "t is a free term" is necessary in **theorem 4.3.36**.

Example 4.3.37

Let I the interpretation of domain $\{0, 1\}$ with $p_I = \{(0, 1)\}$ and e, a state where $y = 0$. Let A the formula $\exists y p(x, y)$ and t the term y. This term is not free for x in A

The condition on t is necessary :

The condition "t is a free term" is necessary in **theorem 4.3.36**.

Example 4.3.37

Let I the interpretation of domain $\{0, 1\}$ with $p_I = \{(0, 1)\}$ and e, a state where $y = 0$. Let A the formula $\exists y p(x, y)$ and t the term y. This term is not free for x in A

- $A < x := t >=$

The condition on t is necessary :

The condition “t is a free term” is necessary in **theorem 4.3.36**.

Example 4.3.37

Let I the interpretation of domain $\{0,1\}$ with $p_I = \{(0,1)\}$ and e, a state where $y = 0$. Let A the formula $\exists yp(x,y)$ and t the term y. This term is not free for x in A

- $A < x := t > =$

  $\exists yp(y,y)$

The condition on t is necessary :

The condition "t is a free term" is necessary in **theorem 4.3.36**.

Example 4.3.37

Let I the interpretation of domain $\{0, 1\}$ with $p_I = \{(0, 1)\}$ and e, a state where $y = 0$. Let A the formula $\exists y p(x, y)$ and t the term y. This term is not free for x in A

- $A < x := t >=$

  $\boxed{\exists y p(y, y)}$

  and $[A < x := t >]_{(I,e)} =$

The condition on t is necessary :

The condition "t is a free term" is necessary in **theorem 4.3.36**.

Example 4.3.37

Let I the interpretation of domain $\{0,1\}$ with $p_I = \{(0,1)\}$ and e, a state where $y = 0$. Let A the formula $\exists y p(x,y)$ and t the term y. This term is not free for x in A

- $A < x := t > =$

  $\exists y p(y,y)$

  and $[A < x := t >]_{(I,e)} =$

  $[\exists y p(y,y)]_{(I,e)} = \max\{[p(0,0)]_{(I,e)}, [p(1,1)]_{(I,e)}\} = max\{0,0\} = 0.$

The condition on t is necessary :

Example 4.3.37

- Let $d = [\![t]\!]_{(I,e)} = [\![y]\!]_{(I,e)} = 0$. In the assignment $(I, e[x = d])$, we have $x = 0$. Hence $[A]_{(I,e[x=d])} =$

The condition on t is necessary :

Example 4.3.37

- Let $d = [\![t]\!]_{(I,e)} = [\![y]\!]_{(I,e)} = 0$. In the assignment $(I, e[x = d])$, we have $x = 0$. Hence $[A]_{(I,e[x=d])} =$

  $[\exists y p(x, y)]_{(I,e[x=d])} = \max\{[p(0,0)]_{(I,e)}, [p(0,1)]_{(I,e)}\} = max\{0,1\} = 1.$

The condition on t is necessary :

Example 4.3.37

- Let $d = [\![t]\!]_{(I,e)} = [\![y]\!]_{(I,e)} = 0$. In the assignment $(I, e[x = d])$, we have $x = 0$. Hence $[A]_{(I,e[x=d])} =$

  $[\exists y p(x, y)]_{(I,e[x=d])} = \max\{[p(0, 0)]_{(I,e)}, [p(0, 1)]_{(I,e)}\} = max\{0, 1\} = 1$.

Thus, $[A < x := t >]_{(I,e)} \neq [A]_{(I,e[x=d])}$, for $d = [\![t]\!]_{(I,e)}$.

Overview

Finite model

Definition

A finite model of a closed formula is an interpretation of the formula of finite domain, which makes the formula true.

Finite model

Definition

A finite model of a closed formula is an interpretation of the formula of finite domain, which makes the formula true.

Remark

- The name of the elements of the domain is not important.
- Hence for a model with n elements, we'll use the domain of integers less than n.

Building a finite model

Naive idea : In order to know whether a closed formula has a model of domain $\{0, \ldots, n-1\}$, just

- **enumerate** all the possible interpretations of the associated signature of the formula
- **evaluate** the formula for these interpretations.

Building a finite model

Naive idea : In order to know whether a closed formula has a model of domain $\{0, \ldots, n-1\}$, just

- **enumerate** all the possible interpretations of the associated signature of the formula
- **evaluate** the formula for these interpretations.

Example

Let $\Sigma = \{a^{f0}, f^{f1}, P^{r2}\}$, plus possibly the equality of truth value set.

Building a finite model

Naive idea : In order to know whether a closed formula has a model of domain $\{0, \ldots, n-1\}$, just

- **enumerate** all the possible interpretations of the associated signature of the formula
- **evaluate** the formula for these interpretations.

Example

Let $\Sigma = \{a^{f0}, f^{f1}, P^{r2}\}$, plus possibly the equality of truth value set.

Over a domain of 5 elements, Σ has $5 \times 5^5 \times 2^{25}$ interpretations !

Building a finite model

Naive idea : In order to know whether a closed formula has a model of domain $\{0, \ldots, n-1\}$, just

- **enumerate** all the possible interpretations of the associated signature of the formula
- **evaluate** the formula for these interpretations.

Example

Let $\Sigma = \{a^{f0}, f^{f1}, P^{r2}\}$, plus possibly the equality of truth value set.

Over a domain of 5 elements, Σ has $5 \times 5^5 \times 2^{25}$ interpretations !

This method is unusable in practice.

Software for building a finite model

MACE

- **translation** of first-order formulae in propositional formulae
- **performant algorithms to find the satisfiability** of a propositional formula (e.g., different versions of the `DPLL` algorithm)

`http://www.cs.unm.edu/~mccune/prover9/mace4.pdf`

Method for finding a finite model

Base case : Find models of n elements **by reduction to the propositional case** for a formula with no function symbol and no constant, except representations of integers less than n.

Method for finding a finite model

Base case : Find models of n elements **by reduction to the propositional case** for a formula with no function symbol and no constant, except representations of integers less than n.

Construct the model of n elements

1. eliminate quantifiers by expansion to a domain of n elements,
2. replace equalities with their value
3. search for a model propositional assignment.

Expansion of a formula

Definition 4.3.39

Let A a formula and n an integer. The n-expansion of A is the formula which consists in replacing :

- all sub-formula of A of the form $\forall x B$ with the conjunction $(\prod_{i<n} B < x := \underline{i} >)$
- all sub-formula of A of the form $\exists x B$ with the disjunction $(\sum_{i<n} B < x := \underline{i} >)$

where $\underline{i}$ is the decimal representation of the integer i.

Expansion of a formula

Definition 4.3.39

Let A a formula and n an integer. The n-expansion of A is the formula which consists in replacing :

- all sub-formula of A of the form $\forall x B$ with the conjunction $(\prod_{i<n} B < x := \underline{i} >)$
- all sub-formula of A of the form $\exists x B$ with the disjunction $(\sum_{i<n} B < x := \underline{i} >)$

where $\underline{i}$ is the decimal representation of the integer i.

Example 4.3.40

The 2-expansion of the formula $\exists x P(x) \Rightarrow \forall x P(x)$ is

Expansion of a formula

Definition 4.3.39

Let A a formula and n an integer. The n-expansion of A is the formula which consists in replacing :

- all sub-formula of A of the form $\forall x B$ with the conjunction $(\prod_{i<n} B < x := \underline{i} >)$
- all sub-formula of A of the form $\exists x B$ with the disjunction $(\sum_{i<n} B < x := \underline{i} >)$

where $\underline{i}$ is the decimal representation of the integer i.

Example 4.3.40

The 2-expansion of the formula $\exists x P(x) \Rightarrow \forall x P(x)$ is

$$P(0) \vee P(1) \Rightarrow P(0) \wedge P(1)$$

Property of the *n*-expansion

Theorem 4.3.41

Let n be an integer and A be a formula containing only representations of integers whose value are less than n.
Let B be the n-expansion of A.
All interpretation of domain $\{0, \ldots, n-1\}$ assign the same value to A and B.

Property of the n-expansion

Theorem 4.3.41

Let n be an integer and A be a formula containing only representations of integers whose value are less than n.
Let B be the n-expansion of A.
All interpretation of domain $\{0, \ldots, n-1\}$ assign the same value to A and B.

The condition on A is necessary because if A contains a representation of an integer which is at least equal to n, the value of this representation will not be in the domain of the interpretation.

Property of the *n*-expansion

Theorem 4.3.41

Let n be an integer and A be a formula containing only representations of integers whose value are less than n.
Let B be the n-expansion of A.
All interpretation of domain $\{0, \ldots, n-1\}$ assign the same value to A and B.

The condition on A is necessary because if A contains a representation of an integer which is at least equal to n, the value of this representation will not be in the domain of the interpretation.
The proof of the theorem is by induction on the height of formulae.

Idea of the induction : elimination of a universal quantifier

Reminder : theorem 4.3.36

Let A be a formula and t be a term which is free for the variable x in A. Let I be an interpretation and e be a state of the interpretation. We have $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, where $d = [\![t]\!]_{(I,e)}$.

Idea of the induction : elimination of a universal quantifier

Reminder : theorem 4.3.36

Let A be a formula and t be a term which is free for the variable x in A. Let I be an interpretation and e be a state of the interpretation. We have $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, where $d = [\![t]\!]_{(I,e)}$.

Let (I, e) be an interpretation and a state of domain $\{0, \ldots, n-1\}$ assigning to the representation of an integer the value of the represented integer.

Idea of the induction : elimination of a universal quantifier

Reminder : theorem 4.3.36

Let A be a formula and t be a term which is free for the variable x in A. Let I be an interpretation and e be a state of the interpretation. We have $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, where $d = [\![t]\!]_{(I,e)}$.

Let (I, e) be an interpretation and a state of domain $\{0, \ldots, n-1\}$ assigning to the representation of an integer the value of the represented integer. By definition :

$$[\forall xB]_{(I,e)} = \prod_{i<n} [B]_{(I,e[x=i])}$$

Idea of the induction : elimination of a universal quantifier

Reminder : theorem 4.3.36

Let A be a formula and t be a term which is free for the variable x in A. Let I be an interpretation and e be a state of the interpretation. We have $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, where $d = [\![t]\!]_{(I,e)}$.

Let (I, e) be an interpretation and a state of domain $\{0, \ldots, n-1\}$ assigning to the representation of an integer the value of the represented integer. By definition :

$$[\forall x B]_{(I,e)} = \prod_{i<n} [B]_{(I,e[x=i])}$$

According to theorem 4.3.36 and the fact that the value of the representation of the integer i is i, we have :

$$[B]_{(I,e[x=i])} = [B < x := \underline{i} >]_{(I,e)}$$

Idea of the induction : elimination of a universal quantifier

Reminder : theorem 4.3.36

Let A be a formula and t be a term which is free for the variable x in A. Let I be an interpretation and e be a state of the interpretation. We have $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, where $d = [\![t]\!]_{(I,e)}$.

Let (I, e) be an interpretation and a state of domain $\{0, \dots, n-1\}$ assigning to the representation of an integer the value of the represented integer. By definition :

$$[\forall xB]_{(I,e)} = \prod_{i<n} [B]_{(I,e[x=i])}$$

According to theorem 4.3.36 and the fact that the value of the representation of the integer i is i, we have :

$$[B]_{(I,e[x=i])} = [B < x := \underline{i} >]_{(I,e)}$$

Therefore : $[\forall xB]_{(I,e)} = \prod_{i<n}[B < x := \underline{i} >]_{(I,e)} = [\prod_{i<n} B < x := \underline{i} >]_{(I,e)}$.

From assignment to interpretation

Let n be an integer and A be a closed formula, with no quantifier, no equality, no function symbol, and no constant except the representations of integers less than n. Let P be the set of atomic formulae of A (except $\top$ and $\bot$ whose truth value are fixed).

Theorem 4.3.42

Let v be a propositional assignment of P in $\{0, 1\}$; then there exists an interpretation I of A such that $[A]_I = [P]_v$.

Proof.

See handout course notes. □

Example 4.3.43

Let v the assignment defined by $p(0) = 1, p(1) = 0$.

v gives the value 0 to the formula $(p(0) \vee p(1)) \Rightarrow (p(0) \wedge p(1))$.

Hence the interpretation I defined by $p_I = \{0\}$ also gives the value 0 to the same formula.

This example shows that v and I are two analogous ways of presenting an interpretation, the second one is often more concise.

From interpretation to assignment

Let n an integer and A a closed formula, with no quantifier, no equality, no function symbol, no constant except for the representations of integers less than n. Let P the set of atomic formulae of A (except $\top$ and $\bot$ whose truth value is fixed).

Theorem 4.3.44

Let I an interpretation of A then there exist an assignment v of P such that

$$[A]_I = [P]_v.$$

Proof.

See handout course notes. □

Finding a finite model of a closed formula **without** function symbol

Procedure under the same hypotheses.

1. Replace A by its n-expansion B
2. In B,
   - replace equalities by their truth constants, i.e., $\underline{i} = \underline{j}$ is replaced by $\perp$ if $i \neq j$ and by $\top$ if $i = j$.
   - Simplification using equivalences $x \vee \perp = x, \; x \vee \top = \top, \; x \wedge \perp = \perp, \; x \wedge \top = x$.

   Let C be the obtained formula.
3. Look for a propositional assignment v of the atomic formulae of C, which is a model of C :
   - if such an assignment does not exist, A has no model
   - otherwise the interpretation I deduced from v is a model of A.

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.
   - According to theorem 4.3.41, I is a model of B, hence of C.

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.
   - According to theorem 4.3.41, I is a model of B, hence of C.
   - According to theorem 4.3.44, there is a model propositional assignment of C.

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.
   - According to theorem 4.3.41, I is a model of B, hence of C.
   - According to theorem 4.3.44, there is a model propositional assignment of C.

   From this contradiction, we deduce that A has no model with n elements.

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.
   - According to theorem 4.3.41, I is a model of B, hence of C.
   - According to theorem 4.3.44, there is a model propositional assignment of C.

   From this contradiction, we deduce that A has no model with n elements.
2. Suppose there exist a propositional assignment v of atomic formulae of C which is a model of C.

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.
   - According to theorem 4.3.41, I is a model of B, hence of C.
   - According to theorem 4.3.44, there is a model propositional assignment of C.

   From this contradiction, we deduce that A has no model with n elements.
2. Suppose there exist a propositional assignment v of atomic formulae of C which is a model of C.
   Hence, the interpretation I constructed as indicated in theorem 4.3.42 is a model of C.

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.
   - According to theorem 4.3.41, I is a model of B, hence of C.
   - According to theorem 4.3.44, there is a model propositional assignment of C.

   From this contradiction, we deduce that A has no model with n elements.
2. Suppose there exist a propositional assignment v of atomic formulae of C which is a model of C.
   Hence, the interpretation I constructed as indicated in theorem 4.3.42 is a model of C.
   Hence it is a model of B

Correctness proof of the method

1. Suppose there is no model propositional assignment of C, but that A has a model I.
   - According to theorem 4.3.41, I is a model of B, hence of C.
   - According to theorem 4.3.44, there is a model propositional assignment of C.

   From this contradiction, we deduce that A has no model with n elements.
2. Suppose there exist a propositional assignment v of atomic formulae of C which is a model of C.
   Hence, the interpretation I constructed as indicated in theorem 4.3.42 is a model of C.
   Hence it is a model of B
   Hence according to theorem 4.3.41, it is a model of A.

Example 4.3.45

$$A = \exists x P(x) \land \exists x \neg P(x) \land \forall x \forall y (P(x) \land P(y) \Rightarrow x = y))$$

Example 4.3.45

$$A = \exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y))$$

A has no model of one element, since we have *P* and its negation.

Example 4.3.45

$$A = \exists x P(x) \land \exists x \neg P(x) \land \forall x \forall y (P(x) \land P(y) \Rightarrow x = y))$$

A has no model of one element, since we have P and its negation.

2-expansion de A

Example 4.3.45

$$A = \exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y))$$

A has no model of one element, since we have P and its negation.

2-expansion de A

$$(P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1)) \wedge$$
$$(P(0) \wedge P(0) \Rightarrow 0 = 0) \wedge (P(0) \wedge P(1) \Rightarrow 0 = 1) \wedge$$
$$(P(1) \wedge P(0) \Rightarrow 1 = 0) \wedge (P(1) \wedge P(1) \Rightarrow 1 = 1).$$

Example 4.3.45

$$A = \exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y))$$

A has no model of one element, since we have P and its negation.

2-expansion de A

$$\begin{array}{l} (P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1)) \wedge \\ (P(0) \wedge P(0) \Rightarrow 0 = 0) \wedge (P(0) \wedge P(1) \Rightarrow 0 = 1) \wedge \\ (P(1) \wedge P(0) \Rightarrow 1 = 0) \wedge (P(1) \wedge P(1) \Rightarrow 1 = 1). \end{array}$$

We replace equalities by their truth constants

$$\begin{array}{l} (P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1)) \wedge (P(0) \wedge P(0) \Rightarrow \top). \\ (P(0) \wedge P(1) \Rightarrow \bot) \wedge (P(1) \wedge P(0) \Rightarrow \bot) \wedge (P(1) \wedge P(1) \Rightarrow \top). \end{array}$$

Example 4.3.45

$$A = \exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y))$$

A has no model of one element, since we have P and its negation.

2-expansion de A

$(P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1)) \wedge$
$(P(0) \wedge P(0) \Rightarrow 0 = 0) \wedge (P(0) \wedge P(1) \Rightarrow 0 = 1) \wedge$
$(P(1) \wedge P(0) \Rightarrow 1 = 0) \wedge (P(1) \wedge P(1) \Rightarrow 1 = 1).$

We replace equalities by their truth constants

$(P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1)) \wedge (P(0) \wedge P(0) \Rightarrow \top).$
$(P(0) \wedge P(1) \Rightarrow \bot) \wedge (P(1) \wedge P(0) \Rightarrow \bot) \wedge (P(1) \wedge P(1) \Rightarrow \top).$
Which simplifies to : $(P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1))$

Example 4.3.45

$$A = \exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y))$$

A has no model of one element, since we have P and its negation.

2-expansion de A

$(P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1)) \wedge$
$(P(0) \wedge P(0) \Rightarrow 0 = 0) \wedge (P(0) \wedge P(1) \Rightarrow 0 = 1) \wedge$
$(P(1) \wedge P(0) \Rightarrow 1 = 0) \wedge (P(1) \wedge P(1) \Rightarrow 1 = 1)$.

We replace equalities by their truth constants

$(P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1)) \wedge (P(0) \wedge P(0) \Rightarrow \top)$.
$(P(0) \wedge P(1) \Rightarrow \bot) \wedge (P(1) \wedge P(0) \Rightarrow \bot) \wedge (P(1) \wedge P(1) \Rightarrow \top)$.
Which simplifies to : $(P(0) \vee P(1)) \wedge (\neg P(0) \vee \neg P(1))$
The assignment $P(0) \mapsto 1,\ P(1) \mapsto 0$ is a propositional model of the above formula, hence the interpretation I of domain $\{0, 1\}$ where $P_I = \{0\}$ is a model of A.

Finding a finite model of a closed formula **with** a function symbol

Let A be a closed formula which can contain representations of integers of value less than n.

Procedure

- Replace A by its expansion
- Enumerate the choices of symbol values, by propagating as much as possible each of the realized choices.

Similar to `DPLL` *algorithm*.

Example 4.3.46 : $A = \exists y P(y) \Rightarrow P(a)$

Look for a counter-model with 2 elements.

Example 4.3.46 : $A = \exists y P(y) \Rightarrow P(a)$

Look for a counter-model with 2 elements.

2-expansion of A

Example 4.3.46 : $A = \exists y P(y) \Rightarrow P(a)$

Look for a counter-model with 2 elements.

2-expansion of A

$$P(0) \vee P(1) \Rightarrow P(a)$$

Find the values of $P(0)$, $P(1)$, a. We (arbitrarily) choose $a = 0$.

Example 4.3.46 : $A = \exists y P(y) \Rightarrow P(a)$

Look for a counter-model with 2 elements.

2-expansion of A

$$P(0) \vee P(1) \Rightarrow P(a)$$

Find the values of $P(0)$, $P(1)$, a. We (arbitrarily) choose $a = 0$.

$$P(0) \vee P(1) \Rightarrow P(0)$$

Example 4.3.46 : $A = \exists y P(y) \Rightarrow P(a)$

Look for a counter-model with 2 elements.

2-expansion of A

$$P(0) \vee P(1) \Rightarrow P(a)$$

Find the values of $P(0)$, $P(1)$, a. We (arbitrarily) choose $a = 0$.

$$P(0) \vee P(1) \Rightarrow P(0)$$

$P(0) \mapsto 0, P(1) \mapsto 1$ is a propositional counter-model,
i.e., an interpretation such that $P \mapsto \{1\}$.

Example 4.3.46 : $A = \exists y P(y) \Rightarrow P(a)$

Look for a counter-model with 2 elements.

2-expansion of A

$$P(0) \vee P(1) \Rightarrow P(a)$$

Find the values of $P(0)$, $P(1)$, a. We (arbitrarily) choose $a = 0$.

$$P(0) \vee P(1) \Rightarrow P(0)$$

$P(0) \mapsto 0, P(1) \mapsto 1$ is a propositional counter-model,
i.e., an interpretation such that $P \mapsto \{1\}$.

A counter-model is the interpretation of domain $\{0, 1\}$ such that $P \mapsto \{1\}$ and $a \mapsto 0$.

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.
   $P(a) \mapsto 1$, $(P(0) \Rightarrow P(f(0))) \mapsto 1$, $(P(1) \Rightarrow P(f(1))) \mapsto 1$,
   $P(f(b)) \mapsto 0$
3. Choose $a \mapsto 0$

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.
   $P(a) \mapsto 1$, $(P(0) \Rightarrow P(f(0))) \mapsto 1$, $(P(1) \Rightarrow P(f(1))) \mapsto 1$, $P(f(b)) \mapsto 0$
3. Choose $a \mapsto 0$
   - From $P(a) \mapsto 1$ and $a \mapsto 0$, we deduce : $P(0) \mapsto 1$

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.
   $P(a) \mapsto 1$, $(P(0) \Rightarrow P(f(0))) \mapsto 1$, $(P(1) \Rightarrow P(f(1))) \mapsto 1$,
   $P(f(b)) \mapsto 0$
3. Choose $a \mapsto 0$
   - From $P(a) \mapsto 1$ and $a \mapsto 0$, we deduce : $P(0) \mapsto 1$
   - From $P(0) \mapsto 1$ and $(P(0) \Rightarrow P(f(0))) \mapsto 1$, we deduce : $P(f(0)) \mapsto 1$

Example 4.3.47 : $P(a), \forall x (P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.
   $P(a) \mapsto 1$, $(P(0) \Rightarrow P(f(0))) \mapsto 1$, $(P(1) \Rightarrow P(f(1))) \mapsto 1$, $P(f(b)) \mapsto 0$
3. Choose $a \mapsto 0$
   - From $P(a) \mapsto 1$ and $a \mapsto 0$, we deduce : $P(0) \mapsto 1$
   - From $P(0) \mapsto 1$ and $(P(0) \Rightarrow P(f(0))) \mapsto 1$, we deduce : $P(f(0)) \mapsto 1$
   - From $P(f(b)) \mapsto 0$ and $P(f(0)) \mapsto 1$, we deduce $f(0) \neq f(b)$ therefore $b \neq 0$, hence : $b \mapsto 1$

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.
   $P(a) \mapsto 1$, $(P(0) \Rightarrow P(f(0))) \mapsto 1$, $(P(1) \Rightarrow P(f(1))) \mapsto 1$, $P(f(b)) \mapsto 0$
3. Choose $a \mapsto 0$
   - From $P(a) \mapsto 1$ and $a \mapsto 0$, we deduce : $P(0) \mapsto 1$
   - From $P(0) \mapsto 1$ and $(P(0) \Rightarrow P(f(0))) \mapsto 1$, we deduce : $P(f(0)) \mapsto 1$
   - From $P(f(b)) \mapsto 0$ and $P(f(0)) \mapsto 1$, we deduce $f(0) \neq f(b)$ therefore $b \neq 0$, hence : $b \mapsto 1$
   - From $P(f(b)) \mapsto 0$, $P(0) \mapsto 1$ and $b \mapsto 1$, we deduce $f(b) = f(1) \neq 0$ hence : $f(1) \mapsto 1$ and $P(1) \mapsto 0$

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.
   $P(a) \mapsto 1$, $(P(0) \Rightarrow P(f(0))) \mapsto 1$, $(P(1) \Rightarrow P(f(1))) \mapsto 1$, $P(f(b)) \mapsto 0$
3. Choose $a \mapsto 0$
   - From $P(a) \mapsto 1$ and $a \mapsto 0$, we deduce : $P(0) \mapsto 1$
   - From $P(0) \mapsto 1$ and $(P(0) \Rightarrow P(f(0))) \mapsto 1$, we deduce : $P(f(0)) \mapsto 1$
   - From $P(f(b)) \mapsto 0$ and $P(f(0)) \mapsto 1$, we deduce $f(0) \neq f(b)$ therefore $b \neq 0$, hence : $b \mapsto 1$
   - From $P(f(b)) \mapsto 0$, $P(0) \mapsto 1$ and $b \mapsto 1$, we deduce $f(b) = f(1) \neq 0$ hence : $f(1) \mapsto 1$ and $P(1) \mapsto 0$
   - From $P(f(0)) \mapsto 1$ and $P(1) \mapsto 0$, we deduce : $f(0) \mapsto 0$

Example 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))) \wedge (P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Find values for $P(0)$, $P(1)$, a, b, $f(0)$ and $f(1)$ which provide model of F.
   $P(a) \mapsto 1$, $(P(0) \Rightarrow P(f(0))) \mapsto 1$, $(P(1) \Rightarrow P(f(1))) \mapsto 1$, $P(f(b)) \mapsto 0$
3. Choose $a \mapsto 0$
   - From $P(a) \mapsto 1$ and $a \mapsto 0$, we deduce : $P(0) \mapsto 1$
   - From $P(0) \mapsto 1$ and $(P(0) \Rightarrow P(f(0))) \mapsto 1$, we deduce : $P(f(0)) \mapsto 1$
   - From $P(f(b)) \mapsto 0$ and $P(f(0)) \mapsto 1$, we deduce $f(0) \neq f(b)$ therefore $b \neq 0$, hence : $b \mapsto 1$
   - From $P(f(b)) \mapsto 0$, $P(0) \mapsto 1$ and $b \mapsto 1$, we deduce $f(b) = f(1) \neq 0$ hence : $f(1) \mapsto 1$ and $P(1) \mapsto 0$
   - From $P(f(0)) \mapsto 1$ and $P(1) \mapsto 0$, we deduce : $f(0) \mapsto 0$

Overview

Substitution

Recall that, in propositional logic, substituting a proposition to a valid propositional formula gives a valid formula. This extends to first-order logic.

Substitution

Recall that, in propositional logic, substituting a proposition to a valid propositional formula gives a valid formula. This extends to first-order logic.

Example :

Let $\sigma(p) = \forall x\ q(x)$.
$p \vee \neg p$ is valid, the same holds for

$$\sigma(p \vee \neg p) = \forall x\ q(x) \vee \neg \forall x\ q(x)$$

Replacement

The principle of replacement for propositional logic extends as well to first-order logic since it follows from the following elementary properties :

For all formulae A and B and all variable x :

- $(A \Leftrightarrow B) \models (\forall x A \Leftrightarrow \forall x B)$
- $(A \Leftrightarrow B) \models (\exists x A \Leftrightarrow \exists x B)$

Overview

Relation between $\forall$ and $\exists$

Lemma 4.4.1

Let A be a formula and x be a variable.

1. $\neg\forall xA \equiv \exists x\neg A$
2. $\forall xA \equiv \neg\exists x\neg A$
3. $\neg\exists xA \equiv \forall x\neg A$
4. $\exists xA \equiv \neg\forall x\neg A$

Let us prove the first two equivalences, the other are in exercise 76

Proof of $\neg\forall xA \equiv \exists x\neg A$

Let I be an interpretation of domain D and e be a state
Let us evaluate $[\neg\forall xA]_{(I,e)}$

Proof of $\neg\forall x A \equiv \exists x \neg A$

Let I be an interpretation of domain D and e be a state
Let us evaluate $[\neg\forall x A]_{(I,e)}$
$= \neg[\forall x A]_{(I,e)}$

Proof of $\neg\forall x A \equiv \exists x \neg A$

Let I be an interpretation of domain D and e be a state

Let us evaluate $[\neg\forall x A]_{(I,e)}$

$= \neg[\forall x A]_{(I,e)}$

$= \neg \prod_{d \in D}[A]_{(I,e[x=d])}$ interpretation of $\forall$

Proof of $\neg\forall x A \equiv \exists x \neg A$

Let I be an interpretation of domain D and e be a state

Let us evaluate $[\neg\forall x A]_{(I,e)}$

$= \neg[\forall x A]_{(I,e)}$

$= \neg \prod_{d\in D}[A]_{(I,e[x=d])}$ interpretation of $\forall$

$= \sum_{d\in D} \neg[A]_{(I,e[x=d])}$ generalized de Morgan laws

Proof of $\neg\forall xA \equiv \exists x\neg A$

Let I be an interpretation of domain D and e be a state

Let us evaluate $[\neg\forall xA]_{(I,e)}$

$= \neg[\forall xA]_{(I,e)}$

$= \neg\prod_{d\in D}[A]_{(I,e[x=d])}$ — interpretation of $\forall$

$= \sum_{d\in D}\neg[A]_{(I,e[x=d])}$ — generalized de Morgan laws

$= \sum_{d\in D}[\neg A]_{(I,e[x=d])}$ — interpretation of $\neg$

Proof of $\neg\forall xA \equiv \exists x\neg A$

Let I be an interpretation of domain D and e be a state

Let us evaluate $[\neg\forall xA]_{(I,e)}$

$= \neg[\forall xA]_{(I,e)}$	
$= \neg\prod_{d\in D}[A]_{(I,e[x=d])}$	interpretation of $\forall$
$= \sum_{d\in D}\neg[A]_{(I,e[x=d])}$	generalized de Morgan laws
$= \sum_{d\in D}[\neg A]_{(I,e[x=d])}$	interpretation of $\neg$
$= [\exists x\neg A]_{(I,e)}$	interpretation of $\exists$

Proof of $\forall x A \equiv \neg \exists x \neg A$

Proof of $\forall x A \equiv \neg \exists x \neg A$

Let us evaluate $\forall x A$

Proof of $\forall x A \equiv \neg \exists x \neg A$

Let us evaluate $\forall x A$

$\equiv \neg\neg\forall x A$ double negation equivalence

Proof of $\forall x A \equiv \neg \exists x \neg A$

Let us evaluate $\forall x A$	
$\equiv \neg\neg\forall x A$	double negation equivalence
$\equiv \neg\exists x \neg A$	by equivalence 1

Moving quantifiers

Let x, y be two variables and A, B be two formulae.

1. $\forall x \forall y A \equiv \forall y \forall x A$

Moving quantifiers

Let x, y be two variables and A, B be two formulae.

1. $\forall x \forall y A \equiv \forall y \forall x A$
2. $\exists x \exists y A \equiv \exists y \exists x A$

Moving quantifiers

Let x, y be two variables and A, B be two formulae.

1. $\forall x \forall y A \equiv \forall y \forall x A$
2. $\exists x \exists y A \equiv \exists y \exists x A$
3. $\forall x (A \land B) \equiv (\forall x A \land \forall x B)$

Moving quantifiers

Let x, y be two variables and A, B be two formulae.

1. $\forall x \forall y A \equiv \forall y \forall x A$
2. $\exists x \exists y A \equiv \exists y \exists x A$
3. $\forall x (A \wedge B) \equiv (\forall x A \wedge \forall x B)$
4. $\exists x (A \vee B) \equiv (\exists x A \vee \exists x B)$

Moving quantifiers

Let x, y be two variables and A, B be two formulae.

1. $\forall x \forall y A \equiv \forall y \forall x A$
2. $\exists x \exists y A \equiv \exists y \exists x A$
3. $\forall x (A \wedge B) \equiv (\forall x A \wedge \forall x B)$
4. $\exists x (A \vee B) \equiv (\exists x A \vee \exists x B)$
5. Let Q be a quantifier among $\forall, \exists$, let $\circ$ be an operation among $\wedge, \vee$. Suppose that x is not a free variable of A.
   5.1 $QxA \equiv A$,
   5.2 $Qx(A \circ B) \equiv (A \circ QxB)$

Example 4.4.2

Let us eliminate useless quantifiers from these two formulae :

- $\forall x \exists x P(x) \equiv$

Example 4.4.2

Let us eliminate useless quantifiers from these two formulae :

- $\forall x \exists x P(x) \equiv$

  $\exists x P(x)$

- $\forall x (\exists x P(x) \lor Q(x)) \equiv$

Example 4.4.2

Let us eliminate useless quantifiers from these two formulae :

- $\forall x \exists x P(x) \equiv$

  $\exists x P(x)$

- $\forall x (\exists x P(x) \vee Q(x)) \equiv$

  $\exists x P(x) \vee \forall x Q(x)$

Change of bound variables (1/4)

Theorem 4.4.3

Let Q be a quantifier among $\forall, \exists$. Suppose that y is a variable not occurring in QxA then : $QxA \equiv QyA < x := y >$.

Change of bound variables (1/4)

Theorem 4.4.3

Let Q be a quantifier among $\forall, \exists$. Suppose that y is a variable not occurring in QxA then : $QxA \equiv QyA < x := y >$.

Example 4.4.4

- $\forall x\ p(x,z) \equiv \forall y\ p(y,z)$.
- $\forall x\ p(x,z) \not\equiv \forall z\ p(z,z)$.

Change of bound variables (2/4)

Definition 4.4.5

Two formulae are equal with respect to a change of bound variables if we can obtain one starting from the other by replacing sub-formulae of the form QxA by

$$QyA < x := y >$$

where Q is a quantifier and y is a variable not appearing in QxA.

The two formulae are α-equivalent or a copy of each other, denoted $A =_{\alpha} B$

Change of bound variables (3/4)

Theorem 4.4.6

If two formulae are equal with respect to a change of bound variables then they are equivalent.

Change of bound variables (4/4)

Example 4.4.7

Let us show that the formulae $\forall x \exists y P(x, y)$ and $\forall y \exists x P(y, x)$ are equal with respect to a change of bound variables and therefore that they are equivalent.

Change of bound variables (4/4)

Example 4.4.7

Let us show that the formulae $\forall x \exists y P(x, y)$ and $\forall y \exists x P(y, x)$ are equal with respect to a change of bound variables and therefore that they are equivalent.

$$\forall x \exists y P(x, y)$$

Change of bound variables (4/4)

Example 4.4.7

Let us show that the formulae $\forall x \exists y P(x, y)$ and $\forall y \exists x P(y, x)$ are equal with respect to a change of bound variables and therefore that they are equivalent.

$$\forall x \exists y P(x, y)$$

$$\equiv \forall u \exists y P(u, y)$$

Change of bound variables (4/4)

Example 4.4.7

Let us show that the formulae $\forall x \exists y P(x, y)$ and $\forall y \exists x P(y, x)$ are equal with respect to a change of bound variables and therefore that they are equivalent.

$$\forall x \exists y P(x, y)$$

$$\equiv \forall u \exists y P(u, y)$$

$$\equiv \forall u \exists x P(u, x)$$

Change of bound variables (4/4)

Example 4.4.7

Let us show that the formulae $\forall x \exists y P(x, y)$ and $\forall y \exists x P(y, x)$ are equal with respect to a change of bound variables and therefore that they are equivalent.

$$\forall x \exists y P(x, y)$$

$$\equiv \forall u \exists y P(u, y)$$

$$\equiv \forall u \exists x P(u, x)$$

$$\equiv \forall y \exists x P(y, x)$$

α-equivalence howto

Technique

- Draw lines between each quantifier and the variables that it binds.
- Erase the name of bound variables.

If after this transformation, the two formulae become identical, it means that they are equal with respect to a change of bound variables.

Example 4.4.8

Let $\forall x \exists y P(y, x)$ and $\forall y \exists x P(x, y)$ two formulae.

$$\forall | \exists | P(|, |)$$

Exercise

Compute the transformation for

- $A = \forall x \forall y\ R(x, y, y)$
- $B = \forall x \forall y\ R(x, x, y)$

Are A and B α-equivalent ?

Property $=_\alpha$

Theorem 4.4.9

1. Let A be an atomic formula, $A =_\alpha A'$ if and only if $A' = A$
2. $\neg B =_\alpha A'$ if and only if $A' = \neg B'$ and $B =_\alpha B'$
3. $(B \circ C) =_\alpha A'$ if and only if $A' = (B' \circ C')$ and $B =_\alpha B'$ and $C =_\alpha C'$, where $\circ$ is one of the connectives $\wedge, \vee, \Rightarrow, \Leftrightarrow$.
4. If $\forall x B =_\alpha A'$ then $A' = \forall x' B'$ and for every variable z not in the formulae B and B', we have :
   $B < x := z > =_\alpha B' < x' := z >$.
5. If $\exists x B =_\alpha A'$ then $A' = \exists x' B'$ and for every variable z not in the formulae B and B', we have :
   $B < x := z > =_\alpha B' < x' := z >$.
6. If there is one variable z not in the formulae B and B' such that $B < x := z > =_\alpha B' < x' := z >$ then $\forall x B =_\alpha \forall x' B'$ et $\exists x B =_\alpha \exists x' B'$.

Algorithm for testing alpha-equivalence

The test data are two formulae A and A'.
The result is yes if $A =_\alpha A'$, no if $A \neq_\alpha A'$.

Example 4.4.10

We only study the case where $A = \forall x B$.

1. If A' is not of the form $\forall x' B'$, then, according to point (4) of the theorem, the answer is no.
2. If $A' = \forall x' B'$ then we choose any variable z not in B and B'.
   2.1 If $B < x := z > =_\alpha B' < x' := z >$ then, according to point (6) of the theorem, the answer is yes.
   2.2 If $B < x := z > \neq_\alpha B' < x' := z >$ then, according to point (4) of the theorem, the answer is no.

Conclusion

Thank you for your attention.

Questions ?