Lecture 11

Applications of Herbrand's theorem

Ground resolution proofs, semi-decidability of validity, undecidability of validity

Introduction to Logic for Computer Science

Prof Hongseok Yang
KAIST

These slides are minor variants of those made by Prof Worrell and Prof Haase for their logic course at Oxford.

Recap

Theorem (Herbrand's theorem)

Let $F = \forall x_1 \forall x_2 \ldots \forall x_n\, F^*$ be a Skolem formula. Then F is satisfiable if and only if F has a Herbrand model.

Generalisation of the ground resolution theorem

Theorem

Let $F_1, \ldots, F_n$ be closed rectified formulas in prenex form with Skolem forms $G_1, \ldots, G_n$. Assume each G_i is obtained using different Skolem functions. Then

$$F_1 \wedge F_2 \wedge \cdots \wedge F_n \text{ is satisfiable}$$
$$\text{iff } G_1 \wedge G_2 \wedge \cdots \wedge G_n \text{ is satisfiable.}$$

Generalisation of the ground resolution theorem

Theorem

Let $F_1, \ldots, F_n$ be closed rectified formulas in prenex form with Skolem forms $G_1, \ldots, G_n$. Assume each G_i is obtained using different Skolem functions. Then

$$F_1 \wedge F_2 \wedge \cdots \wedge F_n \text{ is satisfiable}$$
$$\text{iff } G_1 \wedge G_2 \wedge \cdots \wedge G_n \text{ is satisfiable.}$$

Theorem (Ground resolution theorem)

Let $G_1, \ldots, G_n$ be closed formulas in Skolem form whose respective matrices $G_1^*, G_2^*, \ldots, G_n^*$ are in CNF. Then $G_1 \wedge G_2 \wedge \cdots \wedge G_n$ is unsatisfiable if and only if there is a propositional resolution proof of $\Box$ starting from the set of ground instances of clauses from $G_1^*, \ldots, G_n^*$.

Generalisation of the ground resolution theorem

Theorem

Let $F_1, \ldots, F_n$ be closed rectified formulas in prenex form with Skolem forms $G_1, \ldots, G_n$. Assume each G_i is obtained using different Skolem functions. Then

$$F_1 \wedge F_2 \wedge \cdots \wedge F_n \text{ is satisfiable}$$
$$\text{iff } G_1 \wedge G_2 \wedge \cdots \wedge G_n \text{ is satisfiable.}$$

Theorem (Ground resolution theorem)

Let $G_1, \ldots, G_n$ be closed formulas in Skolem form whose respective matrices $G_1^*, G_2^*, \ldots, G_n^*$ are in CNF. Then $G_1 \wedge G_2 \wedge \cdots \wedge G_n$ is unsatisfiable if and only if there is a propositional resolution proof of $\Box$ starting from the set of ground instances of clauses from $G_1^*, \ldots, G_n^*$.

Ex: Prove both theorems.

An example

An example

Example

Consider the following hypothetical scenario:

(a) Everyone at Oriel[a] is lazy, a rower or drunk.

(b) All rowers are lazy.

(c) Someone at Oriel is not drunk.

(d) Someone at Oriel is lazy.

Show that (a), (b) and (c) together entail (d).

[a] Oriel is one of the oldest Oxford colleges. Oxford colleges are like houses in the Harry Potter movie.

An example

Translation into first-order logic:

An example

Translation into first-order logic:

$$
\begin{aligned}
F_1 &:= \forall x(O(x) \to (L(x) \vee R(x) \vee D(x))),\\
F_2 &:= \forall x(R(x) \to L(x)),\\
F_3 &:= \exists x(O(x) \wedge \neg D(x)),\\
F_4 &:= \neg\exists x(O(x) \wedge L(x)).
\end{aligned}
$$

An example

Translation into first-order logic:

$$
\begin{aligned}
F_1 &:= \forall x (O(x) \rightarrow (L(x) \vee R(x) \vee D(x))), \\
F_2 &:= \forall x (R(x) \rightarrow L(x)), \\
F_3 &:= \exists x (O(x) \wedge \neg D(x)), \\
F_4 &:= \neg \exists x (O(x) \wedge L(x)).
\end{aligned}
$$

Ex: Translate $F_1 \wedge F_2 \wedge F_3 \wedge F_4$ into CNF Skolem form. Then, prove that the result is unsat using ground resolution.

Transformation into CNF Skolem form:

$$\begin{aligned}
G_1 &:= \forall x(\neg O(x) \vee L(x) \vee R(x) \vee D(x)),\\
G_2 &:= \forall x(\neg R(x) \vee L(x)),\\
G_3 &:= O(a) \wedge \neg D(a),\\
G_4 &:= \forall x(\neg O(x) \vee \neg L(x)).
\end{aligned}$$

Ground resolution proof for the example:

$$\dfrac{\dfrac{\dfrac{\dfrac{\{\neg R(a), L(a)\} \qquad \{\neg O(a), L(a), R(a), D(a)\}}{\{L(a), \neg O(a), D(a)\}} \qquad \{\neg O(a), \neg L(a)\}}{\{\neg O(a), D(a)\}} \qquad \{\neg D(a)\}}{\{\neg O(a)\}} \qquad \{O(a)\}}{\square}$$

Another example

Show that the following formula is valid:

$$F = \forall x \exists y (P(x) \to Q(y)) \to \exists y \forall x (P(x) \to Q(y)).$$

Another example

Show that the following formula is valid:

$$F = \forall x \exists y (P(x) \rightarrow Q(y)) \rightarrow \exists y \forall x (P(x) \rightarrow Q(y)).$$

F is valid if and only if $\neg F$ is unsatisfiable:

$$\neg F \equiv \forall x \exists y (P(x) \rightarrow Q(y)) \wedge \neg \exists y \forall x (P(x) \rightarrow Q(y)).$$

Another example

Show that the following formula is valid:

$$F = \forall x \exists y (P(x) \to Q(y)) \to \exists y \forall x (P(x) \to Q(y)).$$

F is valid if and only if $\neg F$ is unsatisfiable:

$$\neg F \equiv \forall x \exists y (P(x) \to Q(y)) \land \neg \exists y \forall x (P(x) \to Q(y)).$$

Express $\neg F$ as $F_1 \land F_2$:

$$\begin{aligned} \neg F &\equiv F_1 \land F_2, \\ F_1 &= \forall x \exists y (P(x) \to Q(y)), \\ F_2 &= \neg \exists y \forall x (P(x) \to Q(y)). \end{aligned}$$

Another example

Show that the following formula is valid:

$$F = \forall x \exists y (P(x) \to Q(y)) \to \exists y \forall x (P(x) \to Q(y)).$$

F is valid if and only if $\neg F$ is unsatisfiable:

$$\neg F \equiv \forall x \exists y (P(x) \to Q(y)) \wedge \neg \exists y \forall x (P(x) \to Q(y)).$$

Express $\neg F$ as $F_1 \wedge F_2$:

$$\begin{aligned} \neg F &\equiv F_1 \wedge F_2, \\ F_1 &= \forall x \exists y (P(x) \to Q(y)), \\ F_2 &= \neg \exists y \forall x (P(x) \to Q(y)). \end{aligned}$$

Ex: Prove that $F_1 \wedge F_2$ is unsat via Skolemisation and resolution.

Hint: In this case, Skolemisation does not introduce any constants, so that you won't have any ground terms. To overcome this, introduce a constant symbol *a*. Justify why this introduction is ok.

Skolemise:

$$F_1 = \forall x(\neg P(x) \vee Q(f(x))) \qquad F_2 = \forall y(P(g(y)) \wedge \neg Q(y))$$

Skolemise:

$$F_1 = \forall x(\neg P(x) \vee Q(f(x))) \qquad F_2 = \forall y(P(g(y)) \wedge \neg Q(y))$$

No constant symbols. No ground terms. Introduce a constant a.

Skolemise:

$$F_1 = \forall x(\neg P(x) \vee Q(f(x))) \qquad F_2 = \forall y(P(g(y)) \wedge \neg Q(y))$$

No constant symbols. No ground terms. Introduce a constant *a*.

Ground resolution proof:

$$\dfrac{\dfrac{\{P(g(a))\} \qquad \{\neg P(g(a)), Q(f(g(a)))\}}{\{Q(f(g(a)))\}} \qquad \{\neg Q(f(g(a)))\}}{\square}$$

Semi-decidability of validity

Theorem

Validity of first-order logic is semi-decidable.

Semi-decidability of validity

Theorem

Validity of first-order logic is semi-decidable.

Semi-Decision Procedure for Validity

Input: Closed formula F

Output: Either that F is valid or compute forever

```
Compute a Skolem-form formula $G$ equisatisfiable with $\neg F$
  such that $G$'s quantifier-free part is in CNF
Let $G_1, G_2, \ldots$ be an enumeration of the Herbrand expansion $E(G)$
for $n = 1$ to $\infty$ do
begin
    if $\Box \in \mathrm{Res}^*(G_1 \cup \ldots \cup G_n)$ then stop and output "$F$ is valid"
end
```

Semi-decidability of validity

Theorem

Validity of first-order logic is semi-decidable.

```
Semi-Decision Procedure for Validity
Input: Closed formula $F$
Output: Either that $F$ is valid or compute forever
Compute a Skolem-form formula $G$ equisatisfiable with $\neg F$
   such that $G$'s quantifier-free part is in CNF
Let $G_1, G_2, \ldots$ be an enumeration of the Herbrand expansion $E(G)$
for $n = 1$ to $\infty$ do
begin
    if $\Box \in \mathrm{Res}^*(G_1 \cup \ldots \cup G_n)$ then stop and output "$F$ is valid"
end
```

Ex: Can we do better? Can we design an *algorithm* for validity?

Semi-decidability of validity

Theorem

Validity of first-order logic is semi-decidable.

Semi-Decision Procedure for Validity

Input: Closed formula F

Output: Either that F is valid or compute forever

```
Compute a Skolem-form formula $G$ equisatisfiable with $\neg F$
    such that $G$'s quantifier-free part is in CNF
Let $G_1, G_2, \ldots$ be an enumeration of the Herbrand expansion $E(G)$
for $n = 1$ to $\infty$ do
begin
    if $\Box \in \mathrm{Res}^*(G_1 \cup \ldots \cup G_n)$ then stop and output "$F$ is valid"
end
```

Ex: Can we do better? Can we design an *algorithm* for validity?

Answer: No.

How to show undecidability?

Principle:

- Take an undecidable problem P.
- Provide a computable function f that translates an instance I of P into the validity problem for first order logic $f(I)$.
- Ensure that the answer for I is yes iff $f(I)$ is valid.
- “Validity for first-order logic is at least as difficult as P and hence undecidable.”

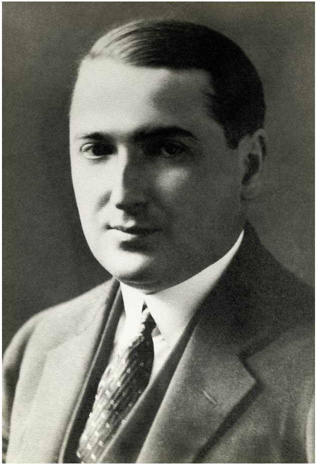
How to show undecidability?

Principle:

- Take an undecidable problem P.
- Provide a computable function f that translates an instance I of P into the validity problem for first order logic $f(I)$.
- Ensure that the answer for I is yes iff $f(I)$ is valid.
- "Validity for first-order logic is at least as difficult as P and hence undecidable."

We choose P to be the **Post Correspondence Problem (PCP).**

Emil Post (1897 – 1954)



The post correspondence problem

In PCP, given a set of **tiles** $(x_i, y_i) \in \{0,1\}^* \times \{0,1\}^*$, e.g.:

$$\left\{ \begin{bmatrix} 1 \\ 101 \end{bmatrix}, \begin{bmatrix} 10 \\ 00 \end{bmatrix}, \begin{bmatrix} 011 \\ 11 \end{bmatrix} \right\}.$$

The post correspondence problem

In PCP, given a set of **tiles** $(x_i, y_i) \in \{0,1\}^* \times \{0,1\}^*$, e.g.:

$$\left\{ \begin{bmatrix} 1 \\ 101 \end{bmatrix}, \begin{bmatrix} 10 \\ 00 \end{bmatrix}, \begin{bmatrix} 011 \\ 11 \end{bmatrix} \right\}.$$

A solution is a sequence of tiles such that the top string equals the bottom string:

$$\begin{bmatrix} 1 \\ 101 \end{bmatrix} \begin{bmatrix} 011 \\ 11 \end{bmatrix} \begin{bmatrix} 10 \\ 00 \end{bmatrix} \begin{bmatrix} 011 \\ 11 \end{bmatrix}.$$

The post correspondence problem

Definition (Post Correspondence Problem (PCP))

An **instance of PCP** is a finite set

$$P = \{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq \{0,1\}^* \times \{0,1\}^*.$$

A **solution of** P is a sequence of indices $i_1, i_2, \ldots, i_n$ such that $i_j \in \{1, \ldots, k\}$ for all $1 \leq j \leq n$, and

$$x_{i_1} x_{i_2} \cdots x_{i_n} = y_{i_1} y_{i_2} \cdots y_{i_n}.$$

Theorem

The PCP is undecidable.

Reduction to first-order logic

$$\left\{ \begin{bmatrix} 1 \\ 101 \end{bmatrix}, \begin{bmatrix} 10 \\ 00 \end{bmatrix}, \begin{bmatrix} 011 \\ 11 \end{bmatrix} \right\}.$$

Reduction to first-order logic

$$\left\{ \begin{bmatrix} 1 \\ 101 \end{bmatrix}, \begin{bmatrix} 10 \\ 00 \end{bmatrix}, \begin{bmatrix} 011 \\ 11 \end{bmatrix} \right\}.$$

Encode strings using terms.

- Introduce constant symbol e.
- Introduce unary function symbols f_0 and f_1.
- Write e.g. $f_{10110}(e)$ instead of $f_1(f_0(f_1(f_1(f_0(e)))))$.

Reduction to first-order logic

$$\left\{ \left[\begin{array}{c} 1 \\ 101 \end{array} \right], \left[\begin{array}{c} 10 \\ 00 \end{array} \right], \left[\begin{array}{c} 011 \\ 11 \end{array} \right] \right\}.$$

Encode strings using terms.

- Introduce constant symbol e.
- Introduce unary function symbols f_0 and f_1.
- Write e.g. $f_{10110}(e)$ instead of $f_1(f_0(f_1(f_1(f_0(e)))))$.

Introduce binary predicate symbol $P(x, y)$.

- Write a formula which expresses that $P(x, y)$ hold iff the pair of strings (x, y) can be built using a sequence of given tiles.

Reduction to first-order logic

$$\left\{ \begin{bmatrix} 1 \\ 101 \end{bmatrix}, \begin{bmatrix} 10 \\ 00 \end{bmatrix}, \begin{bmatrix} 011 \\ 11 \end{bmatrix} \right\}.$$

Encode strings using terms.

- Introduce constant symbol e.
- Introduce unary function symbols f_0 and f_1.
- Write e.g. $f_{10110}(e)$ instead of $f_1(f_0(f_1(f_1(f_0(e)))))$.

Introduce binary predicate symbol $P(x, y)$.

- Write a formula which expresses that $P(x, y)$ hold iff the pair of strings (x, y) can be built using a sequence of given tiles.

Ex1: Find such a formula for the three tiles from above.

Ex2: Using a formula, express the existence of a solution.

$$\left\{ \begin{bmatrix} 1 \\ 101 \end{bmatrix}, \begin{bmatrix} 10 \\ 00 \end{bmatrix}, \begin{bmatrix} 011 \\ 11 \end{bmatrix} \right\}.$$

$$\begin{aligned}
F &= F_1 \wedge F_2 \to F_3, \\
F_1 &= P(f_1(e), f_{101}(e)) \wedge P(f_{10}(e), f_{00}(e)) \wedge P(f_{011}(e), f_{11}(e)), \\
F_2 &= \forall u \forall v\, (P(u,v) \to P(f_1(u), f_{101}(v))) \\
&\qquad\qquad \wedge (P(u,v) \to P(f_{10}(u), f_{00}(v)) \\
&\qquad\qquad \wedge (P(u,v) \to P(f_{011}(u), f_{11}(v))). \\
F_3 &= \exists u\, P(u,u).
\end{aligned}$$

Reduction to first-order logic

Given instance P of PCP

$$P = \{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq \{0, 1\}^* \times \{0, 1\}^*.$$

Reduction to first-order logic

Given instance P of PCP

$$P = \{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq \{0,1\}^* \times \{0,1\}^*.$$

Define

$$\begin{aligned}
F_1 &= \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)), \\
F_2 &= \forall u \forall v \bigwedge_{i=1}^{k} (P(u,v) \rightarrow P(f_{x_i}(u), f_{y_i}(v))), \\
F_3 &= \exists u \, P(u,u).
\end{aligned}$$

Reduction to first-order logic

Given instance P of PCP

$$P = \{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq \{0,1\}^* \times \{0,1\}^*.$$

Define

$$\begin{aligned}
F_1 &= \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)), \\
F_2 &= \forall u \forall v \bigwedge_{i=1}^{k} (P(u,v) \to P(f_{x_i}(u), f_{y_i}(v))), \\
F_3 &= \exists u \, P(u,u).
\end{aligned}$$

Proposition

P has a solution if and only if $F_1 \wedge F_2 \to F_3$ is valid.

Reduction to first-order logic

Given instance P of PCP

$$P = \{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq \{0,1\}^* \times \{0,1\}^*.$$

Define

$$\begin{aligned}
F_1 &= \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)), \\
F_2 &= \forall u \forall v \bigwedge_{i=1}^{k} (P(u,v) \to P(f_{x_i}(u), f_{y_i}(v))), \\
F_3 &= \exists u\, P(u,u).
\end{aligned}$$

Proposition

P has a solution if and only if $F_1 \wedge F_2 \to F_3$ is valid.

Ex: Prove the proposition.

$$F_1 = \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e))$$

$$F_3 = \exists u \, P(u, u)$$

$$F_2 = \forall u \, \forall v \, \bigwedge_{i=1}^{k} (P(u, v) \rightarrow P(f_{x_i}(u), f_{y_i}(v)))$$

$$F_1 = \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)) \qquad\qquad F_3 = \exists u\, P(u,u)$$

$$F_2 = \forall u\, \forall v\, \bigwedge_{i=1}^{k} (P(u,v) \to P(f_{x_i}(u), f_{y_i}(v)))$$

- If $F_1 \wedge F_2 \to F_3$ is valid, consider the Herbrand structure $\mathcal{H}$ with

$$P_{\mathcal{H}} = \{(f_u(e), f_v(e)) : \exists i_1 \ldots \exists i_t \,.\, u = x_{i_1} \ldots x_{i_t} \text{ and } v = y_{i_1} \ldots y_{i_t}\}.$$

$$F_1 = \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)) \qquad F_3 = \exists u\, P(u,u)$$

$$F_2 = \forall u \,\forall v \, \bigwedge_{i=1}^{k} (P(u,v) \to P(f_{x_i}(u), f_{y_i}(v)))$$

- If $F_1 \wedge F_2 \to F_3$ is valid, consider the Herbrand structure $\mathcal{H}$ with

  $$P_{\mathcal{H}} = \{(f_u(e), f_v(e)) : \exists i_1 \ldots \exists i_t \,.\, u = x_{i_1} \ldots x_{i_t} \text{ and } v = y_{i_1} \ldots y_{i_t}\}.$$

  Now $\mathcal{H} \models F_1 \wedge F_2$. So, $\mathcal{H} \models F_3$. But then P has a solution.

$$F_1 = \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)) \qquad\qquad F_3 = \exists u\, P(u,u)$$

$$F_2 = \forall u\, \forall v\, \bigwedge_{i=1}^{k} (P(u,v) \to P(f_{x_i}(u), f_{y_i}(v)))$$

- If $F_1 \wedge F_2 \to F_3$ is valid, consider the Herbrand structure $\mathcal{H}$ with

  $$P_{\mathcal{H}} = \{(f_u(e), f_v(e)) : \exists i_1 \ldots \exists i_t \,.\, u = x_{i_1} \ldots x_{i_t} \text{ and } v = y_{i_1} \ldots y_{i_t}\}.$$

  Now $\mathcal{H} \models F_1 \wedge F_2$. So, $\mathcal{H} \models F_3$. But then P has a solution.

- If P has a solution, consider $\mathcal{A}$ that satisfies $F_1 \wedge F_2$. Show by induction on t that for every sequence of tiles $i_1 \ldots i_t$,

  $$\mathcal{A} \models P(f_u(e), f_v(e)), \text{ where } u = x_{i_1} \ldots x_{i_t} \text{ and } v = y_{i_1} \ldots y_{i_t}.$$

$$F_1 = \bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)) \qquad F_3 = \exists u\, P(u,u)$$

$$F_2 = \forall u \forall v \bigwedge_{i=1}^{k} (P(u,v) \rightarrow P(f_{x_i}(u), f_{y_i}(v)))$$

- If $F_1 \wedge F_2 \rightarrow F_3$ is valid, consider the Herbrand structure $\mathcal{H}$ with

  $$P_{\mathcal{H}} = \{(f_u(e), f_v(e)) : \exists i_1 \ldots \exists i_t \,.\, u = x_{i_1} \ldots x_{i_t} \text{ and } v = y_{i_1} \ldots y_{i_t}\}.$$

  Now $\mathcal{H} \models F_1 \wedge F_2$. So, $\mathcal{H} \models F_3$. But then P has a solution.

- If P has a solution, consider $\mathcal{A}$ that satisfies $F_1 \wedge F_2$. Show by induction on t that for every sequence of tiles $i_1 \ldots i_t$,

  $$\mathcal{A} \models P(f_u(e), f_v(e)), \text{ where } u = x_{i_1} \ldots x_{i_t} \text{ and } v = y_{i_1} \ldots y_{i_t}.$$

  But since P has a solution, $\mathcal{A} \models P(f_u(e), f_u(e))$ for some string u. Thus, $\mathcal{A} \models F_3$.

Proposition

P has a solution if and only if $F_1 \wedge F_2 \rightarrow F_3$ is valid.

Proposition

P has a solution if and only if $F_1 \wedge F_2 \rightarrow F_3$ is valid.

Theorem

Validity in first-order logic is undecidable.

Theorem

Satisfiability in first-order logic is undecidable.

Proposition

P has a solution if and only if $F_1 \wedge F_2 \rightarrow F_3$ is valid.

Theorem

Validity in first-order logic is undecidable.

Theorem

Satisfiability in first-order logic is undecidable.

Ex: Prove these theorems. Hint: Use the proposition and the undecidability of the PCP problem P.

Proposition

P has a solution if and only if $F_1 \wedge F_2 \rightarrow F_3$ is valid.

Theorem

Validity in first-order logic is undecidable.

Theorem

Satisfiability in first-order logic is undecidable.

Ex: Prove these theorems. Hint: Use the proposition and the undecidability of the PCP problem P.

Theorem

Satisfiability in first-order logic is not semi-decidable.

Proposition

P has a solution if and only if $F_1 \wedge F_2 \rightarrow F_3$ is valid.

Theorem

Validity in first-order logic is undecidable.

Theorem

Satisfiability in first-order logic is undecidable.

Ex: Prove these theorems. Hint: Use the proposition and the undecidability of the PCP problem P.

Theorem

Satisfiability in first-order logic is not semi-decidable.

Ex: Prove it. Hint: Use the semi-decidability and the undecidability of validity in first-order logic.