universität innsbruck

SS 2024 lecture 8

Logic

Diana Gründlinger Aart Middeldorp Fabian Mitterwallner

Alexander Montag Johannes Niederhauser Daniel Rainer

Outline

Theorem

$$\neg\forall x\,\varphi \dashv\vdash \exists x\,\neg\varphi \qquad \neg\exists x\,\varphi \dashv\vdash \forall x\,\neg\varphi$$
$$\forall x\,\varphi \wedge \forall x\,\psi \dashv\vdash \forall x\,(\varphi \wedge \psi) \qquad \exists x\,\varphi \vee \exists x\,\psi \dashv\vdash \exists x\,(\varphi \vee \psi)$$
$$\forall x\,\forall y\,\varphi \dashv\vdash \forall y\,\forall x\,\varphi \qquad \exists x\,\exists y\,\varphi \dashv\vdash \exists y\,\exists x\,\varphi$$

if x is not free in ψ then

$$\forall x\,\varphi \wedge \psi \dashv\vdash \forall x\,(\varphi \wedge \psi) \qquad \forall x\,\varphi \vee \psi \dashv\vdash \forall x\,(\varphi \vee \psi)$$
$$\exists x\,\varphi \wedge \psi \dashv\vdash \exists x\,(\varphi \wedge \psi) \qquad \exists x\,\varphi \vee \psi \dashv\vdash \exists x\,(\varphi \vee \psi)$$
$$\psi \to \forall x\,\varphi \dashv\vdash \forall x\,(\psi \to \varphi) \qquad \exists x\,\varphi \to \psi \dashv\vdash \forall x\,(\varphi \to \psi)$$
$$\psi \to \exists x\,\varphi \dashv\vdash \exists x\,(\psi \to \varphi) \qquad \forall x\,\varphi \to \psi \dashv\vdash \exists x\,(\varphi \to \psi)$$

Definitions

- substitution is set of variable bindings $\theta = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ with pairwise different variables $x_1, \dots, x_n$ and terms $t_1, \dots, t_n$
- given substitution $\theta = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ and expression E, instance $E\theta$ of E is obtained by simultaneously replacing each occurrence of x_i in E by t_i
- composition of substitutions $\theta = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ and $\sigma = \{y_1 \mapsto s_1, \dots, y_k \mapsto s_k\}$ is substitution $\theta\sigma = \{x_1 \mapsto t_1\sigma, \dots, x_n \mapsto t_n\sigma\} \cup \{y_i \mapsto s_i \mid y_i \neq x_j \text{ for all } 1 \leqslant j \leqslant n\}$
- substitution θ is at least as general as substitution σ if $\theta\mu = \sigma$ for some substitution μ
- unifier of terms s and t is substitution θ such that $s\theta = t\theta$
- most general unifier (mgu) is at least as general as any other unifier

Theorem

unifiable terms have mgu which can be computed by unification algorithm

Unification Algorithm

d decomposition
$$\frac{E_1,\ f(s_1,\dots,s_n) \approx f(t_1,\dots,t_n),\ E_2}{E_1,\ s_1 \approx t_1,\dots,s_n \approx t_n,\ E_2}$$

t removal of trivial equations
$$\frac{E_1,\ t \approx t,\ E_2}{E_1,\ E_2}$$

v variable elimination
$$\frac{E_1,\ x \approx t,\ E_2}{(E_1,\ E_2)\{x \mapsto t\}} \quad \text{and} \quad \frac{E_1,\ t \approx x,\ E_2}{(E_1,\ E_2)\{x \mapsto t\}}$$

if x does not occur in t (occurs check)

Theorem

- there are no infinite derivations $U \Rightarrow_{\theta_1} V \Rightarrow_{\theta_2} \cdots$
- if s and t are unifiable then for every maximal derivation $s \approx t \Rightarrow_{\theta_1} E_1 \Rightarrow_{\theta_2} \cdots \Rightarrow_{\theta_n} E_n$
  $E_n = \square$ and $\theta_1\theta_2\cdots\theta_n$ is mgu of s and t

Definitions

- prenex normal form is predicate logic formula
$$Q_1 x_1\, Q_2 x_2\, \dots\, Q_n x_n\, \varphi$$
  with $Q_i \in \{\forall, \exists\}$ and φ quantifier-free
- Skolem normal form is closed (no free variables) prenex normal form
$$\forall x_1\, \forall x_2\, \dots\, \forall x_n\, \varphi$$
  with φ quantifier-free CNF

Theorem

for every formula φ there exists prenex normal form ψ such that $\varphi \equiv \psi$

Theorem

for every sentence φ there exists Skolem normal form ψ such that $\varphi \approx \psi$

Proof (Skolemization)

① transform φ into closed prenex normal form $Q_1 x_1\, Q_2 x_2 \dots Q_n x_n\, \chi$ with χ in CNF

② repeatedly replace $\forall x_1 \dots \forall x_{i-1}\, \exists x_i\, Q_{i+1} x_{i+1} \dots Q_n x_n\, \psi$ by
$$\forall x_1 \dots \forall x_{i-1}\, Q_{i+1} x_{i+1} \dots Q_n x_n\, \psi[f(x_1,\dots,x_{i-1})/x_i]$$
where f is new function symbol of arity $i-1$

Part I: Propositional Logic

algebraic normal forms, binary decision diagrams, conjunctive normal forms, DPLL, Horn formulas, natural deduction, Post's adequacy theorem, resolution, SAT, semantics, sorting networks, soundness and completeness, syntax, Tseitin's transformation

Part II: Predicate Logic

natural deduction, quantifier equivalences, resolution, semantics, Skolemization, syntax, undecidability, unification

Part III: Model Checking

adequacy, branching-time temporal logic, CTL*, fairness, linear-time temporal logic, model checking algorithms, symbolic model checking

Outline

Definitions

- literal is atom p or negation of atom $\neg p$
- clause is set of literals $\{\ell_1, \dots, \ell_n\}$
- $\square$ denotes empty clause
- clausal form is set of clauses $\{C_1, \dots, C_m\}$
- $\ell^c = \begin{cases} \neg p & \text{if } \ell = p \\ p & \text{if } \ell = \neg p \end{cases}$
- clauses C_1 and C_2 clash on literal ℓ if $\ell \in C_1$ and $\ell^c \in C_2$
- resolvent of clauses C_1 and C_2 clashing on literal ℓ is clause $(C_1 \setminus \{\ell\}) \cup (C_2 \setminus \{\ell^c\})$

Resolution

input: clausal form S

output: yes if S is satisfiable no if S is unsatisfiable

① repeatedly add (new) resolvents of clashing clauses in S

② return no as soon as empty clause is derived

③ return yes if all clashing clauses have been resolved

Definition

refutation of S is resolution derivation of $\square$ from S

Theorem

resolution is sound and complete for propositional logic:

clausal form S is unsatisfiable if and only if S admits refutation

Outline

Definitions

- atomic formula: $P \mid P(t,\dots,t) \mid t = t$
- literal is atomic formula or negation of atomic formula
- clause is set of literals $\{\ell_1,\dots,\ell_n\}$
- clausal form is set of clauses $\{C_1,\dots,C_m\}$, representing $\forall\,(C_1 \wedge \cdots \wedge C_m)$
- clauses C_1 and C_2 without common variables clash on literals $\ell_1 \in C_1$ and $\ell_2 \in C_2$ if ℓ_1 and ℓ_2^c are unifiable
- resolvent of clauses C_1 and C_2 clashing on literals $\ell_1 \in C_1$ and $\ell_2 \in C_2$ is clause

$$((C_1 \setminus \{\ell_1\}) \cup (C_2 \setminus \{\ell_2\}))\theta$$

where θ is mgu of ℓ_1 and ℓ_2^c

Example ❶

1	$\{\neg P(x),\ Q(x),\ R(x,f(x))\}$		
2	$\{\neg P(x),\ Q(x),\ S(f(x))\}$		
3	$\{T(a)\}$		
4	$\{P(a)\}$		
5	$\{\neg R(a,y),\ T(y)\}$		
6	$\{\neg T(x),\ \neg Q(x)\}$		
7	$\{\neg T(x),\ \neg S(x)\}$		
8	$\{\neg Q(a)\}$	resolve 3, 6	$\{x \mapsto a\}$
9	$\{Q(a),\ S(f(a))\}$	resolve 2, 4	$\{x \mapsto a\}$
10	$\{Q(a),\ R(a,f(a))\}$	resolve 1, 4	$\{x \mapsto a\}$
11	$\{S(f(a))\}$	resolve 8, 9	
12	$\{R(a,f(a))\}$	resolve 8, 10	
13	$\{T(f(a))\}$	resolve 5, 12	$\{y \mapsto f(a)\}$
14	$\{\neg S(f(a))\}$	resolve 7, 13	$\{x \mapsto f(a)\}$
15	$\square$	resolve 11, 14	

Example ❷

1	$\{\neg P(x,y),\ P(y,x)\}$		
2	$\{\neg P(x,y),\ \neg P(y,z),\ P(x,z)\}$		
3	$\{P(x,f(x))\}$		
4	$\{\neg P(x,x)\}$		
3′	$\{P(x',f(x'))\}$	rename 3	
5	$\{P(f(x),x)\}$	resolve 1, 3′	$\{y \mapsto f(x),\ x' \mapsto x\}$
6	$\{\neg P(f(x),z),\ P(x,z)\}$	resolve 2, 3′	$\{y \mapsto f(x),\ x' \mapsto x\}$
5′	$\{P(f(x'),x')\}$	rename 5	
7	$\{P(z,z)\}$	resolve 6, 5′	$\{x \mapsto z,\ x' \mapsto z\}$
8	$\square$	resolve 4, 7	$\{x \mapsto z\}$

$$\forall x\,\forall y\,\forall z\,((\neg P(x,y) \vee P(y,x)) \wedge (\neg P(x,y) \vee \neg P(y,z) \vee P(x,z)) \wedge P(x,f(x)) \wedge \neg P(x,x))$$

Theorem

resolution is sound for predicate logic: clausal form S is unsatisfiable if S admits refutation

Problem

resolution is incomplete for predicate logic

Example

1 $\{P(x),\ P(y)\}$

2 $\{\neg P(x'),\ \neg P(y')\}$

3 $\{P(y),\ \neg P(y')\}$ resolve 1, 2 $\{x \mapsto x'\}$

unsatisfiable but no refutation

Solution

incorporate factoring: $C\theta$ is factor of C if two or more literals in C have mgu θ

Example

1 $\{P(x), P(y)\}$

2 $\{\neg P(x'), \neg P(y')\}$

3 $\{P(x)\}$ factor 1

4 $\{\neg P(x')\}$ factor 2

5 $\Box$ resolve 3, 4

Resolution with Factoring

input: clausal form S

output: yes if S is satisfiable

no if S is unsatisfiable

∞ if S is satisfiable (or unsatisfiable)

① repeatedly add resolvents (renaming clauses if necessary) and factors

② return no as soon as empty clause $\Box$ is derived

③ return yes if all clashing clauses have been resolved and factoring produces no new clauses (modulo renaming)

Example

1 $\{R(x), Q(f(x))\}$

2 $\{\neg R(f(x)), Q(f(y))\}$

3 $\{\neg Q(f(f(f(a))))\}$

1′ $\{R(x'), Q(f(x'))\}$ rename 1

4 $\{Q(f(y)), Q(f(f(x)))\}$ resolve 1′, 2 $\{x' \mapsto f(x)\}$

5 $\{Q(f(f(x)))\}$ factor 4 $\{y \mapsto f(x)\}$

6 $\Box$ resolve 3, 5 $\{x \mapsto f(a)\}$

Theorem

resolution with factoring is sound and complete:

clausal form S is unsatisfiable if and only if S admits refutation

Example

1 $\{\neg P(x), P(f(x))\}$

2 $\{P(a)\}$

3 $\{P(f(a))\}$ resolve 1, 2 $\{x \mapsto a\}$

4 $\{P(f(f(a)))\}$ resolve 1, 3 $\{x \mapsto f(a)\}$

5 $\{P(f(f(f(a))))\}$ resolve 1, 4 $\{x \mapsto f(f(a))\}$

6 $\{P(f(f(f(f(a)))))\}$ resolve 1, 5 $\{x \mapsto f(f(f(a)))\}$

$\vdots$

Example

1 $\{a = b\}$	4 $\{x \neq y,\ y \neq z,\ x = z\}$		
2 $\{b = c\}$	5 $\{b \neq z,\ a = z\}$	resolve 1, 4	$\{x \mapsto a,\ y \mapsto b\}$
3 $\{a \neq c\}$	6 $\{a = c\}$	resolve 2, 5	$\{z \mapsto c\}$
	7 $\Box$	resolve 3, 6	

unsatisfiable but no refutation

Remark

equality needs special treatment: add equality axioms, e.g.

$$\{x \neq y,\ y \neq z,\ x = z\}$$

for transitivity

Satisfiability Procedure

sentence φ

① transform φ into Skolem normal form ψ

② extract clausal form S from ψ

③ apply resolution (with factoring) to S

④ φ is satisfiable if and only if empty clause cannot be derived

Validity Procedure

sentence φ

① transform $\neg\varphi$ into Skolem normal form ψ

② extract clausal form S from ψ

③ apply resolution (with factoring) to S

④ φ is valid if and only if empty clause can be derived

Outline

particify with session ID **0992 9580**

Question

Which of the following statements are true ?

A $\{P(a,b)\}$ is a factor of $\{P(x,b),\ \neg P(a,y)\}$.

B The literals $R(x,x,a)$ and $\neg R(f(b),g(y),y)$ do not clash.

C $\{Q(f(x)),\ R(y,z)\}$ is a resolvent of $\{\neg Q(y),\ R(y,z)\}$ and $\{Q(x),\ Q(f(x))\}$.

D A clause cannot have a factor if it contains at least two literals which are not unifiable.

Outline

Church's Theorem

validity in predicate logic is undecidable: there is no algorithm

input: formula φ in predicate logic

output: yes if $\vDash \varphi$ holds

no if $\vDash \varphi$ does not hold

Idea

reduction from Post correspondence problem

Post Correspondence Problem

instance: finite sequence of pairs $(s_1, t_1), \dots, (s_k, t_k)$ of non-empty bit strings

question: is there sequence $(i_1, i_2, \dots, i_n)$ with $n \geqslant 1$ such that $s_{i_1} s_{i_2} \dots s_{i_n} = t_{i_1} t_{i_2} \dots t_{i_n}$?

Examples

1.

	1	2	3
s_i:	1	10111	10
t_i:	11	101	01

solution

	2	1	1	
s	10111	1	1	= 1011111
t	101	11	11	= 1011111

2.

	1	2	3
s_i:	10	011	101
t_i:	101	11	011

no solution

3.

	1	2	3
s_i:	01	1	0
t_i:	0	101	1

solution 1 3 1 1 3 1 3 1 1 3 1 1 2 1 1 2 2 1 3 3 2 1
1 3 1 2 1 1 3 3 1 2 1 1 1 3 2 1 2 1 2 2 3 2

Theorem (Post, 1946)

Post correspondence problem is undecidable

Theorem (Church, 1936)

validity in predicate logic is undecidable

Idea

translate PCP instance C into predicate logic formula φ such that

$$\vDash \varphi \iff C \text{ has solution}$$

Proof

$C = ((s_1, t_1), (s_2, t_2), \dots, (s_k, t_k))$

- function symbols e: constant f_0, f_1: arity 1
  predicate symbol P: arity 2
- if $b_1, b_2, \dots, b_n \in \{0,1\}$ then $f_{b_1 b_2 \cdots b_n}(t)$ denotes $f_{b_n}(\cdots(f_{b_2}(f_{b_1}(t)))\cdots)$
- $\varphi = \varphi_1 \wedge \varphi_2 \to \varphi_3$ with

$$\varphi_1 = \bigwedge_{i=1}^{k} P(f_{s_i}(e), f_{t_i}(e))$$

$$\varphi_2 = \forall\, v\, \forall\, w\, \left(P(v,w) \to \bigwedge_{i=1}^{k} P(f_{s_i}(v), f_{t_i}(w))\right)$$

$$\varphi_3 = \exists\, z\, P(z,z)$$

- $\vDash \varphi \iff C$ has solution

Example

- $C = ((10, 101), (011, 11), (10, 0))$
- $\varphi = P(f_0(f_1(e)), f_1(f_0(f_1(e)))) \wedge P(f_1(f_1(f_0(e))), f_1(f_1(e))) \wedge P(f_0(f_1(e)), f_0(e))$

$$\wedge\, \forall\, v\, \forall\, w\, \big(P(v,w) \to P(f_0(f_1(v)), f_1(f_0(f_1(w))))$$
$$\wedge\, P(f_1(f_1(f_0(v))), f_1(f_1(w)))$$
$$\wedge\, P(f_0(f_1(v)), f_0(w))\big)$$
$$\to \exists\, z\, P(z,z)$$

Outline

Definition

set X of boolean functions is called adequate or functionally complete if every boolean function can be expressed using functions from X

Examples

- $\{\,\bar{}\,, \cdot, +\}$ is adequate: truth table gives rise to DNF
- $\{\,\bar{}\,, \cdot\}$ is adequate: $x + y = \overline{\bar{x} \cdot \bar{y}}$
- $\{\cdot, +, \to\}$ with $x \to y = \bar{x} + y$ is not adequate

x	y	$f(x,y)$
0	0	1
0	1	0
1	0	0
1	1	1

$$f(x,y) = \bar{x} \cdot \bar{y} + x \cdot y$$

Definitions

- $x \,|\, y = \overline{x \cdot y}$ (nand)
- ite$(x, y, z) = (\overline{x} + y) \cdot (x + z)$ (if-then-else)

Examples

- $\{\,|\,\}$ is adequate: $\overline{x} = x \,|\, x$, $\quad x \cdot y = (x \,|\, y) \,|\, (x \,|\, y)$
- $\{$ite$, 0, 1\}$ is adequate: $\overline{x} = \text{ite}(x, 0, 1)$, $\quad x \cdot y = \text{ite}(x, y, 0)$
- $\{\,\overline{}, \leftrightarrow\}$ with $x \leftrightarrow y = (\overline{x} + y) \cdot (x + \overline{y})$ is not adequate

Outline

Theorem (Algebraic Normal Form, ANF)

every boolean function $f\colon \{0,1\}^n \to \{0,1\}$ can be uniquely written as

$$f(x_1, \dots, x_n) = \bigoplus_{A \subseteq \{1,\dots,n\}} c_A \cdot \prod_{i \in A} x_i$$

with $c_A \in \{0,1\}$ for all $A \subseteq \{1, \dots, n\}$

Corollary

every binary boolean function $f\colon \{0,1\}^2 \to \{0,1\}$ can be uniquely written as

$$f(x_1, x_2) = c_\varnothing \oplus c_{\{1\}} x_1 \oplus c_{\{2\}} x_2 \oplus c_{\{1,2\}} x_1 x_2 = \bigoplus_{A \subseteq \{1,2\}} c_A \cdot \prod_{i \in A} x_i$$

with $c_\varnothing, c_{\{1\}}, c_{\{2\}}, c_{\{1,2\}} \in \{0,1\}$

Theorem (Algebraic Normal Form, ANF)

every boolean function $f\colon \{0,1\}^n \to \{0,1\}$ can be uniquely written as

$$f(x_1, \dots, x_n) = \bigoplus_{A \subseteq \{1,\dots,n\}} c_A \cdot \prod_{i \in A} x_i$$

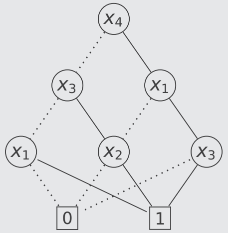
with $c_A \in \{0,1\}$ for all $A \subseteq \{1, \dots, n\}$

Proof sketch

- $n = 0$: easy
- $n > 0$: $f = f[0/x] \oplus (f[0/x] \oplus f[1/x])\, x$

$$\begin{aligned} f &= \overline{x}\, f[0/x] + x\, f[1/x] = f[0/x]\,\overline{x} + f[1/x]\, x && \text{(Shannon expansion)} \\ &= f[0/x]\,\overline{x} \oplus f[1/x]\, x \oplus f[0/x]\,\overline{x}\, f[1/x]\, x && (y + z = y \oplus z \oplus yz) \\ &= f[0/x]\,\overline{x} \oplus f[1/x]\, x = f[0/x]\,(1 \oplus x) \oplus f[1/x]\, x && (\overline{x} = 1 \oplus x) \\ &= f[0/x] \oplus f[0/x]\, x \oplus f[1/x]\, x = f[0/x] \oplus (f[0/x] \oplus f[1/x])\, x \end{aligned}$$

Example (Algebraic Normal Form of HWB_4)



$$
\begin{aligned}
x + y &= x \oplus y \oplus xy \\
\overline{x}x &= 0 \\
\overline{x} &= x \oplus 1 \\
(x \oplus y)z &= xz \oplus yz \\
1x &= x \\
&\cdots
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{HWB}_4(x_1,x_2,x_3,x_4) &= \overline{x}_4(\overline{x}_3x_1 + x_3x_2) + x_4(\overline{x}_1x_2 + x_1x_3) \\
&= \overline{x}_4(\overline{x}_3x_1 \oplus x_3x_2) \oplus x_4(\overline{x}_1x_2 \oplus x_1x_3) \\
&= \overline{x}_4(x_1 \oplus x_1x_3 \oplus x_3x_2) \oplus x_4(\overline{x}_1x_2 \oplus x_1x_3) \\
&= \overline{x}_4(x_1 \oplus x_1x_3 \oplus x_3x_2) \oplus x_4(x_2 \oplus x_1x_2 \oplus x_1x_3) \\
&= x_1 \oplus x_1x_3 \oplus x_3x_2 \oplus x_4(x_2 \oplus x_1x_2 \oplus x_1x_3) \oplus x_4(x_1 \oplus x_1x_3 \oplus x_3x_2) \\
&= x_1 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_2x_3x_4
\end{aligned}
$$

Outline

Huth and Ryan

- Section 2.5

Resolution

- Wikipedia [accessed January 25, 2024]

Algebraic Normal Form

- Wikipedia [accessed January 25, 2024]

Important Concepts

- adequacy
- algebraic normal form (ANF)
- Church's theorem
- clashing
- factor
- factoring
- functional completeness
- nand
- Post correspondence problem
- resolvent

homework for May 16