Logik für Informatiker
Logic for computer scientists

Till Mossakowski

WiSe 2013/14

First-order resolution

First-order resolution

- generalises propositional resolution to first-order logic
- is a proof system that is well-suited for efficient implementation
- many automated first-order provers are based on resolution: SPASS, Prover9, Vampire
- also interactive provers for higher-order logic are based on resolution: Isabelle, HOL, HOL-light

Satisfiability and logical consequence

Logical consequence can be reduced to (un)satisfiability:

> *The logical consequence* $\mathcal{T} \models S$ *holds*
> *if and only if*
> $\mathcal{T} \cup \{\neg S\}$ *is unsatisfiable.*

Note: Resolution is about satisfiability.

Skolemization

The sentence

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

is logically equivalent to the second-order sentence

$$\exists f \forall x Neighbor(x, f(x))$$

In first-order logic, we have the Skolem normal form

$$\forall x Neighbor(x, f(x))$$

Theorem about Skolem normal form

Theorem

A sentence $S \equiv \forall x \exists y P(x, y)$ is satisfiable iff its Skolem normal form $\forall x P(x, f(x))$ is.

Every structure satisfying the Skolem normal form also satisfies S. Moreover, every structure satisfying S can be turned into one satisfying the Skolem normal form. This is done by interpreting f by a function which picks out, for any object b in the domain, some object c such that they satisfy $P(x, y)$.

Unification of terms

$$\{P(f(a)), \forall x\ \neg P(f(g(x)))\}$$

is satisfiable, but

$$\{P(f(g(a))), \forall x\ \neg P(f(x))\}$$

is not. This can be seen with unification.
Terms $t_1, \ldots, t_n$ are unifiable, if there is a substitution of terms for some or all the variables in $t_1, \ldots, t_n$ such that the terms that result from the substitution are syntactically identical terms.

Example

$$f(g(z), x), \quad f(y, x), \quad f(y, h(a))$$

are unifiable by substituting $h(a)$ for x and $g(z)$ for y.

Recall Prenex Form: Rules for conjunctions and disjunctions

$\forall x Q \wedge P \rightsquigarrow \forall x(Q \wedge P)$ $\quad$ $\exists x Q \wedge P \rightsquigarrow \exists x(Q \wedge P)$

$P \wedge \forall x Q \rightsquigarrow \forall x(P \wedge Q)$ $\quad$ $P \wedge \exists x Q \rightsquigarrow \exists x(P \wedge Q)$

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

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

Note that x must not be a free variable in P.
If x is a free variable in P, we can achieve this condition by the following rule:

$\forall x Q \rightsquigarrow \forall y Q[y/x]$
Here, $Q[y/x]$ is Q with all free occurrences of x replaced by y.

Recall Prenex Form: Rules for negations, implications, equivalences

$\neg\forall x P \leadsto \exists x \neg P$ $\qquad$ $\neg\exists x P \leadsto \forall x \neg P$

$\forall x Q \to P \leadsto \exists x(Q \to P)$ $\qquad$ $\exists x Q \to P \leadsto \forall x(Q \to P)$

$P \to \forall x Q \leadsto \forall x(P \to Q)$ $\qquad$ $P \to \exists x Q \leadsto \exists x(P \to Q)$

$P \leftrightarrow Q \leadsto (P \to Q) \wedge (Q \to P)$

Note that for the second and third line, x must not be a free variable in P.

Alpha-renaming (change of bound variables)

The Prenex normal form algorithm assumes that all variables in a formula are distinct. This can be achieved by α-renaming:

$\forall x P(x) \rightsquigarrow \forall y P(y)$

$\exists x P(x) \rightsquigarrow \exists y P(y)$

Resolution for FOL

How to show unsatisfiability of set $\mathcal{T}$ of sentences?

1. Put each sentence in $\mathcal{T}$ into prenex form, say
$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots P(x_1, y_1, x_2, y_2, \ldots)$$
2. Skolemize each of the resulting sentences, say
$$\forall x_1 \forall x_2 \ldots P(x_1, f_1(x_1), x_2, f_2(x_1, x_2), \ldots)$$
using different Skolem functions for different sentences.
3. Put each quantifier free matrix P into conjunctive normal form, say
$$P_1 \wedge P_2 \wedge \ldots \wedge P_n$$
where each P_i is a disjunction of literals.
4. Distribute the universal quantifiers in each sentence across the conjunctions and drop the conjunction signs:
$$\forall x_1 \forall x_2 \ldots P_i$$

5. Change the bound variables in each of the resulting sentences so that no variable appears in two of them.
6. Turn each of the resulting sentences into a set of literals by dropping the universal quantifiers and disjunction signs. In this way we end up with a set of resolution clauses.
7. Use resolution and unification to resolve this set of clauses

$$\frac{\{C_1, \dots, C_m\}, \ \{\neg D_1, \dots, D_n\}}{\{C_2\theta, \dots C_m\theta, D_2\theta, \dots, D_n\theta\}}$$

if $C_1\theta = D_1\theta$ (θ is a unifier of C_1 and D_1)

Example I

Is the following argument valid?

$$\begin{array}{|l} \forall x(\mathrm{Cube}(x) \vee \mathrm{Tet}(x)) \\ \exists x \neg \mathrm{Cube}(x) \\ \hline \exists x \mathrm{Tet}(x) \end{array}$$

Reformulated: is the following set unsatisfiable?

$$\forall x(Cube(x) \vee Tet(x))$$
$$\exists x \neg Cube(x)$$
$$\neg \exists x Tet(x)$$

Step 1: Prenex normal form

$$\forall x(Cube(x) \vee Tet(x))$$
$$\exists x \neg Cube(x)$$
$$\forall x \neg Tet(x)$$

Step 2: Skolemization

$$\forall x(Cube(x) \vee Tet(x))$$
$$\neg Cube(c)$$
$$\forall x \neg Tet(x)$$

Since the existential quantifier was not preceeded by any universal quantifier, we need a 0-ary function symbol, that is, an individual constant c.

Step 3: This is already in conjunctive normal form.
Step 4: Drop conjunctions: nothing to do either.

Step 5: change bound variables

$$\forall x(Cube(x) \lor Tet(x))$$
$$\neg Cube(c)$$
$$\forall y \neg Tet(y)$$

Step 6: Drop universal quantifiers and disjunctions, and step 7: do resolution

1. $\{Cube(x),\ Tet(x)\}$
2. $\{\neg Cube(c)\}$
3. $\{\neg Tet(y)\}$
4. $\{Tet(c)\}$ 1,2 with c for x
5. $\square$ 3,4 with c for y

Example II

Is the following argument valid?

$$\begin{array}{|l} \forall x(\mathrm{Cube}(x) \to \exists y \mathrm{BackOf}(y,x)) \\ \forall x \forall y(\mathrm{BackOf}(x,y) \to \mathrm{Large}(y)) \\ \hline \forall x(\mathrm{Cube}(x) \to \mathrm{Large}(x)) \end{array}$$

Reformulated: is the following set unsatisfiable?

$$\forall x(Cube(x) \to \exists y BackOf(y,x))$$
$$\forall x \forall y(BackOf(x,y) \to Large(y))$$
$$\neg \forall x(Cube(x) \to Large(x))$$

Step 1: Prenex normal form

$$\forall x \exists y (Cube(x) \rightarrow BackOf(y, x))$$
$$\forall x \forall y (BackOf(x, y) \rightarrow Large(y))$$
$$\exists x \neg (Cube(x) \rightarrow Large(x))$$

Step 2: Skolemization

$$\forall x(Cube(x) \rightarrow BackOf(f(x), x))$$
$$\forall x \forall y(BackOf(x, y) \rightarrow Large(y))$$
$$\neg(Cube(c) \rightarrow Large(c))$$

Since the first existential quantifier was preceeded by a universal quantifier, we need a 1-ary function symbol f. For the other one, an individual constant c suffices.

Step 3: Conjunctive normal form

$$\forall x(\neg Cube(x) \vee BackOf(f(x), x))$$
$$\forall x \forall y(\neg BackOf(x, y) \vee Large(y))$$
$$Cube(c) \wedge \neg Large(c)$$

Step 4: Drop conjunctions

$$\forall x(\neg Cube(x) \vee BackOf(f(x), x))$$
$$\forall x \forall y(\neg BackOf(x, y) \vee Large(y))$$
$$Cube(c)$$
$$\neg Large(c)$$

Step 5: change bound variables

$$\forall z(\neg Cube(z) \vee BackOf(f(z), z))$$
$$\forall x \forall y(\neg BackOf(x, y) \vee Large(y))$$
$$Cube(c)$$
$$\neg Large(c)$$

Step 6: Drop universal quantifiers and disjunctions, and step 7: do resolution

1. $\{\neg Cube(z),\ BackOf(f(z), z)\}$
2. $\{\neg BackOf(x, y),\ Large(y)\}$
3. $\{Cube(c)\}$
4. $\{\neg Large(c)\}$
5. $\{BackOf(f(c), c)\}$ 1,3 with c for z
6. $\{Large(c)\}$ 2,5 with c for y, $f(c)$ for x
7. $\square$ 4,6

Example III

Is the following argument valid?

$$\begin{array}{|l} \forall x(P(x,b) \vee Q(x)) \\ \forall y(\neg P(f(y),b) \vee Q(y)) \\ \hline \forall y(Q(y) \vee Q(f(y))) \end{array}$$

Reformulated: is the following set unsatisfiable?

$$\begin{array}{l} \forall x(P(x,b) \vee Q(x)) \\ \forall y(\neg P(f(y),b) \vee Q(y)) \\ \neg\forall y(Q(y) \vee Q(f(y))) \end{array}$$

Step 1: Prenex normal form

$$\forall x(P(x,b) \lor Q(x))$$
$$\forall y(\ \neg P(f(y),b) \lor Q(y))$$
$$\exists y \neg(Q(y) \lor Q(f(y))$$

Step 2: Skolemization

$$\forall x(P(x,b) \vee Q(x))$$
$$\forall y(\neg P(f(y),b) \vee Q(y))$$
$$\neg(Q(c) \vee Q(f(c)))$$

Since the existential quantifier was not preceeded by any universal quantifier, we need a 0-ary function symbol, that is, an individual constant c.

Step 3: Conjunctive normal form

$$\forall x(P(x,b) \lor Q(x))$$
$$\forall y(\ \neg P(f(y),b) \lor Q(y))$$
$$\neg Q(c) \land \neg Q(f(c))$$

Step 4: Drop conjunctions

$$
\begin{array}{l}
\forall x(P(x,b) \vee Q(x)) \\
\forall y(\neg P(f(y),b) \vee Q(y)) \\
\neg Q(c) \\
\neg Q(f(c))
\end{array}
$$

Step 5: change bound variables: nothing to do.

Step 6: Drop universal quantifiers and disjunctions, and step 7: do resolution

1. $\{P(x,b),\ Q(x)\}$
2. $\{\neg P(f(y),b),\ Q(y)\}$
3. $\{\neg Q(c)\}$
4. $\{\neg Q(f(c))\}$

5. $\{Q(y), Q(f(y))\}$ 1,2 with $f(y)$ for x
6. $\{Q(f(c))\}$ 3,5 with c for y
7. $\square$ 4,6

Example IV

Is the following argument valid?

From

"Everyone admires someone who admires them unless they admire Quaid."

we can infer

"There are people who admire each other, at least one of whom admires Quaid."

The formalization

$$\begin{array}{|l} \forall x[\neg A(x,q) \rightarrow \exists y(A(x,y) \wedge A(y,x))] \\ \hline \exists x \exists y[A(x,q) \wedge A(x,y) \wedge A(y,x)] \end{array}$$

Reformulated: is the following set unsatisfiable?

$$\forall x[\neg A(x,q) \rightarrow \exists y(A(x,y) \wedge A(y,x))]$$
$$\neg \exists x \exists y[A(x,q) \wedge A(x,y) \wedge A(y,x)]$$

Step 1: Prenex normal form

$$\forall x \exists y [\neg A(x,q) \to (A(x,y) \land A(y,x))]$$
$$\forall x \forall y \neg [A(x,q) \land A(x,y) \land A(y,x)]$$

Step 2: Skolemization

$$\forall x [\neg A(x,q) \to (A(x,f(x)) \land A(f(x),x))]$$
$$\forall x \forall y \neg [A(x,q) \land A(x,y) \land A(y,x)]$$

Step 3: Conjunctive normal form

$$\forall x [(A(x,q) \lor A(x,f(x))) \land (A(x,q) \lor A(f(x),x))]$$
$$\forall x \forall y [\neg A(x,q) \lor \neg A(x,y) \lor \neg A(y,x)]$$

Step 4: Drop conjunctions

$$\forall x(A(x,q) \vee A(x,f(x)))$$
$$\forall x(A(x,q) \vee A(f(x),x))$$
$$\forall x \forall y[\neg A(x,q) \vee \neg A(x,y) \vee \neg A(y,x)]$$

Step 5: change bound variables.

$$\forall x(A(x,q) \vee A(x,f(x)))$$
$$\forall y(A(y,q) \vee A(f(y),y))$$
$$\forall z \forall w[\neg A(z,q) \vee \neg A(z,w) \vee \neg A(w,z)]$$

Step 6: Drop universal quantifiers and disjunctions, and step 7: do resolution

1. $\{A(x, q), A(x, f(x))\}$
2. $\{A(y, q), A(f(y), y)\}$
3. $\{\neg A(z, q), \neg A(z, w), \neg A(w, z)\}$
4. ... [homework: fill in the rest]

The FO Con routine of Fitch ...

... is based on automated deduction similar to resolution.
However, note: first-order consequence is undecidable (Church).
Hence, the FO Con routine at some inputs does not give a result.

Outlook

Beyond first-order logic

- many-sorted logic (variables, constants, predicates and functions have types)
  E.g.: $\forall n : Nat\ \forall l : List\ head(cons(n, l)) = n$
- partial function logic: $D(f(x))$ ("$f(x)$ is defined")
- higher-order logic: $\forall f : s \to t \ldots, \forall p : Pred(t) \ldots$

$$\begin{aligned}&\forall u \forall v (Path(u, v) \leftrightarrow\\ &\quad \forall R\ \{[\forall x \forall y \forall z (R(x, y) \land R(y, z) \to R(x, z))\\ &\qquad \land \forall x \forall y (DirectWay(x, y) \to R(x, y))]\\ &\qquad\qquad \to R(u, v)\})\end{aligned}$$

Modal and temporal logics

- modal logic:
  $\Box P$ ("necessarily P") and $\Diamond P$ ("possibly P")
  Other readings of $\Box P$:
  It ought to be that P
  It is known that P
  It is provable that P
  Always P (temporal logic)

- temporal logic: $\Box P$ ("always in the future, P"), $\Diamond P$ ("sometimes in the future, P"), and $\bigcirc P$ ("in the next step, P")
  e.g. $\Box bank\_account > 0$ (very unrealistic)

Further modal and temporal logics

- temporal logic of actions (TLA): $\Box[state' = f(state)]_{state}$
  read: always in the future, either the state does not change, or the next state is f applied to the previous state
- dynamic logic:
  $[p]P$ ("after every run of program p, P holds")
  $<p> P$ ("after some run of program p, P holds")

More exotic modal logics

- agent logics, e.g. ATL: agents A and B have the possibility to make a telephone call, if they cooperate
- logics for security, e.g. ABLP: A *controls* P ("agent A has the permission to perform action P")

Logics for knowledge representation/semantic web

- description logics, e.g. $\mathcal{ALC}$:
  $Elephant \doteq Mammal \sqcap \exists bodypart.Trunk \sqcap \forall color.Grey$
  abbreviates
  $$\forall x[Elephant(x) \leftrightarrow (Mammal(x) \wedge \exists y(bodypart(x, y) \wedge Trunk(y)) \wedge \forall z(color(x, z) \rightarrow Grey(z)))]$$

Multi-valued logics

- three-valued logics: truth values are true, false, and undefined
- object constraint logic (OCL)
  (for UML — the unified modeling language)

```
-- Managers get a higher salary than employees
inv Branch2:
  self.employee->forall(e | e <> self.manager
    implies self.manager.salary > e.salary)
```

Multi-valued logics (cont'd)

- fuzzy logic: truth values in the interval $[0,1]$ correspond to different degrees of truth (e.g. Peter is quite tall, is tall, is very tall)

Even more exotic logics

- paraconsistent logics
  for databases, local inconsistency is o.k. and should not lead to global inconsistency
- non-monotonic logics
  new facts make previous arguments invalid, e.g.
  $Bird(x) \vdash CanFly(x)$
  $\{Bird(x), Penguin(x)\} \vdash \neg CanFly(x)$
- linear logic (resource-bounded logic)
  $A \otimes A \vdash B$
  (we can prove B when we are allowed to use A twice)

Why do we need so many logics?

- different aspects of the complex world / of software systems
- one "big" logic covering everything would be too clumsy
- good news: most of the logics are based on propositional or first-order logics
- most of the logics use the same central notions (although always specialised to the logic at hand)
  - satisfaction of a sentence in a model
  - logical consequence
  - validity, satisfiability
  - proof calculus and its soundness and completeness