COMP30026 Models of Computation

Predicate Logic: Clausal Form

Lecture Week 4 Part 2 (Zo

Semester 2, 2021

Resolution for Predicate Logic

The resolution technique generalises to first-order predicate logic.

As for propositional logic, it assumes clausal form, that is, having a formula p

quantifi

Again, it consists on generating logical consequences (resolvents), trying to derive an empty clause, thereby proving th rmula unsatisfiable.

Existential quantifiers are eliminated in a process called Skolemization.

Consider the formula $F = \exists x \forall y \; P(x, y)$ under some interpretation $\mathcal{I}$.

F is satisfiable iff some valuation σ makes $\forall y \; P(x, y)$ true. Say that σ, with

Now con $, y))$.

This formula is satisfiable iff F is. If $\mathcal{I}$ dd to $\mathcal{I}$ the mapping of a to d_0 — this exten

$\forall y \; P(a, y)$.

If $\forall y \; P(x, y)$ is unsatisfiable, there is no valuation that will make $\forall y \; P(x, y)$ true. Hence no interpretation will make $\forall y \; P(a, y)$ true.

Skolem Constants and Functions

Now consider the formula $G = \forall y \exists x \; P(x, y)$.

We cannot conclude that $\forall y \; P(a, y)$ is satisfiable iff G is.

Since $\exists$ [illegible] $P(x, y)$
holds ma[illegible] x is a
function

But then we can generate the formula [illegible]
fresh function symbol f.

For reasons similar to those outlined on slide 4, this formula is satisfiable iff G is.

Skolemization

We call a (on slide 4) a Skolem constant, and f (on slide 5) a Skolem function.

Skolem functions can be of arbitrary arity. To eliminate $\exists y$ in
$\forall x_1, x_2,$ … e by
$f(x_1, x_2,$ …

Namely, y may depend on all three xs.

Each introduced Skolem constant or function …

Recall also our convention: We use letters from the start of the alphabet (a, b, c, …) for constants, and letters from the end of the alphabet (u, v, x, y …) for variables.

Skolemization Example

This formula has three existential quantifiers—we remove them one by one:

$\exists u \forall v \exists x \forall y \exists z$

$))$

Skolemization Example

This formula has three existential quantifiers—we remove them one by one:

$\exists u \forall v \exists x \forall y \exists z$

))

$\rightsquigarrow$ $\forall v$

)

Skolemization Example

This formula has three existential quantifiers—we remove them one by one:

$$\exists u \forall v \exists x \forall y \exists z$$

$$))$$

$$\leadsto \quad \forall v$$

$$)$$

$$\leadsto \quad \forall v \forall y \exists z$$

$$((\neg P(c, f(v), h(v), b) \vee R(g(h \quad)))$$

Skolemization Example

This formula has three existential quantifiers—we remove them one by one:

$\exists u \forall v \exists x \forall y \exists z$

$))$

$\rightsquigarrow \quad \forall v$

$)$

$\rightsquigarrow \quad \forall v \forall y \exists z$

$((\neg P(c, f(v), h(v), b) \vee R(g(h \quad)))$

$\rightsquigarrow \quad \forall v \forall y$

$((\neg P(c, f(v), h(v), b) \vee R(g(h \quad , \quad , \quad , \quad , (v, y))))$

Skolemization Example

This formula has three existential quantifiers—we remove them one by one:

$$\exists u \forall v \exists x \forall y \exists z$$

$$))$$

$\rightsquigarrow$ $\forall v$

$$)$$

$\rightsquigarrow$ $\forall v \forall y \exists z$

$$((\neg P(c, f(v), h(v), b) \vee R(g(h \quad)))$$

$\rightsquigarrow$ $\forall v \forall y$

$$((\neg P(c, f(v), h(v), b) \vee R(g(h \quad , \quad , \quad , \quad , (v, y)))))$$

Instead of $j(v, y)$ we could have chosen $k(v, y)$, or even $j(y, v)$—as long as we replace each occurrence of z by the same term, of course.

From Predicate Logic Formulas to Clausal Form

The process is similar to what we did with propositional formulas:

1. Re
2. Dri
3. Sta
4. Eliminate existential quantifiers (Skole
5. Eliminate universal quantifiers (just remo
6. Bring to CNF (using the distributive laws).

Clausal Form: Step 1—Use Just $\vee$, $\wedge$, $\neg$

Let us use this running example:

$$\forall x \; (P(x) \Leftrightarrow \exists y \; (R(x,y) \wedge \forall z \; R(z,y)))$$

First use t d $\oplus$):

$(\exists y \; (R(x,y) \wedge \forall z \; R$

which then becomes:

$$\forall x \; \left(\begin{array}{l} (\neg P(x) \vee \exists y \; (R(x,y) \wedge \forall z \; R(z,y))) \wedge \\ \qquad (\neg \exists y \; (R(x,y) \wedge \forall z \; R(z,y)) \vee P(x)) \end{array} \right)$$

Clausal Form: Step 2—Push Negation

Next drive negation in.

then beco

$$\forall x \left(\begin{array}{l} (\neg P(x) \vee \exists y\ (P(x, y) \\ \quad (\forall y\ (\neg R(x, y)) \vee \exists z \end{array} \right.$$

Clausal Form: Step 3—Standardize Apart

Now rename variables so that no two quantifiers use the same variable name. With that,

$$\Bigg)$$

turns into, say

$$\forall x \left(\begin{array}{l} (\neg P(x) \vee \exists y\ (R(x, y)) \\ \quad (\forall u\ (\neg R(x, u) \vee \exists v \end{array} \right.$$

Clausal Form: Step 4—Skolemize

Let us highlight the existentially quantified variables:

$$\forall x \left(\begin{array}{c} (\neg P(x) \vee \exists y \; (R(x, y) \wedge \forall z \; R(z, y))) \wedge \\ (\forall u \; (\neg R(x, u) \vee \exists v \; \neg R(v, u)) \vee P(x)) \end{array} \right)$$

The exist [illegible] ace it by $f(x)$.

Clausal Form: Step 4—Skolemize

Let us highlight the existentially quantified variables:

$$\forall x \left(\begin{array}{c} (\neg P(x) \vee \exists y\ (R(x,y) \wedge \forall z\ R(z,y))) \wedge \\ (\forall u\ (\neg R(x,u) \vee \exists v\ \neg R(v,u)) \vee P(x)) \end{array} \right)$$

The exist ... ace it by $f(x)$.

The existentially quantified v is in the scope ... $\forall u$. So we replace it by $g(u,x)$.

$$\forall x \left(\begin{array}{c} (\neg P(x) \vee (R(x,f(x)) \wedge \forall z\ R(z,f(x)))) \wedge \\ (\forall u\ (\neg R(x,u) \vee \neg R(g(u,x),u)) \vee P(x)) \end{array} \right)$$

Clausal Form: Step 5—Drop Universal Quantifiers

Eliminating universal quantifiers is easy:

$$(\neg P(x) \vee (R(x, f(x)) \wedge \forall z\ R(z, f(x)))) \wedge \quad) \Big)$$

become

$$(\neg P(x) \vee (R(x, f(x)) \wedge R(z, f(x)))) \wedge (\neg \qquad \vee P(x))$$

It is understood that all variables are now universall . If you prefer, you can think of all the universal quantifiers as sitting in front of the formula.

$$(\neg P(x) \vee (R(x, f(x)) \wedge R(z, f(x)))) \wedge (\neg R(x, u) \vee \neg R(g(u, x), u) \vee P(x))$$

becomes, using distribution:

$$(\neg R(x, u) \vee \neg R(g(u, x$$

or, written as a set of sets of literals:

$$\left\{ \begin{array}{c} \{\neg P(x), R(x, f(x))\}, \\ \{(\neg P(x), R(z, f(x))\}, \\ \{(\neg R(x, u), \neg R(g(u, x), u), P(x))\} \end{array} \right\}$$

Justifying Skolemization

Note that Skolemization of a formula does not produce a logically equivalent formula.

For exam

If we inter f as the

"successor" function $(+1)$, and P as $>$, then the original formula is satisfied, but the second is not.

However, Skolemization does produce an a—one that is satisfiable iff the original was—and this is all we care about for the purposes of resolution proofs.

We wish to develop the resolution principle for predicate logic with function symbols.

Howeve … on atomic fo … ion symbols

Simple cases seem easy enough, for example, from

$$\neg B(x) \vee M(x) \quad \text{and}$$

we would like to conclude $\neg B(c)$. ("Every borogove is mimsy" and "Colin is not mimsy" entails "Colin is not a borogove.")

Resolution for Predicate Logic

Note that all variables in

$\neg B(x) \vee M(x)$ and $\neg M(c)$

are unive

In particu) $\vee$ $M(c)$,
then we will be resolving the two clauses on
just as happened in the propositional case.

The resolvent then comes out as $\neg B(c$

Next we will develop this idea and define resolution deduction for arbitrary sets of clauses.

Next Week

How reso

Plus: Ind