Computational Logic

Standardization of Formulæ

Damiano Zanardini

UPM European Master in Computational Logic (EMCL)
School of Computer Science
Technical University of Madrid
damiano@fi.upm.es

Academic Year 2009/2010

Satisfiability as deduction

Our main observation

- by the Validity, Completeness and Deduction Theorems:

$$T[F_1, .., F_n] \models G \quad \text{iff} \quad T[F_1, .., F_n] \vdash G \quad \text{iff} \quad UNSAT(F_1 \wedge .. \wedge F_n \wedge \neg G)$$

- ☞ this means that we can reduce a problem of deduction (whether a formula can be derived by a set of premises) to a problem of satisfiability
- ☞ also, satisfiability can tell us if a set of sentences is contradictory
- the techniques we are going to see deal with satisfiability

Satisfiability as deduction

An example: how to obtain documents in Spain...

- in order to get the Health Care number (*número de Seguridad Social*), I was told I needed a regular job contract

Satisfiability as deduction

An example: how to obtain documents in Spain...

- in order to get the Health Care number (*número de Seguridad Social*), I was told I needed a regular job contract
- when I was about to sign the contract, they told me I needed the Health Care number

Satisfiability as deduction

An example: how to obtain documents in Spain...

- in order to get the Health Care number (*número de Seguridad Social*), I was told I needed a regular job contract
- when I was about to sign the contract, they told me I needed the Health Care number

¿  ?

Satisfiability as deduction

An example: how to obtain documents in Spain...

- in order to get the Health Care number (*número de Seguridad Social*), I was told I needed a regular job contract
- when I was about to sign the contract, they told me I needed the Health Care number

- how can we write it down formally?
  - constants: *hcn*, *jc*, 0 (initial state)
  - functions: $f/1$ (next state)
  - predicates:
    - *wants*/1 (the document I want)
    - *has*/2 (the document I have in a certain state)
    - *gets*/2 (the document I get in a certain state)

Skolem normal form

Our goal: simplifying formulæ

We want to obtain, by means of a series of *transformations*, a formula which is easier to deal with automatically, yet retains certain properties of the original one

- this is called *standardization*, and produces first the *Skolem Normal Form*, then the *Clause Form*

Running example

$$\forall y \ (\ \ (\exists x p(x, f(y)) \rightarrow (\ q(y) \wedge q(z) \ ) \ ) \ \ \vee \ \ \neg \forall w r(g(w), y) \ \ )$$

Skolem normal form

How to get the Skolem Normal Form (SNF)

1. all quantifiers to the head of the formula (*prenex form*)
   - move quantifiers by means of *equivalence rules*
2. no free occurrences of variables
   - do the *existential closure*
3. the *matrix* of the formula is in *conjunctive normal form* (CNF): a conjunction of disjunctions of literals
   - transform the formula by means of equivalence rules
4. only universal quantifiers
   - remove existential quantifiers by introducing *Skolem functions*

Skolem normal form

What does this transformation preserve?

- it preserves the satisfiability
- but *not* all the models: the result is *not semantically equivalent* to the original

Preservation

Take a transformation from F to F'

- to preserve the semantics means that, for every interpretation $\mathcal{I}$, $\mathcal{I}$ is a model of F iff it is a model of F'
  - $\forall x p(x)$ is semantically equivalent to $\neg\exists x \neg p(x)$
- to preserve the satisfiability means that there exists a model $\mathcal{I}$ for F iff there exists a model $\mathcal{I}'$ (probably not the same) for F'
  - $SAT(\exists x p(x))$ iff $SAT(p(a))$

Skolem normal form

1 Prenex form: all quantifiers at the beginning

Getting a prenex form relies on the following rules for *moving* quantifiers towards the head:

- renaming of bounded occurrences (if y does not occur free in F)
  $\vdash \forall x F(x) \leftrightarrow \forall y F(x/y) \qquad \vdash \exists x F(x) \leftrightarrow \exists y F(x/y)$
- interdefintion of quantifiers
  $\vdash \neg\forall x F(x) \leftrightarrow \exists x \neg F(x) \qquad \vdash \neg\exists x F(x) \leftrightarrow \forall x \neg F(x)$
- connectives vs. quantifiers (if x does not appear free in the other formula)
  $\vdash \forall x F \wedge G \leftrightarrow \forall x (F \wedge G) \qquad \vdash (\forall x F \rightarrow G) \leftrightarrow \exists x (F \rightarrow G)$
  $\vdash \exists x F \wedge G \leftrightarrow \exists x (F \wedge G) \qquad \vdash (\exists x F \rightarrow G) \leftrightarrow \forall x (F \rightarrow G)$
  $\vdash \forall x F \vee G \leftrightarrow \forall x (F \vee G) \qquad \vdash (F \rightarrow \forall x G) \leftrightarrow \forall x (F \rightarrow G)$
  $\vdash \exists x F \vee G \leftrightarrow \exists x (F \vee G) \qquad \vdash (F \rightarrow \exists x G) \leftrightarrow \exists x (F \rightarrow G)$
- connectives vs. quantifiers (more)
  $\vdash (\forall x F \wedge \forall x G) \leftrightarrow \forall x (F \wedge G) \qquad \vdash (\exists x F \vee \exists x G) \leftrightarrow \exists x (F \vee G)$
  ☞ there are only these two, not the dual ones

Skolem normal form

Lemma

The prenex form of a formula always exists, although it could be non-unique

Proof.

- how could we prove it?

Lemma

Every formula F is equivalent to its prenex form(s):

$$\vdash F \leftrightarrow Prenex(F)$$

Proof.

- easy because all steps leading to $Prenex(F)$ are equivalencies

Skolem normal form

2 Existential closure: no more free variable occurrences

Variables which occur free in the formula are existentially quantified

$$\begin{array}{lcl} \forall y(x \land q(y)) & \rightsquigarrow & \exists x(\forall y(x \land q(y))) \\ \forall y \exists x(p(x) \land q(y) \rightarrow r(f(z), x)) & \rightsquigarrow & \exists z \forall y \exists x(p(x) \land q(y) \rightarrow r(f(z), x)) \end{array}$$

Lemma

- *the closure does not affect satisfiability: $F(x)$ is satisfiable iff $\exists x F(x)$ is*
- *by extension, $SAT(F)$ iff $SAT(\exists - closure(F))$*

Skolem normal form

3. Conjunctive normal form (CNF): the *matrix* becomes a conjunction of disjunctions of literals

- connectives
  $\vdash (F \rightarrow G) \leftrightarrow (\neg F \vee G)$
  $\vdash (F \leftrightarrow G) \leftrightarrow (F \rightarrow G) \wedge (G \rightarrow F)$
- De Morgan
  $\vdash \neg(F \wedge G) \leftrightarrow \neg F \vee \neg G$ $\quad$ $\vdash \neg(F \vee G) \leftrightarrow \neg F \wedge \neg G$
- distributivity of $\wedge$ and $\vee$
  $\vdash F \wedge (G \vee H) \leftrightarrow (F \wedge G) \vee (F \wedge H)$
  $\vdash F \vee (G \wedge H) \leftrightarrow (F \vee G) \wedge (F \vee H)$

Skolem normal form

Lemma

The conjunctive normal form of a (quantifier-free) formula always exists

Proof.

- (exercise)

Lemma

For every formula F, $\vdash F \leftrightarrow CNF(F)$

Proof.

- easy because all steps leading to $CNF(F)$ are equivalencies

Skolem normal form

4. $\exists$-elimination: no more existential quantifiers

An existential quantifier can be removed by replacing the variable it bounds by a *Skolem function* of the form $f(x_1, ..x_n)$, where:

- f is a fresh function symbol
- $x_1, .., x_n$ are the variables which are universally quantified *before* the quantifier to be removed

$$
\begin{array}{lcl}
\forall x \exists y (p(x) \rightarrow \neg q(y)) & \rightsquigarrow & \forall x (p(x) \rightarrow \neg q(f(x))) \\
\exists x \forall z (q(x,z) \vee r(a,x)) & \rightsquigarrow & \forall z (q(b,z) \vee r(a,b)) \\
\exists x \forall y \exists z (p(x) \wedge q(y) \rightarrow r(f(z),x)) & \rightsquigarrow & \forall y (p(a) \wedge q(y) \rightarrow r(f(g(y)),a))
\end{array}
$$

Lemma

A formula F is satisfiable iff Skolem(F) is

Skolem normal form

Definition

$$\begin{aligned} Q.M &= \exists - closure(Prenex(F)) \\ SNF(F) &= Skolem(Q.CNF(M)) \end{aligned} \qquad Q.M = \text{[quantifiers].[matrix]}$$

Theorem

F is satisfiable iff $SNF(F)$ is

Proof.

1. F is satisfiable iff $Prenex(F)$ is
2. $Prenex(F)$ is satisfiable iff $\exists - closure(Prenex(F))$ is
3. M is satisfiable iff $CNF(M)$ is
4. $Q.M$ is satisfiable iff $Q.CNF(M)$ is (from ❸)
5. $Q.CNF(M)$ is satisfiable iff $Skolem(Q.CNF(M))$ is

Skolem normal form

Conclusion

- we are basically interested in satisfiability
- $SNF(F)$ exists for every F
- $SNF(F)$ preserves satisfiability

☞ therefore, we can restrict ourselves to only formulæ in Skolem normal form

- the Skolem normal form is named after the Norwegian mathematician Thoralf Albert Skolem (1887 - 1963)
- it was introduced in this context by Martin Davis and Hilary Putnam in 1960

Advantages

- no internal quantifiers
- only universal quantifiers, only in the head
- no free variable occurrences
- only $\wedge$ and $\vee$, suitably arranged

Clause form

It is easier to work on the *Clause Form* $CF(F)$

- *clause*: disjunction of literals
- the clause form of F is the set of clauses of $SNF(F)$, where the set means conjunction, and all variables are *universally quantified*

$$\begin{aligned} F &= \forall x(p(x) \wedge \forall y(\neg q(y) \rightarrow r(z,x))) \\ SNF(F) &= \forall x \forall y(p(x) \wedge (q(y) \vee r(a,x))) \\ CF(F) &= \{p(x), q(y) \vee r(a,x)\} \end{aligned}$$

Theorem

F is satisfiable iff $CF(F)$ is

Clause form

Clause form of a deduction

A deduction $[F_1, .., F_n] \vdash G$ is correct iff $F_1 \wedge .. \wedge F_n \wedge \neg G$ is not satisfiable

- get the clause form of every F_i
- get the clause form of $\neg G$

☞ important: we cannot use the same Skolem functions in different formulæ of the deduction (always *new* names)!

☞ more important: $CF(\neg G)$, not $\neg(CF(G))$!!! (ex. $\exists x p(x)$)

- compute the union of all sets of clauses
- check the satisfiability

☞ this is what we should do when asked to verify that a deduction of a formula from some premises is correct

☞ we will see several methods for proving the unsatisfiability of a clause set

Clause form

Example: $[\exists x f(x), \exists x g(x)] \vdash \exists x(f(x) \wedge g(x))$

$$\begin{array}{lcl} CF(\exists x f(x)) & = & \{f(a)\} \\ CF(\exists x g(x)) & = & \{g(b)\} \\ CF(\neg(\exists x(f(x) \wedge g(x)))) & = & \{\neg f(x) \vee \neg g(x)\} \end{array}$$

Here, there exists an interpretation which is a model:

- $D = \{0, 1\}$
- $I(a) = 0$
- $I(b) = 1$
- $I(f(a)) = \mathcal{F}(I(a)) = \mathcal{F}(0) = \mathbf{t}$
- $I(g(b)) = \mathcal{G}(I(b)) = \mathcal{G}(1) = \mathbf{t}$
- $I(f(b)) = \mathcal{F}(I(b)) = \mathcal{F}(1) = \mathbf{f}$
- $I(g(a)) = \mathcal{G}(I(a)) = \mathcal{G}(0) = \mathbf{f}$

therefore, the deduction is not correct