Resolution-Based Theorem Proving for Modal and Description Logic

Renate Schmidt
The University of Manchester, UK

Overview

- The logics and translation to FOL
- First-order resolution
- Resolution decision procedures
- Other applications

Introduction

- Aim:
  - To study modal and description logics as fragments of first-order logic
  - To use techniques from first-order resolution for deciding modal and description logics
  - To mention some other applications

- Remarks:
  - Content not as detailed as in ordinary lectures
  - Feel free to ask questions!

Part I

The logics and translation to FOL
Basic modal logic

- Basic modal logic $K_{(m)} = \text{propositional logic plus } \langle r_1, r_2, \ldots \rangle$
  $Ac = \{r_1, r_2, \ldots \}$ (index set)
- Modal formulae: $\phi, \psi \rightarrow p_i \mid \neg \phi \mid \phi \lor \psi \mid \langle \alpha \rangle \phi$
  Actions: $\alpha, \beta \rightarrow r_j$
- $[\alpha] \phi \overset{\text{def}}{=} \neg \langle \alpha \rangle \neg \phi$
- Semantics: Kripke model $\mathcal{M} = (W, \{R_j \mid r_j \in Ac\}, \nu)$
  $\mathcal{M}, x \models p_i$ iff $x \in \nu(p_i)$
  $\mathcal{M}, x \models \neg \phi$ iff $\mathcal{M}, x \not\models \phi$
  $\mathcal{M}, x \models \phi \lor \psi$ iff $\mathcal{M}, x \models \phi$ or $\mathcal{M}, x \models \psi$
  $\mathcal{M}, x \models \langle r_j \rangle \phi$ iff for some $R_{r_j}$-successor $y$ of $x$ $\mathcal{M}, y \models \phi$
  $\mathcal{M}, x \models [r_j] \phi$ iff for all $R_{r_j}$-successors $y$ of $x$ $\mathcal{M}, y \models \phi$

Extensions of the basic modal logic

- Traditional MLs: extension of $K_{(m)}$ with extra modal axioms
  - epistemic ML, doxastic ML, …
- Dynamic MLs: extensions of $K_{(m)}$ with operators on actions
  - dynamic logic PDL = $K_{(m)}(\lor, \cdot, *, ?)$
  - description logics with role operators

<table>
<thead>
<tr>
<th>Reading of $[r_j] \phi$</th>
<th>Notation</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$ is necessary</td>
<td>$\square \phi$</td>
<td>basic modal logic $K$</td>
</tr>
<tr>
<td>agent $j$ knows $\phi$</td>
<td>$K_j \phi$</td>
<td>epistemic logic $KT45_{(m)}$</td>
</tr>
<tr>
<td>agent $j$ believes $\phi$</td>
<td>$B_j \phi$</td>
<td>doxastic logic $KD45_{(m)}$</td>
</tr>
<tr>
<td>action $r_j$ causes $\phi$</td>
<td>$[r_j] \phi$</td>
<td>dynamic logic PDL</td>
</tr>
<tr>
<td>$R_{r_j}$-relatives of only $C_{\phi}$s</td>
<td>$\forall R_{r_j}, C_{\phi}$</td>
<td>description logics, $ALC$ family</td>
</tr>
</tbody>
</table>

Dynamic modal logics

- $K_{(m)}$ plus action-forming operators
- Actions: $\alpha, \beta \rightarrow r_j \mid \neg \alpha \mid \alpha \lor \beta \mid \alpha \cdot \beta \mid \phi^c \mid id$
- Semantics:
  - $R_{\neg \alpha} \overset{\text{def}}{=} (W \times W \backslash R_{\alpha})$
  - $R_{\alpha \lor \beta} \overset{\text{def}}{=} R_{\alpha} \cup R_{\beta}$
  - $R_{\alpha \cdot \beta} \overset{\text{def}}{=} \{(x, y) \mid (y, x) \in R_{\alpha}\}$
  - $R_{\phi^c} \overset{\text{def}}{=} \{(x, y) \mid x \in R_{\phi}\}$
  - $R_{id} \overset{\text{def}}{=} Id_W$
- … defines Peirce logic
- Very expressive; undecidable; has many decidable sublogics
- $K_{(m)}(*_1, \ldots, *_n) = K_{(m)}$ extended with $*_1, \ldots, *_n$

Lattice of dynamic modal logics

decidable DMLs/DLs without relational $\neg$
decidable DMLs/DLs with relational $\neg$
this talk

$BML = K_{(m)}(\lor, \cdot, *)$
Why are extensions with ¬ and ⌜interesting?

- $K(m)(⌜)$ = tense logic; $⌜[α⌜]$ is past operator
- $K(m)(¬)$ = the logic of ‘some’, ‘all’ and ‘only’

| possibility op | $⌜(α)ϕ$ | ‘ϕ is true at some $α$-successor’ |
| necessity op   | $¬(⌜α)¬ϕ$ | ‘ϕ is true at all $α$-successors’ |
| sufficiency op | $¬⌜(¬α)ϕ$ | ‘ϕ is true at only $α$-successors’ |

- Also definable (by non-logical axioms) are:
  - left cyl., right cyl., cross product, domain/range restriction

- Standard tableau methods decide $K(m)(⌜)$
- $K(m)(¬)$ and $K(m)(¬,⌜)$ do not have the tree model property
- Using unrestricted blocking $K(m)(¬)$ and $K(m)(¬,⌜)$ can be decided with tableau

First-order logic: Background

- Most important ‘unifying’ formalism for knowledge representation and reasoning in CS and AI
- Very Expressive: most domain knowledge can be represented with ease
- Sound and complete deduction calculi exist
- Many reasoning tools available
- Reasoning is undecidable (not a problem!)
- Many decidable fragments
- Many practical decision procedures for decidable fragments

Syntax of first-order logic

- Terms:
  - $s, t, u → x$ (first-order variable)
  - $a$ (constant)
  - $f(s_1, \ldots, s_n), n > 0$ (functional term)

- Atoms:
  - $A, B → P(s_1, \ldots, s_n), n ≥ 0$ (non-equational atom)
  - $s ≈ t$ (equational atom)

- First-order formulae:
  - $F, G → ⊥ | ⊤ | ϕ$ (atomic formula)
  - $¬F | (F ⋆ G) ⋆ ∈ \{∧, ∨, →, ↔\}$ (quantified formulae)

Standard translation of MLs into FOL

- Translation mapping: $L \xrightarrow{Π} FOL$
  - sound & complete, efficiently computable (linear/polynomial)
  - Standard translation based on semantics of source logic $L$
- Question: $Γ \models φ$ ?
  - Where $Γ \equiv \{p → q\}$
  - $ϕ \equiv [r](r)p$
- Equivalent to: $Π(Γ) \models_{FOL} Π(ϕ)$?
  - $Π(Γ) = ∀x [Q_p(x) → Q_q(x)]$
  - $Π(ϕ) = ∀x ∀y (R(x, y) → ∃z (R(y, z) ∧ Q_p(z)))$
- Now give to any FOL prover
Using translation to FOL

- Let \( L \) be given DML/DL
  \[ F_L = \Pi(L) \] corresponding FO fragment
- \( \Pi \) sound & complete \( \Rightarrow \) any FOL prover can be used
- \( \Pi \) efficiently computable \( \Rightarrow \) if \( L \) decidable then \( F_L \) decidable
- FO methods are not automatically decision procedures for \( F_L \)
  - Identify decidable FO fragment \( G \) encompassing \( F_L \) and use decision procedure of \( G \)
- \( F_L \) not necessarily subfragment of known decidable FO fragm.
  - Develop FO decision procedure for \( F_L \)
- Decision procedure of \( G \) might not be suitable for purpose
  - Develop suitable refinement for purpose of \( F_L \)

Resolution

- Refutation approach, testing (un)satisfiability
- Operates on clauses
- Two rules: resolution and factoring
- No branching rules required \( \Rightarrow \) derivations are linear

\[
\begin{align*}
\text{Resolution:} & \quad \frac{C \lor A \quad \neg A \lor D}{C \lor D} \\
\text{Factoring:} & \quad \frac{C \lor A \quad A}{C \lor A}
\end{align*}
\]

Theorem 1

\( \text{Res} \) is sound and (refutationally) complete for propositional and ground clause logic

Clause logic

- Language of resolution is that of clause logic
- Literals:
  \[ L \quad A \quad \text{ (positive literal, atom)} \]
  \[ | \quad \neg A \quad \text{ (negative literal)} \]
- Clauses:
  \[ C, D \quad \perp \quad \text{ (empty clause)} \]
  \[ | \quad L_1 \lor \ldots \lor L_k, \quad k \geq 1 \quad \text{ (non-empty clause)} \]
- Free variables interpreted as implicitly universally quantified
- Clauses regarded as multi-sets of literals
  - \( P(a) \lor P(a) \lor Q(x) \) is not the same as \( P(a) \lor Q(x) \)
Transformation to clausal form

- Basic algorithm (too naive):
  1. Transform into prenex normal form (PNF): move quantifiers to the front
     \[ Q_1 \wedge \ldots \wedge Q_n G \quad (G \text{ quantifier-free}) \]
  2. Skolemisation: eliminate quantifiers
     \[ \leadsto \quad \text{quantifier-free formula} \]
  3. Transform into conjunctive normal form (CNF)
     \[ \leadsto \quad C_1 \wedge \ldots \wedge C_n \]
  4. Classify
     \[ \leadsto \quad \text{set of clauses } N = \{ C_1, \ldots, C_n \} \]
- For any \( F \): \( F \) is satisfiable iff \( \text{Cls}(F) \) is satisfiable
- Various standard optimisations exist (see later)

Running example: Transformation to clausal form

- Take \( \phi \overset{\text{def}}{=} [r](\neg p \vee \langle r \rangle p) \); \( \phi \) is satisfiable in \( K(m) \)
- FO translation:
  \[ \exists x[\forall y (R(x, y) \rightarrow (\neg Q_p(y) \vee \exists z (R(y, z) \wedge Q_p(z))))] \]
- Prenex normal form:
  \[ \exists x \forall y \exists z [\neg R(x, y) \vee \neg Q_p(y) \vee (R(y, z) \wedge Q_p(z))] \]
- Skolemisation:
  \[ \neg R(a, y) \vee \neg Q_p(y) \vee (R(y, f(y)) \wedge Q_p(f(y))) \]
  \[ \neg \text{Sk. const. for } \exists x \quad \neg \text{Sk. term for } \exists z \]
- CNF:
  \[ (\neg R(a, y) \vee \neg Q_p(y) \vee R(y, f(y))) \wedge \]
  \[ (\neg R(a, y) \vee \neg Q_p(y) \vee Q_p(f(y))) \]
- Clausal form: drop \( \wedge \) and outer (, )

Basic resolution calculus \( \text{Res} \) for FO clause logic

- \( \text{Res} \) for ground clause logic plus unification
  \[ C \vee A \quad \neg B \vee D \quad (C \vee D)_{\sigma} \quad \text{if } \sigma = \text{mgu}(A \equiv B) \]
- Resolution:
  \[ \frac{C \vee A \vee B}{(C \vee A)_{\sigma}} \quad \text{if } \sigma = \text{mgu}(A \equiv B) \]
- Factoring:
  \[ \frac{C \vee A \vee B}{(C \vee A)_{\sigma}} \quad \text{if } \sigma = \text{mgu}(A \equiv B) \]
- Example:
  \[ Q(y) \vee P(f(y)) \quad \neg P(z) \vee R(z, a) \quad \sigma = \{ z/f(y) \} \]

Theorem 2
\( \text{Res} \) is sound and (refutationally) complete for FO clause logic

- Problem: Extremely prolific at generating new clauses

Running example (cont’d): Applying basic resolution

- Clausal form:
  \[ 1. \quad \neg R(a, y) \vee \neg Q_p(y) \vee R(y, f(y)) \quad \text{given} \]
  \[ 2. \quad \neg R(a, y) \vee \neg Q_p(y) \vee Q_p(f(y)) \quad \text{given} \]
- Resolvents under \( \text{Res} \):
  \[ 3. \quad \neg R(a, a) \vee \neg Q_p(a) \vee \neg Q_p(f(a)) \vee \neg Q_p(f^2(a)) \quad (1.3, 2.1) \]
  \[ 4. \quad \neg R(a, f(y)) \vee R(f(y), f^2(y)) \vee \neg R(a, y) \vee \neg Q_p(y) \quad (1.2, 2.3) \]
  \[ 5. \quad \neg R(a, f^2(y)) \vee R(f^2(y), f^3(y)) \vee \neg R(a, f(y)) \quad (2.3, 4.4) \]
  \[ \vee \neg R(a, y) \vee \neg Q_p(y) \quad \text{etc} \]
- Problem: Termination for satisfiable formulae
  - Clauses expand in width and depth
Modern resolution framework

... = resolution calculus Res + restrictions + control

- Guiding principle: Avoid unnecessary inferences whenever possible
- Local restrictions: control inferences performed via
  - Admissible ordering \( \succ \)
  - Selection function \( S \)
- Global restrictions of search space via
  - General notion of redundancy
- Important for implementation: strategies & heuristics, fairness

Ordered resolution calculus with selection \( Res^\succ \)

- Assume: \( \succ \) admissible atom ordering; \( S \) selection function
- Ordered resolution with selection rule:
  \[
  \frac{C \lor A}{(C \lor D)\sigma}
  \]
  provided \( \sigma = \text{mgu}(A \equiv B) \) and
  (i) \( A\sigma \) strictly maximal wrt. \( C\sigma \);
  (ii) nothing selected in \( C \) by \( S \);
  (iii) either \( \neg B \) selected,
    or else nothing selected in \( \neg B \lor D \)
    and \( \neg B\sigma \) maximal wrt. \( D\sigma \)

- Note: variables of premises must be renamed apart

Ordered resolution calculus with selection \( Res^SV \) (cont’d)

- Ordered factoring rule:
  \[
  \frac{C \lor A \lor B}{(C \lor A)\sigma}
  \]
  provided \( \sigma = \text{mgu}(A \equiv B) \) and
  (i) \( A\sigma \) is maximal wrt. \( C\sigma \);
  (ii) nothing is selected in \( C \)

Theorem 3
\( Res^SV \) is sound and (refutationally) complete for FO clause logic

Local search control parameters

- Admissible ordering \( \succ \)
  - total, well-founded on ground terms and atoms
  - on ground literals: \( \ldots \succ \neg A \succ A \succ \neg B \succ B \succ \ldots \)
  - stable under substitutions
- Selection function \( S \): selects only negative literals
  - \( S(C) = \) possibly empty multi-set of negative literal occurrences in \( C \)
  - Example of selection with selected literals indicated as \( L \):
    \[
    \neg A \lor \neg A \lor B \lor \neg B_0 \lor \neg B_1 \lor A
    \]
- Idea:
  - Inferences restricted to \( \succ\)-maximal or \( S\)-selected literals
  - \( S \) overrides \( \succ \)

M4M School, Copenhagen, Nov. 2009 – p.21
Running example (cont’d): Using ordered resolution

- Recall using Res clauses expand in width and depth
- Use ordering and/or selection function to prevent this

1. \( \neg R(a, y) \lor \neg Q_p(y) \lor R(y, f(y)) \) given
2. \( \neg R(a, y) \lor \neg Q_p(y) \lor Q_p(f(y)) \) given

- Let \( \succ \) extension of subterm ordering + no selection f. \( (S = \emptyset) \)
- \( f(t) \succ t \); precedence on pred. symbols: \( R \succ Q_p \)
- first criterion: \( \succ \) on maximal arguments
- No inference steps possible in Res\( ^{\succ} \) !

1. \( \neg R(a, y) \lor \neg Q_p(y) \lor R(y, f(y)) \) given
2. \( \neg R(a, y) \lor \neg Q_p(y) \lor Q_p(f(y)) \) given

Search spaces become smaller

- Assume \( P \succ Q \succ R \succ T \) and nothing is selected, i.e. \( S = \emptyset \)

1. \( \neg T \lor P \lor Q \) given
2. \( \neg P \lor \neg R \) given
3. \( \neg Q \) given
4. \( \neg T \lor Q \lor \neg R \) Res 1, 2
5. \( \neg T \lor \neg R \) Res 3, 4

- Derivation is completely deterministic
- Generally, proof search still non-deterministic but search space is much smaller than with unrestricted resolution
- Exercise: Choose selection function so that no inferences are possible

Decidability of \( K(m) \) by ordered resolution

- How to show that Res\( ^{\succ} \) decides \( K(m) \)?
  - Characterise a class of clauses closed under Res\( ^{\succ} \) into which any \( K(m) \)-problem can be mapped
  - Show the class is bounded when defined over a bounded signature of predicate and function symbols
- Required: structural transformation . . .
**Structural transformation of first-order formulae**

**Theorem 4**

Let $Q$ be a fresh predicate symbol. Then

$$ F[G(\bar{x})] \text{ satisf. iff } F[Q(\bar{x})] \land \forall \bar{x} (Q(\bar{x}) \leftrightarrow G(\bar{x})) \text{ satisf.} $$

- **Structural transformation rewrite rule:**
  $$ F[G(\bar{x})] \Rightarrow F[Q(\bar{x})] \land \forall \bar{x} (Q(\bar{x}) \leftrightarrow G(\bar{x})) $$
  - Introduces new pred. symbol $Q$ for subformula $G(\bar{x})$ of $F$
  - View $Q(\bar{x})$ as an abbreviation for $G(\bar{x})$.

- Small overhead; efficient transformation to CNF
- Our case: Introduce new $Q_\phi \forall$ non-negated complex $\phi$
  Take polarity of subformulae into account

**Structural transformation for running example**

- **FO translation of $\phi = [r](\neg p \lor (r)p):$$
  $$ \exists x \left[ \forall y (\neg R(x, y) \lor (\neg Q_p(y) \lor \exists z (R(y, z) \land Q_p(z))) \right] $$
  - $Q_\phi(y)$
  - $Q_{\neg p\lor(r)p}(y)$
  - $Q_{\forall r, (\neg p\lor(r)p)}(x)$

- **Clausal form Cls $\exists \neg \Pi(\phi)$:**
  $$ \neg Q_{(r)p}(x) \lor R(x, f(x)) $$
  $$ \neg Q_{(r)p}(x) \lor Q_p(f(x)) $$
  $$ \neg Q_{\neg p\lor(r)p}(x) \lor \neg Q_p(x) \lor Q_{(r)p}(x) $$
  $$ \neg Q_{(r)p}(\neg p\lor(r)p)(x) \lor \neg R(x, y) \lor Q_{\neg p\lor(r)p}(y) $$
  $$ Q_{(r)p}(\neg p\lor(r)p)(a) $$

**General form of input clauses**

- **Form of input clauses for $K_{(m)}$:**
  $$ (-)Q_\phi(a) $$
  $$ R(a, b) $$
  $$ (-)Q_\phi(x) \lor (-)Q_1(x) \lor \ldots \lor (-)Q_n(x) $$
  $$ \geq 1 \text{ max. lits} $$
  $$ (-)Q_\phi(x) \lor \neg R(x, y) \lor (-)Q(y) $$
  $$ (-)Q_\phi(x) \lor R(x, f_\phi(x)) $$
  $$ (-)Q_\phi(x) \lor (-)Q(f_\phi(x)) $$

- **Ordering: binary literals $\succ$ unary literals**
  - depth 2 literals $\succ$ depth 1 literals
- **Step 1:** In each clause what are the maximal literals?
- **Step 2:** What do the resolvents & factors look like?
Clausal class $MC$

- General form of derived clauses
  - ground clauses with only unary literals
    - $(\neg)Q(x) \lor (\neg)Q_1(x) \lor \ldots \lor (\neg)Q_n(x)$ ($0 \leq n$)
    - $(\neg)Q(x) \lor (\neg)Q_1(x) \lor \ldots \lor (\neg)Q_n(x)$
      \[\lor (\neg)Q_1(f(x)) \lor \ldots \lor (\neg)Q_m(f(x))\] ($0 \leq n, m$)
- Let $MC$ = class of these clauses:
  - ground unary clauses
    - $R(a, b)$
  - non-ground unary clauses with arguments $x$ or $f(x)$
    - $(\neg)Q(x) \lor \neg R(x, y) \lor (\neg)Q(y)$
    - $(\neg)Q(x) \lor R(x, f(x))$

Decidability of $K(m)$ by ordered resolution

Lemma 5
For any finite clause set $N$ in $MC$:
1. Any derived clause belongs to $MC$
2. Any $Res^\succ$-derivation from $N$ terminates in EXPTIME

Theorem 6
Assume $\phi$ any formula and any set $\Gamma$ in $K(m)$;
let $N = \text{Cls} \Xi(\Pi(\Gamma) \land \neg\Pi(\phi))$
1. Any $Res^\succ$-derivation from $N$ terminates in EXPTIME
2. $\Gamma \models \phi$ iff $Res^\succ$ derives $\bot$ from $N$
- Complexity is optimal for $\Gamma \neq \emptyset$

Generalisation

- Clausal class $MC$ :
  - ground unary clauses
  - $R(a, b)$
  - non-ground unary clauses with arguments $x$ or $f(x)$
    - $(\neg)Q(x) \lor \neg R(x, y) \lor (\neg)Q(y)$
    - $(\neg)Q(x) \lor R(x, f(x))$
  - What if binary literals are negated ?
  - Lemma true for the extended class
  - Thus, the theorem is true for $K(m)(\neg)$ !
  - What if arguments in binary literals can be swapped ?
Generalisation

- Clause class $MC^*$:
  - ground unary clauses
  - $(\neg)R^a(a, b)$
  - non-ground unary clauses with arguments $x$ or $f(x)$
  - $(\neg)Q\phi(x) \lor (\neg)R^a(x, y) \lor (\neg)Q(y)$
  - $(\neg)Q\phi(x) \lor (\neg)R^a(x, f\phi(x))$
- Lemma true for the extended class
- Thus, the theorem is true for $K(m)(\neg)$!
- And for $K(m)(\neg, \cdash)$!

Ordered resolution decides $K(m)(\neg, \cdash)$

Theorem 7
$Res^\succ$ is decision procedure for any logic between $K(m)$ and $K(m)(\neg, \cdash)$ and has (optimal) EXPTIME complexity

- Also true for any logic between $K(m)$ and $K(m)(\neg, \cdash)\cup \{\|, \vee, \cdot, \star, \times\}$
- Using the axiomatic translation translation many traditional MLs, incl. $KD45, S4, \ldots$, can be efficiently embedded into $MC^*$
- Gives complexity optimal decision procedures

Generalisation to $BML$ and beyond

- Ordered resolution decides wider clausal class: $DL^*$
  $$MC^* \subseteq DL^*, \quad MC^* \subseteq DL^*, \quad BML \subseteq DL^*, \quad BML(\neg, \cdash, pos) \subseteq DL^*$$
  $$FO^2 \subseteq DL^*, \quad FO^3 \cap DL^* \neq \emptyset$$
- $DL^*$ subsumes many DLs
- $DL^*$ is NEXPTIME-complete

Theorem 8
$Res^\succ$ + condensing, or splitting, decides $DL^*$, and hence all subsumed logics, incl. $BML$ and $BML(\neg, \cdash, pos)$

Generalisation to decidable fragments of FOL

- Numerous ways of defining decidable subclasses of FOL

<table>
<thead>
<tr>
<th>Restrict . . .</th>
<th>Decidable classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>arity of predicate symbols</td>
<td>monadic class</td>
</tr>
<tr>
<td>quantifier prefixes</td>
<td>$\exists^<em>, \forall^</em>, \exists\forall\exists^* $</td>
</tr>
<tr>
<td>number of variables</td>
<td>$FO^2$</td>
</tr>
<tr>
<td>ordering on variables</td>
<td>fluted logic</td>
</tr>
<tr>
<td>quantification by relativisation</td>
<td>guarded fragments</td>
</tr>
<tr>
<td>$\forall$ quantification</td>
<td>Maslov’s dual class $\overline{\mathcal{K}}, \overline{DK}$</td>
</tr>
</tbody>
</table>

- All decidable by resolution (with 1 exception based on extensions of $Res^\succ$)
Automated correspondence theory

- Given: traditional ML with extra axioms/rules, e.g. $K(m)\Delta$
- Problem: What are first-order frame correspondence properties for axioms/rules in $\Delta$?
- Second-order quantifier elimination methods solve the problem
  - E.g. SCAN (based on resolution)
  - $\forall p[\Box p \rightarrow \Box\Box p] \leadsto$ transitivity of $R$
- Main issue: successful termination
  - SCAN solves problem for all Sahlqvist formulae and inductive formulae
  - Automatic solution possible for even wider class

Part IV
Other applications and conclusion

Some other applications

- Simulating, generating, implementing and studying different deduction approaches (Thursday)
- Automatically generating models, incl. minimal models
- Second-order quantifier elimination
  - Reasoning with second-order formulae (e.g. modal axioms, rules)
  - Automatically computing correspondence properties

Concluding remarks

- Combination of translation and resolution has practical and theoretical advantages
- Translation is a core technique in computer science
- Resolution provides a powerful and versatile framework
  - for developing practical decision procedures
  - for other applications
- Well-developed implementation: SPASS 3.5
Resolution decision procedures


Part V

Selected references

Surveys