Using Redundancy and Basicness for Obtaining Decision Procedures for Fragments of FO-logic

Yevgeny Kazakov
Plan of the Talk

I. Fragments of FO-logic

II. Decision procedures

III. Redundancy and Basicness
I. Fragments of FO-logic
Many problems from different fields can be naturally represented in FO-logic:

- Knowledge representation (description logics)
- Planning
- Formal linguistics
- Relational databases
- …
Fragments of FO-logic

Example. The Basic Description Logic:

\[ \text{ALC} ::= A | C_1 \sqcap C_2 | C_1 \sqcup C_2 | \neg C | \forall R.C | \exists R.C. \]

- where \( C, C_1, C_2 \) - concepts (unary relations)
- build from:
  - \( A \) – basic concepts (initial sets) and
  - \( R \) – roles (binary relations).

• Reasoning problem:
  Concept Subsumption:

\[ \text{question mark} \quad C \sqsubseteq D \]
Fragments of FO-logic

Description logic as FO-fragment:

\[ \text{\textit{ALC}} ::= A \mid C_1 \land C_2 \mid C_1 \lor C_2 \mid \neg C \mid \forall R.C \mid \exists R.C. \]

\[ \begin{align*}
A(x) & \equiv \text{\textit{FO}}[\text{\textit{ALC}}] \\
C_1(x) \land C_2(x) & \\
C_1(x) \lor C_2(x) & \\
\neg C(x) & \\
\forall y. (R(x, y) \rightarrow C(y)) & \\
\exists y. (R(x, y) \land C(y)) & 
\end{align*} \]

Subsumption problem: \[ C \sqsubseteq D \]

Entailment problem: \[ C(x) \rightarrow D(x) \]
II. Decision procedures
Decision Procedures

\( \mathcal{ALC} \) – PSPACE-complete

\( \mathcal{FO} \) – UNDECIDABLE

- How to explain good computational properties of description logics?
  - “Good” model properties:
    - Finite model property
    - Tree model property

- Basis for Tableau-based decision procedures.
Extensions of description-like logics are harder to handle:

- **ALC** + $O$ – nominals (one-element sets)
  + $S \sqsubseteq R$ – role hierarchies
  + Transitive($R$) – transitive roles
  + $\exists<nR.C$, $\exists\geq nR.C$ – counting
- **UML** class diagramms
- **OWL** – ontology language for semantic web
Decision Procedures

- Extensions of description-like logics are harder to handle:
  - $\mathcal{ALC} + \text{counting} - \text{no finite model property}$;
  - $\mathcal{ALC} + \text{transitive roles} - \text{no tree model property}$;
  - $\mathcal{ALC} + \text{counting} + \text{transitive roles} + \text{unrestricted role hierarchies} - \text{undecidable}$.

- Decision procedures rely on heavy model-theoretic analysis:
  - “Good” model representation property
Alternative approach: use general theorem provers for FO-logic.

- Advantage:
  - No need to invent anything;
  - Soundness and completeness are guaranteed;
  - Easy to implement: just write a translator to FO-logic and use existing theorem provers.

- However:
  - Still need to prove termination.
  - Relatively slow in comparison to specialized decision procedures.
Using Redundancy and Basicness for Obtaining Decision Procedures for Fragments of FO-logic

Satisfaction-based Decision Procedures

Problem: \( \vdash F \) ?

Yes

\[ A(x) \lor \neg B(fx) \land C(y) ; \quad B(y) \lor \neg D(z) \land E(x, z) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad B(y) \lor \neg D(z) \lor E(x, z) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad B(y) \lor \neg D(z) \lor E(x, z) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad \ldots \]

\[ A(x) \lor \neg B(fx) \land C(y) ; \quad B(y) \lor \neg D(z) \land E(x, z) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad B(y) \lor \neg D(z) \lor E(x, z) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad B(y) \lor \neg D(z) \lor E(x, z) ; \quad \neg B(c) \lor \neg H(f(x), x) \lor \neg A(x) ; \quad \ldots \]

Saturation

10-12 December, 2003
AG-2 Logic Seminar, Schloß Ringberg
Using Redundancy and Basicness for Obtaining Decision Procedures for Fragments of FO-logic

**Ordered Paramodulation Calculus**

**Ordered Resolution.**

\[
\frac{C \lor A \quad D \lor \neg B}{C \sigma \lor D \sigma}
\]

where (i) \( \sigma = \text{mgu}(A, B) \), (ii) \( A \) and \( \neg B \) are eligible.

**Ordered Paramodulation.**

\[
\frac{C \lor s \sim t \quad D \lor L[s']}{C \sigma \lor D \sigma \lor L[t] \sigma}
\]

where (i) \( \sigma = \text{mgu}(s, s') \), (ii) \( s \sim t \) and \( L[s'] \) are eligible, (iii) \( t \sigma \not\sim s \sigma \), and (iv) \( s' \) is not a variable.

**Equality Factoring.**

\[
\frac{C \lor u \not\sim v \lor u' \not\sim v'}{C \sigma \lor v' \sigma \not\sim v \sigma \lor u \sigma \sim v \sigma}
\]

where (i) \( \sigma = \text{mgu}(u, u') \), (ii) \( u \sim v \) is eligible.

**Ordered Factoring.**

\[
\frac{C \lor A \lor A'}{C \sigma \lor A \sigma}
\]

where (i) \( \sigma = \text{mgu}(A, A') \), (ii) \( A \) is eligible.

**Reflexivity Resolution.**

\[
\frac{C \lor u \not\sim v}{C \sigma}
\]

where (i) \( \sigma = \text{mgu}(u, v) \), (ii) \( u \not\sim v \) is eligible.
The Guarded Fragment

\[
\begin{align*}
\mathcal{GF} &::= A | \mathcal{FO}[ALC] \\
F_1 \land F_2 &| C_1(x) \land C_2(x) \\
F_1 \lor F_2 &| C_1(x) \lor C_2(x) \\
\neg F_1 &| \neg C(x) \\
\forall \overline{x}. (G \rightarrow F_1) &| \forall y. (R(x, y) \rightarrow C(y)) \\
\exists \overline{x}. (G \land F_1) &| \exists y. (R(x, y) \land C(y))
\end{align*}
\]

\[
\begin{align*}
G - \text{"guard"}: FV(F_1) &\subseteq FV(G) \\
\text{Guarded Formula:} &\exists x. (n(x) \land \forall y. [a(x, y) \rightarrow \\
\forall z. \{ x \sim z \rightarrow \exists x. [a(x, z) \land (\neg b(z, z) \lor \neg c(x, x))] \}])
\end{align*}
\]

non-Guarded Formulae:

\[
\begin{align*}
\text{Transitivity:} &\forall x, y, z. (xT y \land yT z \rightarrow xT z) \\
\text{Functionality:} &\forall x, y, z. (xF y \land xF z \rightarrow y \sim z)
\end{align*}
\]
Saturating the Guarded Fragment

\[ \mathcal{GF} ::= A | F_1 \land F_2 | F_1 \lor F_2 | \neg F_1 | \forall \overline{x}. (G \rightarrow F_1) | \exists \overline{x}. (G \land F_1). \]

**Guarded Clause Fragment:**
1. \( \alpha(\widehat{c}) \);
2. \( \neg \widehat{a}(\widehat{x}) \lor \alpha(\widehat{f}(\widehat{x}), \widehat{x}) \);
3. \( \alpha(x) \).

Saturation terminates for every \( \mathcal{GF} \)-formula.
Guarded Fragment With Transitivity

- Transitivity and functionality axioms are outside the Guarded Fragment.
- Does GF lose decidability when some predicates are allowed to be transitive, or functional?
  - YES [Grädel, 1999]: GF³ with one functional or transitive predicate is undecidable.
- How to explain decidability of modal and description logics with transitivity?
  - [Ganzinger et al., 1999]: GF²[T] is undecidable, but monadic-GF²[T] is decidable.
Guarded Fragment With Transitivity

- Is GF decidable when transitive predicates can appear in guards only? $\Rightarrow [\text{GF+TG}]$?
- What is the complexity of monadic-GF[T]?
- \[\text{[Szwast, Tendera, 2001]}: \ [\text{GF+TG}] \text{ is in DEXPTIME, monadic-GF[T] is NEXPTIME-hard.}\]
- \[\text{[Kierionski, 2002, 2003]}: \ [\text{GF+TG}^{-}] \text{ is EXPSPACE-hard, } [\text{GF+TG}] \text{ is DEXPTIME-hard.}\]
III. Redundancy and Basicness
Why Transitivity Is Hard?

Consider the resolution inferences with transitivity:

1. $\neg xTy \vee \neg yTz \vee xTz$;
2. $\neg \alpha(x) \vee f_T(x)Tx$;

OR\[1;2\]: 3. $\neg \alpha(x) \vee \neg xTz_1 \vee f_T(x)Tz_1$;
OR\[1;3\]: 4. $\neg \alpha(x) \vee \neg xTz_1 \vee \neg z_1Tz_2 \vee f_T(x)Tz_2$;
OR\[1;4\]: 5. $\neg \alpha(x) \vee \neg xTz_1 \vee \neg z_1Tz_2 \vee \neg z_2Tz_3 \vee f_T(x)Tz_3$;

The clause 4 can be obtained another way:

1. $\neg xTz_1 \vee \neg z_1Tz_2 \vee xTz_2$;
3. $\neg \alpha(x) \vee \neg xTz_1 \vee f_T(x)Tz_1$;
   $\Rightarrow$ 4. $\neg \alpha(x) \vee \neg xTz_1 \vee \neg z_1Tz_2 \vee f_T(x)Tz_2$;

With the smaller instance of transitivity clause!
Redundancy

Abstract notion of redundancy

[Bachmair, Ganzinger, 1990]:

- A ground clause $C$ is redundant w.r.t. a set of ground clauses $N$ if $N \prec C \vdash C$.
- An inference $C_1, C_2 \vdash C$ is redundant w.r.t. $N$ if $N \prec_{\text{max}} (C_1, C_2) \vdash C$.

How to show that inference is redundant?

Lemma [Four Clauses] The inference

$C_1 \lor C_2 \lor A, \; D_1 \lor D_2 \lor \neg A \vdash C_1 \lor C_2 \lor D_1 \lor D_2$

is redundant w.r.t.

$C_1 \lor D_1 \lor B, \; C_2 \lor D_2 \lor \neg B \in N$ with $A \prec B$. 
Using Redundancy and Basicness for Obtaining Decision Procedures for Fragments of FO-logic

Redundancy

1. \( \neg xTy \lor \neg yTz \lor xTz \);  
3. \( \neg \alpha(x) \lor \neg xTz_1 \lor \neg f_T(x)Tz_1 \);  

**OR**[1;3]:  
1a: \( \neg f_T(s)Tt \lor \neg tTh \lor f_T(s)Th \);  
3a: \( \neg \alpha(s) \lor \neg sTt \lor f_T(s)Tt \);  

\( \Rightarrow \neg \alpha(s) \lor \neg sTt \lor \neg tTh \lor f_T(s)Th \);  

And by:  
1b: \( \neg sTt \lor \neg tTh \lor sTh \);  
3b: \( \neg \alpha(s) \lor \neg sTh \lor f_T(s)Th \);  

\( \Rightarrow \) Inferece redundant by  

**Lemma** [Four Clauses] since  

\( f_T(s)Tt \succ sTt \)!
Using Redundancy and Basicness for Obtaining Decision Procedures for Fragments of FO-logic

Basicness

Ordered Paramodulation.

\[ \text{OP : } \frac{C \lor s \cong t \quad D \lor L[s']} {C \sigma \lor D \sigma \lor L[t] \sigma} \]

where (i) \( \sigma = \text{mgu}(s, s') \), (ii) \( s \cong t \) and \( L[s'] \) are eligible, (iii) \( t \sigma \not\subseteq s \sigma \), and (iv) \( s' \) is not a variable.

\( (iv)' \) \( s' \) is not below a substitutional position.

- This restriction can be strengthened to basicness:

1. \( \neg xT y \lor \neg yT z \lor xT z \);
2. \( \neg \alpha(x) \lor f_T(x)T x \); ← "Source" of \( f_T \)

OR [1;2]:
3. \( \neg \alpha(x) \lor \neg xT z_1 \lor f_T(x)T z_1 \);
Basicness

➢ Only paramodulation to the “source” of Skolem function is needed.

- Helps to avoid the “dangerous” paramodulation inferences:

3. \( \neg \alpha(x) \lor f_T(x) \; \neg x \lor f_T(x) \equiv y \);  
C : \( \neg x T y \lor \alpha(x) \lor \beta(y) \lor f_T(x) \equiv y \);  
D : \( \neg x T z \lor \alpha'(x) \lor \beta'(z) \lor f_T(x) \equiv z \);

- Eligible paramodulation inferences produce redundant clauses only.
Conclusions

- Using advanced refinements of saturation-based procedures it is possible establish decidability and complexity results for very expressive fragments of FO-logic.
  
  - In particular, decidability of $[GF+TG]$ can be established using redundancy and basicness.
  
  - Basicness is important: allowing conjunctions of transitive relations in guards leads to undecidability.

- New perspectives for designing saturation-based decision procedures.
Thank you!