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

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:

$$\mathcal{ALC} ::= A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \forall R.C \mid \exists R.C.$$

- where C, C_1, C_2 - concepts (unary relations) build from:

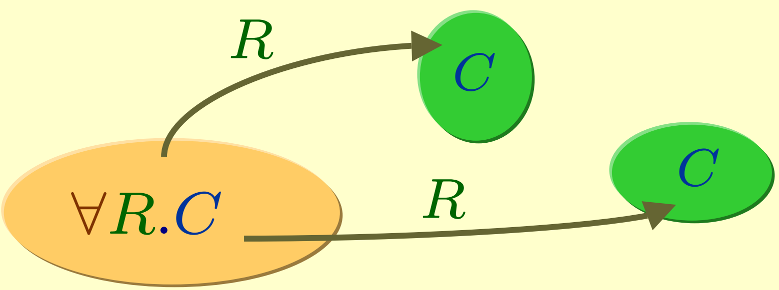
A – basic concepts (initial sets) and

R – roles (binary relations).

- Reasoning problem:

Concept Subsumption:

$$C \overset{?}{\sqsubseteq} D$$



Fragments of FO-logic

- Description logic as FO-fragment:

$$\mathcal{ALC} ::= A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \forall R.C \mid \exists R.C.$$

⇒

$$\mid A(x) \mid C_1(x) \wedge C_2(x) \mid C_1(x) \vee C_2(x) \mid \neg C(x) \mid \forall y.(R(x,y) \rightarrow C(y)) \mid \exists y.(R(x,y) \wedge C(y)). \quad =:: \mathcal{FO}[\mathcal{ALC}]$$

Subsumption problem: $C \sqsubseteq D$ ⇒ Entailment Problem: $C(x) \rightarrow D(x)$

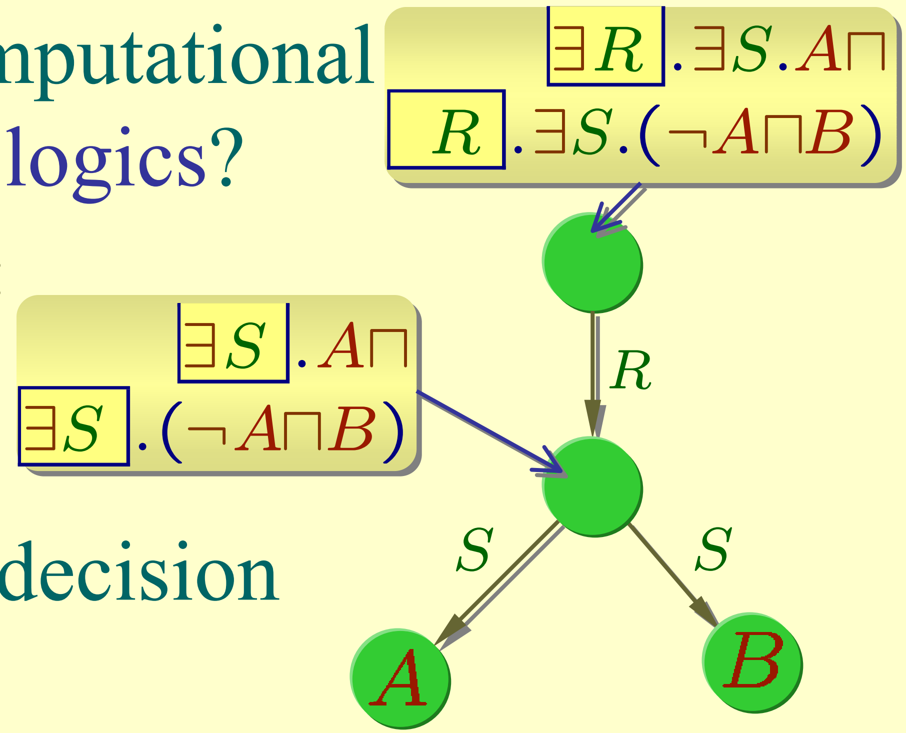
II. Decision procedures

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.



Decision Procedures

- Extensions of description-like logics are harder to handle:
  - $\mathcal{ALC}$ $+$ O – nominals (one-element sets)
    $+$ $S \sqsubseteq R$ – role hierarchies
    $+$ Transitive(R) – transitive roles
    $+$ $\exists^{<n} R.C$, $\exists^{\geq n} R.C$ – counting
  - $\mathcal{UML}$ class diagramms
  - $\mathcal{OWL}$ – ontology language for semantic web

Decision Procedures

- Extensions of description-like logics are harder to handle:
  - $\mathcal{ALC}$ + counting – no finite model property;
  - $\mathcal{ALC}$ + transitive roles – no tree model property;
  - $\mathcal{ALC}$ + counting + transitive roles + un-restricted role hierarhcies – undecidable.
- Decision procedures rely on heavy model-theoretic analysis:
  - "Good" model representation property

Decision Procedures

- Alternative approach: use general theorem provers for FO-logic.
  - Advantage:
    - No need to invent anything;
    - Soundness and completeness are guarantied;
    - 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.

Saturation-based Decision Procedures

Problem: $\overset{?}{\vdash} F$

CNF transformation ⇒

N
Clause Set

$A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);$
$\neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
$C(fx)\vee\neg B(gy)\vee E(x,y);$
...

$$\frac{C\vee \underline{A} \qquad D\vee\neg \underline{B}}{C\sigma\vee D\sigma}$$

Saturation

$\bot$?

Yes

No

Saturated Set

$A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);\ \neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
$C(fx)\vee\neg B(gy)\vee E(x,y);\ A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);$
$\neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);\ C(fx)\vee\neg B(gy)\vee E(x,y);$
$A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);\ \neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
[illegible]
$C(fx)\vee\neg B(gy)\vee E(x,y);\ A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);$
$\neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);\ C(fx)\vee\neg B(gy)\vee E(x,y);$
$C(fx)\vee\neg B(gy)\vee E(x,y);\ A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);$
$\neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);\ C(fx)\vee\neg B(gy)\vee E(x,y);$
$A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);\ \neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
...

$B(y)\vee\neg D(z)\vee E(x,z);\ \neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
$C(fx)\vee\neg B(gy)\vee E(x,y);\ A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);$
$\neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);\ A(x)\vee\neg B(fx)\vee C(y);$
$B(y)\vee\neg D(z)\vee E(x,z);\ \neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
[illegible]
$C(fx)\vee\neg B(gy)\vee E(x,y);\ C(fx)\vee\neg B(gy)\vee E(x,y);$
$A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);\ \neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
$C(fx)\vee\neg B(gy)\vee E(x,y);\ A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);$
$\neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);\ C(fx)\vee\neg B(gy)\vee E(x,y);$
$A(x)\vee\neg B(fx)\vee C(y);\ B(y)\vee\neg D(z)\vee E(x,z);\ \neg B(c)\vee\neg H(f(x),x)\vee\neg A(x);$
...

Ordered Paramodulation Calculus

Ordered Resolution.

$$\text{OR}: \frac{C \vee \underline{A} \qquad D \vee \neg \underline{B}}{C\sigma \vee D\sigma}$$

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

Ordered Factoring.

$$\text{OF}: \frac{C \vee \underline{A} \vee \underline{A'}}{C\sigma \vee A\sigma}$$

where (i) $\sigma = mgu(A, A')$, (ii) A is eligible.

Ordered Paramodulation.

$$\text{OP}: \frac{C \vee \underline{s} \simeq t \qquad D \vee L[\underline{s'}]}{C\sigma \vee D\sigma \vee L[t]\sigma}$$

where (i) $\sigma = mgu(s, s')$, (ii) $s \simeq t$ and $L[s']$ are eligible, (iii) $t\sigma \not\succ s\sigma$, and (iv) s' is not a variable.

Reflexivity Resolution.

$$\text{RR}: \frac{C \vee \underline{u} \not\simeq \underline{v}}{C\sigma}$$

where (i) $\sigma = mgu(u, v)$, (ii) $u \not\simeq v$ is eligible.

Equality Factoring.

$$\text{EF}: \frac{C \vee \underline{u} \simeq v \vee \underline{u'} \simeq v'}{C\sigma \vee v'\sigma \not\simeq v\sigma \vee u\sigma \simeq v\sigma}$$

where (i) $\sigma = mgu(u, u')$, (ii) $u \simeq v$ is eligible.

The Guarded Fragment

$$\mathcal{GF} ::= A \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \neg F_1 \mid \forall \overline{x}.(G \rightarrow F_1) \mid \exists \overline{x}.(G \wedge F_1) .$$

$$\mid A(x) \mid C_1(x) \wedge C_2(x) \mid C_1(x) \vee C_2(x) \mid \neg C(x) \mid \forall y.(R(x,y) \rightarrow C(y)) . \exists y.(R(x,y) \wedge C(y)) =:: \mathcal{FO}[\mathcal{ALC}]$$

G – "guard":

$FV(F_1) \subseteq FV(G)$

non-Guarded Formulae:

Transitivity:
$\forall x,y,z.(xTy \wedge yTz \rightarrow xTz)$

Functionality:
$\forall x,y,z.(xFy \wedge xFz \rightarrow y \simeq z)$

Guarded Formula:

$\exists x.(n(x) \wedge \forall y.[a(x,y) \rightarrow$
$\forall z.\{x \simeq z \rightarrow \exists x.[a(x,z) \wedge (\neg b(z,z) \vee \neg c(x,x))]\}])$

Saturating the Guarded Fragment

$$
\begin{aligned}
\mathcal{GF} ::= {} & A \mid \\
& F_1 \wedge F_2 \mid \\
& F_1 \vee F_2 \mid \\
& \neg F_1 \mid \\
& \forall \overline{x}.(G \to F_1) \mid \\
& \exists \overline{x}.(G \wedge F_1)\ .
\end{aligned}
$$

CNF transformation

1. $\alpha(\hat{c})$;
2. $\neg \hat{a}(\hat{\boldsymbol{x}}) \vee \alpha(\hat{f}(\hat{\boldsymbol{x}}), \hat{x})$.

$\mathcal{OP}$ Saturation

Guarded Clause Fragment:

1. $\alpha(\hat{c})$;
2. $\neg \hat{a}(\hat{\boldsymbol{x}}) \vee \alpha(\hat{f}(\hat{\boldsymbol{x}}), \hat{x})$.
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 loose decidability when some predicates are allowed to be transitive, or functional?
  - YES [Grädel,1999]: GF^3 with one functional or transitive predicate is undecidable.
- How to explain decidability of modal and description logics with transitivity?
  - [Ganzinger et al.,1999]: GF^2[T] is undecidable, but monadic-GF^2[T] is decidable.

Guarded Fragment With Transitivity

- Is GF decidable when transitive predicates can appear in guards only? => [GF+TG]?
- What is the complexity of monadic-GF[T]?
- [Szwast,Tendera,2001]: [GF+TG] is in DEXPTIME, monadic-GF[T] is NEXPTIME-hard.
- [Kierionski,2002,2003]: [GF+TG$^{\rightarrow}$] is EXPSPACE-hard, [GF+TG] 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. $\neg xTz_1 \vee \neg z_1Tz_2 \vee xTz_2$; $\prec$ $\neg f_T(x)Tz_1 \vee \neg z_1Tz_2 \vee f_T(x)Tz_2$;

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

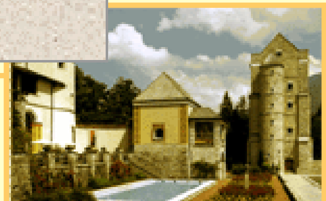
- Abstract notion of redundancy [Bachmair,Ganzinger,1990]:
  - A ground clause C is redundant w.r.t. a set of ground clauses N if $N_{\prec C} \vdash C$;
  - An inference $C_1, C_2 \vdash C$ is redundunt w.r.t. N if $N_{\prec max(C_1,C_2)} \vdash C$.
- How to show that inference is redundant?

Lemma [Four Clauses] The inference

$$C_1 \vee C_2 \vee \underline{A},\ \ D_1 \vee D_2 \vee \neg \underline{A} \vdash C_1 \vee C_2 \vee D_1 \vee D_2$$

is redundant w.r.t.

$C_1 \vee D_1 \vee B,\ C_2 \vee D_2 \vee \neg B \in N$ with $A \prec B$.



Redundancy

1. $\neg \underline{xTy} \vee \neg yTz \vee xTz$;
3. $\neg\alpha(x) \vee \neg xTz_1 \vee \underline{f_T(x)Tz_1}$;

OR[1;3] : 1a : $\neg \underline{f_T(s)Tt} \vee \neg tTh \vee f_T(s)Th$;

3a : $\neg\alpha(s) \vee \neg sTt \vee \underline{f_T(s)Tt}$;

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

And by : 1b : $\neg sTt \vee \neg tTh \vee \underline{sTh}$;

3b : $\neg\alpha(s) \vee \neg \underline{sTh} \vee f_T(s)Th$;

$\Rightarrow$ Inferece redundant by Lemma [Four Clauses] since $f_T(s)Tt \succ sTt$!

Basicness

Ordered Paramodulation.

$$\text{OP}: \frac{C \vee \underline{s} \simeq t \qquad D \vee L[\underline{s'}]}{C\sigma \vee D\sigma \vee L[t]\sigma}$$

where (i) $\sigma = mgu(s, s')$, (ii) $s \simeq t$ and $L[s']$ are eligible, (iii) $t\sigma \not\succ s\sigma$, and (iv) s' is not a variable.

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

- This restriction can be strengthen to basicness:.

1. $\neg \underline{xTy} \vee \neg yTz \vee xTz$;
2. $\neg\alpha(x) \vee \underline{f_T(x)Tx}$; ← "Source" of f_T

OR[1;2]: 3. $\neg\alpha(x) \vee \neg xTz_1 \vee \underline{f_T(x)Tz_1}$;

Basicness

- Only paramodulation to the "source" of Skolem function is needed.
  - Helps to avoid the "dangerous" paramodulation inferences:

$$\begin{array}{ll} 3. & \neg a(x) \vee \underline{f_T(x) T x}; \\ C: & \neg xTy \vee \alpha(x) \vee \beta(y) \vee \underline{f_T(x) \simeq y}; \\ D: & \neg xTz \vee \alpha'(x) \vee \beta'(z) \vee \underline{f_T(x) \simeq z}; \end{array}$$

  - 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 !