SATURATION-BASED DECISION PROCEDURES:

FROM SIMPLE DESCRIPTION LOGICS TO EXPRESSIVE EXTENSIONS OF THE GUARDED FRAGMENT AND BACK TO IMPLEMENTATION

Yevgeny Kazakov

Max-Planck Institute für Informatik

March 17, 2006

Outline

Outline

Tree (Unrelated?) Questions

1. How to use automated theorem provers for obtaining decision procedures?
2. Why some fragments of first-order logics are decidable and others are not?
3. How to design practical and complexity-optimal procedures for reasoning in description logics?

What are Description Logics?

"... [formalisms] for providing high level description of the world that can be effectively used to build intelegent applications." *(Nardi & Brachman, 2003).*

- A family of languages for knowledge representation which:
- Provide a logic-based descriptions of concepts by means of their mutual relationships
- Distinguished by a formal semantics which gives unambiguous reading for these descriptions
- Have effective procedures to identify logical consequences of descriptions and answer queries

What are Description Logics?

"... [formalisms] for providing high level description of the world that can be effectively used to build intelegent applications." *(Nardi & Brachman, 2003).*

- A family of languages for knowledge representation which:
- Provide a logic-based descriptions of concepts by means of their mutual relationships
- Distinguished by a formal semantics which gives unambiguous reading for these descriptions
- Have effective procedures to identify logical consequences of descriptions and answer queries

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- ?– PhDStudent(X) $\wedge$ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
-
  - X = Hans de Nivelle, Y = Ruzica Piskac
  - X = Gert Smolka, Y = Yevgeny Kazakov
  - X = Hans de Nivelle, Y = Yevgeny Kazakov

CLASSICAL APPROACH

DATA (COLLECTION OF FACTS):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

QUERIES:

- $?-$ PhDStudent$(X) \wedge$ D2Member(X).
  - $X =$ Yevgeny Kazakov
  - $X =$ Ruzica Piskac
-
  - $X =$ Hans de Nivelle, $Y =$ Ruzica Piskac
  - $X =$ Gert Smolka, $Y =$ Yevgeny Kazakov
  - $X =$ Hans de Nivelle, $Y =$ Yevgeny Kazakov

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- ?– PhDStudent(X) $\wedge$ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
-
  - X = Hans de Nivelle, Y = Ruzica Piskac
  - X = Gert Smolka, Y = Yevgeny Kazakov
  - X = Hans de Nivelle, Y = Yevgeny Kazakov

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- $?-$ PhDStudent(X) $\wedge$ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
-
  - X = Hans de Nivelle, Y = Ruzica Piskac
  - X = Gert Smolka, Y = Yevgeny Kazakov
  - X = Hans de Nivelle, Y = Yevgeny Kazakov

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- ?– PhDStudent(X) ∧ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
- ?– Supervisor(X) ∧ hasStudent(X, Y) ∧ D2Member(Y).
  - X = Hans de Nivelle, Y = Ruzica Piskac
  - X = Gert Smolka, Y = Yevgeny Kazakov
  - X = Hans de Nivelle, Y = Yevgeny Kazakov

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- ?− PhDStudent(X) ∧ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
- ?− Supervisor(X) ∧ hasStudent(X, Y) ∧ D2Member(Y).
  - X = Hans de Nivelle, Y = Ruzica Piskac
  - X = Gert Smolka, Y = Yevgeny Kazakov
  - X = Hans de Nivelle, Y = Yevgeny Kazakov

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- ?− PhDStudent(X) $\wedge$ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
- ?− Supervisor(X) $\wedge$ hasStudent(X, Y) $\wedge$ D2Member(Y).
  - X = Hans de Nivelle, Y = Ruzica Piskac
  - X = Gert Smolka, Y = Yevgeny Kazakov
  - X = Hans de Nivelle, Y = Yevgeny Kazakov

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- ?– PhDStudent(X) $\wedge$ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
- ?– Supervisor(X) $\wedge$ hasStudent(X, Y) $\wedge$ D2Member(Y).
  - X = Hans de Nivelle, Y = Ruzica Piskac
  - X = Gert Smolka, Y = Yevgeny Kazakov
  - X = Hans de Nivelle, Y = Yevgeny Kazakov

Classical Approach

Data (Collection of Facts):

PhDStudent	Supervisor
...	...
Yevgeny Kazakov	Hans de Nivelle
Yevgeny Kazakov	Gert Smolka
Ruzica Piskac	Hans de Nivelle

D2Member	Email
...	
Hans de Nivelle	...
Yevgeny Kazakov	...
Ruzica Piskac	...

Queries:

- ?– PhDStudent(X) $\wedge$ D2Member(X).
  - X = Yevgeny Kazakov
  - X = Ruzica Piskac
- ?– Supervisor(X) $\sqcap$ $\exists$hasStudent.D2Member(X).
  - X = Hans de Nivelle,
  - X = Gert Smolka,
  - X = Hans de Nivelle,

DL-Based Approach

Data ABox (Collection of Facts):

PhDStudent	Supervisor
...	...

D2Member	Email
...	

Metadata TBox (Properties of Classes and Relations):

$$\text{Supervisor} \doteq \exists \text{hasStudent}.\text{PhDStudent}$$
$$\text{PhDStudent} \sqcap \text{Supervisor} \sqsubseteq \bot$$
$$\text{PhDStudent} \sqcap \text{D2Member} \sqsubseteq \exists \text{hasSupervisor}.\text{D2Member}$$
$$\text{hasStudent} \doteq (\text{hasSupervisor})^{-}$$

- Gives a more expressive query language:
  - $? - \exists \text{hasStudent}.\forall \text{hasSupervisor}.\text{D2Member}(X).$
- Enables query optimisations:
  - $? - \text{Supervisor}(X) \sqcap \text{D2Member}(X).$

DL-Based Approach

Data ABox (Collection of Facts):

PhDStudent	Supervisor
...	...

D2Member	Email
...	

Metadata TBox (Properties of Classes and Relations):

$$\begin{aligned} \text{Supervisor} &\doteq \exists \text{hasStudent}.\text{PhDStudent} \\ \text{PhDStudent} \sqcap \text{Supervisor} &\sqsubseteq \bot \\ \text{PhDStudent} \sqcap \text{D2Member} &\sqsubseteq \exists \text{hasSupervisor}.\text{D2Member} \\ \text{hasStudent} &\doteq (\text{hasSupervisor})^{-} \end{aligned}$$

- Gives a more expressive query language:
  - $?\!-\, \exists \text{hasStudent}.\forall \text{hasSupervisor}.\text{D2Member}(X).$
- Enables query optimisations:
  - $?\!-\, \text{Supervisor}(X) \sqcap \text{D2Member}(X).$

DL-Based Approach

Data ABox (Collection of Facts):

PhDStudent	Supervisor
. . .	. . .

D2Member	Email
. . .	

Metadata TBox (Properties of Classes and Relations):

$$\begin{aligned} \text{Supervisor} &\doteq \exists \text{hasStudent}.\text{PhDStudent} \\ \text{PhDStudent} \sqcap \text{Supervisor} &\sqsubseteq \bot \\ \text{PhDStudent} \sqcap \text{D2Member} &\sqsubseteq \exists \text{hasSupervisor}.\text{D2Member} \\ \text{hasStudent} &\doteq (\text{hasSupervisor})^{-} \end{aligned}$$

- Gives a more expressive query language:
  - $?- \exists \text{hasStudent}.\forall \text{hasSupervisor}.\text{D2Member}(X).$
- Enables query optimisations:
  - $?- \text{Supervisor}(X) \sqcap \text{D2Member}(X).$

DL-Based Approach

Data ABox (Collection of Facts):

PhDStudent	Supervisor
...	...

D2Member	Email
...	

Metadata TBox (Properties of Classes and Relations):

$$\text{Supervisor} \doteq \exists \text{hasStudent}.\text{PhDStudent}$$
$$\text{PhDStudent} \sqcap \text{Supervisor} \sqsubseteq \bot$$
$$\text{PhDStudent} \sqcap \text{D2Member} \sqsubseteq \exists \text{hasSupervisor}.\text{D2Member}$$
$$\text{hasStudent} \doteq (\text{hasSupervisor})^{-}$$

- Gives a more expressive query language:
  - $?- \exists \text{hasStudent}.\forall \text{hasSupervisor}.\text{D2Member}(X).$
- Enables query optimisations:
  - $?- \text{Supervisor}(X) \sqcap \text{D2Member}(X).$

THE LANGUAGE OF DLS

- PRIMITIVE CONCEPTS (unary relations): PhDStudent
  Supervisor
  D2Member

The Language of DLs

- Primitive Concepts (unary relations): PhDStudent
  Supervisor
  D2Member
- Primitive Roles (binary relations): hasStudent
  hasSupervisor

The Language of DLs

- Primitive Concepts (unary relations): PhDStudent, Supervisor, D2Member
- Primitive Roles (binary relations): hasStudent, hasSupervisor
- Individuals (elements): "Gert Smolka", "Hans de Nivelle"

THE LANGUAGE OF DLs

- PRIMITIVE CONCEPTS (unary relations): PhDStudent, Supervisor, D2Member
- PRIMITIVE ROLES (binary relations): hasStudent, hasSupervisor
- INDIVIDUALS (elements): "Gert Smolka", "Hans de Nivelle"
- OPERATORS to form new concepts from existing ones:

$(C_1 \sqcap C_2)$	Conjunction:	$\text{PhDStudent} \sqcap \text{D2Member}$
$(C_1 \sqcup C_2)$	Disjunction:	$\text{PhDStudent} \sqcup \text{Supervisor}$
$(\exists R.C_1)$	Existential Restriction:	$\exists \text{hasStudent}.\text{D2Member}$
$(\forall R.C_1)$	Value Restriction:	$\forall \text{hasSupervisor}.\text{D2Member}$
$(\geqslant n\, R)$	At least restriction:	$\geqslant 2\ \text{hasStudent}$

Reasoning Problems of DLs

TBox (Terminology)

$$\text{Supervisor} \doteq \exists \text{hasStudent}.\text{PhDStudent}$$
$$\text{PhDStudent} \sqcap \text{Supervisor} \sqsubseteq \bot$$
$$\text{PhDStudent} \sqcap \text{D2Member} \sqsubseteq \exists \text{hasSupervisor}.\text{D2Member}$$

ABox (Assertions)

D2Member(Hans de Nivelle)
PhDStudent(Ruzica Piskas)
hasStudent(Hans de Nivelle, Ruzica Piakas)

Reasoning Problems of DLs

TBox (Terminology)

$$\text{Supervisor} \doteq \exists \text{hasStudent}.\text{PhDStudent}$$
$$\text{PhDStudent} \sqcap \text{Supervisor} \sqsubseteq \bot$$
$$\text{PhDStudent} \sqcap \text{D2Member} \sqsubseteq \exists \text{hasSupervisor}.\text{D2Member}$$

ABox (Assertions)

$$\text{D2Member}(\text{Hans de Nivelle})$$
$$\text{PhDStudent}(\text{Ruzica Piskas})$$
$$\text{hasStudent}(\text{Hans de Nivelle}, \text{Ruzica Piakas})$$

Queries (Reasoning Problems)

$?-\ \exists \text{hasStudent}.\text{D2Member} \sqsubseteq \text{Supervisor}$ (subsumption)

$?-\ \text{Supervisor}(\text{Hans de Nivelle})$ (instance)

$?-\ (\text{PhDStudent} \sqcap \text{D2Member})(X)$ (retrieval)

Some Applications of DLs

- Databases: integration of conceptual schemata ($\sim$ **TBox**), query subsumption, configuration,...

SOME APPLICATIONS OF **DL**s

- Databases: integration of conceptual schemata ($\sim$ **TBox**), query subsumption, configuration,...
- Semantic Web:

  "the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications." *[W3C Semantic Web vision]*

Some Applications of DLs

- Databases: integration of conceptual schemata ($\sim$ **TBox**), query subsumption, configuration,...
- Semantic Web:

  "the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications." *[W3C Semantic Web vision]*

  - The present Web is syntactic (HTML), is designed to be readable by humans
  - The new Web must be readable by programs (a search engine should "understand" the web content)
  - A **DL**-based language $\mathcal{OWL}$ has been recommended by W3C as an ontology language for the Semantic Web

SOME APPLICATIONS OF DLs

- **Databases**: integration of conceptual schemata ($\sim$ **TBox**), query subsumption, configuration,. . .
- **Semantic Web**:

"the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications." *[W3C Semantic Web vision]*

HOMEPAGE OF YEVGENY KAZAKOV

Yevgeny Kazakov

$\mathcal{OWL}$-ONLOLOGY

PhDStudent $\doteq$ · · ·

  - The present Web is syntactic (HTML), is designed to be readable by humans
  - The new Web must be readable by programs (a search engine should "understand" the web content)
  - A **DL**-based language $\mathcal{OWL}$ has been recommended by W3C as an ontology language for the Semantic Web

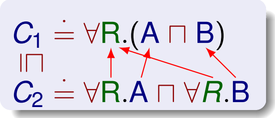
Outline

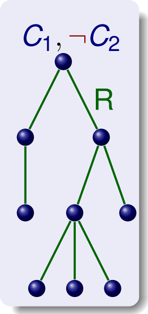
Reasoning Procedures for DLs

$$C_1 \doteq \forall R.(A \sqcap B)$$
$$\sqsupseteq$$
$$C_2 \doteq \forall R.A \sqcap \forall R.B$$

1985–1990 Incomplete reasoning procedures based on structural subsumption algorithms (KL-ONE (Brachman & Schmolze, 1985), systems: BLACK, CLASSIC, LOOM . . .)

REASONING PROCEDURES FOR DLS

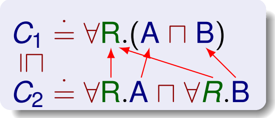


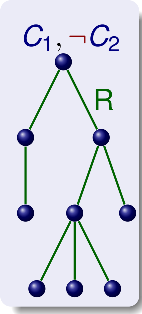


1985–1990 Incomplete reasoning procedures based on structural subsumption algorithms (KL-ONE (Brachman & Schmolze, 1985), systems: BLACK, CLASSIC, LOOM . . .)

1991-1996 Complete tableau-based procedures for DLs closed under negation ($\mathcal{ALC}$ Schmidt-Schauß & Smolka (1991), systems: KRIS (Baader & Hollunder, 1991)), CRACK)

REASONING PROCEDURES FOR DLS



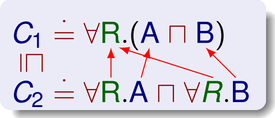


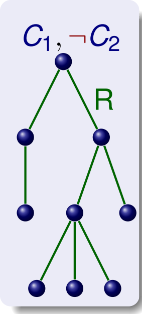
1985–1990 Incomplete reasoning procedures based on structural subsumption algorithms (KL-ONE (Brachman & Schmolze, 1985), systems: BLACK, CLASSIC, LOOM . . .)

1991-1996 Complete tableau-based procedures for DLs closed under negation ($\mathcal{ALC}$ Schmidt-Schauß & Smolka (1991), systems: KRIS (Baader & Hollunder, 1991)), CRACK)

1997-PRESENT Highly optimized implementations for very expressive DLs (FACT (Horrocks, 1998), RACER (Haarslev & Möller, 2001))

REASONING PROCEDURES FOR DLS





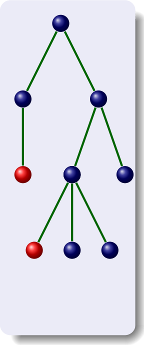
1985–1990 Incomplete reasoning procedures based on structural subsumption algorithms (KL-ONE (Brachman & Schmolze, 1985), systems: BLACK, CLASSIC, LOOM . . .)

1991-1996 Complete tableau-based procedures for DLs closed under negation ($\mathcal{ALC}$ Schmidt-Schauß & Smolka (1991), systems: KRIS (Baader & Hollunder, 1991)), CRACK)

1997-PRESENT Highly optimized implementations for very expressive DLs (FACT (Horrocks, 1998), RACER (Haarslev & Möller, 2001))

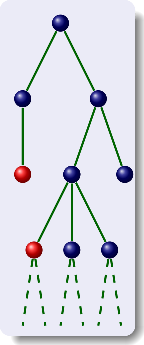
???? What is next?

Tableau-Based Procedures and Complexity



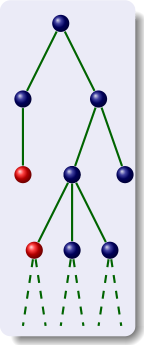
- $\mathcal{ALC}$=($\sqcap$,$\sqcup$,$\neg$,$\exists$.,$\forall$.) - concept subsumtpion. Tableau procedure runs in PSPACE (optimal).
- $\mathcal{ALC}$ with general TBox-es requires cycle detection. Theoretical complexity: EXPTIME, Tableau worst case: EXPSPACE.
- Adding number restrictions ($\geqslant n.R$), and ($\leqslant n.R$) makes the worst case 2EXPSPACE.
- Tree-model property of DLs is the reason behind their decidability, however:
  - Transitive roles $T \circ T \sqsubseteq T$ destroy the tree model property. Instead, tableau proceadures search for a tree-representation of a model.
  - Nominals $O \doteq \{c\}$ can break even this underlying tree-structure. Dealing with nominals is tricky.

Tableau-Based Procedures and Complexity



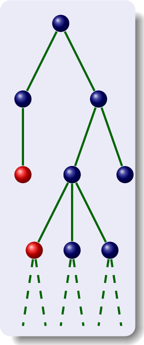
- $\mathcal{ALC}$=($\sqcap$,$\sqcup$,$\neg$,$\exists$.,$\forall$.) - concept subsumtpion. Tableau procedure runs in PSPACE (optimal).
- $\mathcal{ALC}$ with general **TBox**-es requires cycle detection. Theoretical complexity: EXPTIME, Tableau worst case: EXPSPACE.
- Adding number restrictions ($\geqslant n.R$), and ($\leqslant n.R$) makes the worst case 2EXPSPACE.
- Tree-model property of **DL**s is the reason behind their decidability, however:
  - Transitive roles $T \circ T \sqsubseteq T$ destroy the tree model property. Instead, tableau proceadures search for a tree-representation of a model.
  - Nominals $O \doteq \{c\}$ can break even this underlying tree-structure. Dealing with nominals is tricky.

TABLEAU-BASED PROCEDURES AND COMPLEXITY



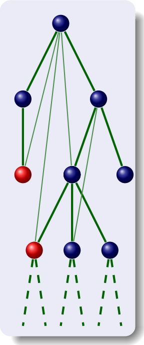
- $\mathcal{ALC}$=($\sqcap$,$\sqcup$,$\neg$,$\exists$.,$\forall$.) - concept subsumtpion. Tableau procedure runs in PSPACE (optimal).
- $\mathcal{ALC}$ with general **TBox**-es requires cycle detection. Theoretical complexity: EXPTIME, Tableau worst case: EXPSPACE.
- Adding number restrictions ($\geqslant n.R$), and ($\leqslant n.R$) makes the worst case 2EXPSPACE.
- Tree-model property of DLs is the reason behind their decidability, however:
  - Transitive roles $T \circ T \sqsubseteq T$ destroy the tree model property. Instead, tableau proceadures search for a tree-representation of a model.
  - Nominals $O \doteq \{c\}$ can break even this underlying tree-structure. Dealing with nominals is tricky.

TABLEAU-BASED PROCEDURES AND COMPLEXITY



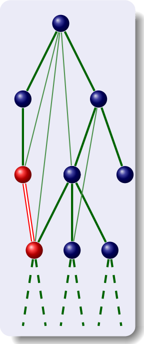
- $\mathcal{ALC}$=($\sqcap$,$\sqcup$,$\neg$,$\exists$.,$\forall$.) - concept subsumtpion. Tableau procedure runs in PSPACE (optimal).
- $\mathcal{ALC}$ with general **TBox**-es requires cycle detection. Theoretical complexity: EXPTIME, Tableau worst case: EXPSPACE.
- Adding number restrictions ($\geqslant n.R$), and ($\leqslant n.R$) makes the worst case 2EXPSPACE.
- Tree-model property of **DL**s is the reason behind their decidability, however:
  - Transitive roles $\mathsf{T} \circ \mathsf{T} \sqsubseteq \mathsf{T}$ destroy the tree model property. Instead, tableau proceadures search for a tree-representation of a model.
  - Nominals $\mathsf{O} \doteq \{\mathrm{c}\}$ can break even this underlying tree-structure. Dealing with nominals is tricky.

TABLEAU-BASED PROCEDURES AND COMPLEXITY



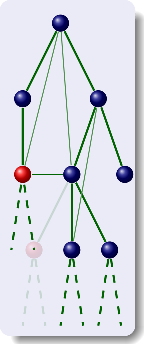
- $\mathcal{ALC}$=($\sqcap$,$\sqcup$,$\neg$,$\exists$.,$\forall$.) - concept subsumtpion. Tableau procedure runs in PSPACE (optimal).
- $\mathcal{ALC}$ with general **TBox**-es requires cycle detection. Theoretical complexity: EXPTIME, Tableau worst case: EXPSPACE.
- Adding number restrictions ($\geqslant n.R$), and ($\leqslant n.R$) makes the worst case 2EXPSPACE.
- Tree-model property of **DL**s is the reason behind their decidability, however:
  - Transitive roles $\mathsf{T} \circ \mathsf{T} \sqsubseteq \mathsf{T}$ destroy the tree model property. Instead, tableau proceadures search for a tree-representation of a model.
  - Nominals $\mathsf{O} \doteq \{\mathrm{c}\}$ can break even this underlying tree-structure. Dealing with nominals is tricky.

Tableau-Based Procedures and Complexity



- $\mathcal{ALC}$=($\sqcap$,$\sqcup$,$\neg$,$\exists$.,$\forall$.) - concept subsumtpion. Tableau procedure runs in PSPACE (optimal).
- $\mathcal{ALC}$ with general **TBox**-es requires cycle detection. Theoretical complexity: EXPTIME, Tableau worst case: EXPSPACE.
- Adding number restrictions ($\geqslant n.R$), and ($\leqslant n.R$) makes the worst case 2EXPSPACE.
- Tree-model property of **DL**s is the reason behind their decidability, however:
  - Transitive roles $\mathsf{T} \circ \mathsf{T} \sqsubseteq \mathsf{T}$ destroy the tree model property. Instead, tableau proceadures search for a tree-representation of a model.
  - Nominals $\mathsf{O} \doteq \{c\}$ can break even this underlying tree-structure. Dealing with nominals is tricky.

TABLEAU-BASED PROCEDURES AND COMPLEXITY



- $\mathcal{ALC}$=($\sqcap$,$\sqcup$,$\neg$,$\exists$.,$\forall$.) - concept subsumtpion. Tableau procedure runs in PSPACE (optimal).
- $\mathcal{ALC}$ with general **TBox**-es requires cycle detection. Theoretical complexity: EXPTIME, Tableau worst case: EXPSPACE.
- Adding number restrictions ($\geqslant n.R$), and ($\leqslant n.R$) makes the worst case 2EXPSPACE.
- Tree-model property of **DL**s is the reason behind their decidability, however:
  - Transitive roles $\mathsf{T} \circ \mathsf{T} \sqsubseteq \mathsf{T}$ destroy the tree model property. Instead, tableau proceadures search for a tree-representation of a model.
  - Nominals $\mathsf{O} \doteq \{c\}$ can break even this underlying tree-structure. Dealing with nominals is tricky.

An Alternative Approach

- Use a general-purpose automated first-order theorem prover (e.g. SPASS or VAMPIRE) to solve reasoning problems in **DL**s:
    - Translate **TBox** + **ABox** + **Query** to clauses according to the semantics of **DL**.
    - Run a theorem prover on the resulted set of clauses.
    - Tweak the parameters of a prover to ensure termination.
- We demonstrate this approach on a simple description logic $\mathcal{EL}$.

AN ALTERNATIVE APPROACH

- Use a general-purpose automated first-order theorem prover (e.g. SPASS or VAMPIRE) to solve reasoning problems in **DL**s:
  - Translate **TBox** + **ABox** + **Query** to clauses according to the semantics of **DL**.
  - Run a theorem prover on the resulted set of clauses.
  - Tweak the parameters of a prover to ensure termination.
- We demonstrate this approach on a simple description logic $\mathcal{EL}$.

An Alternative Approach

- Use a general-purpose automated first-order theorem prover (e.g. SPASS or VAMPIRE) to solve reasoning problems in **DL**s:
  - Translate **TBox** + **ABox** + **Query** to clauses according to the semantics of **DL**.
  - Run a theorem prover on the resulted set of clauses.
  - Tweak the parameters of a prover to ensure termination.
- We demonstrate this approach on a simple description logic $\mathcal{EL}$.

An Alternative Approach

- Use a general-purpose automated first-order theorem prover (e.g. SPASS or VAMPIRE) to solve reasoning problems in **DL**s:
  - Translate **TBox** + **ABox** + **Query** to clauses according to the semantics of **DL**.
  - Run a theorem prover on the resulted set of clauses.
  - Tweak the parameters of a prover to ensure termination.
- We demonstrate this approach on a simple description logic $\mathcal{EL}$.

An Alternative Approach

- Use a general-purpose automated first-order theorem prover (e.g. SPASS or VAMPIRE) to solve reasoning problems in **DL**s:
  - Translate **TBox** + **ABox** + **Query** to clauses according to the semantics of **DL**.
  - Run a theorem prover on the resulted set of clauses.
  - Tweak the parameters of a prover to ensure termination.
- We demonstrate this approach on a simple description logic $\mathcal{EL}$.

SUBBOOLEAN DLs

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

Subsumption w.r.t. $\mathcal{ALC}$ **TBox**-es is EXPTIME-complete

Subboolean DLs

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

$$\mathcal{FL}_0 ::= A \mid C_1 \sqcap C_2 \mid \cancel{C_1 \sqcup C_2} \mid \cancel{\neg C_1} \mid \cancel{\exists R.C_1} \mid \forall R.C_1 \;.$$

Subsumption w.r.t. $\mathcal{ALC}$ **TBox**-es is EXPTIME-complete

Subboolean DLs

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

$\mathcal{FL}_0 ::= A \mid C_1 \sqcap C_2 \mid$ ~~$C_1 \sqcup C_2$~~ $\mid$ ~~$\neg C_1$~~ $\mid$ ~~$\exists R.C_1$~~ $\mid \forall R.C_1$.

Subsumption w.r.t. $\mathcal{ALC}$ **TBox**-es is EXPTIME-complete

Theorem (Baader (1996), Kazakov & de Nivelle (2003))

*Subsumption w.r.t. $\mathcal{FL}_0$ **TBox**-es is PSPACE-complete*

SUBBOOLEAN DLs

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

$$\mathcal{FL}_0 ::= A \mid C_1 \sqcap C_2 \mid \cancel{C_1 \sqcup C_2} \mid \cancel{\neg C_1} \mid \cancel{\exists R.C_1} \mid \forall R.C_1 .$$

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

Subsumption w.r.t. $\mathcal{ALC}$ **TBox**-es is EXPTIME-complete

THEOREM (BAADER (1996), KAZAKOV & DE NIVELLE (2003))

Subsumption w.r.t. $\mathcal{FL}_0$ **TBox***-es is PSPACE-complete*

Subboolean DLs

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

$\mathcal{FL}_0 ::= A \mid C_1 \sqcap C_2 \mid$ ~~$C_1 \sqcup C_2$~~ $\mid$ ~~$\neg C_1$~~ $\mid$ ~~$\exists R.C_1$~~ $\mid \forall R.C_1$.

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

Subsumption w.r.t. $\mathcal{ALC}$ **TBox**-es is EXPTIME-complete

Theorem (Baader (1996), Kazakov & de Nivelle (2003))

*Subsumption w.r.t. $\mathcal{FL}_0$ **TBox**-es is PSPACE-complete*

Theorem (Baader (2002))

*Subsumption w.r.t. $\mathcal{EL}$ **TBox**-es is polynomially solvable*

A Resolution Decision Procedure for $\mathcal{EL}$

TBox

$$A \doteq C$$

$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$
$$\text{Parent} \doteq \text{Human} \sqcap \exists \text{has-child}.\text{Human}$$
$$\text{Father} \doteq \text{Man} \sqcap \exists \text{has-child}.\text{Human}$$
$$\text{Grandfather} \doteq \text{Man} \sqcap \exists \text{has-child}.\text{Parent}$$

Subsumption Query

$$?- C_1 \sqsubseteq C_2$$

$$?- \text{Father} \sqsubseteq \text{Parent}$$
$$?- \text{Grandfather} \sqsubseteq \text{Father}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

TBox

$$A \doteq C$$

$$\begin{aligned} \text{Man} &\doteq \text{Human} \sqcap \text{Male} \\ \text{Parent} &\doteq \text{Human} \sqcap \exists \text{has-child}.\text{Human} \\ \text{Father} &\doteq \text{Man} \sqcap \exists \text{has-child}.\text{Human} \\ \text{Grandfather} &\doteq \text{Man} \sqcap \exists \text{has-child}.\text{Parent} \end{aligned}$$

Subsumption Query

$$?- C_1 \sqsubseteq C_2$$

$$\begin{aligned} ?- \text{Father} &\sqsubseteq \text{Parent} \\ ?- \text{Grandfather} &\sqsubseteq \text{Father} \end{aligned}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Take a compound concept

TBox

$$A \doteq C$$

$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$

$$\text{Parent} \doteq \text{Human} \sqcap \exists \text{has-child}.\text{Human}$$

$$\text{Father} \doteq \text{Man} \sqcap \exists \text{has-child}.\text{Human}$$

$$\text{Grandfather} \doteq \text{Man} \sqcap \exists \text{has-child}.\text{Parent}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. TBox-simplification
2. FO-translation
3. Clausification
4. Saturation in ATP

- Take a compound concept
- Replace by a new concept name

TBox

$$A \doteq C$$
$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$
$$\text{Parent} \doteq \text{Human} \sqcap \underline{\text{N1}}$$
$$\text{Father} \doteq \text{Man} \sqcap \underline{\text{N1}}$$
$$\text{Grandfather} \doteq \text{Man} \sqcap \exists\text{has-child}.\text{Parent}$$
$$\underline{\text{N1}} \doteq \exists\text{has-child}.\text{Human}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Take a compound concept
- Replace by a new concept name

TBox

$$A \doteq C$$

$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$

$$\text{Parent} \doteq \text{Human} \sqcap \text{N1}$$

$$\text{Father} \doteq \text{Man} \sqcap \text{N1}$$

$$\text{Grandfather} \doteq \text{Man} \sqcap \exists \text{has-child}.\text{Parent}$$

$$\text{N1} \doteq \exists \text{has-child}.\text{Human}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Take a compound concept
- Replace by a new concept name

TBox

$$A \doteq C$$
$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$
$$\text{Parent} \doteq \text{Human} \sqcap \text{N1}$$
$$\text{Father} \doteq \text{Man} \sqcap \text{N1}$$
$$\text{Grandfather} \doteq \text{Man} \sqcap \underline{\text{N2}}$$
$$\text{N1} \doteq \exists\text{has-child}.\text{Human}$$
$$\underline{\text{N2}} \doteq \exists\text{has-child}.\text{Parent}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Take a compound concept
- Replace by a new concept name
- After simplifications all definitions have the form:

TBox

$$A \doteq C$$
$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$
$$\text{Parent} \doteq \text{Human} \sqcap \text{N1}$$
$$\text{Father} \doteq \text{Man} \sqcap \text{N1}$$
$$\text{Grandfather} \doteq \text{Man} \sqcap \text{N2}$$
$$\text{N1} \doteq \exists \text{has-child}.\text{Human}$$
$$\text{N2} \doteq \exists \text{has-child}.\text{Parent}$$

Simplified concept definitions

$$A \doteq B \sqcap C$$
$$A \doteq \exists R.B$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Take a compound concept
- Replace by a new concept name
- After simplifications all definitions have the form:

TBox

$$A \doteq C$$
$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$
$$\text{Parent} \doteq \text{Human} \sqcap \text{N1}$$
$$\text{Father} \doteq \text{Man} \sqcap \text{N1}$$
$$\text{Grandfather} \doteq \text{Man} \sqcap \text{N2}$$
$$\text{N1} \doteq \exists \text{has-child}.\text{Human}$$
$$\text{N2} \doteq \exists \text{has-child}.\text{Parent}$$

Simplified concept definitions

$$A \doteq B \sqcap C$$
$$A \doteq \exists R.B$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Translate simplified definitions according to the semantics of **DL**:

TBox

$$A \doteq C$$
$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$
$$\text{Parent} \doteq \text{Human} \sqcap \text{N1}$$
$$\text{Father} \doteq \text{Man} \sqcap \text{N1}$$
$$\text{Grandfather} \doteq \text{Man} \sqcap \text{N2}$$
$$\text{N1} \doteq \exists \text{has-child}.\text{Human}$$
$$\text{N2} \doteq \exists \text{has-child}.\text{Parent}$$

First-Order Translation

$A \doteq B \sqcap C$	$A(x) \leftrightarrow B(x) \wedge C(x)$
$A \doteq \exists R.B$	$A(x) \leftrightarrow \exists y.[R(x,y) \wedge B(y)]$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Translate simplified definitions according to the semantics of **DL**:

TBox

$$A \doteq C$$
$$\text{Man} \doteq \text{Human} \sqcap \text{Male}$$
$$\text{Parent} \doteq \text{Human} \sqcap \text{N1}$$
$$\text{Father} \doteq \text{Man} \sqcap \text{N1}$$
$$\text{Grandfather} \doteq \text{Man} \sqcap \text{N2}$$
$$\text{N1} \doteq \exists \text{has-child}.\text{Human}$$
$$\text{N2} \doteq \exists \text{has-child}.\text{Parent}$$

First-Order Translation

$A \doteq B \sqcap C$	$A(x) \leftrightarrow B(x) \wedge C(x)$
$A \doteq \exists R.B$	$A(x) \leftrightarrow \exists y.[R(x,y) \wedge B(y)]$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Apply standard Skolemization and clause normal form transformations

Clause types

T1. $\neg A(x) \vee B(x)$

Clausification

$$(\Rightarrow) \quad A(x) \leftrightarrow B(x) \wedge C(x)$$
$$A(x) \leftrightarrow \exists y.[R(x,y) \wedge B(y)]$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. TBox-simplification
2. FO-translation
3. Clausification
4. Saturation in ATP

- Apply standard Skolemization and clause normal form transformations

Clause types

T1. $\neg A(x) \vee B(x)$

T2. $\neg B(x) \vee \neg C(x) \vee A(x)$

Clausification

$(\Leftarrow)$ $A(x) \leftrightarrow B(x) \wedge C(x)$

$A(x) \leftrightarrow \exists y.[R(x,y) \wedge B(y)]$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Apply standard Skolemization and clause normal form transformations

Clause types

T1. $\neg A(x) \vee B(x)$

T2. $\neg B(x) \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee R(x, f_A(x))$

Clausification

$$A(x) \leftrightarrow B(x) \wedge C(x)$$

$$(\Rightarrow) \quad A(x) \leftrightarrow \exists y.[R(x,y) \wedge B(y)]$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Apply standard Skolemization and clause normal form transformations

Clause types

T1. $\neg A(x) \vee B(x)$

T2. $\neg B(x) \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee R(x, f_A(x))$

T4. $\neg A(x) \vee B(f_A(x))$

Clausification

$$A(x) \leftrightarrow B(x) \wedge C(x)$$

$$(\Rightarrow) \quad A(x) \leftrightarrow \exists y.[R(x, y) \wedge B(y)]$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Apply standard Skolemization and clause normal form transformations

Clause types

T1. $\neg A(x) \vee B(x)$

T2. $\neg B(x) \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee R(x, f_A(x))$

T4. $\neg A(x) \vee B(f_A(x))$

T5. $\neg R(x, y) \vee \neg B(y) \vee A(x)$

Clausification

$$A(x) \leftrightarrow B(x) \wedge C(x)$$

$$(\Leftarrow) \quad A(x) \leftrightarrow \exists y.[R(x, y) \wedge B(y)]$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Consider all possible inferences between clauses

Clause types

T1. $\neg A(x) \vee B(x)$

T2. $\neg B(x) \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee R(x, f_A(x))$

T4. $\neg A(x) \vee B(f_A(x))$

T5. $\neg R(x, y) \vee \neg B(y) \vee A(x)$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

Resolution

$$\frac{C \vee \underline{A} \quad D \vee \neg \underline{B}}{(C \vee D)\sigma}$$

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

Clause types

T1. $\neg A(x) \vee B(x)$

T2. $\neg B(x) \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee R(x, f_A(x))$

T4. $\neg A(x) \vee B(f_A(x))$

T5. $\neg R(x, y) \vee \neg B(y) \vee A(x)$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

Resolution

$$\frac{C \vee \underline{A} \quad D \vee \neg \underline{B}}{(C \vee D)\sigma}$$

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

Clause types

$\Rightarrow$ T1. $\neg \underline{A(x)} \vee B(x)$

T2. $\neg \underline{B(x)} \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee \underline{R(x, f_A(x))}$

$\Rightarrow$ T4. $\neg A(x) \vee \underline{B(f_A(x))}$

T5. $\underline{\neg R(x, y)} \vee \neg B(y) \vee A(x)$

Possible Inference

$$\frac{\neg A(x) \vee \underline{B(f_A(x))} \quad \underline{\neg B(x)} \vee C(x)}{\neg A(x) \vee C(f_A(x)) \;\Rightarrow\; \text{T4}}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

Resolution

$$\frac{C \vee \underline{A} \quad D \vee \neg \underline{B}}{(C \vee D)\sigma}$$

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

Clause types

T1. $\neg \underline{A(x)} \vee B(x)$

⇒T2. $\neg \underline{B(x)} \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee \underline{R(x, f_A(x))}$

⇒T4. $\neg A(x) \vee \underline{B(f_A(x))}$

T5. $\underline{\neg R(x, y)} \vee \neg B(y) \vee A(x)$

T6. $\neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(f_A(x))$

Possible Inference

$$\frac{\neg A(x) \vee \underline{B(f_A(x))} \quad \underline{\neg B(x)} \vee \neg C(x) \vee D(x)}{\neg A(x) \vee \neg C(f_A(x)) \vee D(f_A(x)) \Rightarrow \text{T6}}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

Resolution

$$\frac{C \vee \underline{A} \quad D \vee \neg \underline{B}}{(C \vee D)\sigma}$$

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

Clause types

T1. $\neg \underline{A(x)} \vee B(x)$

T2. $\neg \underline{B(x)} \vee \neg C(x) \vee A(x)$

⇒T3. $\neg A(x) \vee \underline{R(x, f_A(x))}$

T4. $\neg A(x) \vee \underline{B(f_A(x))}$

⇒T5. $\neg \underline{R(x, y)} \vee \neg B(y) \vee A(x)$

T6. $\neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(f_A(x))$

T7. $\neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(x)$

Possible Inference

$$\frac{\neg A(x) \vee \underline{R(x, f_A(x))} \quad \neg \underline{R(x, y)} \vee \neg B(y) \vee C(x)}{\neg A(x) \vee \neg B(f_A(x)) \vee C(x)} \Rightarrow \text{T7}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

Resolution

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

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

Clause types

T1. $\neg \underline{A(x)} \vee B(x)$

T2. $\neg \underline{B(x)} \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee \underline{R(x, f_A(x))}$

⇒T4. $\neg A(x) \vee \underline{B(f_A(x))}$

T5. $\underline{\neg R(x, y)} \vee \neg B(y) \vee A(x)$

⇒T6. $\neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(f_A(x))$

T7. $\neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(x)$

Possible Inference

$$\frac{\neg A(x) \vee \underline{B(f_A(x))} \qquad \neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(f_A(x))}{\neg A(x) \vee C(f_A(x)) \;\Rightarrow\; \text{T4}}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

Resolution

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

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

Clause types

T1. $\neg \underline{A(x)} \vee B(x)$

T2. $\neg \underline{B(x)} \vee \neg C(x) \vee A(x)$

T3. $\neg A(x) \vee \underline{R(x, f_A(x))}$

$\Rightarrow$ T4. $\neg A(x) \vee \underline{B(f_A(x))}$

T5. $\underline{\neg R(x, y)} \vee \neg B(y) \vee A(x)$

T6. $\neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(f_A(x))$

$\Rightarrow$ T7. $\neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(x)$

Possible Inference

$$\frac{\neg A(x) \vee \underline{B(f_A(x))} \qquad \neg A(x) \vee \neg \underline{B(f_A(x))} \vee C(x)}{\neg A(x) \vee C(x)} \Rightarrow \text{T1}$$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Since there are at most finitely many clauses of types T1 – T7, the saturation procedure is guaranteed to terminate

Clause types

T1. $\neg\underline{A(x)} \lor B(x)$

T2. $\neg\underline{B(x)} \lor \neg C(x) \lor A(x)$

T3. $\neg A(x) \lor \underline{R(x, f_A(x))}$

T4. $\neg A(x) \lor \underline{B(f_A(x))}$

T5. $\underline{\neg R(x, y)} \lor \neg B(y) \lor A(x)$

T6. $\neg A(x) \lor \neg\underline{B(f_A(x))} \lor C(f_A(x))$

T7. $\neg A(x) \lor \neg\underline{B(f_A(x))} \lor C(x)$

A Resolution Decision Procedure for $\mathcal{EL}$

1. **TBox**-simplification
2. FO-translation
3. Clausification
4. Saturation in **ATP**

- Subsumption quieries are handled in a similar way together with **TBox**

Subsumption Query

$?- \ C_1 \sqsubseteq C_2$

$?- \ \text{Father} \sqsubseteq \text{Parent}$

$?- \ \text{Grandfather} \sqsubseteq \text{Father}$

THE GENERAL RECIPE

- Saturation-Based decision procedures have been invented by Joyner Jr. (1976)
- The general strategy can be described as follows:
  1. Define an appropriate clause class for the target fragment
  2. Insure that this class is closed under inferences
  3. Demonstrate that the class is finite for a fixed signature
- Many decision procedures based on this principle have been found later on.

  (clause classes ($\mathcal{E}$, $\mathcal{S}^+$ $\mathcal{E}^+$,...) (Fermüller, Leitsch, Tammet & Zamov, 1993), modal logics (Schmidt, 1997; Hustadt, 1999; Hustadt, de Nivelle & Schmidt, 2000), fragments of first-order logic (Bachmair, Ganzinger & Waldmann, 1993; Ganzinger & de Nivelle, 1999; de Nivelle & Pratt-Hartmann, 2001)

THE GENERAL RECIPE

- Saturation-Based decision procedures have been invented by Joyner Jr. (1976)
- The general strategy can be described as follows:
  1. Define an appropriate clause class for the target fragment
  2. Insure that this class is closed under inferences
  3. Demonstrate that the class is finite for a fixed signature
- Many decision procedures based on this principle have been found later on.

  (clause classes ($\mathcal{E}$, $\mathcal{S}^+$ $\mathcal{E}^+$,...) (Fermüller et al., 1993), modal logics (Schmidt, 1997; Hustadt, 1999; Hustadt et al., 2000), fragments of first-order logic (Bachmair et al., 1993; Ganzinger & de Nivelle, 1999; de Nivelle & Pratt-Hartmann, 2001)

The General Recipe

- Saturation-Based decision procedures have been invented by Joyner Jr. (1976)
- The general strategy can be described as follows:
  1. Define an appropriate clause class for the target fragment
  2. Insure that this class is closed under inferences
  3. Demonstrate that the class is finite for a fixed signature
- Many decision procedures based on this principle have been found later on.

  (clause classes ($\mathcal{E}$, $\mathcal{S}^{+}$ $\mathcal{E}^{+}$,...) (Fermüller et al., 1993), modal logics (Schmidt, 1997; Hustadt, 1999; Hustadt et al., 2000), fragments of first-order logic (Bachmair et al., 1993; Ganzinger & de Nivelle, 1999; de Nivelle & Pratt-Hartmann, 2001)

Novel Techniques

- We extend the approach of Joyner Jr. (1976) using several techniques and refinements known in automated theorem proving, namely:
  1. The general notion of redundancy introduced by Bachmair & Ganzinger (1990, 1994)
  2. Structure-preserving transformations
  3. Dynamic renaming based on semantical properties
- This allows one to design custom simplification rules to improve termination behaviour, which results in that:
  - more expressive fragments can be handled
  - in a modular way
  - the procedures are of optimal complexity

NOVEL TECHNIQUES

- We extend the approach of Joyner Jr. (1976) using several techniques and refinements known in automated theorem proving, namely:
  1. The general notion of redundancy introduced by Bachmair & Ganzinger (1990, 1994)
  2. Structure-preserving transformations
  3. Dynamic renaming based on semantical properties
- This allows one to design custom simplification rules to improve termination behaviour, which results in that:
  - more expressive fragments can be handled
  - in a modular way
  - the procedures are of optimal complexity

Novel Techniques

- We extend the approach of Joyner Jr. (1976) using several techniques and refinements known in automated theorem proving, namely:
  1. The general notion of redundancy introduced by Bachmair & Ganzinger (1990, 1994)
  2. Structure-preserving transformations
  3. Dynamic renaming based on semantical properties
- This allows one to design custom simplification rules to improve termination behaviour, which results in that:
  - more expressive fragments can be handled
  - in a modular way
  - the procedures are of optimal complexity

Novel Techniques

- We extend the approach of Joyner Jr. (1976) using several techniques and refinements known in automated theorem proving, namely:
  1. The general notion of redundancy introduced by Bachmair & Ganzinger (1990, 1994)
  2. Structure-preserving transformations
  3. Dynamic renaming based on semantical properties
- This allows one to design custom simplification rules to improve termination behaviour, which results in that:
  - more expressive fragments can be handled
  - in a modular way
  - the procedures are of optimal complexity

Novel Techniques

- We extend the approach of Joyner Jr. (1976) using several techniques and refinements known in automated theorem proving, namely:
  1. The general notion of redundancy introduced by Bachmair & Ganzinger (1990, 1994)
  2. Structure-preserving transformations
  3. Dynamic renaming based on semantical properties
- This allows one to design custom simplification rules to improve termination behaviour, which results in that:
  - more expressive fragments can be handled
  - in a modular way
  - the procedures are of optimal complexity

Outline

The Guarded Fragment

- Was introduced by Andréka, van Benthem & Németi (1996, 1998) to transfer good computational properties of modal logics to first-order level

The Basic Description Logic and its First-Order Variant

$$\mathcal{ALC} ::= A \mid C_1 \sqcap C_2 \mid \neg C_1 \mid \exists R.C_1 .$$
$$F(\mathcal{ALC}) ::= A(x) \mid C_1(x) \wedge C_2(x) \mid \neg C_1(x) \mid \exists y.[R(x,y) \wedge C_1(y)] .$$

- The range of quantified variables is bounded by atoms-guards

The Guarded Fragment

$$\mathcal{GF} ::= A(\vec{x}) \mid F_1 \wedge F_2 \mid \neg F_1 \mid \exists \vec{y}.[G(\vec{x},\vec{y}) \wedge F_1(\vec{x},\vec{y})] .$$

- $\mathcal{GF}$ was shown to be decidable by resolution in de Nivelle (1998); de Nivelle & de Rijke (2003)

THE GUARDED FRAGMENT

- Was introduced by Andréka et al. (1996, 1998) to transfer good computational properties of modal logics to first-order level

THE BASIC DESCRIPTION LOGIC AND ITS FIRST-ORDER VARIANT

$$\mathcal{ALC} ::= A \mid C_1 \sqcap C_2 \mid \neg C_1 \mid \exists R.C_1 \,.$$
$$F(\mathcal{ALC}) ::= A(x) \mid C_1(x) \wedge C_2(x) \mid \neg C_1(x) \mid \exists y.[R(x,y) \wedge C_1(y)] \,.$$

- The range of quantified variables is bounded by atoms-guards

THE GUARDED FRAGMENT

$$\mathcal{GF} ::= A(\vec{x}) \mid F_1 \wedge F_2 \mid \neg F_1 \mid \exists \vec{y}.[G(\vec{x},\vec{y}) \wedge F_1(\vec{x},\vec{y})] \,.$$

- $\mathcal{GF}$ was shown to be decidable by resolution in de Nivelle (1998); de Nivelle & de Rijke (2003)

The Guarded Fragment

- Was introduced by Andréka et al. (1996, 1998) to transfer good computational properties of modal logics to first-order level

The Basic Description Logic and its First-Order Variant

$$\mathcal{ALC} ::= A \mid C_1 \sqcap C_2 \mid \neg C_1 \mid \exists R.C_1 \,.$$
$$F(\mathcal{ALC}) ::= A(x) \mid C_1(x) \wedge C_2(x) \mid \neg C_1(x) \mid \exists y.[R(x,y) \wedge C_1(y)] \,.$$

- The range of quantified variables is bounded by atoms-guards

The Guarded Fragment

$$\mathcal{GF} ::= A(\vec{x}) \mid F_1 \wedge F_2 \mid \neg F_1 \mid \exists \vec{y}.[G(\vec{x},\vec{y}) \wedge F_1(\vec{x},\vec{y})] \,.$$

- $\mathcal{GF}$ was shown to be decidable by resolution in de Nivelle (1998); de Nivelle & de Rijke (2003)

Two-Variable and Monadic Fragments

- Other useful fragments studied before include:

The Two-Variable Fragment

$$\mathcal{FO}^2 ::= A[x,y] \mid T_1 \wedge T_2 \mid \neg T_1 \mid \exists y . T_1[x,y] \ .$$

The (Full) Monadic Fragment

$$\mathcal{MF} ::= A[x] \mid M_1[x] \cdot \{x/f(x)\} \mid M_1 \wedge M_2 \mid \neg M_1 \mid \exists y . M_1 \ .$$

- Decidability of the two-variable and monadic fragments by resolution was known before

Two-Variable and Monadic Fragments

- Other useful fragments studied before include:

The Two-Variable Fragment

$\mathcal{FO}^2 ::= \mathsf{A}[x, y] \mid T_1 \wedge T_2 \mid \neg T_1 \mid \exists y . T_1[x, y]$.

The (Full) Monadic Fragment

$\mathcal{MF} ::= \mathsf{A}[x] \mid M_1[x] \cdot \{x/f(x)\} \mid M_1 \wedge M_2 \mid \neg M_1 \mid \exists y . M_1$.

- Decidability of the two-variable and monadic fragments by resolution was known before

Combinations of Decidable Fragments

- We studied combinations of fragments $\mathcal{GF}$, $\mathcal{FO}^2$ and $\mathcal{MF}$ in which their constructors are joint:

Example

$$\forall xy.[\mathrm{Nat}(x) \wedge \mathrm{Nat}(y) \rightarrow \underbrace{\exists z.(\mathrm{Sum}(x,y,z) \wedge \mathrm{Nat}(z))}_{\mathrm{Summable}(x,y)\in\mathcal{GF}}] \in \mathcal{GF}|\mathcal{FO}^2$$

- Results:
  1. Every combination of these fragments is decidable by resolution
  2. Retains the complexity of its components (i.e. the procedures are optimal)
  3. Decidability results, however, do not hold with equality

Combinations of Decidable Fragments

- We studied combinations of fragments $\mathcal{GF}$, $\mathcal{FO}^2$ and $\mathcal{MF}$ in which their constructors are joint:

Example

$$\forall xy.[\mathrm{Nat}(x) \wedge \mathrm{Nat}(y) \rightarrow \underbrace{\exists z.(\mathrm{Sum}(x,y,z) \wedge \mathrm{Nat}(z))}_{\mathrm{Summable}(x,y) \in \mathcal{GF}}] \in \mathcal{GF}|\mathcal{FO}^2$$

- Results:
  1. Every combination of these fragments is decidable by resolution
  2. Retains the complexity of its components (i.e. the procedures are optimal)
  3. Decidability results, however, do not hold with equality

Extensions of the Guarded Fragment

- $\mathcal{GF}$ captures only relatively simple description logics $\mathcal{ALCIH}$
- Functionality, Transitivity and Nominals are not expressible in $\mathcal{GF}$.
- We extend the paramodulation-based decision procedure for $\mathcal{GF}_{\simeq}$ (Ganzinger & de Nivelle, 1999) to capture those constructors.

Extensions of the Guarded Fragment

- $\mathcal{GF}$ captures only relatively simple description logics $\mathcal{ALCIH}$
- Functionality, Transitivity and Nominals are not expressible in $\mathcal{GF}$.
- We extend the paramodulation-based decision procedure for $\mathcal{GF}_{\simeq}$ (Ganzinger & de Nivelle, 1999) to capture those constructors.

Extensions of the Guarded Fragment

- $\mathcal{GF}$ captures only relatively simple description logics $\mathcal{ALCIH}$
- Functionality, Transitivity and Nominals are not expressible in $\mathcal{GF}$.
- We extend the paramodulation-based decision procedure for $\mathcal{GF}_{\simeq}$ (Ganzinger & de Nivelle, 1999) to capture those constructors.

The Guarded Fragment with Constants

- Nominals can be expressed using a guarded formula with constant: $\mathsf{O} \sqsubseteq \{c\} \;\Rightarrow\; \forall x.[\mathsf{O}(x) \rightarrow x \simeq c]$
- We found two paramodulation-based procedures for $\mathcal{GF}_{\simeq}$ with constants:
  1. Using elimination of constants proposed by Grädel (1999) combined with elimination of equational guards, and
  2. A procedure that handles constants directly
- Both procedures have theoretically optimal complexity both with bounded and unbounded number of variable names (EXPTIME and 2EXPTIME respectively).

The Guarded Fragment with Constants

- Nominals can be expressed using a guarded formula with constant: $O \sqsubseteq \{c\} \;\Rightarrow\; \forall x.[O(x) \rightarrow x \simeq c]$
- We found two paramodulation-based procedures for $\mathcal{GF}_{\simeq}$ with constants:
  1. Using elimination of constants proposed by Grädel (1999) combined with elimination of equational guards, and
  2. A procedure that handles constants directly
- Both procedures have theoretically optimal complexity both with bounded and unbounded number of variable names (EXPTIME and 2EXPTIME respectively).

THE GUARDED FRAGMENT WITH CONSTANTS

- Nominals can be expressed using a guarded formula with constant: $\mathsf{O} \sqsubseteq \{c\} \;\Rightarrow\; \forall x.[\mathsf{O}(x) \rightarrow x \simeq c]$
- We found two paramodulation-based procedures for $\mathcal{GF}_{\simeq}$ with constants:
  1. Using elimination of constants proposed by Grädel (1999) combined with elimination of equational guards, and
  2. A procedure that handles constants directly
- Both procedures have theoretically optimal complexity both with bounded and unbounded number of variable names (EXPTIME and 2EXPTIME respectively).

The Guarded Fragment with Functionality

- Functionality of binary relations is not expressible in $\mathcal{GF}_{\simeq}$: $\forall xyz.[F(x,y) \wedge F(x,z) \rightarrow y \simeq z]$
- Moreover, the guarded fragment with functionality is undecidable (Grädel, 1999)
- We consider a syntactical restriction $\mathcal{GF}_{\simeq}[FG]$ of $\mathcal{GF}_{\simeq}$, when functional relations may appear in guards only.
- Results:
  - $\mathcal{GF}_{\simeq}[FG]$ is decidable by paramodulation with a custom simplification rule:
  - complexity of the procedure is optimal (EXPTIME/2EXPTIME)

The Guarded Fragment with Functionality

- Functionality of binary relations is not expressible in $\mathcal{GF}_{\simeq}$: $\forall xyz.[F(x,y) \wedge F(x,z) \rightarrow y \simeq z]$
- Moreover, the guarded fragment with functionality is undecidable (Grädel, 1999)
- We consider a syntactical restriction $\mathcal{GF}_{\simeq}[FG]$ of $\mathcal{GF}_{\simeq}$, when functional relations may appear in guards only.
- Results:
  - $\mathcal{GF}_{\simeq}[FG]$ is decidable by paramodulation with a custom simplification rule;
  - complexity of the procedure is optimal (EXPTIME/2EXPTIME)

The Guarded Fragment with Functionality

- Functionality of binary relations is not expressible in $\mathcal{GF}_{\simeq}$:
  $\forall xyz.[\mathsf{F}(x, y) \wedge \mathsf{F}(x, z) \rightarrow y \simeq z]$
- Moreover, the guarded fragment with functionality is undecidable (Grädel, 1999)
- We consider a syntactical restriction $\mathcal{GF}_{\simeq}[FG]$ of $\mathcal{GF}_{\simeq}$, when functional relations may appear in guards only.
- Results:
  - $\mathcal{GF}_{\simeq}[FG]$ is decidable by paramodulation with a custom simplification rule:
  - complexity of the procedure is optimal (EXPTIME/2EXPTIME)

The Guarded Fragment with Functionality

- Functionality of binary relations is not expressible in $\mathcal{GF}_{\simeq}$: $\forall xyz.[\mathrm{F}(x, y) \wedge \mathrm{F}(x, z) \rightarrow y \simeq z]$
- Moreover, the guarded fragment with functionality is undecidable (Grädel, 1999)
- We consider a syntactical restriction $\mathcal{GF}_{\simeq}[FG]$ of $\mathcal{GF}_{\simeq}$, when functional relations may appear in guards only.
- Results:
  - $\mathcal{GF}_{\simeq}[FG]$ is decidable by paramodulation with a custom simplification rule:

    Literal Projection

    $$\frac{[C \vee f(x) \simeq g(x)]}{\begin{array}{c} C \vee A(x) \\ \neg A(x) \vee f(x) \simeq g(x) \end{array}}$$

  - complexity of the procedure is optimal (EXPTIME/2EXPTIME)

From Functionality to Counting

- Our procedure for $\mathcal{GF}_{\simeq}[FG]$ can be extended for counting restrictions: $\forall x.\exists y^{\leq n}.R(x, y)$ and $\forall x.\exists y^{\geq n}.R(x, y)$
- Gives the same complexity as for $\mathcal{GF}_{\simeq}[FG]$ assuming unary coding of numbers
- An alternative procedure which is optimal for binary coding of numbers has been described in Kazakov (2004):

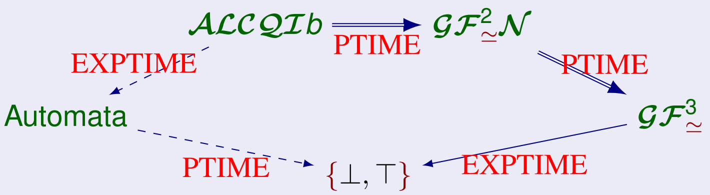
From Functionality to Counting

- Our procedure for $\mathcal{GF}_{\simeq}[FG]$ can be extended for counting restrictions: $\forall x.\exists y^{\leq n}.R(x, y)$ and $\forall x.\exists y^{\geq n}.R(x, y)$
- Gives the same complexity as for $\mathcal{GF}_{\simeq}[FG]$ assuming unary coding of numbers
- An alternative procedure which is optimal for binary coding of numbers has been described in Kazakov (2004):

From Functionality to Counting

- Our procedure for $\mathcal{GF}_{\simeq}[FG]$ can be extended for counting restrictions: $\forall x.\exists y^{\leq n}.\mathrm{R}(x, y)$ and $\forall x.\exists y^{\geq n}.\mathrm{R}(x, y)$
- Gives the same complexity as for $\mathcal{GF}_{\simeq}[FG]$ assuming unary coding of numbers
- An alternative procedure which is optimal for binary coding of numbers has been described in Kazakov (2004):

Polynomial Translation from $\mathcal{GF}^2_{\simeq}\mathcal{N}$ to $\mathcal{GF}^3_{\simeq}$



Simple Compositional Axioms

- Many useful properties are expressible using:

Simple Compositional Axioms

$$S \circ T \sqsubseteq H_1 \sqcup \cdots \sqcup H_n$$

Temporal Properties

If (x before y) and (y before z) then (x before z)

Orderings

If ($x<y$) and ($y<z$) then ($x<z$)

Topological and Distance Relations

(x is a part of y) $\circ$ (y is located in z) $\rightarrow$ (x is located in z)
(x distance $\geq$ 5 y) $\circ$ (y distance $<$ 2 z) $\rightarrow$ (x distance $\geq$ 3 z)

Simple Compositional Axioms

- Many useful properties are expressible using:

Simple Compositional Axioms

$$S \circ T \sqsubseteq H_1 \sqcup \cdots \sqcup H_n$$

Temporal Properties

If (x before y) and (y before z) then (x before z)

Orderings

If ($x<y$) and ($y<z$) then ($x<z$)

Topological and Distance Relations

(x is a part of y) $\circ$ (y is located in z) $\rightarrow$ (x is located in z)

(x distance $\geq$ 5 y) $\circ$ (y distance $<$ 2 z) $\rightarrow$ (x distance $\geq$ 3 z)

Simple Compositional Axioms

- Many useful properties are expressible using:

Simple Compositional Axioms

$$\mathsf{S} \circ \mathsf{T} \sqsubseteq \mathsf{H}_1 \sqcup \cdots \sqcup \mathsf{H}_n$$

Temporal Properties

If (x before y) and (y before z) then (x before z)

Orderings

If ($x < y$) and ($y < z$) then ($x < z$)

Topological and Distance Relations

(x is a part of y) $\circ$ (y is located in z) $\rightarrow$ (x is located in z)

(x distance $\geq$ 5 y) $\circ$ (y distance $<$ 2 z) $\rightarrow$ (x distance $\geq$ 3 z)

Simple Compositional Axioms

- Many useful properties are expressible using:

Simple Compositional Axioms

$$\mathsf{S} \circ \mathsf{T} \sqsubseteq \mathsf{H}_1 \sqcup \cdots \sqcup \mathsf{H}_n$$

Temporal Properties

If (x before y) and (y before z) then (x before z)

Orderings

If ($x{<}y$) and ($y{<}z$) then ($x{<}z$)

Topological and Distance Relations

(x is a part of y) $\circ$ (y is located in z) $\rightarrow$ (x is located in z)
(x distance $\geq$ 5 y) $\circ$ (y distance $<$ 2 z) $\rightarrow$ (x distance $\geq$ 3 z)

Complex Compositional Axioms

Complex Compositional Axioms

$$S \circ T \sqsubseteq H_1 \sqcup \cdots \sqcup H_n$$

Region Connection Calculi RCC5, RCC8

$$\mathrm{TPPI} \circ \mathrm{NTPP} \sqsubseteq \mathrm{PO} \sqcup \mathrm{TPP} \sqcup \mathrm{NTPP}$$

Allen's (1983) Interval Algebra

x before y

x meets y

x overlaps y

x starts y

x during y

x finishes y

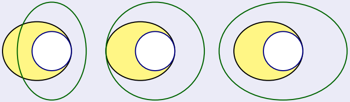
Complex Compositional Axioms

Complex Compositional Axioms

$$\mathsf{S} \circ \mathsf{T} \sqsubseteq \mathsf{H}_1 \sqcup \cdots \sqcup \mathsf{H}_n$$

Region Connection Calculi RCC5, RCC8

$$\mathsf{TPPI} \circ \mathsf{NTPP} \sqsubseteq \mathsf{PO} \sqcup \mathsf{TPP} \sqcup \mathsf{NTPP}$$



Allen's (1983) Interval Algebra

x before y

x meets y

x overlaps y

x starts y

x during y

x finishes y

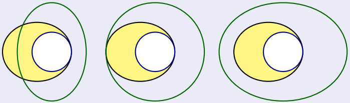
Complex Compositional Axioms

Complex Compositional Axioms

$$S \circ T \sqsubseteq H_1 \sqcup \cdots \sqcup H_n$$

Region Connection Calculi RCC5, RCC8

$$\mathrm{TPPI} \circ \mathrm{NTPP} \sqsubseteq \mathrm{PO} \sqcup \mathrm{TPP} \sqcup \mathrm{NTPP}$$



Allen's (1983) Interval Algebra

x before *y*

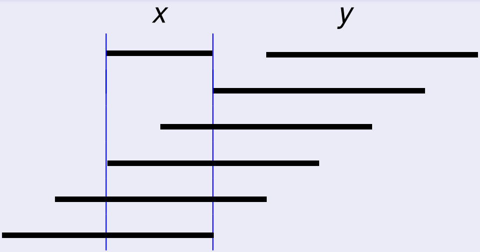
x meets *y*

x overlaps *y*

x starts *y*

x during *y*

x finishes *y*



Theories over Compositional Axioms

- Applications:
  1. (Interval) temporal reasoning
  2. Medical informatics, in particular, anatomical ontologies
  3. Qualitative and quantitative spatial reasoning (GIS)
  4. ...
- Integration into DLs is highly demanded

Theories over Compositional Axioms

- Applications:
  1. (Interval) temporal reasoning
  2. Medical informatics, in particular, anatomical ontologies
  3. Qualitative and quantitative spatial reasoning (GIS)
  4. ...
- Integration into **DL**s is highly demanded

The Guarded Fragment with Transitive Guards

- Transitivity $T \circ T \sqsubseteq T$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger, Meyer & Veanes, 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

THE GUARDED FRAGMENT WITH TRANSITIVE GUARDS

- Transitivity $T \circ T \sqsubseteq T$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger et al., 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

The Guarded Fragment with Transitive Guards

- Transitivity $T \circ T \sqsubseteq T$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger et al., 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

The Guarded Fragment with Transitive Guards

- Transitivity $\mathsf{T} \circ \mathsf{T} \sqsubseteq \mathsf{T}$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger et al., 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

The Guarded Fragment with Transitive Guards

- Transitivity $T \circ T \sqsubseteq T$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger et al., 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

The Guarded Fragment with Transitive Guards

- Transitivity $\mathsf{T} \circ \mathsf{T} \sqsubseteq \mathsf{T}$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger et al., 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

Transitive Closure

$$\frac{\neg(x\mathsf{T}y) \vee \neg\alpha(x) \vee \beta(y)}{\begin{array}{c}\neg(x\mathsf{T}y) \vee \neg\alpha(x) \vee u_\alpha^{\mathsf{T}}(y) \\ \neg(x\mathsf{T}y) \vee \neg u_\alpha^{\mathsf{T}}(x) \vee u_\alpha^{\mathsf{T}}(y) \\ \neg u_\alpha^{\mathsf{T}}(y) \vee \beta(y)\end{array}}$$

The Guarded Fragment with Transitive Guards

- Transitivity $T \circ T \sqsubseteq T$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger et al., 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

Transitive Closure

$$\frac{\neg(xTy) \vee \neg\alpha(x) \vee \beta(y)}{\begin{array}{l}\neg(xTy) \vee \neg\alpha(x) \vee u^T_\alpha(y) \\ \neg(xTy) \vee \neg u^T_\alpha(x) \vee u^T_\alpha(y) \\ \neg u^T_\alpha(y) \vee \beta(y)\end{array}}$$

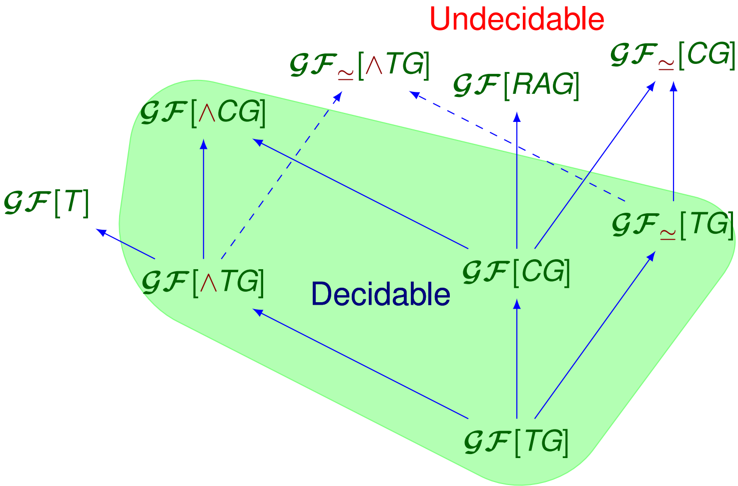
The Guarded Fragment with Transitive Guards

- Transitivity $T \circ T \sqsubseteq T$ is the simplest compositional axiom
- The guarded fragment with transitivity is undecidable (Grädel, 1999; Ganzinger et al., 1999)
  - We have sharpened these results and demonstrated that already two transitive relations makes $\mathcal{GF}^2$ undecidable.
- Szwast & Tendera (2001) and later Kieronski (2003) demonstrated that a restriction $\mathcal{GF}[TG]$ is decidable.
- In (Kazakov & de Nivelle, 2004) we obtained the first practical resolution-based decision procedure for $\mathcal{GF}[TG]$.
- Our procedure employs a custom simplification rule:

Transitive Closure

$$\frac{\neg(xTy) \vee \neg\alpha(x) \vee \beta(y)}{\begin{array}{r} \neg(xTy) \vee \neg\alpha(x) \vee u_\alpha^T(y) \\ \neg(xTy) \vee \neg u_\alpha^T(x) \vee u_\alpha^T(y) \\ \neg u_\alpha^T(y) \vee \beta(y) \end{array}}$$

Classification for $\mathcal{GF}$ over Compositional Theories



Outline

How to Implement Saturation-Based Procedures?

- Adopt a theorem prover to your strategy
- Difficult for complicated strategies (which employ non-standard orderings and custom simplification rules)
- Even if implemented, the it is mostly overkill because:
  - the clauses to deal with are usually shallow
  - indexing in theorem provers is not optimized for such clauses
  - most inferences are trivial and can be precomputed

How to Implement Saturation-Based Procedures?

- Adopt a theorem prover to your strategy
- Difficult for complicated strategies (which employ non-standard orderings and custom simplification rules)
- Even if implemented, the it is mostly overkill because:
  - the clauses to deal with are usually shallow
  - indexing in theorem provers is not optimized for such clauses
  - most inferences are trivial and can be precomputed

How to Implement Saturation-Based Procedures?

- Adopt a theorem prover to your strategy
- Difficult for complicated strategies (which employ non-standard orderings and custom simplification rules)
- Even if implemented, the it is mostly overkill because:
  - the clauses to deal with are usually shallow
  - indexing in theorem provers is not optimized for such clauses
  - most inferences are trivial and can be precomputed

Back to DL $\mathcal{EL}$

- The types of inferences we had for **DL** $\mathcal{EL}$ can be written as follows:

Classification of $\mathcal{EL}$-TBox-es

$$T4(A,B,f_A), T1(B,C) \vdash T4(A,C,f_A)$$
$$T4(A,B,f_A), T2(B,C,D) \vdash T6(A,C,f_A,D)$$
$$T3(A,R,f_A), T5(R,B,A) \vdash T7(A,B,f_A,C)$$
$$T4(A,B,f_A), T6(A,B,f_A,C) \vdash T4(A,C,f_A)$$
$$T4(A,B,f_A), T7(A,B,f_A,C) \vdash T1(A,C)$$

Clause types

T1. $\neg A(x) \vee B(x)$
T2. $\neg B(x) \vee \neg C(x) \vee A(x)$
T3. $\neg A(x) \vee R(x, f_A(x))$
T4. $\neg A(x) \vee B(f_A(x))$
T5. $\neg R(x,y) \vee \neg B(y) \vee A(x)$
T6. $\neg A(x) \vee \neg B(f_A(x)) \vee C(f_A(x))$
T7. $\neg A(x) \vee \neg B(f_A(x)) \vee C(x)$

- Conclusions:
  1. The procedure for $\mathcal{EL}$ can be implemented in datalog
  2. Runs in polynomial time

Back to DL $\mathcal{EL}$

- The types of inferences we had for **DL** $\mathcal{EL}$ can be written as follows:

Clause types

T1. $\neg A(x) \vee B(x)$
T2. $\neg B(x) \vee \neg C(x) \vee A(x)$
T3. $\neg A(x) \vee R(x, f_A(x))$
T4. $\neg A(x) \vee B(f_A(x))$
T5. $\neg R(x, y) \vee \neg B(y) \vee A(x)$
T6. $\neg A(x) \vee \neg B(f_A(x)) \vee C(f_A(x))$
T7. $\neg A(x) \vee \neg B(f_A(x)) \vee C(x)$

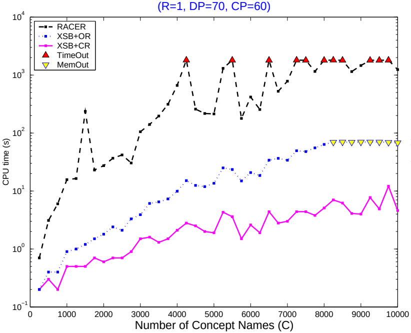
Classification of $\mathcal{EL}$-TBox-es

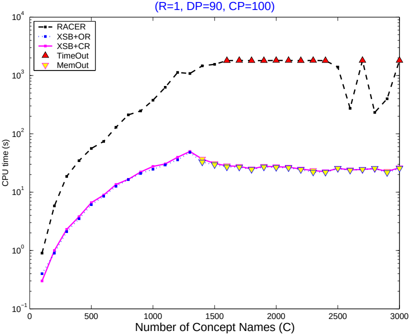
$$\mathrm{T4}(A, B, f_A), \mathrm{T1}(B, C) \vdash \mathrm{T4}(A, C, f_A)$$
$$\mathrm{T4}(A, B, f_A), \mathrm{T2}(B, C, D) \vdash \mathrm{T6}(A, C, f_A, D)$$
$$\mathrm{T3}(A, R, f_A), \mathrm{T5}(R, B, A) \vdash \mathrm{T7}(A, B, f_A, C)$$
$$\mathrm{T4}(A, B, f_A), \mathrm{T6}(A, B, f_A, C) \vdash \mathrm{T4}(A, C, f_A)$$
$$\mathrm{T4}(A, B, f_A), \mathrm{T7}(A, B, f_A, C) \vdash \mathrm{T1}(A, C)$$

- Conclusions:
  1. The procedure for $\mathcal{EL}$ can be implemented in datalog
  2. Runs in polynomial time

Empirical Evaluation

- We have performed a series of tests on randomly generated $\mathcal{EL}$-**TBox**-es (up to 10.000 concepts) using our procedure in XSB-system vs RACER system:





Outline

CONTRIBUTIONS

- We obtained many (un)decidability, complexity results and decision procedures for first-order fragments relevant to knowledge representation languages. Most important:
  1. Polynomial saturation-based decision procedures for $\mathcal{EL}$ and its extensions (most studied in (Baader, Brandt & Lutz, 2005) and new). Empirical evaluation demonstrates that our approach is promising.
  2. Combination of the guarded, two-variable and monadic fragments. Optimal complexity results.
  3. Paramodulation-based decision procedures for extensions of the guarded fragment with constants, functionality and number restrictions. Optimal complexity results.
  4. Full classification of (un)decidability results for the guarded fragment over compositional theories. Saturation-based decision procedures. Optimal complexities.

In Memoriam Harald Ganzinger (1950-2004)

- Most of our the results are based on a theory of saturation-based theorem proving developed by Prof. Harald Ganzinger and would not have been possible without his scientific achievements.

Thank you for your attention

Thank you for your attention!

Allen, J. F. [1983], 'Maintaining knowledge about temporal intervals.', *Commun. ACM* **26**(11), 832–843.

Andréka, H., van Benthem, J. & Németi, I. [1996], Modal languages and bounded fragments of predicate logic, Technical Report ML-1996-03, ILLC.

Andréka, H., van Benthem, J. & Németi, I. [1998], 'Modal languages and bounded fragments of predicate logic', *Journal of Philosophical Logic* **27**, 217–274.

Baader, F. [1996], 'Using automata theory for characterizing the semantics of terminological cycles.', *Ann. Math. Artif. Intell.* **18**(2-4), 175–219.

Baader, F. [2002], Terminological cycles in a description logic with existential restrictions, LTCS-Report LTCS-02-02, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany.

Baader, F. & Hollunder, B. [1991], 'Kris: Knowledge representation and inference system.', *SIGART Bulletin* **2**(3), 8–14.

Baader, F., Brandt, S. & Lutz, C. [2005], Pushing the el envelope, LTCS-Report LTCS-05-01, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany.

Bachmair, L. & Ganzinger, H. [1990], On restrictions of ordered paramodulation with simplification, *in* 'Proceedings of the tenth international conference on Automated deduction', Springer-Verlag New York, Inc., pp. 427–441.

Bachmair, L. & Ganzinger, H. [1994], 'Rewrite-based equational theorem proving with selection and simplification', *Journal of Logic and Computation* **4**(3), 217–247.

Bachmair, L., Ganzinger, H. & Waldmann, U. [1993], Superposition with simplification as a decision procedure for the monadic class with equality, *in* G. Gottlob, A. Leitsch & D. Mundici, eds, 'Computational Logic and Proof Theory, Third Kurt Gödel Colloquium, KGC'93', Vol. 713 of *Lecture Notes in Computer Science*, Springer, Brno, Czech Republic, pp. 83–96.

Brachman, R. J. & Schmolze, J. G. [1985], 'An overview of the KL-ONE knowledge representation system.', *Cognitive Science* **9**(2), 171–216.

de Nivelle, H. [1998], A resolution decision procedure for the guarded fragment, *in* C. Kirchner & H. Kirchner, eds, 'Proceedings of the 15th International Conference on Automated Deduction (CADE-14)', Vol. 1421 of *Lecture Notes in Artificial Intelligence*, Springer Verlag, Lindau, Germany, pp. 191–204.

de Nivelle, H. & de Rijke, M. [2003], 'Deciding the guarded fragments by resolution', *Journal of Symbolic Computation* **35**, 21–58.

de Nivelle, H. & Pratt-Hartmann, I. [2001], A resolution-based decision procedure for the two-variable fragment with equality., *in* T. N. R. Goré, A. Leitsch, ed., 'In: Proc. 1st Int. Joint Conf. on Automated Reasoning (IJCAR-2001)', Vol. 2083 of *Lect. Notes Artif. Intell.*, Springer, Berlin, pp. 211–225.

Fermüller, C., Leitsch, A., Tammet, T. & Zamov, N. [1993], *Resolution Methods for the Decision Problem*, Vol. 679 of *LNAI*, Springer, Berlin, Heidelberg.

Ganzinger, H. & de Nivelle, H. [1999], A superposition decision procedure for the guarded fragment with equality, *in* 'Proc. 14th IEEE Symposium on Logic in Computer Science', IEEE Computer Society Press, pp. 295–305.

Ganzinger, H., Meyer, C. & Veanes, M. [1999], The two-variable guarded fragment with transitive relations, *in* 'Proc. 14th IEEE Symposium on Logic in Computer Science', IEEE Computer Society Press, pp. 24–34.

Grädel, E. [1999], 'On the restraining power of guards', *Journal of Symbolic Logic* **64(4)**, 1719–1742.

Haarslev, V. & Möller, R. [2001], RACER system description, *in* 'IJCAR '01: Proceedings of the First International Joint Conference on Automated Reasoning', Springer-Verlag, pp. 701–706.

Horrocks, I. [1998], Using an expressive description logic: FaCT or fiction?, *in* 'Principles of Knowledge Representation and Reasoning:Proceedings 6th International Conference (KR'98)', Morgan Kauffman, pp. 636–647.

Hustadt, U. [1999], Resolution-Based Decision Procedures for Subclasses of First-Order Logic, PhD thesis, Universität des Saarlandes, Saarbrücken, Germany.

Hustadt, U., de Nivelle, H. & Schmidt, R. A. [2000], 'Resolution-based methods for modal logics', *Logic Journal of the IGPL* **8**(3), 265–292.

Joyner Jr., W. H. [1976], 'Resolution strategies as decision procedures', *Journal of the ACM* **23**(3), 398–417.

Kazakov, Y. [2004], A polynomial translation from the two-variable guarded fragment with number restrictions to the guarded fragment., *in* J. J. Alferes & J. A. Leite, eds, 'JELIA', Vol. 3229 of *Lecture Notes in Computer Science*, Springer, pp. 372–384.

Kazakov, Y. & de Nivelle, H. [2003], Subsumption of concepts in $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete., *in* D. Calvanese, G. D. Giacomo & E. Franconi, eds, 'Description Logics', Vol. 81 of *CEUR Workshop Proceedings*.

Kazakov, Y. & de Nivelle, H. [2004], A resolution decision procedure for the guarded fragment with transitive guards., *in* D. A. Basin & M. Rusinowitch, eds, 'IJCAR', Vol. 3097 of *Lecture Notes in Computer Science*, Springer, pp. 122–136.

Kieronski, E. [2003], The two-variable guarded fragment with transitive guards is 2EXPTIME-hard, *in* A. D. Gordon, ed., 'FoSSaCS', Vol. 2620 of *Lecture Notes in Computer Science*, Springer, pp. 299–312.

Nardi, D. & Brachman, R. J. [2003], An introduction to description logics., *in* F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi & P. F. Patel-Schneider, eds, 'Description Logic Handbook', Cambridge University Press, pp. 1–40.

Schmidt, R. A. [1997], Optimised Modal Translation and Resolution, PhD thesis, Universität des Saarlandes, Saarbrücken, Germany.

Schmidt-Schauß, M. & Smolka, G. [1991], 'Attributive concept descriptions with complements.', *Artif. Intell.* **48**(1), 1–26.

Szwast, W. & Tendera, L. [2001], On the decision problem for the guarded fragment with transitivity., *in* 'LICS', pp. 147–156.