Consequence-Driven Reasoning for Horn-SHIQ Ontologies

Yevgeny Kazakov

Oxford University Computing Laboratory

May 12, 2009

Outline

The Outline of this Talk

- Bio-medical ontologies and reasoning problems
- Description logics $\mathcal{SHIQ}$ and Horn-$\mathcal{SHIQ}$
- Model-building reasoning procedures and their liminations
- Completion-based reasoning procedure for $\mathcal{EL}^+$
- Consequence-based reasoning procedure for Horn-$\mathcal{SHIQ}$
- Impressive experiment results

BIO-MEDICAL ONTOLOGIES

- Galen, Snomed, FMA, NCI, OBO

Bio-Medical Ontologies

- Galen, Snomed, FMA, NCI, OBO
- Define rich schemata of various term:
  - Anatomical system
  - Drugs
  - Chemical processes
  - Clinical procedures
  - . . .

Bio-Medical Ontologies

- Galen, Snomed, FMA, NCI, OBO
- Define rich schemata of various term:
  - Anatomical system
  - Drugs
  - Chemical processes
  - Clinical procedures
  - . . .
- Applications:
  - Medical expert systems [demo]
  - Annotation of the data (patient records,. . .)
  - Both

Reasoning Problems: Informally

- Ontology modeling:
  - Ontology consistency checking: check whether $\mathcal{O}$ has any implicit contradictions
  - Class consistency checking: given a class expression C, check if it is possible to have instances of C

Reasoning Problems: Informally

- Ontology modeling:
  - Ontology consistency checking: check whether $\mathcal{O}$ has any implicit contradictions
  - Class consistency checking: given a class expression C, check if it is possible to have instances of C
- Ontology use:
  - Classification: compute class subsumption hierarchy
  - Query answering: given a class expression C, compute all implicit instances of C

Reasoning Problems: Informally

- Ontology modeling:
  - Ontology consistency checking: check whether $\mathcal{O}$ has any implicit contradictions
  - Class consistency checking: given a class expression C, check if it is possible to have instances of C
- Ontology use:
  - Classification: compute class subsumption hierarchy
  - Query answering: given a class expression C, compute all implicit instances of C
- Classification times for various reasoners (in seconds):

	Classes:	FaCT++	Pellet	HermiT	CEL	CB
GO	20465	15.24	72.02	199.52	1.84	1.17
NCI	27652	6.05	26.47	169.47	5.76	3.57
GalenS	2748	465.35	—	45.72	*n/a*	0.32
GalenB	23136	—	—	—	*n/a*	9.58
Snomed	389472	650.37	—	—	1185.70	49.44

Examples of Axioms in Galen

- $\mathsf{KidneyExamination} \equiv \mathsf{ClinicalAct} \sqcap \exists \mathsf{hasSubprocess}.(\mathsf{ExaminingProcess} \sqcap \exists \mathsf{involves}.\mathsf{Kidney})$

Examples of Axioms in Galen

- $\mathsf{KidneyExamination} \equiv \mathsf{ClinicalAct} \sqcap \exists \mathsf{hasSubprocess}.(\mathsf{ExaminingProcess} \sqcap \exists \mathsf{involves}.\mathsf{Kidney})$
- $\mathsf{BasilarArtery} \sqsubseteq \exists \mathsf{isBranchOf}.\mathsf{VertebralArtery}$

Examples of Axioms in Galen

- $\mathsf{KidneyExamination} \equiv \mathsf{ClinicalAct} \sqcap \exists \mathsf{hasSubprocess}.(\mathsf{ExaminingProcess} \sqcap \exists \mathsf{involves}.\mathsf{Kidney})$
- $\mathsf{BasilarArtery} \sqsubseteq \exists \mathsf{isBranchOf}.\mathsf{VertebralArtery}$
  - Atomic concepts (Classes)

Examples of Axioms in Galen

- $\mathsf{KidneyExamination} \equiv \mathsf{ClinicalAct} \sqcap \exists \mathsf{hasSubprocess}.(\mathsf{ExaminingProcess} \sqcap \exists \mathsf{involves}.\mathsf{Kidney})$
- $\mathsf{BasilarArtery} \sqsubseteq \exists \mathsf{isBranchOf}.\mathsf{VertebralArtery}$
  - Atomic concepts (Classes)
  - Atomic roles (Properties)

Examples of Axioms in Galen

- $\mathsf{KidneyExamination} \equiv \mathsf{ClinicalAct} \sqcap \exists \mathsf{hasSubprocess}.(\mathsf{ExaminingProcess} \sqcap \exists \mathsf{involves}.\mathsf{Kidney})$
- $\mathsf{BasilarArtery} \sqsubseteq \exists \mathsf{isBranchOf}.\mathsf{VertebralArtery}$
  - Atomic concepts (Classes)
  - Atomic roles (Properties)
  - Constructors

Examples of Axioms in Galen

- $\mathsf{KidneyExamination} \equiv \mathsf{ClinicalAct} \sqcap \exists \mathsf{hasSubprocess}.(\mathsf{ExaminingProcess} \sqcap \exists \mathsf{involves}.\mathsf{Kidney})$
- $\mathsf{BasilarArtery} \sqsubseteq \exists \mathsf{isBranchOf}.\mathsf{VertebralArtery}$
  - Atomic concepts (Classes)
  - Atomic roles (Properties)
  - Constructors
- $\mathsf{BodyStructure} \sqcap \exists \mathsf{isContainedIn}.\mathsf{SpinalCanal} \sqsubseteq \exists \mathsf{isStructuralComponentOf}.\mathsf{NervousSystem}$

Examples of Axioms in Galen

- $\mathsf{KidneyExamination} \equiv \mathsf{ClinicalAct} \sqcap \exists \mathsf{hasSubprocess}.(\mathsf{ExaminingProcess} \sqcap \exists \mathsf{involves}.\mathsf{Kidney})$
- $\mathsf{BasilarArtery} \sqsubseteq \exists \mathsf{isBranchOf}.\mathsf{VertebralArtery}$
  - Atomic concepts (Classes)
  - Atomic roles (Properties)
  - Constructors
- $\mathsf{BodyStructure} \sqcap \exists \mathsf{isContainedIn}.\mathsf{SpinalCanal} \sqsubseteq \exists \mathsf{isStructuralComponentOf}.\mathsf{NervousSystem}$
- Reciprocal links:

$$\mathsf{SpinalCanal} \sqsubseteq \exists \mathsf{nonPartitivelyContains}.\mathsf{PeriSpinalSpace}$$

$$\mathsf{PeriSpinalSpace} \sqsubseteq \exists \mathsf{isNonPartitivelyContainedIn}.\mathsf{SpinalCanal}$$

$$\mathsf{nonPartitivelyContains} \sqsubseteq \mathsf{isNonPartitivelyContainedIn}^{-}$$

Syntax and Semantics of $\mathcal{SHIQ}$: Constructors

- Concept constructors:

Atomic Concept	A	$A(x)$
Top Concept	$\top$	$\top$
Bottom Concept	$\bot$	$\top$
Conjunction	$C \sqcap D$	$C(x) \wedge D(x)$
Disjunction	$C \sqcup D$	$C(x) \vee D(x)$
Negation	$\neg C$	$\neg C(x)$
Existential Restriction	$\exists R.C$	$\exists y.(R(x,y) \wedge C(y))$
Universal Restriction	$\forall R.C$	$\forall y.(R(x,y) \rightarrow C(y))$
"At least" Restriction	$\geqslant n\, S.C$	$\exists^{\geq n} y.(S(x,y) \wedge C(y))$
"At most" Restriction	$\leqslant n\, S.C$	$\exists^{\leq n} y.(S(x,y) \wedge C(y))$

Syntax and Semantics of $\mathcal{SHIQ}$: Constructors

- Concept constructors:

Atomic Concept	A	$A(x)$
Top Concept	$\top$	$\top$
Bottom Concept	$\bot$	$\top$
Conjunction	$C \sqcap D$	$C(x) \wedge D(x)$
Disjunction	$C \sqcup D$	$C(x) \vee D(x)$
Negation	$\neg C$	$\neg C(x)$
Existential Restriction	$\exists R.C$	$\exists y.(R(x,y) \wedge C(y))$
Universal Restriction	$\forall R.C$	$\forall y.(R(x,y) \rightarrow C(y))$
"At least" Restriction	$\geqslant n\, S.C$	$\exists^{\geq n} y.(S(x,y) \wedge C(y))$
"At most" Restriction	$\leqslant n\, S.C$	$\exists^{\leq n} y.(S(x,y) \wedge C(y))$

- Role Constructor:

Atomic Role	R	$R(x,y)$
Inverse Role	R^-	$R(y,x)$

Syntax and Semantics of $\mathcal{SHIQ}$: Axioms

- Concept axioms:

Concept Inclusion	$C \sqsubseteq D$	$\forall x.(C(x) \rightarrow D(x))$
Concept Equivalence	$C \equiv D$	$\forall x.(C(x) \leftrightarrow D(x))$

Syntax and Semantics of $\mathcal{SHIQ}$: Axioms

- Concept axioms:

Concept Inclusion	$C \sqsubseteq D$	$\forall x.(C(x) \rightarrow D(x))$
Concept Equivalence	$C \equiv D$	$\forall x.(C(x) \leftrightarrow D(x))$

- Role axioms:

Role Inclusion	$R_1 \sqsubseteq R_2$	$\forall xy.(R_1(x,y) \rightarrow R_2(x,y))$
Transitivity	$Tra(R)$	$\forall xyz.(R(x,y) \wedge R(x,y) \rightarrow R(x,z))$
Functionality	$Fun(S)$	$\forall xyz.(S(x,y) \wedge S(x,z) \rightarrow y \simeq z)$

Syntax and Semantics of $\mathcal{SHIQ}$: Axioms

- Concept axioms:

Concept Inclusion	$C \sqsubseteq D$	$\forall x.(C(x) \rightarrow D(x))$
Concept Equivalence	$C \equiv D$	$\forall x.(C(x) \leftrightarrow D(x))$

- Role axioms:

Role Inclusion	$R_1 \sqsubseteq R_2$	$\forall xy.(R_1(x,y) \rightarrow R_2(x,y))$
Transitivity	$Tra(R)$	$\forall xyz.(R(x,y) \wedge R(x,y) \rightarrow R(x,z))$
Functionality	$Fun(S)$	$\forall xyz.(S(x,y) \wedge S(x,z) \rightarrow y \simeq z)$

- Assertions (not considered in this talk):

Concept Assertion	$C(a)$	$C(a)$
Role Assertion	$R(a,b)$	$R(a,b)$

$\mathcal{SHIQ}$ Reasoning Problems Formalized

- Below:
  - $\mathcal{O}$ is a ($\mathcal{SHIQ}$) ontology,
  - α an axiom,
  - C a concept,
  - $\mathcal{I}$ a (first-order) interpretation
- axiom entailment: $\mathcal{O} \models \alpha$ iff $\forall \mathcal{I}.(\mathcal{I} \models \mathcal{O} \Rightarrow \mathcal{I} \models \alpha)$
- ontology consistency:
  given $\mathcal{O}$, check whether $\mathcal{O} \models \top \sqsubseteq \bot$
- concept consistency:
  given $\mathcal{O}$ and C, check whether $\mathcal{O} \models C \sqsubseteq \bot$
- classification:
  given $\mathcal{O}$, compute all pairs $\langle A, B \rangle$ such that $\mathcal{O} \models A \sqsubseteq B$
- instance retrieval:
  given $\mathcal{O}$ and C, compute all a such that $\mathcal{O} \models C(a)$

HORN-$\mathcal{SHIQ}$

- A restriction of $\mathcal{SHIQ}$ that can be translated to the Horn fragment of First-Order logic

HORN-$\mathcal{SHIQ}$

- A restriction of $\mathcal{SHIQ}$ that can be translated to the Horn fragment of First-Order logic
- Originally considered in [Hustadt et al., 2007] due to tractable data complexity

HORN-$\mathcal{SHIQ}$

- A restriction of $\mathcal{SHIQ}$ that can be translated to the Horn fragment of First-Order logic
- Originally considered in [Hustadt et al., 2007] due to tractable data complexity
- No positive occurrences of:
  - $C \sqcup D$ and
  - $\leqslant n\, R.C$ when $n \geq 2$.

HORN-$\mathcal{SHIQ}$

- A restriction of $\mathcal{SHIQ}$ that can be translated to the Horn fragment of First-Order logic
- Originally considered in [Hustadt et al., 2007] due to tractable data complexity
- No positive occurrences of:
  - $C \sqcup D$ and
  - $\leqslant n\, R.C$ when $n \geq 2$.
- No negative occurrences of:
  - $\neg C$,
  - $\forall R.C$,
  - $\geqslant n\, R.C$ when $n \geq 2$
  - $\leqslant m\, R.C$.

HORN-$\mathcal{SHIQ}$

- A restriction of $\mathcal{SHIQ}$ that can be translated to the Horn fragment of First-Order logic
- Originally considered in [Hustadt et al., 2007] due to tractable data complexity
- No positive occurrences of:
  - $C \sqcup D$ and
  - $\leqslant n\, R.C$ when $n \geq 2$.
- No negative occurrences of:
  - $\neg C$,
  - $\forall R.C$,
  - $\geqslant n\, R.C$ when $n \geq 2$
  - $\leqslant m\, R.C$.
- For example:
  - $A \sqsubseteq \forall R.(\neg B)$, $\exists R^{-}.A \sqsubseteq \leqslant 1\, R.(B \sqcup C)$ are OK
  - $A \sqsubseteq B \sqcup C$, $\forall R.A \sqsubseteq B$, $A \sqsubseteq \leqslant 2\, R.B$ are not OK

Outline

The Basic Idea

- Traditional reasoning procedures for most DLs
- Employed by FaCT++, Hermit, Pellet, and Racer
- Given a conjecture $?- \mathcal{O} \models \alpha$ to prove, try to construct a (representation of a) model $\mathcal{I} \models \mathcal{O}$ such that $\mathcal{I} \not\models \alpha$.
- If such an $\mathcal{I}$ is found then $\mathcal{O} \not\models \alpha$; otherwise $\mathcal{O} \models \alpha$

The Basic Idea

- Traditional reasoning procedures for most DLs
- Employed by FaCT++, Hermit, Pellet, and Racer
- Given a conjecture $?- \mathcal{O} \models \alpha$ to prove, try to construct a (representation of a) model $\mathcal{I} \models \mathcal{O}$ such that $\mathcal{I} \not\models \alpha$.
- If such an $\mathcal{I}$ is found then $\mathcal{O} \not\models \alpha$; otherwise $\mathcal{O} \models \alpha$
- Model expansion rules:
  1. Create an initial element i_0, label with $\neg\alpha(i_o)$
  2. Put all concepts into the negation normal form
  3. Expand conjunctions $C \sqcap D$ into C and D
  4. Guess disjunctions $C \sqcup D$ by selecting C or D
  5. Use $\exists R.C$ to create R-successors with C
  6. Use $\forall R.C$ to propagate C to R-successors
  7. Backtrack when a clash, e.g., $\{C, \neg C\}$ is detected

Example 1

Ontology $\mathcal{O}$

$$\frac{\begin{array}{r} A \sqsubseteq \exists R.(B \sqcap C) \\ \exists R.C \sqsubseteq D \end{array}}{?- A \sqsubseteq D}$$

Example 1

Ontology $\mathcal{O}$

$$\frac{\begin{array}{r} A \sqsubseteq \exists R.(B \sqcap C) \\ \exists R.C \sqsubseteq D \end{array}}{?- A \sqsubseteq D}$$

1. Convert Axioms to Negation Normal Form

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$

1. Convert Axioms to Negation Normal Form

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture

- $A, \neg D$

Example 1

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$

$\top \sqsubseteq \forall R.(\neg C) \sqcup D$

$?- A \sqsubseteq D$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors

- $A, \neg D$

Example 1

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$

$\top \sqsubseteq \forall R.(\neg C) \sqcup D$

$?- A \sqsubseteq D$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 1

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$

$\top \sqsubseteq \forall R.(\neg C) \sqcup D$

$?- A \sqsubseteq D$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

- $A, \neg D, \exists R.(B \sqcap C)$

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

- $A, \neg D, \exists R.(B \sqcap C), \forall R.(\neg C)$

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

- $A, \neg D, \exists R.(B \sqcap C), \forall R.(\neg C)$

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

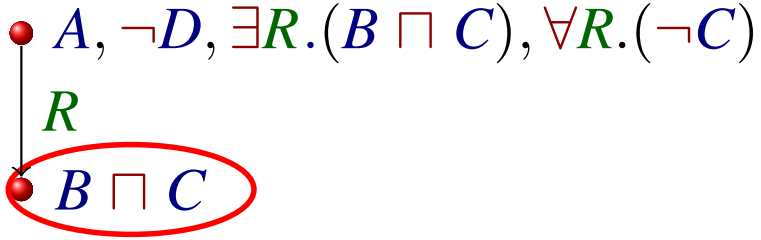
- $A, \neg D, \exists R.(B \sqcap C), \forall R.(\neg C)$

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$



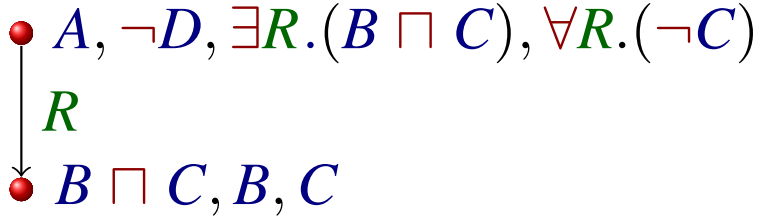
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$



1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

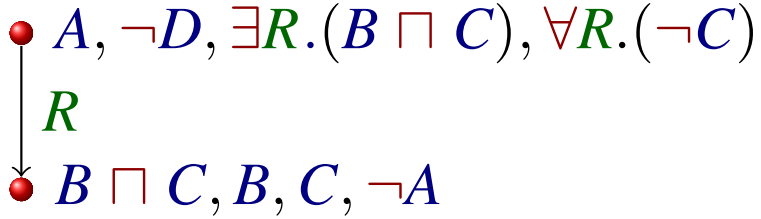
Example 1

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$

$\top \sqsubseteq \forall R.(\neg C) \sqcup D$

$?- A \sqsubseteq D$



1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$

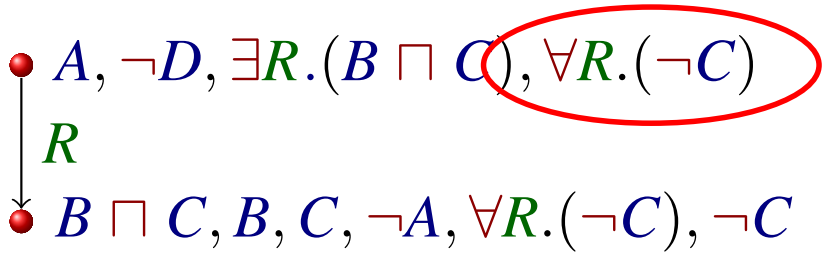
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$



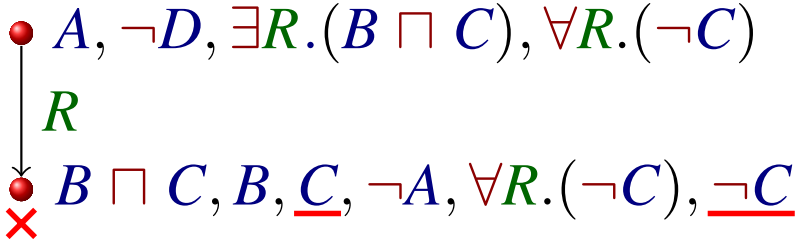
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$



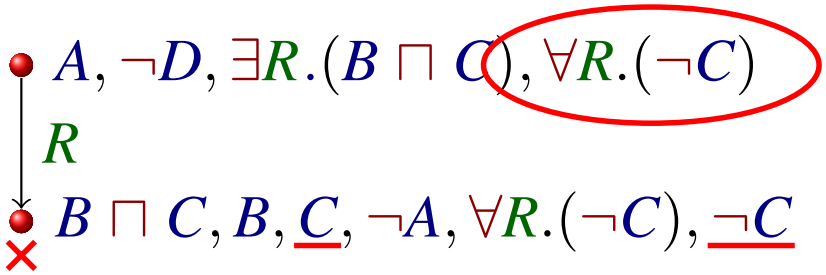
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$$?- A \sqsubseteq D$$



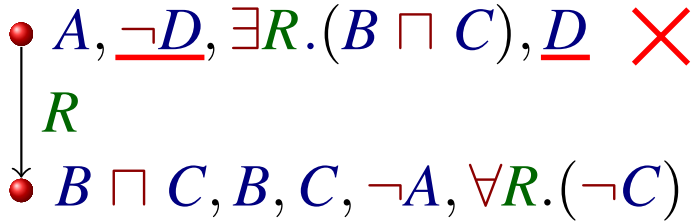
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 1

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.(B \sqcap C)$$
$$\top \sqsubseteq \forall R.(\neg C) \sqcup D$$

$?- A \sqsubseteq D$ *Yes*



1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\frac{\begin{array}{l} A \sqsubseteq \exists R.B \\ B \sqsubseteq \exists R^{-}.(A \sqcap C) \end{array}}{?- A \sqsubseteq C}$$

Example 2

Ontology $\mathcal{O}$

$$\frac{\begin{array}{l} A \sqsubseteq \exists R.B \\ B \sqsubseteq \exists R^{-}.(A \sqcap C) \end{array}}{?- A \sqsubseteq C}$$

1. Convert Axioms to Negation Normal Form

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture

- $A, \neg C$

Example 2

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.B$

$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$

$?- A \sqsubseteq C$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors

- $A, \neg C$

Example 2

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.B$

$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$

$?- A \sqsubseteq C$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.B$

$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$

$?- A \sqsubseteq C$

- $A, \neg C, \exists R.B$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

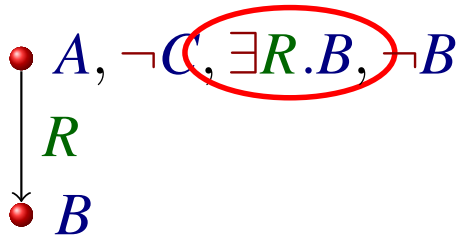
- $A, \neg C, \exists R.B, \neg B$

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$



1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

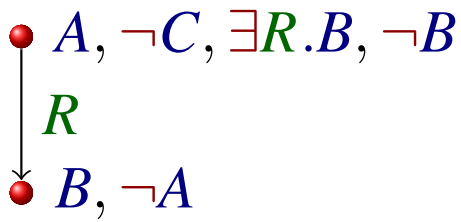
Example 2

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.B$

$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$

$?- A \sqsubseteq C$



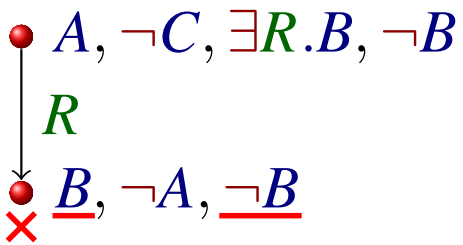
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$



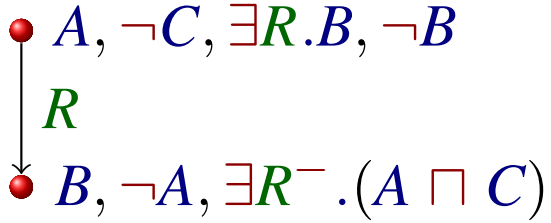
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$



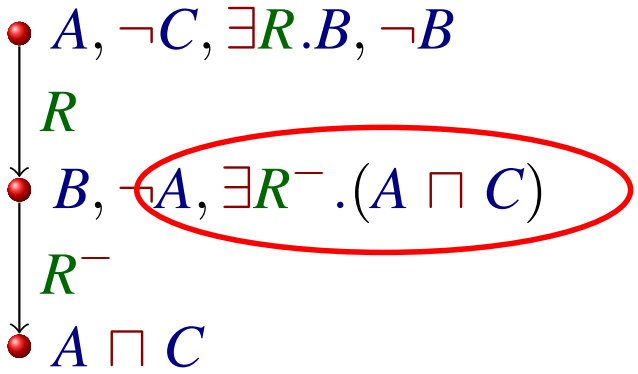
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$



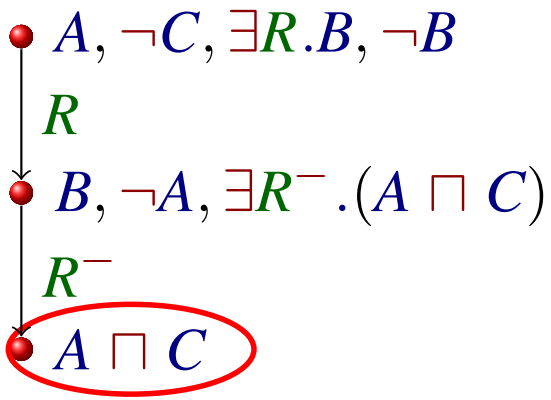
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$



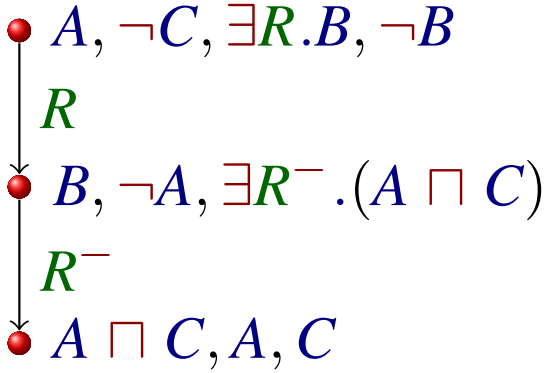
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$



1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

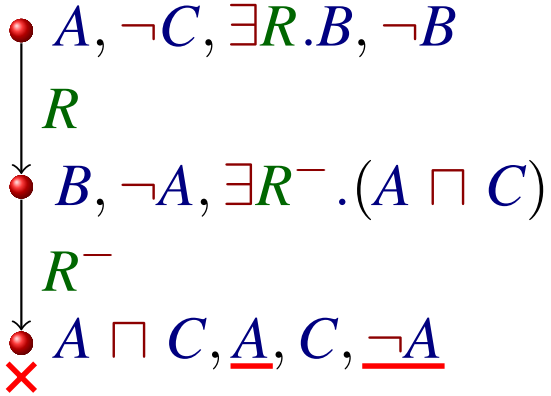
Example 2

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.B$

$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$

$?- A \sqsubseteq C$



1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

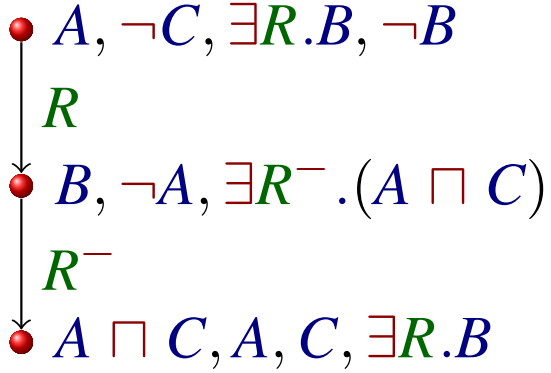
Example 2

Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.B$

$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$

$?- A \sqsubseteq C$



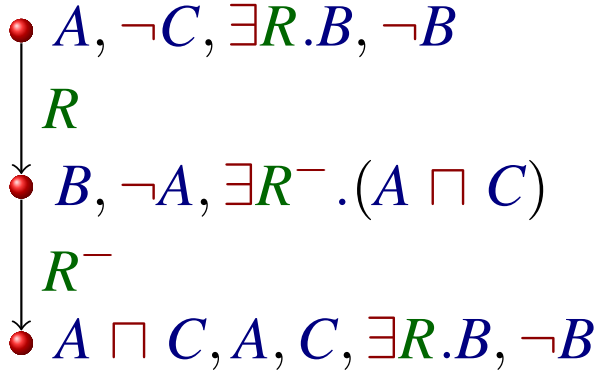
1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$



1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

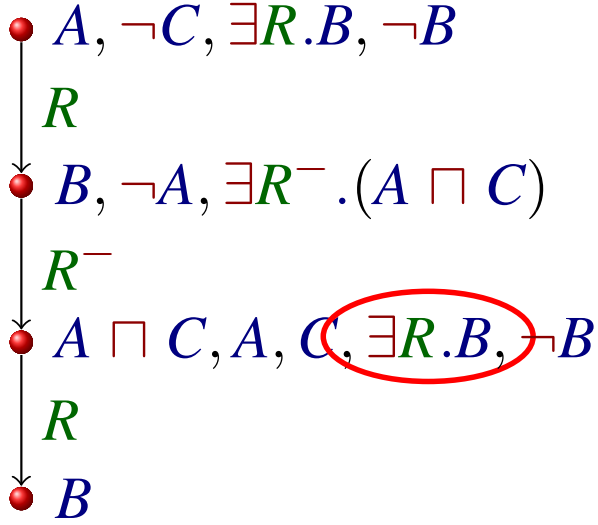
Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash



Example 2

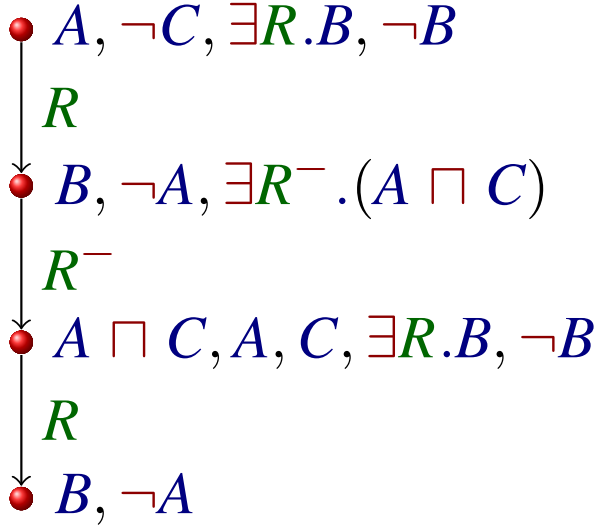
Ontology $\mathcal{O}$

$\top \sqsubseteq \neg A \sqcup \exists R.B$

$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$

$?- A \sqsubseteq C$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash



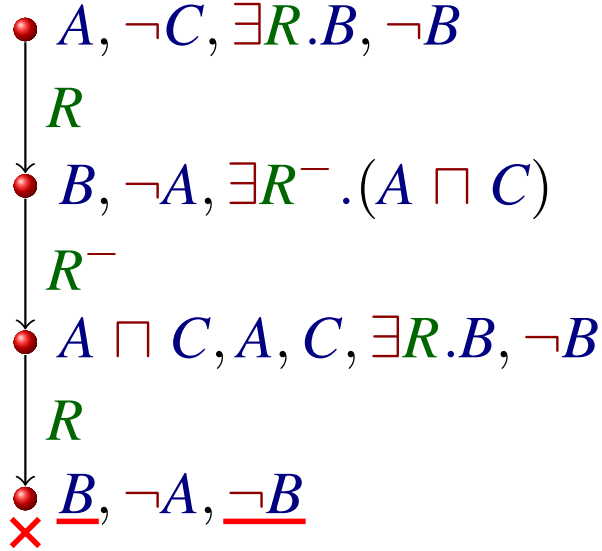
Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash

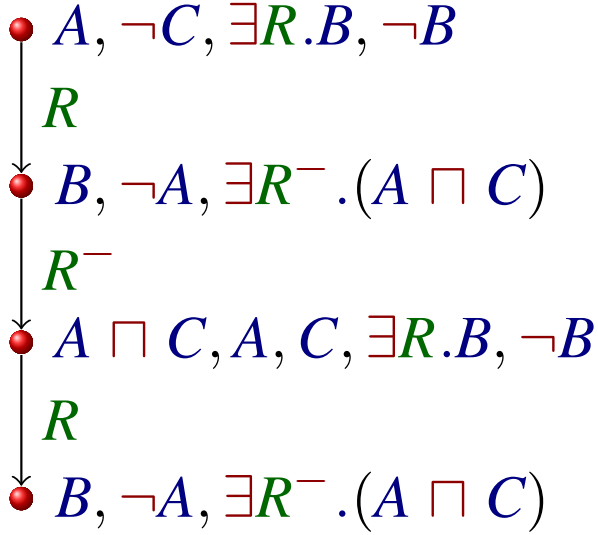


Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$
$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash



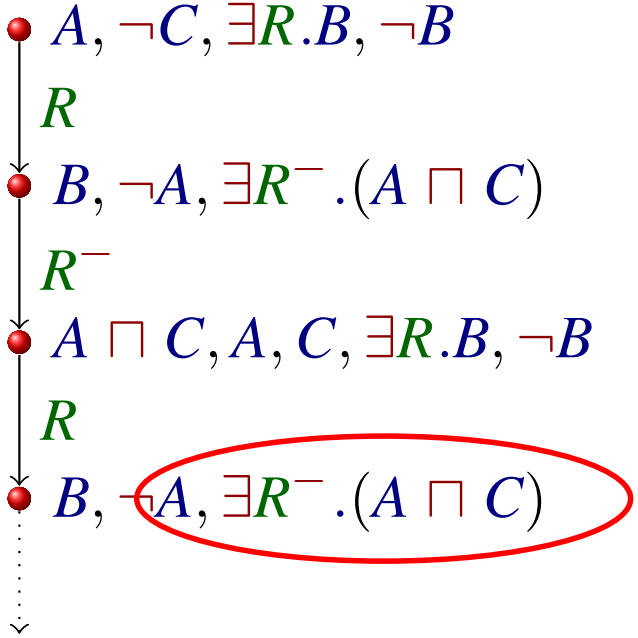
Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash



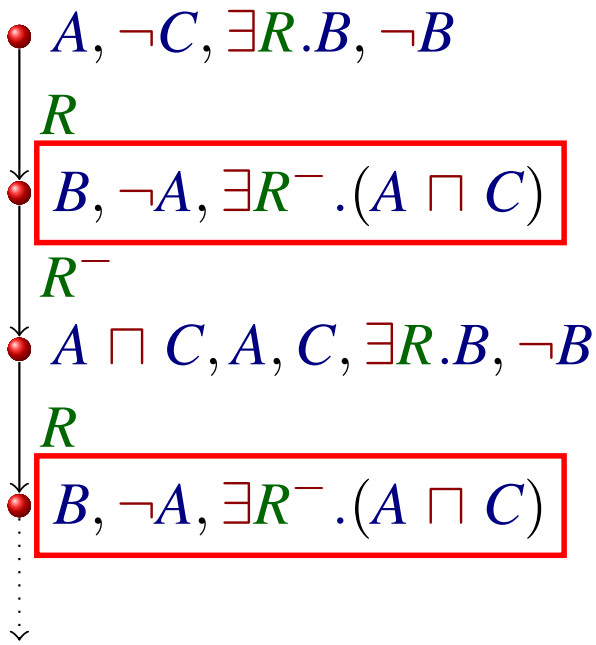
Example 2

Ontology $\mathcal{O}$

$$\top \sqsubseteq \neg A \sqcup \exists R.B$$
$$\top \sqsubseteq \neg B \sqcup \exists R^{-}.(A \sqcap C)$$

$?- A \sqsubseteq C$ *No*

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the conjecture
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash
5. Block



The Drawbacks

1. Excessive non-determinism:
   - Every axiom $C \sqsubseteq D$ results in a disjunction $\top \sqsubseteq \neg C \sqcup D$
   - The choice has to be done for every concept inclusion and every element of the constructed model
   - Partially addressed using (binary) absorption and in "hyper-tableau" procedures

The Drawbacks

1. Excessive non-determinism:
   - Every axiom $C \sqsubseteq D$ results in a disjunction $\top \sqsubseteq \neg C \sqcup D$
   - The choice has to be done for every concept inclusion and every element of the constructed model
   - Partially addressed using (binary) absorption and in "hyper-tableau" procedures
2. Large models:
   - Axioms of the form $A \sqsubseteq \exists R.B$ cause cyclic dependencies
   - Blocking is not efficient since labels could be large and sensitive to the order of rule applications.
   - Both are especially apparent in case of Galen.

The Drawbacks

1. Excessive non-determinism:
   - Every axiom $C \sqsubseteq D$ results in a disjunction $\top \sqsubseteq \neg C \sqcup D$
   - The choice has to be done for every concept inclusion and every element of the constructed model
   - Partially addressed using (binary) absorption and in "hyper-tableau" procedures
2. Large models:
   - Axioms of the form $A \sqsubseteq \exists R.B$ cause cyclic dependencies
   - Blocking is not efficient since labels could be large and sensitive to the order of rule applications.
   - Both are especially apparent in case of Galen.
3. Classification requires enumeration:
   - Every subsumption $A \sqsubseteq B$ has to be checked separately
   - $> 99\%$ of subsumption tests fail and require the construction of the full model (consequently the issues with model sizes)
   - Some optimisations exist

Outline

The Basic Idea

- Originally proposed for the DL $\mathcal{EL}^{++}$ [Baader et al., 2005]
- Implemented in CEL, performs well for Snomed and Galen$^-$
- Uses a number of inference rules to derive consequences of (normalized) axioms—hence the name of the method.

The Basic Idea

- Originally proposed for the DL $\mathcal{EL}^{++}$ [Baader et al., 2005]
- Implemented in CEL, performs well for Snomed and Galen$^-$
- Uses a number of inference rules to derive consequences of (normalized) axioms—hence the name of the method.

The inference rules for $\mathcal{ELH}$

CR1 $$\frac{A \sqsubseteq B \quad B \sqsubseteq C \in \mathcal{O}}{A \sqsubseteq C}$$

CR2 $$\frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D \in \mathcal{O}}{A \sqsubseteq D}$$

CR3 $$\frac{A \sqsubseteq B \quad B \sqsubseteq \exists R.C \in \mathcal{O}}{A \sqsubseteq \exists R.C}$$

CR4 $$\frac{A \sqsubseteq \exists R_1.B \quad R_1 \sqsubseteq R_2 \in \mathcal{O}}{A \sqsubseteq \exists R_2.B}$$

CR5 $$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D \in \mathcal{O}}{A \sqsubseteq D}$$

Example 1

Ontology $\mathcal{O}$

$$\frac{\begin{array}{r} A \sqsubseteq \exists R.(B \sqcap C) \\ \exists R.C \sqsubseteq D \end{array}}{?- A \sqsubseteq D}$$

1. Normalisation (structural transformation)
2. Saturation under the inference rules

Example 1

Ontology $\mathcal{O}$

$$\frac{A \sqsubseteq \exists R.(B \sqcap C) \qquad \exists R.C \sqsubseteq D}{? - A \sqsubseteq D}$$

1. Normalisation (structural transformation)
2. Saturation under the inference rules

Example 1

Ontology $\mathcal{O}$

$$\frac{A \sqsubseteq \exists R.\underline{E} \quad \underline{E} \sqsubseteq B \sqcap C \qquad \exists R.C \sqsubseteq D}{?- A \sqsubseteq D}$$

1. Normalisation (structural transformation)
2. Saturation under the inference rules

Example 1

Ontology $\mathcal{O}$

$$\frac{A \sqsubseteq \exists R.E \quad E \sqsubseteq B \sqcap C \quad \exists R.C \sqsubseteq D}{?- A \sqsubseteq D}$$

1. Normalisation (structural transformation)
2. Saturation under the inference rules

Example 1

Ontology $\mathcal{O}$

$$\frac{\begin{array}{ll} A \sqsubseteq \exists R.E & E \sqsubseteq B \\ \exists R.C \sqsubseteq D & E \sqsubseteq C \end{array}}{?- A \sqsubseteq D}$$

1. Normalisation (structural transformation)
2. Saturation under the inference rules

Example 1

Ontology $\mathcal{O}$

$$\frac{A \sqsubseteq \exists R.E \quad E \sqsubseteq B \qquad \exists R.C \sqsubseteq D \quad E \sqsubseteq C}{?- A \sqsubseteq D \qquad Yes}$$

1. Normalisation (structural transformation)
2. Saturation under the inference rules

CR5:

$$\frac{A \sqsubseteq \exists R.E \quad E \sqsubseteq C \quad \exists R.C \sqsubseteq D \in \mathcal{O}}{A \sqsubseteq D}$$

Advantages

- Performs the full classification “in one pass”
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- The order of rule applications is irrelevant
- Computationally optimal (for $\mathcal{ELH}$)
- Easy to make incremental, add explanations, progress bar

Inverse Roles and Functionality

- Galen uses two constructors that are not in $\mathcal{ELH}$: inverse roles and role functionality:
  - BasilArtery $\sqsubseteq \exists$isBranchOf.VertebalArtery
  - VertebalArtery $\sqsubseteq \exists$hasBranch.BasilArtery
  - isBranchOf $\sqsubseteq$ hasBranch$^-$
  - Fun(isBranchOf)

Inverse Roles and Functionality

- Galen uses two constructors that are not in $\mathcal{ELH}$: inverse roles and role functionality:
  - BasilArtery $\sqsubseteq \exists$isBranchOf.VertebalArtery
  - VertebalArtery $\sqsubseteq \exists$hasBranch.BasilArtery
  - isBranchOf $\sqsubseteq$ hasBranch$^-$
  - Fun(isBranchOf)

Inverse Roles and Functionality

- Galen uses two constructors that are not in $\mathcal{ELH}$: inverse roles and role functionality:
  - BasilArtery $\sqsubseteq \exists$isBranchOf.VertebalArtery
  - VertebalArtery $\sqsubseteq \exists$hasBranch.BasilArtery
  - isBranchOf $\sqsubseteq$ hasBranch$^-$
  - Fun(isBranchOf)

Inverse Roles and Functionality

- Galen uses two constructors that are not in $\mathcal{ELH}$: inverse roles and role functionality:
  - $\text{BasilArtery} \sqsubseteq \exists \text{isBranchOf}.\text{VertebalArtery}$
  - $\text{VertebalArtery} \sqsubseteq \exists \text{hasBranch}.\text{BasilArtery}$
  - $\text{isBranchOf} \sqsubseteq \text{hasBranch}^{-}$
  - $\mathit{Fun}(\text{isBranchOf})$
- In [Baader et al., 2005; 2008] it was shown that:

$\mathcal{EL}$ + functionality $\vee$ number restrictions = NP-hard

Inverse Roles and Functionality

- Galen uses two constructors that are not in $\mathcal{ELH}$: inverse roles and role functionality:
  - BasilArtery $\sqsubseteq \exists$isBranchOf.VertebalArtery
  - VertebalArtery $\sqsubseteq \exists$hasBranch.BasilArtery
  - isBranchOf $\sqsubseteq$ hasBranch$^-$
  - Fun(isBranchOf)
- In [Baader et al., 2005; 2008] it was shown that:

  $\mathcal{EL}$ + functionality $\vee$ number restrictions = NP-hard

- Using inverse roles one can express positive universal restrictions:

$$C \sqsubseteq \forall R.D \quad \Leftrightarrow \quad \exists R^-.C \sqsubseteq D$$

Inverse Roles and Functionality

- Galen uses two constructors that are not in $\mathcal{ELH}$: inverse roles and role functionality:
  - BasilArtery $\sqsubseteq \exists$isBranchOf.VertebalArtery
  - VertebalArtery $\sqsubseteq \exists$hasBranch.BasilArtery
  - isBranchOf $\sqsubseteq$ hasBranch$^-$
  - Fun(isBranchOf)
- In [Baader et al., 2005; 2008] it was shown that:

  $\mathcal{EL}$ + functionality $\vee$ number restrictions = NP-hard
- Using inverse roles one can express positive universal restrictions:

$$C \sqsubseteq \forall R.D \quad \Leftrightarrow \quad \exists R^-.C \sqsubseteq D$$

- Qualified functional restrictions generalize functional roles:

$$Fun(S) \quad \Leftrightarrow \quad \top \sqsubseteq \leqslant 1\, S.\top$$

New Interactions Between Axioms

- Two types of interactions between existential and universal restrictions:

1 $$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq \forall R^{-}.C}{A \sqsubseteq C} \quad \leftrightsquigarrow \quad \frac{A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C}{A \sqsubseteq C}$$ **(CR5)**

2 $$\frac{A \sqsubseteq \exists R.B \quad C \sqsubseteq \forall R.D}{A \sqcap C \sqsubseteq \exists R.(B \sqcap D)} \quad \leftrightsquigarrow$$ no analogue in $\mathcal{ELH}$.

New Interactions Between Axioms

- Two types of interactions between existential and universal restrictions:

1 $$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq \forall R^{-}.C}{A \sqsubseteq C} \quad \leftrightsquigarrow \quad \frac{A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C}{A \sqsubseteq C}$$ **(CR5)**

2 $$\frac{A \sqsubseteq \exists R.B \quad C \sqsubseteq \forall R.D}{A \sqcap C \sqsubseteq \exists R.(B \sqcap D)} \quad \leftrightsquigarrow$$ no analogue in $\mathcal{ELH}$.

- Repeated application of the last can produce exponentially many axioms of the form $\bigsqcap A_i \sqsubseteq \exists R. \bigsqcap B_j$

New Interactions Between Axioms

- Two types of interactions between existential and universal restrictions:

1 $$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq \forall R^-.C}{A \sqsubseteq C} \quad \leftrightsquigarrow \quad \frac{A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C}{A \sqsubseteq C}$$ **(CR5)**

2 $$\frac{A \sqsubseteq \exists R.B \quad C \sqsubseteq \forall R.D}{A \sqcap C \sqsubseteq \exists R.(B \sqcap D)} \quad \not\leftrightsquigarrow$$ no analogue in $\mathcal{ELH}$.

- Repeated application of the last can produce exponentially many axioms of the form $\bigsqcap A_i \sqsubseteq \exists R. \bigsqcap B_j$
- Similar interactions take place for functional restrictions:

1 $$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq \exists R^-.C \quad B \sqsubseteq \leqslant 1\, R^-.\top}{A \sqsubseteq C}$$

2 $$\frac{A \sqsubseteq \exists R.B \quad C \sqsubseteq \exists R.D \quad A \sqsubseteq \leqslant 1\, S.\top}{A \sqcap C \sqsubseteq \exists R.(B \sqcap D)}$$

The Procedure for Horn-SHIQ

1. Preprocessing by structural transformation to the form:
   - $\bigsqcap A_i \sqsubseteq C$, where $C = \bot, A, \exists R.A, \forall R.A$, or $\leqslant 1\, S.A$,
   - $R_1 \sqsubseteq R_2$,
   - $Tra(R)$.

The Procedure for Horn-SHIQ

1. Preprocessing by structural transformation to the form:
   - $\bigsqcap A_i \sqsubseteq C$, where $C = \bot, A, \exists R.A, \forall R.A$, or $\leqslant 1\, S.A$,
   - $R_1 \sqsubseteq R_2$,
   - $Tra(R)$.
2. Elimination of transitivity:
   - For every $\bigsqcap A_i \sqsubseteq \forall R.B$ and $Tra(T)$ with $T \sqsubseteq_{\mathcal{O}} R$ add:

     $\bigsqcap A_i \sqsubseteq \forall T.B^T$, $\quad B^T \sqsubseteq \forall T.B^T$, and $B^T \sqsubseteq B$.
   - Remove all axioms $Tra(T)$

The Procedure for Horn-SHIQ

1. Preprocessing by structural transformation to the form:
   - $\bigsqcap A_i \sqsubseteq C$, where $C = \bot, A, \exists R.A, \forall R.A$, or $\leqslant 1\, S.A$,
   - $R_1 \sqsubseteq R_2$,
   - $Tra(R)$.
2. Elimination of transitivity:
   - For every $\bigsqcap A_i \sqsubseteq \forall R.B$ and $Tra(T)$ with $T \sqsubseteq_{\mathcal{O}} R$ add:

     $\bigsqcap A_i \sqsubseteq \forall T.B^T$, $\quad B^T \sqsubseteq \forall T.B^T$, and $B^T \sqsubseteq B$.
   - Remove all axioms $Tra(T)$
3. Saturation under the inference rules ($M, N = \bigsqcap A_i$):

The inference rules for Horn-$\mathcal{SHIQ}$

R1 $$\frac{M \sqsubseteq A_i \quad \bigsqcap A_i \sqsubseteq C \in \mathcal{O}}{M \sqsubseteq C}$$

R3 $$\frac{M \sqsubseteq \exists R_1.N \quad M \sqsubseteq \forall R_2.A \quad R_1 \sqsubseteq_{\mathcal{O}} R_2}{M \sqsubseteq \exists R.(N \sqcap A)}$$

R2 $$\frac{M \sqsubseteq \exists R.N \quad N \sqsubseteq \bot}{M \sqsubseteq \bot}$$

R4 $$\frac{M \sqsubseteq \exists R_1.N \quad N \sqsubseteq \forall R_2.A \quad R_1 \sqsubseteq_{\mathcal{O}} R_2^-}{M \sqsubseteq A}$$

Example 2

Ontology $\mathcal{O}$

$$A \sqsubseteq \exists R.B$$
$$B \sqsubseteq \exists R^{-}.(A \sqcap C)$$

$$?- A \sqsubseteq C$$

1. Preprocessing
2. Saturation

Example 2

Ontology $\mathcal{O}$

$$A \sqsubseteq \exists R.B$$
$$B \sqsubseteq \exists R^{-}.(A \sqcap C)$$

$$?-A \sqsubseteq C$$

1. Preprocessing
2. Saturation

Example 2

Ontology $\mathcal{O}$

$$A \sqsubseteq \exists R.B$$

$$B \sqsubseteq \exists R^{-}.D \qquad D \sqsubseteq A \sqcap C$$

$$?- A \sqsubseteq C$$

1. Preprocessing
2. Saturation

Example 2

Ontology $\mathcal{O}$

$$A \sqsubseteq \exists R.B$$

$$B \sqsubseteq \exists R^{-}.D \qquad D \sqsubseteq A \sqcap C$$

$$?- A \sqsubseteq C$$

1. Preprocessing
2. Saturation

Example 2

Ontology $\mathcal{O}$

$$A \sqsubseteq \exists R.B \qquad D \sqsubseteq A$$
$$B \sqsubseteq \exists R^{-}.D \qquad D \sqsubseteq C$$

$?- A \sqsubseteq C$

1. Preprocessing
2. Saturation

Example 2

Ontology $\mathcal{O}$

$$A \sqsubseteq \exists R.B \qquad D \sqsubseteq A$$
$$B \sqsubseteq \exists R^{-}.D \qquad D \sqsubseteq C$$

$?- A \sqsubseteq C$ *No*

1. Preprocessing
2. Saturation

No inference is possible

Soundness, Completeness, Complexity

Soundness is due to the fact that:

1. preprocessing preserves entailment over the original singature: if $\mathcal{O} \Rightarrow \mathcal{O}'$ then $\mathcal{O} \models \alpha$ if and only if $\mathcal{O}' \models \alpha$, and
2. inference rules produce consequences of the axioms in their premises.

Soundness, Completeness, Complexity

Soundness is due to the fact that:

1. preprocessing preserves entailment over the original singature: if $\mathcal{O} \Rightarrow \mathcal{O}'$ then $\mathcal{O} \models \alpha$ if and only if $\mathcal{O}' \models \alpha$, and
2. inference rules produce consequences of the axioms in their premises.

Theorem (Completeness)

If $\mathcal{O}'$ is the saturation of under the inference rules then:
$\mathcal{O} \models M \sqsubseteq B$ *iff* $M \sqsubseteq \bot \in \mathcal{O}'$ *or* $M \sqsubseteq B \in \mathcal{O}'$.

Soundness, Completeness, Complexity

Soundness is due to the fact that:

1. preprocessing preserves entailment over the original singature: if $\mathcal{O} \Rightarrow \mathcal{O}'$ then $\mathcal{O} \models \alpha$ if and only if $\mathcal{O}' \models \alpha$, and
2. inference rules produce consequences of the axioms in their premises.

Theorem (Completeness)

If $\mathcal{O}'$ is the saturation of under the inference rules then:
$\mathcal{O} \models M \sqsubseteq B$ *iff* $M \sqsubseteq \bot \in \mathcal{O}'$ *or* $M \sqsubseteq B \in \mathcal{O}'$.

- Since the procedure produces only axioms of the forms: $\bigsqcap A_i \sqsubseteq \exists R.\bigsqcap B_j$ and $\bigsqcap A_i \sqsubseteq C$, it runs in exponential time

Soundness, Completeness, Complexity

Soundness is due to the fact that:

1. preprocessing preserves entailment over the original singature: if $\mathcal{O} \Rightarrow \mathcal{O}'$ then $\mathcal{O} \models \alpha$ if and only if $\mathcal{O}' \models \alpha$, and
2. inference rules produce consequences of the axioms in their premises.

Theorem (Completeness)

If $\mathcal{O}'$ is the saturation of under the inference rules then:
$\mathcal{O} \models M \sqsubseteq B$ *iff* $M \sqsubseteq \bot \in \mathcal{O}'$ *or* $M \sqsubseteq B \in \mathcal{O}'$.

- Since the procedure produces only axioms of the forms: $\bigsqcap A_i \sqsubseteq \exists R.\bigsqcap B_j$ and $\bigsqcap A_i \sqsubseteq C$, it runs in exponential time
- For $\mathcal{ELH}$ the procedure produces only axioms of the forms: $A \sqsubseteq \exists R.B$ and $A \sqsubseteq C$, and runs in polynomial time

Experimental Results

- We have implemented a procedure for Horn-$\mathcal{SHIF}$ in a reasoner CB:
  `http://code.google.com/p/cb-reasoner/`
- Tests are performed on a 2GHz PC with 1.5 GB RAM
- Time limit 1 hour, java heap space = 1GB

	Classes:	FaCT++	Pellet	HermiT	CEL	CB
GO	20465	15.24	72.02	199.52	1.84	1.17
NCI	27652	6.05	26.47	169.47	5.76	3.57
GalenS$^-$	2748	3166.74	133.25	91.98	3.27	0.26
GalenS	2748	465.35	—	45.72	*n/a*	0.32
GalenB$^-$	23136	—	—	—	189.12	4.07
GalenB	23136	—	—	—	*n/a*	9.58
Snomed	389472	650.37	—	—	1185.70	49.44

Outline

Summary

- We described and implemented a novel consequence-based procedure for Horn-$\mathcal{SHIQ}$ ontologies.
- The procedure have several advantages over traditional model-building procedures:
  1. 100 % deterministic
  2. 1-pass classification
  3. computationally optimal for both Horn-$\mathcal{SHIQ}$ and $\mathcal{ELH}$.
- The implementation demonstrates a substantial performance improvement over other reasoners
- Thank you for your attention!