Consequence-Driven Reasoning for Horn-SHIQ Ontologies

Yevgeny Kazakov

Oxford University Computing Laboratory

July 16, 2009

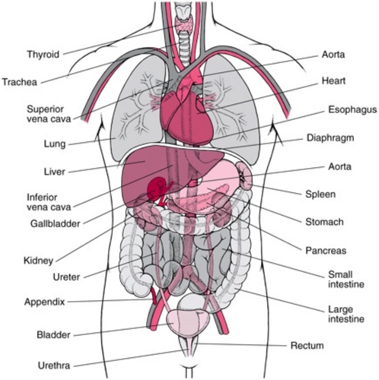
Outline

ONTOLOGIES...

- ...are formal vocabularies of terms covering specific subjects

ONTOLOGIES. . .

- . . . are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy



ONTOLOGIES...

- ...are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses

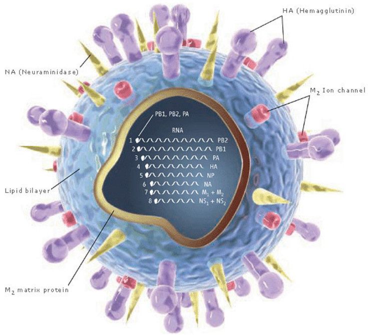


Illustration: Chris Bickel/Science. Reprinted with permission from Science Vol. 312, page 380 (21 April 2006) © 2006 by AAAS

Ontologies. . .

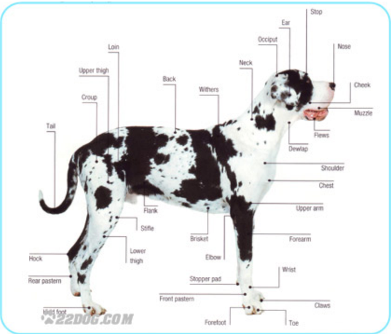
- . . . are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines

Ontologies. . .

- . . . are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs

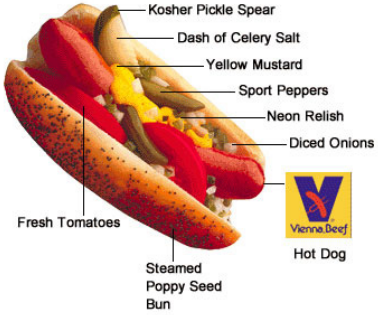
Ontologies...

- ...are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs
  - Dogs



ONTOLOGIES...

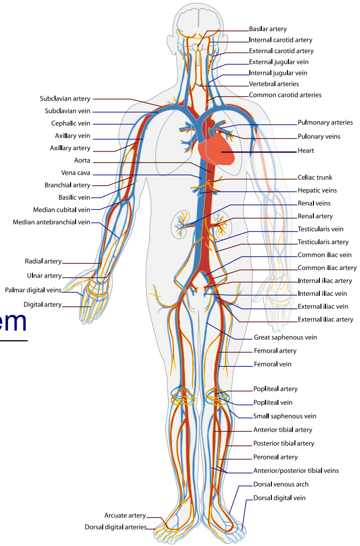
- ...are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs
  - Dogs
  - Hot dogs...



ONTOLOGIES...

- ...are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs
  - Dogs
  - Hot dogs...
- Axioms in ontologies define the relations between terms:

  Heart is a muscular organ
  that is a part of circulatory system



Ontologies...

- ...are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs
  - Dogs
  - Hot dogs...
- Axioms in ontologies define the relations between terms:

  > Heart is a muscular organ
  > that is a part of circulatory system

- Applications:

ONTOLOGIES. . .

- . . . are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs
  - Dogs
  - Hot dogs. . .
- Axioms in ontologies define the relations between terms:

  > Heart is a muscular organ
  > that is a part of circulatory system

- Applications:
  - Reference Databases (e.g., medical references)

Ontologies...

- ...are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs
  - Dogs
  - Hot dogs...
- Axioms in ontologies define the relations between terms:

  Heart is a muscular organ
  that is a part of circulatory system
- Applications:
  - Reference Databases (e.g., medical references)
  - Expert systems (e.g., symptom diagnosis)

ONTOLOGIES. . .

- . . . are formal vocabularies of terms covering specific subjects such as:
  - Human Anatomy
  - Viruses
  - Wines
  - Drugs
  - Dogs
  - Hot dogs. . .
- Axioms in ontologies define the relations between terms:

  > Heart is a muscular organ
  > that is a part of circulatory system
- Applications:
  - Reference Databases (e.g., medical references)
  - Expert systems (e.g., symptom diagnosis)
  - Data exchange (e.g., patient record annotations)

ONTOLOGY LANGUAGES...

- ...define syntax and semantics of ontologies

ONTOLOGY LANGUAGES...

- ...define syntax and semantics of ontologies
- OWL and OWL 2 are based on Description Logics:

EXAMPLE

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

Ontology Languages...

- ...define syntax and semantics of ontologies
- OWL and OWL 2 are based on Description Logics:

Example

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

- Syntax:
  - Atomic concepts

Ontology Languages...

- ...define syntax and semantics of ontologies
- OWL and OWL 2 are based on Description Logics:

Example

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

- Syntax:
  - Atomic concepts
  - Atomic roles

Ontology Languages...

- ...define syntax and semantics of ontologies
- OWL and OWL 2 are based on Description Logics:

Example

$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

- Syntax:
  - Atomic concepts
  - Atomic roles
  - Constructors

Ontology Languages...

- ...define syntax and semantics of ontologies
- OWL and OWL 2 are based on Description Logics:

Example

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

$\text{Heart}(x)$ $\text{MuscularOrgan}(x)$ $\text{CirculatorySystem}(y)$

- Syntax:
  - Atomic concepts
  - Atomic roles
  - Constructors
- Semantics:
  - $\rightsquigarrow$ Unary predicates

Ontology Languages...

- ...define syntax and semantics of ontologies
- OWL and OWL 2 are based on Description Logics:

Example

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

$$\text{Heart}(x) \quad \text{MuscularOrgan}(x) \quad \text{isPartOf}(x, y) \quad \text{CirculatorySystem}(y)$$

- Syntax:
  - Atomic concepts $\rightsquigarrow$ Unary predicates
  - Atomic roles $\rightsquigarrow$ Binary predicates
  - Constructors
- Semantics:

Ontology Languages...

- ...define syntax and semantics of ontologies
- OWL and OWL 2 are based on Description Logics:

Example

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

$$\text{Heart}(x) \leftrightarrow \text{MuscularOrgan}(x) \land \exists y.(\text{isPartOf}(x, y) \land \text{CirculatorySystem}(y))$$

Syntax:		Semantics:
Atomic concepts	$\rightsquigarrow$	Unary predicates
Atomic roles	$\rightsquigarrow$	Binary predicates
Constructors	$\rightsquigarrow$	Logical connectives

Ontology Reasoning

- One advantage of the logic-based ontology languages is the automated reasoning support — ability to draw conclusions from given axioms

Example

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$$

$$\mathsf{Heart} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$$

Ontology Reasoning

- One advantage of the logic-based ontology languages is the automated reasoning support — ability to draw conclusions from given axioms

Example

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$
$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$$

$$\text{Heart} \sqsubseteq \exists \text{isPartOf}.\text{MuscularSystem}$$

- Makes it possible to model ontologies without duplicating redundant information

Ontology Reasoning

- One advantage of the logic-based ontology languages is the automated reasoning support — ability to draw conclusions from given axioms

Example

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$
$$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$$
$$\overline{\mathsf{Heart} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}}$$

- Makes it possible to model ontologies without duplicating redundant information
- Obtaining implicit axioms from explicit ones is a task of ontology reasoners

CLASSIFICATION

ONTOLOGY

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$
$$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$$
$$\mathsf{MuscularSystem} \sqsubseteq \mathsf{BodySystem}$$
$$\mathsf{CirculatorySystem} \sqsubseteq \mathsf{BodySystem}$$

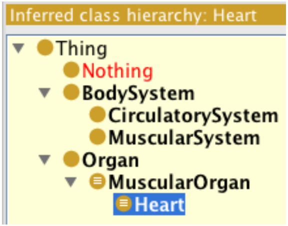
- Ontology developers do not work with axioms directly

Classification

Ontology

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$
$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$$
$$\text{MuscularSystem} \sqsubseteq \text{BodySystem}$$
$$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$$

- Ontology developers do not work with axioms directly
- They navigate within the ontology using a concept taxonomy



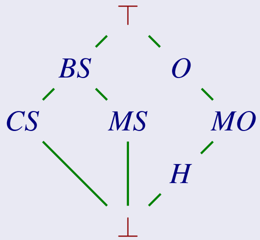
Classification

Ontology

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$
$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$$
$$\text{MuscularSystem} \sqsubseteq \text{BodySystem}$$
$$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$$

- Ontology developers do not work with axioms directly
- They navigate within the ontology using a concept taxonomy
- One of the main reasoning problem for ontologies is classification which goal is to compute the taxonomy

Taxonomy



Results Overview

- Classification times for some well-known large ontologies:

	GO	NCI	Galen v.0	Galen v.7	SNOMED
Concepts:	20465	27652	2748	23136	389472
FaCT++	15.24	6.05	465.35	—	650.37
HermiT	199.52	169.47	45.72	—	—
Pellet	72.02	26.47	—	—	—
CEL	1.84	5.76	—	—	1185.70

Results Overview

- Classification times for some well-known large ontologies:

	GO	NCI	Galen v.0	Galen v.7	SNOMED
Concepts:	20465	27652	2748	23136	389472
FaCT++	15.24	6.05	465.35	—	650.37
HermiT	199.52	169.47	45.72	—	—
Pellet	72.02	26.47	—	—	—
CEL	1.84	5.76	—	—	1185.70
CB	1.17	3.57	0.32	9.58	49.44
Speed-Up:	1.57X	1.61X	143X	∞	13.15X

- The improvement is obtained using a novel consequence-based reasoning procedure

Outline

The Classical Classification Procedure

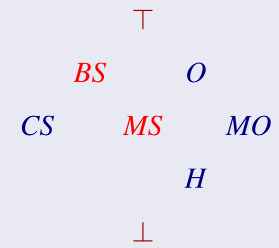
Ontology

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$
$$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$$
$$\mathsf{MuscularSystem} \sqsubseteq \mathsf{BodySystem}$$
$$\mathsf{CirculatorySystem} \sqsubseteq \mathsf{BodySystem}$$

$$?- \mathsf{MuscularSystem} \sqsubseteq \mathsf{BodySystem}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



The Classical Classification Procedure

Ontology

$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$

$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$

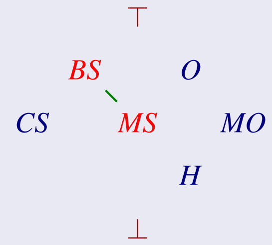
➤ $\text{MuscularSystem} \sqsubseteq \text{BodySystem}$

$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$

$?- \text{MuscularSystem} \sqsubseteq \text{BodySystem}$ *Yes*

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



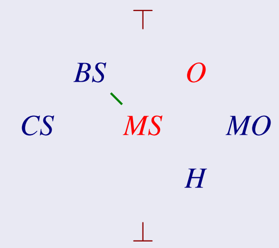
The Classical Classification Procedure

Ontology

$$\begin{aligned}
\mathsf{Heart} &\equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem} \\
\mathsf{MuscularOrgan} &\equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem} \\
\mathsf{MuscularSystem} &\sqsubseteq \mathsf{BodySystem} \\
\mathsf{CirculatorySystem} &\sqsubseteq \mathsf{BodySystem} \\
\hline
?-\ \mathsf{MuscularSystem} &\sqsubseteq \mathsf{Organ}
\end{aligned}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



The Classical Classification Procedure

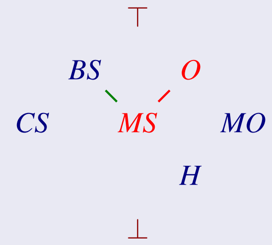
Ontology

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$
$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$$
$$\text{MuscularSystem} \sqsubseteq \text{BodySystem}$$
$$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$$

$$?- \text{MuscularSystem} \sqsubseteq \text{Organ} \qquad \mathit{No}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



The Classical Classification Procedure

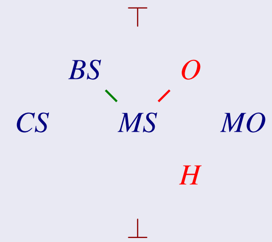
Ontology

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$
$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$$
$$\text{MuscularSystem} \sqsubseteq \text{BodySystem}$$
$$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$$

$$?- \text{Heart} \sqsubseteq \text{Organ}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



The Classical Classification Procedure

Ontology

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$

$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$$

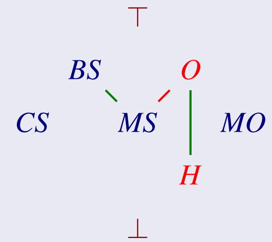
$$\text{MuscularSystem} \sqsubseteq \text{BodySystem}$$

$$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$$

$$?- \text{Heart} \sqsubseteq \text{Organ} \qquad \textit{Yes}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



The Classical Classification Procedure

Ontology

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

$$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$$

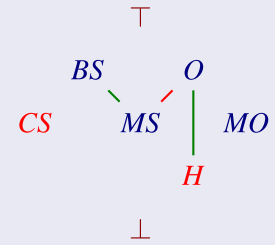
$$\mathsf{MuscularSystem} \sqsubseteq \mathsf{BodySystem}$$

$$\mathsf{CirculatorySystem} \sqsubseteq \mathsf{BodySystem}$$

$$?- \mathsf{Heart} \sqsubseteq \mathsf{CirculatorySystem}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



The Classical Classification Procedure

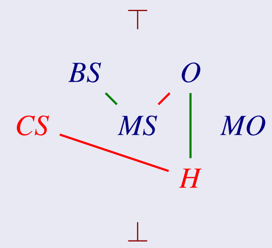
Ontology

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists\text{isPartOf}.\text{CirculatorySystem}$$
$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists\text{isPartOf}.\text{MuscularSystem}$$
$$\text{MuscularSystem} \sqsubseteq \text{BodySystem}$$
$$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$$

$$?- \text{Heart} \sqsubseteq \text{CirculatorySystem} \qquad \textit{No}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation

Taxonomy



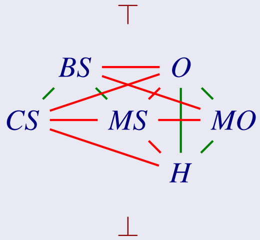
The Classical Classification Procedure

Ontology

$$\mathsf{Heart} \equiv \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$
$$\mathsf{MuscularOrgan} \equiv \mathsf{Organ} \sqcap \exists \mathsf{isPartOf}.\mathsf{MuscularSystem}$$
$$\mathsf{MuscularSystem} \sqsubseteq \mathsf{BodySystem}$$
$$\mathsf{CirculatorySystem} \sqsubseteq \mathsf{BodySystem}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation
3. The taxonomy is computed after all subsumption relations are tested

Taxonomy



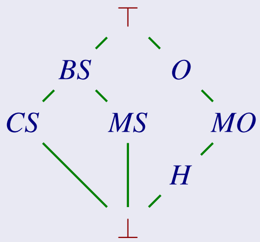
The Classical Classification Procedure

Ontology

$$\text{Heart} \equiv \text{MuscularOrgan} \sqcap \exists \text{isPartOf}.\text{CirculatorySystem}$$
$$\text{MuscularOrgan} \equiv \text{Organ} \sqcap \exists \text{isPartOf}.\text{MuscularSystem}$$
$$\text{MuscularSystem} \sqsubseteq \text{BodySystem}$$
$$\text{CirculatorySystem} \sqsubseteq \text{BodySystem}$$

1. Enumerating pairs of atomic concepts
2. Testing subsumption relation
3. The taxonomy is computed after all subsumption relations are tested

Taxonomy



Testing Individual Subsumptions

Ontology

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

$$\exists R.C \sqsubseteq D$$

$$?- A \sqsubseteq D$$

More Examples

Testing Individual Subsumptions

Ontology

$$A \sqsubseteq \exists R.(B \sqcap C)$$
$$\exists R.C \sqsubseteq D$$

$$?- A \sqsubseteq D$$

1. Convert Axioms to Negation Normal Form

More Examples

Testing Individual Subsumptions

Ontology

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

$?- A \sqsubseteq D$

1. Convert Axioms to Negation Normal Form

More Examples

Testing Individual Subsumptions

Ontology

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

$?-\ A \sqsubseteq D$

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

- $A, \neg D$

More Examples

Testing Individual Subsumptions

Ontology

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

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

$?-\ A \sqsubseteq D$

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

- $A, \neg D$

More Examples

Testing Individual Subsumptions

Ontology

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

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

$?-\ A \sqsubseteq D$

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

$A, \neg D, \neg A$ ✗

More Examples

Testing Individual Subsumptions

Ontology

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

$?- A \sqsubseteq D$

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

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

More Examples

Testing Individual Subsumptions

Ontology

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

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

?- $A \sqsubseteq D$

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

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

More Examples

Testing Individual Subsumptions

Ontology

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

$?- A \sqsubseteq D$

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

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

More Examples

Testing Individual Subsumptions

Ontology

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

$?-\ A \sqsubseteq D$

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

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

R

$B \sqcap C$

More Examples

TESTING INDIVIDUAL SUBSUMPTIONS

ONTOLOGY

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

$$?- A \sqsubseteq D$$

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

R

$B \sqcap C$

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

More Examples

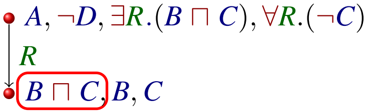
TESTING INDIVIDUAL SUBSUMPTIONS

ONTOLOGY

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

$$?- A \sqsubseteq D$$

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



More Examples

Testing Individual Subsumptions

Ontology

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

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

$?- A \sqsubseteq D$

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

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

$\downarrow R$

$B \sqcap C, B, C, \neg A$

More Examples

Testing Individual Subsumptions

Ontology

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

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

$?- A \sqsubseteq D$

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

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

$\downarrow R$

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

▸ More Examples

Testing Individual Subsumptions

Ontology

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

$$?- A \sqsubseteq D$$

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

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

$\downarrow R$

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

More Examples

Testing Individual Subsumptions

Ontology

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

$$?- A \sqsubseteq D$$

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

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

$\downarrow R$

$B \sqcap C, B, \underline{C}, \neg A, \forall R.(\neg C), \underline{\neg C}$ ✕

More Examples

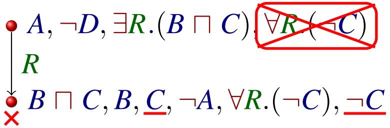
Testing Individual Subsumptions

Ontology

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

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

$?- A \sqsubseteq D$



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

More Examples

Testing Individual Subsumptions

Ontology

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

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

$?- A \sqsubseteq D$

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

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

$\downarrow R$

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

▸ More Examples

Testing Individual Subsumptions

Ontology

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

$?- A \sqsubseteq D$ *Yes*

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

$\downarrow R$

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

1. Convert Axioms to Negation Normal Form
2. Create an element that violates the subsumption
3. Expand labels according to the constructors
4. Backtrack whenever encounter a clash
5. If no bactracking possible, return Yes

More Examples

PROBLEMS

1. Excessive non-determinism from disjunctions $\top \sqsubseteq \neg C \sqcup D$ that appear from axioms $C \sqsubseteq D$

PROBLEMS

1. Excessive non-determinism from disjunctions $\top \sqsubseteq \neg C \sqcup D$ that appear from axioms $C \sqsubseteq D$
2. Large and highly cyclic models caused by existential dependencies $C \sqsubseteq \exists R.D$

ONTOLOGY

$$\mathsf{Heart} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$
$$\mathsf{CirculatorySystem} \sqsubseteq \exists \mathsf{hasPart}.\mathsf{LeftLung}$$
$$\mathsf{LeftLung} \sqsubseteq \exists \mathsf{isPartOf}.\mathsf{RespiratorySystem}$$
$$\mathsf{RespiratorySystem} \sqsubseteq \exists \mathsf{hasPart}.\mathsf{Trachea}$$

and so on . . .

▸ Blocking

Problems

1. Excessive non-determinism from disjunctions $\top \sqsubseteq \neg C \sqcup D$ that appear from axioms $C \sqsubseteq D$
2. Large and highly cyclic models caused by existential dependencies $C \sqsubseteq \exists R.D$

Ontology

$$\begin{aligned} \text{Heart} &\sqsubseteq \exists \text{isPartOf}.\text{CirculatorySystem} \\ \text{CirculatorySystem} &\sqsubseteq \exists \text{hasPart}.\text{LeftLung} \\ \text{LeftLung} &\sqsubseteq \exists \text{isPartOf}.\text{RespiratorySystem} \\ \text{RespiratorySystem} &\sqsubseteq \exists \text{hasPart}.\text{Trachea} \\ &\text{and so on} \ldots \end{aligned}$$

Blocking

3. Classification requires enumeration:
   - Every subsumption $A \sqsubseteq B$ has to be checked separately
   - Typically $> 99\%$ of subsumptions do not hold

Outline

$\mathcal{EL}$ Family of DLs

- $\mathcal{EL}$ [Baader et al.,IJCAI 2003,2005] is a simple DL:
  - concept constructors are $C \sqcap D$ and $\exists R.C$
  - axioms are $C \sqsubseteq D$ and $C \equiv D$

$\mathcal{EL}$ Family of DLs

- $\mathcal{EL}$ [Baader et al.,IJCAI 2003,2005] is a simple DL:
  - concept constructors are $C \sqcap D$ and $\exists R.C$
  - axioms are $C \sqsubseteq D$ and $C \equiv D$
- Interesting due to its polynomial-time complexity

$\mathcal{EL}$ Family of DLs

- $\mathcal{EL}$ [Baader et al.,IJCAI 2003,2005] is a simple DL:
  - concept constructors are $C \sqcap D$ and $\exists R.C$
  - axioms are $C \sqsubseteq D$ and $C \equiv D$
- Interesting due to its polynomial-time complexity
- Surprisingly useful:

GO	NCI	Galen v.0	Galen v.7	SNOMED
✓	✓	✗	✗	✓

$\mathcal{EL}$ Family of DLs

- $\mathcal{EL}$ [Baader et al.,IJCAI 2003,2005] is a simple DL:
  - concept constructors are $C \sqcap D$ and $\exists R.C$
  - axioms are $C \sqsubseteq D$ and $C \equiv D$
- Interesting due to its polynomial-time complexity
- Surprisingly useful:

GO	NCI	Galen v.0	Galen v.7	SNOMED
✓	✓	✗	✗	✓

Example

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

$\mathcal{EL}$ Classification Procedure

1. Normalization to simple axioms of the forms:
   $A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$

$\mathcal{EL}$ Classification Procedure

1. Normalization to simple axioms of the forms:

$A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$

Example

$A \sqsubseteq \exists R.\underline{(B \sqcap C)} \quad \rightsquigarrow$

$\mathcal{EL}$ Classification Procedure

1. Normalization to simple axioms of the forms:
   $A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$

Example

$A \sqsubseteq \exists R.(B \sqcap C) \quad \rightsquigarrow \quad A \sqsubseteq \exists R.\underline{D} \quad \underline{D} \sqsubseteq B \sqcap C$

$\mathcal{EL}$ Classification Procedure

1. Normalization to simple axioms of the forms:
   $A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$

Example

$A \sqsubseteq \exists R.(B \sqcap C) \quad \rightsquigarrow \quad A \sqsubseteq \exists R.D \quad D \sqsubseteq \underline{B \sqcap C}$

$\mathcal{EL}$ Classification Procedure

1. Normalization to simple axioms of the forms:
   $A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$

Example

$$A \sqsubseteq \exists R.(B \sqcap C) \quad \rightsquigarrow \quad A \sqsubseteq \exists R.D \quad D \sqsubseteq B \quad D \sqsubseteq C$$

$\mathcal{EL}$ Classification Procedure

1. Normalization to simple axioms of the forms:

$A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$

Example

$A \sqsubseteq \exists R.(B \sqcap C) \quad \rightsquigarrow \quad A \sqsubseteq \exists R.D \quad D \sqsubseteq B \quad D \sqsubseteq C$

2. Saturation under the inference rules:

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C} \qquad \frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

– Derives only polynomial-many new axioms

Advantages

- Performs the full classification "in one pass"
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- Computationally optimal
- Existential dependencies $C \sqsubseteq \exists R.D$ alone do not trigger any inferences
- Easy to make incremental, add explanations, progress bar

Advantages

- Performs the full classification "in one pass"
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- Computationally optimal
- Existential dependencies $C \sqsubseteq \exists R.D$ alone do not trigger any inferences
- Easy to make incremental, add explanations, progress bar

Advantages

- Performs the full classification "in one pass"
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- Computationally optimal
- Existential dependencies $C \sqsubseteq \exists R.D$ alone do not trigger any inferences
- Easy to make incremental, add explanations, progress bar

ADVANTAGES

- Performs the full classification “in one pass”
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- Computationally optimal
- Existential dependencies $C \sqsubseteq \exists R.D$ alone do not trigger any inferences
- Easy to make incremental, add explanations, progress bar

ADVANTAGES

- Performs the full classification "in one pass"
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- Computationally optimal
- Existential dependencies $C \sqsubseteq \exists R.D$ alone do not trigger any inferences
- Easy to make incremental, add explanations, progress bar

ADVANTAGES

- Performs the full classification "in one pass"
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- Computationally optimal
- Existential dependencies $C \sqsubseteq \exists R.D$ alone do not trigger any inferences
- Easy to make incremental, add explanations, progress bar

ADVANTAGES

- Performs the full classification "in one pass"
- Derives only subsumptions that are implied ($< 1\%$ of all)
- No non-determinism, no backtracking
- Computationally optimal
- Existential dependencies $C \sqsubseteq \exists R.D$ alone do not trigger any inferences
- Easy to make incremental, add explanations, progress bar

The only disadvantage:

- The ontology language is too restricted

AN EXTENSION TO HORN $\mathcal{SHIQ}$

- Many new constructors:

	positive	negative
universal quantification:	$A \sqsubseteq \forall R.B$	—

An Extension to Horn $\mathcal{SHIQ}$

- Many new constructors:

	positive	negative
universal quantification:	$A \sqsubseteq \forall R.B$	–
at least restriction:	$A \sqsubseteq \geqslant n\, R.B$	$\geqslant 1\, R.B \sqsubseteq A$

An Extension to Horn $\mathcal{SHIQ}$

- Many new constructors:

	positive	negative
universal quantification:	$A \sqsubseteq \forall R.B$	–
at least restriction:	$A \sqsubseteq \geqslant n\, R.B$	$\geqslant 1\, R.B \sqsubseteq A$
functionality restriction:	$A \sqsubseteq \leqslant 1\, R.B$	–

An Extension to Horn $\mathcal{SHIQ}$

- Many new constructors:

	positive	negative
universal quantification:	$A \sqsubseteq \forall R.B$	–
at least restriction:	$A \sqsubseteq \geqslant n\, R.B$	$\geqslant 1\, R.B \sqsubseteq A$
functionality restriction:	$A \sqsubseteq \leqslant 1\, R.B$	–
inverse roles:	$S \equiv R^{-}$	

An Extension to Horn $\mathcal{SHIQ}$

- Many new constructors:

	positive	negative
universal quantification:	$A \sqsubseteq \forall R.B$	–
at least restriction:	$A \sqsubseteq \geqslant n\, R.B$	$\geqslant 1\, R.B \sqsubseteq A$
functionality restriction:	$A \sqsubseteq \leqslant 1\, R.B$	–
inverse roles:	$S \equiv R^-$	
functional roles:	$Fun(R)$	

An Extension to Horn $\mathcal{SHIQ}$

- Many new constructors:

	positive	negative
universal quantification:	$A \sqsubseteq \forall R.B$	–
at least restriction:	$A \sqsubseteq \geqslant n\, R.B$	$\geqslant 1\, R.B \sqsubseteq A$
functionality restriction:	$A \sqsubseteq \leqslant 1\, R.B$	–
inverse roles:	$S \equiv R^{-}$	
functional roles:	$Fun(R)$	

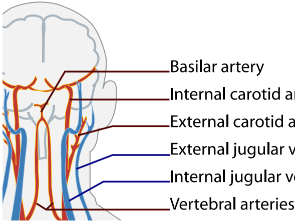
Example (Galen)

BasilarArtery $\sqsubseteq$ $\exists$isBranchOf.VertebalArtery

VertebalArtery $\sqsubseteq$ $\exists$hasBranch.BasilarArtery

isBranchOf $\equiv$ hasBranch^{-}

Fun(isBranchOf)



AN EXTENSION TO HORN $\mathcal{SHIQ}$

- Many new constructors:

	positive	negative
universal quantification:	$A \sqsubseteq \forall R.B$	–
at least restriction:	$A \sqsubseteq \geqslant n\, R.B$	$\geqslant 1\, R.B \sqsubseteq A$
functionality restriction:	$A \sqsubseteq \leqslant 1\, R.B$	–
inverse roles:	$S \equiv R^{-}$	
functional roles:	$Fun(R)$	

- Many new inference rules:

$$\frac{C \sqsubseteq \exists R.D \quad D \sqsubseteq \forall S.A \quad S \equiv R^{-}}{C \sqsubseteq A} \qquad \frac{C \sqsubseteq \exists R.D \quad C \sqsubseteq \forall R.A}{C \sqsubseteq \exists R.(D \sqcap A)}$$

$$\frac{C \sqsubseteq \exists R.D \quad C \sqsubseteq \exists R.E \quad Fun(R)}{C \sqsubseteq \exists R.(D \sqcap E)}$$

All Rules

- The general form of derived axioms:

$$\bigsqcap A_i \sqsubseteq B \qquad \bigsqcap A_i \sqsubseteq \exists R.\bigsqcap B_j$$

RESULTS

- A novel classification procedure for Horn $\mathcal{SHIQ}$ ontologies
- Has several advantages over model-building procedures:
  1. deterministic
  2. 1-pass classification
  3. computationally optimal for Horn-$\mathcal{SHIQ}$ and $\mathcal{ELH}$. ▸ More
- The implementation exhibits a significant speedup:

	GO	NCI	Galen v.0	Galen v.7	SNOMED
FACT++	15.24	6.05	465.35	—	650.37
HERMIT	199.52	169.47	45.72	—	—
PELLET	72.02	26.47	—	—	—
CEL	1.84	5.76	—	—	1185.70
CB	1.17	3.57	0.32	9.58	49.44
Speed-Up:	1.57X	1.61X	143X	∞	13.15X

available at `http://code.google.com/p/cb-reasoner/`

The Inference Rules for Horn $\mathcal{SHIQ}$

$$\frac{}{M \sqcap A \sqsubseteq A} \qquad \frac{}{M \sqsubseteq \top} \qquad \frac{M \sqsubseteq A_1 \quad \dots \quad M \sqsubseteq A_n}{M \sqsubseteq C} : \prod_{i=1}^{n} A_i \sqsubseteq C \in \mathcal{O}$$

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

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

$$\frac{\begin{array}{ll} M \sqsubseteq \exists R_1.N_1 & N_1 \sqsubseteq B \\ M \sqsubseteq \exists R_2.N_2 & N_2 \sqsubseteq B \\ M \sqsubseteq \leqslant 1\, S.B & \end{array}}{M \sqsubseteq \exists R_1.(N_1 \sqcap N_2)} : \begin{array}{l} R_1 \sqsubseteq_{\mathcal{O}} S \\ R_2 \sqsubseteq_{\mathcal{O}} S \end{array} \qquad \frac{\begin{array}{ll} M \sqsubseteq \exists R_1.N_1 & M \sqsubseteq B \\ N_1 \sqsubseteq \exists R_2.(N_2 \sqcap A) & \\ N_1 \sqsubseteq \leqslant 1\, S.B & N_2 \sqcap A \sqsubseteq B \end{array}}{M \sqsubseteq A \quad M \sqsubseteq \exists R_2^-.N_1} : \begin{array}{l} R_1 \sqsubseteq_{\mathcal{O}} S^- \\ R_2 \sqsubseteq_{\mathcal{O}} S \end{array}$$

Where $M, N = \prod A_i$

Blocking

Ontology

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

$$?- A \sqsubseteq C$$

Blocking

Ontology

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

$$?- A \sqsubseteq C$$

1. Convert Axioms to Negation Normal Form

BLOCKING

ONTOLOGY

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

$?- A \sqsubseteq C$

1. Convert Axioms to Negation Normal Form

Blocking

Ontology

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

$?-\ A \sqsubseteq C$

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

- $A, \neg C$

Blocking

Ontology

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

$?- A \sqsubseteq C$

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

- $A, \neg C$

Blocking

Ontology

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

$?- A \sqsubseteq C$

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

- $A, \neg C, \neg A$ ✗

Blocking

Ontology

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

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

?– $A \sqsubseteq C$

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

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

Blocking

Ontology

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

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

$$?- A \sqsubseteq C$$

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

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

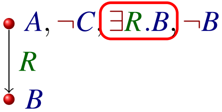
Blocking

Ontology

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

$$?-\ A \sqsubseteq C$$

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



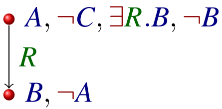
BLOCKING

ONTOLOGY

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

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

$?- A \sqsubseteq C$



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

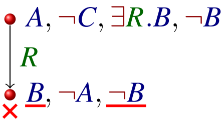
Blocking

Ontology

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

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

$?- A \sqsubseteq C$



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

Blocking

Ontology

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

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

$\downarrow R$

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

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

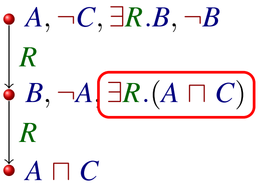
Blocking

Ontology

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

$?- A \sqsubseteq C$

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



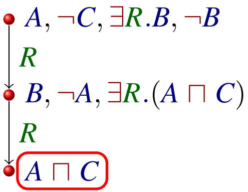
Blocking

Ontology

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

$?- A \sqsubseteq C$

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



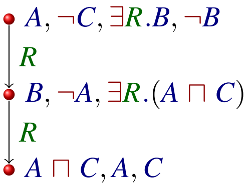
Blocking

Ontology

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

$?- A \sqsubseteq C$

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



Blocking

Ontology

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

$?- A \sqsubseteq C$

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

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

$\downarrow R$

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

$\downarrow R$

$A \sqcap C, \underline{A}, C, \underline{\neg A}$ ✕

BLOCKING

ONTOLOGY

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

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

$?- A \sqsubseteq C$

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

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

R

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

R

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

Blocking

Ontology

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

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

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

$\downarrow R$

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

$\downarrow R$

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

Blocking

Ontology

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

$?- A \sqsubseteq C$

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

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

R

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

R

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

R

B

Blocking

Ontology

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

$?- A \sqsubseteq C$

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

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

$\downarrow R$

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

$\downarrow R$

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

$\downarrow R$

$B, \neg A$

Blocking

Ontology

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

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

?− $A \sqsubseteq C$

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

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

R

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

R

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

R

$B, \neg A, \neg B$ ✗

Blocking

Ontology

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

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

$?- A \sqsubseteq C$

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

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

R

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

R

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

R

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

Blocking

Ontology

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

$?- A \sqsubseteq C$

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

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

R

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

R

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

R

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

Blocking

Ontology

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

$?- A \sqsubseteq C$ *No*

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