Part 2: Reasoning in Description Logic

Peter Radics
CS6204 - Usable Security
Fall 2009
Dr. Dennis Kafura
Virginia Tech

September 24, 2009
Goal

Goal of Presentation

- Demonstrate the power (or lack thereof) of reasoning (what can be reasoned about?)
- Introduce an algorithm for reasoning (how can the computer reason?)
Recap

Three main building blocks
- Concepts
- Relationships
- Individuals
Further building blocks

- Union
- Intersection
- Complement
- Existential quantification
- Universal quantification
- Number restriction
Introducing formality:

We will write:

- $A, B$ for *atomic concepts*
- $R$ for *atomic roles*
- $C, D$ for *concept descriptions* (concepts that are defined through combination of other concepts)
Attributive Languages (cont’d)

The basic description language $\mathcal{AL}$

Definition

$C, D \rightarrow A$ | (atomic concept)
$\top$ | (universal concept)
$\bot$ | (bottom concept)
$\neg A$ | (atomic negation)
$C \sqcap D$ | (intersection)
$\forall R.C$ | (value restriction)
$\exists R.\top$ | (limited existential quantification)
Definition (Interpretations)

An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^\mathcal{I}$ (the domain of the interpretation) and an interpretation function $\cdot^\mathcal{I}$, which assigns to every atomic concept $A$ a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and to every atomic role $R$ a binary relation $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. $\mathcal{I}$ furthermore maps every individual $a$ to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$.

Therefore:

**Definition**

\[
\begin{align*}
\top^\mathcal{I} &= \Delta^\mathcal{I}. \\
\perp^\mathcal{I} &= \emptyset. \\
(-A)^\mathcal{I} &= \Delta^\mathcal{I} \setminus A^\mathcal{I}. \\
(C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I}. \\
(\forall R.C)^\mathcal{I} &= \{ a \in \Delta^\mathcal{I} \mid \forall b. (a, b) \in R^\mathcal{I} \rightarrow b \in C^\mathcal{I} \}. \\
(\exists R.C)^\mathcal{I} &= \{ a \in \Delta^\mathcal{I} \mid \exists b. (a, b) \in R^\mathcal{I} \}. 
\end{align*}
\]

We say \( C \equiv D \) iff \( C^\mathcal{I} = D^\mathcal{I} \) for all interpretations \( \mathcal{I} \).
Extensions of \( \mathcal{AL} \):

**Definition \( \mathcal{AL} [\mathcal{U}] \) (Union)**

\[
(C \sqcup D)^I = C^I \cup D^I.
\]

**Definition \( \mathcal{AL} [\mathcal{E}] \) (Full existential quantification)**

\[
(\exists R. C)^I = a \in \Delta^I \mid \exists b (a, b) \in R^I \land b \in C^I.
\]

**Definition \( \mathcal{AL} [\mathcal{N}] \) (Number restrictions)**

\[
(\geq nR)^I = \{ a \in \Delta^I \mid \| \{ b \mid (a, b) \in R^I \} \| \geq n \}.
\]

\[
(\leq nR)^I = \{ a \in \Delta^I \mid \| \{ b \mid (a, b) \in R^I \} \| \leq n \}.
\]
Definition

\( C \sqsubseteq D \) and \( R \sqsubseteq S \) are called \textit{inclusions}.
\( C \equiv D \) and \( R \equiv S \) are called \textit{equalities}.
\( A \equiv C \) is called a \textit{definition}.

Definition

\( C (a) \) is called a \textit{concept assertion}.
\( R (a, b) \) is called a \textit{role assertion}.

Definition

The interpretation \( \mathcal{I} \) \textit{satisfies} the concept assertion \( C (a) \) if \( a^\mathcal{I} \in C^\mathcal{I} \), and it \textit{satisfies} the role assertion \( R (a, b) \) if \( (a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I} \).
Example scenario

Example

Atomic concepts:
- Store
- Issuer
- Credential
- GovernmentAgency

Atomic roles:
- HasCredential
- IssuedBy
- ControlledBy
Example

Definitions:
- \( \text{UntrustedIssuer} \equiv \text{Issuer} \land \neg \exists \text{ControlledBy.GovernmentAgency} \)
- \( \text{TrustedIssuer} \equiv \neg \text{UntrustedIssuer} \)
- \( \text{UntrustedCredential} \equiv \text{Credential} \land \neg \exists \text{IssuedBy.TrustedIssuer} \)
- \( \text{TrustedCredential} \equiv \text{Credential} \land \exists \text{IssuedBy.TrustedIssuer} \)
- \( \text{TrustedStore} \equiv \text{Store} \land \exists \text{HasCredential.TrustedCredential} \)
Example

**Concept assertions:**
- Store(amazon)
- Store(malroysShadyEmporium)
- Issuer(veriSign)
- Issuer(malroysShadyEmporium)
- GovernmentAgency(nsa)
- Credential(sslCertificate_amazon)
- Credential(sslCertificate_malroysShadyEmporium)
Example scenario (cont’d)

Example

Role assertions:

- HasCredential(amazon, sslCertificate_amazon)
- HasCredential(malroysShadyEmporium, sslCertificate_malroysShadyEmporium)
- IssuedBy(sslCertificate_amazon, veriSign)
- IssuedBy(sslCertificate_malroysShadyEmporium, malroysShadyEmporium)
- ControlledBy(veriSign, nsa)
There are four reasoning tasks for TBoxes:

- Satisfiability (Consistency)
- Subsumption
- Equivalence
- Disjointness
Definition (Satisfiability)

A Concept $C$ is *satisfiable* with respect to a TBox $\mathcal{T}$ if a model $\mathcal{I}$ of $\mathcal{T}$ exists such that $C^\mathcal{I}$ is not empty. In this case, we say that $\mathcal{I}$ is a *model* of $C$.

Definition (Subsumption)

A concept $C$ is *subsumed* by a concept $D$ with respect to $\mathcal{T}$ if $C^\mathcal{I} \sqsubseteq D^\mathcal{I}$ for every model $\mathcal{I}$ of $\mathcal{T}$. In this case we write $C \sqsubseteq_T D$ or $\mathcal{T} \models C \sqsubseteq D$. 
**Definition (Equivalence)**

Two concepts $C$ and $D$ are *equivalent* with respect to $\mathcal{T}$ if $C^\mathcal{I} = D^\mathcal{I}$ for every model $\mathcal{I}$ of $\mathcal{T}$. In this case we write $C \equiv_\mathcal{T} D$ or $\mathcal{T} \models C \equiv D$.

**Definition (Disjointness)**

Two concepts $C$ and $D$ are disjoint with respect to $\mathcal{T}$ if $C^\mathcal{I} \cap D^\mathcal{I} = \emptyset$ for every model $\mathcal{I}$ of $\mathcal{T}$. 
Theorem

All reasoning questions for TBoxes can be reduced to satisfiability!

Corollary

1. C is subsumed by D ⇔ C ∩ ¬D is unsatisfiable;
2. C and D are equivalent ⇔ both (C ∩ ¬D) and (¬C ∩ D) are unsatisfiable;
3. C and D are disjoint ⇔ C ∩ D is unsatisfiable.
Reasoning for ABoxes:

- Satisfiability (Consistency)
- Instance Check (Entailment)

**Definition**

An ABox $A$ is consistent with respect to a TBox $T$, if there is an interpretation that is a model of both $A$ and $T$.

**Definition (Entailment)**

An assertion $a$ is entailed by $A$ and we write $A \models a$ if every interpretation that satisfies $A$, that is, every model of $A$, also satisfies $a$. 
Theorem

All reasoning questions for ABoxes can be reduced to consistency!

Corollary

\( \mathcal{A} \models C(a) \) iff \( \mathcal{A} \cup \{\neg C(a)\} \) is inconsistent.

\( C \) is satisfiable iff \( \{C(a)\} \) is consistent.
Algorithm

Tableau Calculus

Definition

1. Formulate query
2. Expand query
3. Bring query into negative normal form
4. Start with ABox $\mathcal{A} = \{ C_0 (x_0) \}$
5. Iterate transformations on ABox (see next slide)
6. Check consistency on transformed ABoxes
Algorithm

**The → \( \square \)-rule**
*Condition:* \( A \) contains \((C_1 \sqcap C_2)(x)\), but it does not contain both \( C_1(x) \) and \( C_2(x) \).
*Action:* \( A' = A \cup \{C_1(x), C_2(x)\} \).

**The → \( \sqcup \)-rule**
*Condition:* \( A \) contains \((C_1 \sqcup C_2)(x)\), but neither \( C_1(x) \) nor \( C_2(x) \).
*Action:* \( A' = A \cup \{C_1(x)\} \), \( A'' = A \cup \{C_2(x)\} \).

**The → \( \exists \)-rule**
*Condition:* \( A \) contains \((\exists R.C)(x)\), but there is no individual name \( z \) such that \( C(z) \) and \( R(x,z) \) are in \( A \).
*Action:* \( A' = A \cup \{C(y), R(x,y)\} \) where \( y \) is an individual name not occurring in \( A \).

**The → \( \forall \)-rule**
*Condition:* \( A \) contains \((\forall R.C)(x)\) and \( R(x,y) \), but it does not contain \( C(y) \).
*Action:* \( A' = A \cup C(y) \).
The $\rightarrow \geq$-rule

**Condition:** $\mathcal{A}$ contains $(\geq nR)(x)$, and there are no individual names $z_1 \ldots z_n$ such that $R(x, z_i) (1 \leq i \leq n)$ and $z_i \neq z_j (1 \leq i < j \leq n)$ are contained in $\mathcal{A}$.

**Action:** $\mathcal{A}' = \mathcal{A} \cup \{R(x, y_i) \mid (1 \leq i \leq n)\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\}$, where $y_1 \ldots y_n$ are distinct individual names not occurring in $\mathcal{A}$.

The $\rightarrow \leq$-rule

**Condition:** $\mathcal{A}$ contains distinct individual names $y_1 \ldots y_n + 1$ such that $(\leq nR)(x)$ and $R(x, y_1) \ldots R(x, y_n + 1)$ are in $\mathcal{A}$, and $y_i \neq y_j$ is not in $\mathcal{A}$ for some $i \leq j$.

**Action:** For each pair $y_i, y_j$ such that $i > j$ and $y_i \neq y_j$ is not in $\mathcal{A}$, the ABox $\mathcal{A}_{i,j} = [y_i/y_j] \mathcal{A}$ is obtained from $\mathcal{A}$ by replacing each occurrence of $y_i$ by $y_j$. 
Help me try to reason, using the algorithm and the example defined earlier, whether malroysShadyEmporium is a TrustedStore!
Discussion

- What is the impact of reasoning on the usefulness of Description Logics?
- What uses do you see for reasoning in a Usable Security context?