Lecture 3: Description Logics 2

Daria Stepanova

slides based on Reasoning Web 2011 tutorial “Foundations of Description Logics and OWL” by S. Rudolph

Max Planck Institute for Informatics
D5: Databases and Information Systems group

WS 2017/18
Unit Outline

Semantics of Description Logics

DL Nomenclature

Equivalences
Semantics of Description Logics

“Now! *That* should clear up a few things around here!”
Interpretations

Semantics for DLs is defined in a model theoretic way, i.e., based on “abstract possible worlds”, called interpretations.

Def.: An interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of:

- a nonempty set $\Delta^\mathcal{I}$, called the interpretation domain (of $\mathcal{I}$)
- an interpretation function $\cdot^\mathcal{I}$, which maps
  - each atomic concept $A$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$
  - each atomic role $r$ to a subset $r^\mathcal{I}$ of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

individual names $N_I$
... $a$ ...

class names $N_C$
... $C$ ...

role names $N_R$
... $r$ ...

Interpretations

Semantics for DLs is defined in a **model theoretic** way, i.e., based on "abstract possible worlds", called interpretations.

**Def.:** An interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of:

- a nonempty set $\Delta^\mathcal{I}$, called the interpretation domain (of $\mathcal{I}$)
- an interpretation function $\cdot^\mathcal{I}$, which maps
  - each atomic concept $A$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$
  - each atomic role $r$ to a subset $r^\mathcal{I}$ of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

![Interpretation Diagram](image-url)
Semantics of Description Logics

Interpretations

Semantics for DLs is defined in a model theoretic way, i.e., based on "abstract possible worlds", called interpretations.

Def.: An interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of:

- a nonempty set $\Delta^\mathcal{I}$, called the interpretation domain (of $\mathcal{I}$)
- an interpretation function $\cdot^\mathcal{I}$, which maps
  - each atomic concept $A$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$
  - each atomic role $r$ to a subset $r^\mathcal{I}$ of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

![Diagram](https://via.placeholder.com/150)

- individual names $N_I$
  - $\ldots \; a \; \ldots$
- class names $N_C$
  - $\ldots \; C \; \ldots$
- role names $N_R$
  - $\ldots \; r \; \ldots$
- $\Delta^\mathcal{I}$
- $\cdot^\mathcal{I}$
- $a^\mathcal{I}$
Interpretations

Semantics for DLs is defined in a **model theoretic** way, i.e., based on ”abstract possible worlds”, called **interpretations**.

Def.: An **interpretation** $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of:

- a nonempty set $\Delta^\mathcal{I}$, called the interpretation domain (of $\mathcal{I}$)
- an interpretation function $\cdot^\mathcal{I}$, which maps
  - each atomic concept $A$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$
  - each atomic role $r$ to a subset $r^\mathcal{I}$ of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.
Interpretations

Semantics for DLs is defined in a model theoretic way, i.e., based on "abstract possible worlds", called interpretations.

Def.: An interpretation \( \mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I}) \) consists of:

- a nonempty set \( \Delta^\mathcal{I} \), called the interpretation domain (of \( \mathcal{I} \))
- an interpretation function \( \cdot^\mathcal{I} \), which maps
  - each atomic concept \( A \) to a subset \( A^\mathcal{I} \) of \( \Delta^\mathcal{I} \)
  - each atomic role \( r \) to a subset \( r^\mathcal{I} \) of \( \Delta^\mathcal{I} \times \Delta^\mathcal{I} \).
Interpretations: an Example

\[ N_I = \{\text{sun, morning\_star, evening\_star, moon, home}\} \]
\[ N_C = \{\text{Planet, Star}\} \]
\[ N_R = \{\text{orbitsAround, shinesOn}\} \]

\[ \Delta^I = \{\odot, \varnothing, \varphi, \varphi, \delta, \alpha, \sigma, \iota, \eta, \delta, \psi, \iota, \} \]

\[ \text{sun}^I = \odot \]
\[ \text{morning\_star}^I = \varnothing \]
\[ \text{evening\_star}^I = \varphi \]
\[ \text{moon}^I = \alpha \]
\[ \text{home}^I = \delta \]

\[ \text{Planet}^I = \{\varnothing, \varphi, \delta, \sigma, \iota, \eta, \delta, \psi, \iota, \} \]
\[ \text{Star}^I = \{\odot\} \]
\[ \text{orbitsAround}^I = \{\langle \varnothing, \odot \rangle, \langle \varnothing, \varnothing \rangle, \langle \delta, \odot \rangle, \langle \sigma, \odot \rangle, \langle \iota, \odot \rangle, \langle \eta, \odot \rangle, \langle \delta, \odot \rangle, \langle \psi, \odot \rangle, \langle \iota, \odot \rangle, \langle \alpha, \delta \rangle \} \]
\[ \text{shinesOn}^I = \{\langle \odot, \varnothing \rangle, \langle \odot, \varnothing \rangle, \langle \odot, \delta \rangle, \langle \odot, \alpha \rangle, \langle \odot, \sigma \rangle, \langle \odot, \varphi \rangle, \langle \odot, \eta \rangle, \langle \odot, \delta \rangle, \langle \odot, \psi \rangle, \langle \odot, \varnothing \rangle \} \]
Interpretation of Individuals

Unique Name Assumption (UNA)

If $c_1$ and $c_2$ are two individuals such that $c_1 \neq c_2$, then $c^I_1 \neq c^I_2$.

Note: When the UNA holds, equality and distinctness assertions are meaningless. In DLs one can drop UNA.

Example: absence of UNA

Two fathers ($f_1, f_2$) and two sons ($s_1, s_2$) went to a pizzeria and bought three pizzas for picnic lunch. When they started their lunch, everyone had a whole pizza. How could this happen?
Interpretation of Individuals

**Unique Name Assumption (UNA)**

If $c_1$ and $c_2$ are two individuals such that $c_1 \neq c_2$, then $c_1^I \neq c_2^I$

**Note:** When the UNA holds, equality and distinctness assertions are meaningless. In DLs one can drop UNA.

**Standard Name Assumption (SNA)**

The UNA holds, and moreover individuals are interpreted in the same way in all interpretations. Hence, we can assume that $\Delta^I$ contains the set of individuals, and that for each interpretation $\mathcal{I}$, we have that $c^I = c$ (then $c$ is called standard name)
# Interpretation of Concept Expressions

<table>
<thead>
<tr>
<th>Construct</th>
<th>Syntax</th>
<th>Example</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>atomic concept</td>
<td>$A$</td>
<td>$\text{Doctor}$</td>
<td>$A^I \subseteq \Delta^I$</td>
</tr>
<tr>
<td>atomic role</td>
<td>$r$</td>
<td>$\text{hasChild}$</td>
<td>$r^I \subseteq \Delta^I \times \Delta^I$</td>
</tr>
<tr>
<td>atomic negation</td>
<td>$\neg A$</td>
<td>$\neg \text{Doctor}$</td>
<td>$\Delta^I \setminus A^I$</td>
</tr>
<tr>
<td>conjunction</td>
<td>$C \sqcap D$</td>
<td>$\text{Human} \sqcap \text{Male}$</td>
<td>$C^I \cap D^I$</td>
</tr>
<tr>
<td>unqual. exist. res.$^1$</td>
<td>$\exists r$</td>
<td>$\exists \text{hasChild}$</td>
<td>${ o \mid \exists o'.(o, o') \in r^I }$</td>
</tr>
<tr>
<td>value res.</td>
<td>$\forall r.C$</td>
<td>$\forall \text{hasChild}.\text{Male}$</td>
<td>${ o \mid \forall o'.(o, o') \in r^I \rightarrow o' \in C^I }$</td>
</tr>
<tr>
<td>bottom</td>
<td>$\bot$</td>
<td></td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

$C, D$ denote arbitrary concepts and $r$ denotes an arbitrary role.

The above constructs form the basic language $\mathcal{AL}$

---

$^1$Unqualified existential restriction
Many different DL constructs and their combinations have been investigated. Combining various constructs we obtain a concrete DL fragment, i.e., language (see slide 26 for further details).

2 Qualified existential restriction
Boolean Concept Expressions
Boolean Concept Expressions

¬ Politician
Boolean Concept Expressions

\[ \neg \text{Politician} \]
Boolean Concept Expressions

\[ \neg \text{Politician} \]

\[ I \text{Politician} \cup I \text{Actor} \]

\[ I \neg \text{Politician} \cap I \text{Actor} \]

\[ I \text{Politician} \cap I \text{Actor} \]

\[ I \neg \text{Politician} \cup I \text{Actor} \]

\[ I \text{Politician} \cap I \text{Actor} \]

\[ I \neg \text{Politician} \]

\[ I \text{Politician} \]

\[ I \text{Actor} \]

\[ I \neg \text{Politician} \]

\[ I \text{Actor} \]

\[ I \neg \text{Politician} \]

\[ I \text{Actor} \]
Boolean Concept Expressions

\neg Politician

Actor \sqcap Politician

Semantics of Description Logics
DL Nomenclature
Equivalences
Boolean Concept Expressions

\[
\neg \text{Politician} \\
\text{Actor} \sqcap \text{Politician}
\]
Semantics of Description Logics

Boolean Concept Expressions

$\neg Politician$

$Politician \sqcap Actor$

$Actor \sqcap Politician$

$\neg Politician$

$Politician \sqcup Actor$

$Actor \sqcup Politician$

$\neg Politician$

$Politician \sqcap Actor$

$Actor \sqcap Politician$

$\neg Politician$

$Politician \sqcup Actor$

$Actor \sqcup Politician$
Boolean Concept Expressions

\[ \neg \text{Politician} \]

\[ \text{Politician} \sqcup \text{Actor} \]

Actor

Politician
Boolean Concept Expressions

¬ Politician

Politician ⊓ Actor

Politician ⊔ Actor

Actor ⊑ Politician

Actor ⊑ Politician

Politician
Boolean Concept Expressions

\[ \neg Politician \]

\[ \square \]

\[ \text{Actor} \]

\[ \text{Politician} \]
Boolean Concept Expressions

\[ \neg \text{Politician} \]

\[ \text{Politician} \sqcap \text{Actor} \]

\[ \text{Politician} \sqcup \text{Actor} \]
Boolean Concept Expressions

\[ \neg \text{Politician} \]

\[ \text{Politician} \sqcup \text{Actor} \]

\[ \text{Actor} \sqsubseteq \text{Politician} \]
Existential Role Restrictions

\[ \exists \text{parentOf. Male} \]
Universal Role Restrictions

∀ parentOf. Male
Qualified Number Restrictions

\[ \geq 2 \text{ parentOf } \text{ Male} \]
∃ \textit{killed} \quad \text{Self}
## Interpretation of Role Expressions

<table>
<thead>
<tr>
<th>Construct</th>
<th>Syntax</th>
<th>Example</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>atomic role</td>
<td>( r )</td>
<td>( \text{hasChild} )</td>
<td>( r^I \subseteq \Delta^I \times \Delta^I )</td>
</tr>
<tr>
<td>role negation</td>
<td>( \neg r )</td>
<td>( \neg \text{hasSister} )</td>
<td>( \Delta^I \times \Delta^I \setminus {(o, o') \in r^I } )</td>
</tr>
<tr>
<td>inverse role</td>
<td>( r^- )</td>
<td>( \text{hasParent}^- )</td>
<td>( {(o, o') \mid (o', o) \in r^I } )</td>
</tr>
<tr>
<td>transitivity</td>
<td>( r \circ r' )</td>
<td>( \text{hasChild} \circ \text{hasParent} )</td>
<td>( {(o, o') \mid (o, o'') \in r^I, (o'', o') \in r'^I } )</td>
</tr>
</tbody>
</table>
Inverse Role

\[ \text{childOf}^- = \text{parentOf} \]
Role Chain

\[ \text{childOf} \circ \text{parentOf} \]
Semantics of Axioms

Given a way to determine a semantic counterpart for all expressions, we now define the criteria for checking whether an interpretation $\mathcal{I}$ satisfies an axiom alpha $\alpha$ (written: $\mathcal{I} \models \alpha$).

- $\mathcal{I} \models r_1 \circ \ldots \circ r_n \sqsubseteq r$ if $(r_1 \circ \ldots \circ r_n)^\mathcal{I} \subseteq r^\mathcal{I}$
- $\mathcal{I} \models \text{Dis}(s_1, s_2)$ if $s_1^\mathcal{I} \cap s_2^\mathcal{I} = \emptyset$
- $\mathcal{I} \models C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$
- $\mathcal{I} \models C(a)$ if $a^\mathcal{I} \in D^\mathcal{I}$
- $\mathcal{I} \models r(a, b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$
- $\mathcal{I} \models \neg r(a, b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \notin r^\mathcal{I}$
- $\mathcal{I} \models a \approx b$ if $a^\mathcal{I} = b^\mathcal{I}$
- $\mathcal{I} \models a \not\approx b$ if $a^\mathcal{I} \neq b^\mathcal{I}$
Concept and Role Membership

\[ \text{Male}(\text{nicolas}) \]

\[ \text{married(}\text{carla, nicolas}\text{)} \]
General Inclusion Axioms

\[ \text{ExPresident} \sqsubseteq \text{Politician} \]
Role Inclusion Axioms

\[ childOf \circ parentOf \subseteq siblingOf \]
## DL to First Order Logic

<table>
<thead>
<tr>
<th>Syntax</th>
<th>FOL formalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1 \sqsubseteq A_2$</td>
<td>$\forall x (A_1(x) \rightarrow A_2(x))$</td>
</tr>
<tr>
<td>$R_1 \sqsubseteq R_2$</td>
<td>$\forall x, y (R_1(x, y) \rightarrow R_2(x, y))$</td>
</tr>
<tr>
<td>$A_1 \sqsubseteq \neg A_2$</td>
<td>$\forall x (A_1(x) \rightarrow \neg A_2(x))$</td>
</tr>
<tr>
<td>$R_1 \sqsubseteq \neg R_2$</td>
<td>$\forall x, y (R_1(x, y) \rightarrow \neg R_2(x, y))$</td>
</tr>
<tr>
<td>$\exists R \sqsubseteq A$</td>
<td>$\forall x (\exists y (R(x, y)) \rightarrow A(x))$</td>
</tr>
<tr>
<td>$\exists R^- \sqsubseteq A$</td>
<td>$\forall x (\exists y (R(y, x)) \rightarrow A(x))$</td>
</tr>
<tr>
<td>$A \sqsubseteq \exists R$</td>
<td>$\forall x (A(x) \rightarrow \exists y (R(x, y)))$</td>
</tr>
<tr>
<td>$\text{funct}(R)$</td>
<td>$\forall x, y, y' (R(x, y) \land R(x, y') \rightarrow y = y')$</td>
</tr>
<tr>
<td>$A_1 \sqcap A_2 \sqsubseteq A_3$</td>
<td>$\forall x A_1(x) \land A_2(x) \rightarrow A_3(x)$</td>
</tr>
<tr>
<td>$\exists R.A_1 \sqsubseteq A_2$</td>
<td>$\forall x (\exists y (R(x, y) \land A_1(y)) \rightarrow A_2(x)$</td>
</tr>
<tr>
<td>$A_1 \sqsubseteq \exists R.A_2$</td>
<td>$\forall x (A(x) \rightarrow \exists y (R(x, y) \land A_2(y)))$</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
</tr>
</tbody>
</table>
Semantics via Translation into FOL

As (common) DLs can be seen as fragments of FOL, one can also define the semantics by providing a translation of DL axioms into FOL formulas.

- $\tau_R(r, x, y)$: produce for $r(x, y)$ a formula with free variables $x, y$
- $\tau_C(C, x)$: produce for $C(x)$ a formula with free variable $x$
- define transformations recursively
Semantics via Translation into FOL

As (common) DLs can be seen as fragments of FOL, one can also define the semantics by providing a translation of DL axioms into FOL formulas.

- \( \tau_R(r, x, y) \): produce for \( r(x, y) \) a formula with free variables \( x, y \)
- \( \tau_C(C, x) \): produce for \( C(x) \) a formula with free variable \( x \)
- define transformations recursively

Bottom rewriting:

\[
\tau_R(u, x_i, x_j) = \text{true}
\]
\[
\tau_R(r, x_i, x_j) = r(x_i, x_j)
\]
\[
\tau_R(r^-, x_i, x_j) = r(x_j, x_i)
\]
Semantics via Translation into FOL

As (common) DLs can be seen as fragments of FOL, one can also define the semantics by providing a translation of DL axioms into FOL formulas.

- $\tau_R(r, x, y)$: produce for $r(x, y)$ a formula with free variables $x, y$
- $\tau_C(C, x)$: produce for $C(x)$ a formula with free variable $x$
- define transformations recursively

Bottom rewriting:

\[
\begin{align*}
\tau_R(u, x_i, x_j) &= \text{true} \\
\tau_R(r, x_i, x_j) &= r(x_i, x_j) \\
\tau_R(r^-, x_i, x_j) &= r(x_j, x_i)
\end{align*}
\]

\[
\begin{align*}
\tau_C(A, x_i) &= A(x_i) \\
\tau_C(\top, x_i) &= \text{true} \\
\tau_C(\bot, x_i) &= \text{false} \\
\tau_C\left(\{a_1, \ldots, a_n\}, x_i\right) &= \bigvee_{j=1}^{n} x_i = a_j
\end{align*}
\]
Semantics of Description Logics

DL Nomenclature

Equivalences

Semantics via Translation into FOL (ctd.)

Complex concepts:

\[ \tau_C(C \cap D, x_i) = \tau_C(C, x_i) \land \tau_C(D, x_i) \]
\[ \tau_C(C \cup D, x_i) = \tau_C(C, x_i) \lor \tau_C(D, x_i) \]
\[ \tau_C(\neg C, x_i) = \neg \tau_C(C, x_i) \]
\[ \tau_C(\exists r.C, x_i) = \exists x_{i+1}.(\tau_R(r, x_i, x_{i+1}) \land \tau_C(C, x_{i+1})) \]
\[ \tau_C(\forall r.C, x_i) = \forall x_{i+1}.(\tau_R(r, x_i, x_{i+1}) \rightarrow \tau_C(C, x_{i+1})) \]
\[ \tau_C(\exists r.\text{Self}, x_i) = \tau_R(r, x_i, x_i) \]
\[ \tau_C(\geq nr.C, x_i) = \exists x_{i+1} \ldots x_{i+n}.( \bigwedge_{j=i+1}^{i+n} \bigwedge_{k=j+1}^{i+n} (x_j \neq x_k) \]
\[ \bigwedge_{j=i+1}^{i+n} \bigwedge_{k=j+1}^{i+n} (\tau_R(r, x_i, x_j) \land \tau_C(C, x_j)) \]
\[ \tau_C(\leq nr.C, x_i) = \neg \tau_C(\geq (n + 1)r.C, x_i) \]
Axioms:

\[ \tau(C \sqsubseteq D) = \forall x_0 (\tau_C(C, x_0) \rightarrow \tau_C(D, x_0)) \]

\[ \tau(r_1 \circ \ldots \circ r_n \sqsubseteq r) = \forall x_0 \ldots x_n (\bigwedge_{i=1}^n \tau_R(r_i, x_{i-1}, x_i)) \rightarrow \tau_R(r, x_0, x_n) \]

\[ \tau(Dis(r, r')) = \forall x_0, x_1 (\tau_R(r, x_0, x_1) \rightarrow \neg \tau_R(r', x_0, x_1)) \]

\[ \tau(Ref(r, r')) = \forall x \tau_R(r, x, x) \]

\[ \tau(Asym(r)) = \forall x_0, x_1 (\tau_R(r, x_0, x_1) \rightarrow \neg \tau_R(r, x_1, x_0)) \]
Semantics via Translation into FOL (ctd.)

Axioms:

\[ \tau(C \sqsubseteq D) = \forall x_0(\tau_C(C, x_0) \rightarrow \tau_C(D, x_0)) \]
\[ \tau(r_1 \circ \ldots \circ r_n \sqsubseteq r) = \forall x_0 \ldots x_n(\bigwedge_{i=1}^{n} \tau_R(r_i, x_{i-1}, x_i)) \rightarrow \tau_R(r, x_0, x_n) \]
\[ \tau(\text{Dis}(r, r')) = \forall x_0, x_1(\tau_R(r, x_0, x_1) \rightarrow \neg \tau_R(r', x_0, x_1)) \]
\[ \tau(\text{Ref}(r, r')) = \forall x \tau_R(r, x, x) \]
\[ \tau(\text{Asym}(r)) = \forall x_0, x_1.(\tau_R(r, x_0, x_1) \rightarrow \neg \tau_R(r, x_1, x_0)) \]

Assertions:

\[ \tau(C(a)) = \tau_C(C, x_0)[x_0/a] \]
\[ \tau(r(a, b)) = \tau_R(r, x_0, x_1)[x_0/a][x_1/b] \]
\[ \tau(\neg r(a, b)) = \neg \tau(r(a, b)) \]
\[ \tau(a \approx b) = a = b \]
\[ \tau(a \not\approx b) = \neg(a = b) \]
Description Logics Nomenclature
Naming Scheme for Expressive DLs

$((\text{ALC} \mid S)[\mathcal{H}] \mid SR)[\mathcal{O}][\mathcal{I}][\mathcal{F} \mid \mathcal{N} \mid \mathcal{Q}]$

- $S$ stands for ALC + role transitivity
- $\mathcal{H}$ stands for role hierarchies
- $\mathcal{O}$ stands for nominals, i.e., closed classes $\{o\}$ such as $\{john, mary, tom\}$
- $\mathcal{I}$ stands for inverse roles (seen soon)
- $\mathcal{F}$ stands for role functionality ($\top \sqsubseteq \leq 1.r$)
- $\mathcal{N}$ ($\mathcal{Q}$) stands for arbitrary (qualified) cardinality restrictions
- $\mathcal{R}$ stands for role box with all kinds of role axioms plus self concepts

Note:

- $S$ subsumes ALC, $SR$ subsumes $(\text{ALC} \mid S)[\mathcal{H}]$ ALC\text{H}
- $\text{SROIQ}$ subsumes all the other description logics in this scheme.
- $\mathcal{N}$ makes $\mathcal{F}$ obsolete
- $\mathcal{Q}$ makes $\mathcal{N}$ (and $\mathcal{F}$) obsolete
### DL Syntax - Overview

#### Concepts

<table>
<thead>
<tr>
<th></th>
<th>Concepts</th>
<th>( ALC )</th>
<th>( Q (\mathcal{N}) )</th>
<th>( O )</th>
<th>( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic</td>
<td>( A, B )</td>
<td>( A, B )</td>
<td>( \geq n ; r.C ) (( \geq n ; r ))</td>
<td>( {i_1, \ldots, i_n} )</td>
<td>( \exists r.\text{Self} )</td>
</tr>
<tr>
<td>Not</td>
<td>( \neg C )</td>
<td>( \neg C )</td>
<td>( \geq n ; r.C ) (( \geq n ; r ))</td>
<td>( {i_1, \ldots, i_n} )</td>
<td>( \exists r.\text{Self} )</td>
</tr>
<tr>
<td>And</td>
<td>( C \sqcap D )</td>
<td>( C \sqcap D )</td>
<td>( \geq n ; r.C ) (( \geq n ; r ))</td>
<td>( {i_1, \ldots, i_n} )</td>
<td>( \exists r.\text{Self} )</td>
</tr>
<tr>
<td>Or</td>
<td>( C \sqcup D )</td>
<td>( C \sqcup D )</td>
<td>( \leq n ; r.C ) (( \leq n ; r ))</td>
<td>( {i_1, \ldots, i_n} )</td>
<td>( \exists r.\text{Self} )</td>
</tr>
<tr>
<td>Exists</td>
<td>( \exists r.C )</td>
<td>( \exists r.C )</td>
<td>( \geq n ; r.C ) (( \geq n ; r ))</td>
<td>( {i_1, \ldots, i_n} )</td>
<td>( \exists r.\text{Self} )</td>
</tr>
<tr>
<td>For all</td>
<td>( \forall r.C )</td>
<td>( \forall r.C )</td>
<td>( \leq n ; r.C ) (( \leq n ; r ))</td>
<td>( {i_1, \ldots, i_n} )</td>
<td>( \exists r.\text{Self} )</td>
</tr>
</tbody>
</table>

#### Roles

<table>
<thead>
<tr>
<th>( I )</th>
<th>Atomic</th>
<th>( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverse</td>
<td>( r^- )</td>
<td></td>
</tr>
</tbody>
</table>
DL Syntax - Overview ctd.

Ontology (=Knowledge Base)

Concept Axioms \( TBox \)

<table>
<thead>
<tr>
<th>Subclass</th>
<th>( C \sqsubseteq D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equivalent</td>
<td>( C \equiv D )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assertional Axioms ( ABox )</th>
<th>Role Axioms ( RBox )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>( C(a) )</td>
</tr>
<tr>
<td>Role</td>
<td>( r(a, b) )</td>
</tr>
<tr>
<td>Same</td>
<td>( a \approx b )</td>
</tr>
<tr>
<td>Different</td>
<td>( a \not\approx b )</td>
</tr>
<tr>
<td>Subrole</td>
<td>( r \sqsubseteq s )</td>
</tr>
<tr>
<td>Transitivity</td>
<td>( \text{Trans}(r) )</td>
</tr>
<tr>
<td>Role Chain</td>
<td>( r \circ r' \sqsubseteq s )</td>
</tr>
<tr>
<td>Role Disjointness</td>
<td>( \text{Disj}(s, r) )</td>
</tr>
</tbody>
</table>

- Transitivity and Disjointness are role characteristics.
- Further characteristics in \( SROIQ \) are asymmetry, \( \text{Asym}(r) \), and reflexivity, \( \text{Ref}(r) \).
Semantics of Description Logics

Concept Equivalences

\( C \equiv D \)

Two concept expressions \( C \) and \( D \) are called equivalent (written: \( C \equiv D \)), if for every interpretation \( I \) holds \( C^I = D^I \).

- **Commutativity, Associativity, Idempotence:**
  
  \[
  \begin{align*}
  C \sqcap D & \equiv D \sqcap C \\
  (C \sqcap D) \sqcap E & \equiv C \sqcap (D \sqcap E) \\
  C \sqcap C & \equiv C \\
  C \sqcup D & \equiv D \sqcup C \\
  (C \sqcup D) \sqcup E & \equiv C \sqcup (D \sqcup E) \\
  C \sqcup C & \equiv C
  \end{align*}
  \]

- **Double Negation:**
  
  \( \neg \neg C \equiv C \)

- **Complement, de Morgan laws:**
  
  \[
  \begin{align*}
  \neg \top & \equiv \bot \\
  C \sqcap \neg C & \equiv \bot \\
  \neg (C \sqcap D) & \equiv \neg D \sqcup \neg C \\
  \neg \bot & \equiv \top \\
  C \sqcup \neg C & \equiv \top \\
  \neg (C \sqcup D) & \equiv \neg D \sqcap \neg C
  \end{align*}
  \]
Concept Equivalences ctd.

• Distributivity, Absorption:

\[(C \sqcup D) \cap E \equiv (C \cap E) \sqcup (D \cap E)\]
\[(C \cap D) \sqcup E \equiv (C \sqcup E) \cap (D \sqcup E)\]
\[C \sqcup (C \cap D) \equiv C\]

\[(C \sqcup D) \cap C \equiv C\]
\[(C \cap D) \sqcup C \equiv C\]
\[C \sqcup (C \cap D) \equiv C\]

• Quantifiers and Counting:

\[\neg \exists r.C \equiv \forall r.\neg C\]
\[\neg \forall r.C \equiv \exists r.\neg C\]
\[\neg \leq nr.C \equiv \geq (n + 1)r.C\]
\[\neg \geq (n + 1)r.C \equiv \leq nr.C\]
\[\geq 0r.C \equiv \top\]
\[\geq 1r.C \equiv \exists r.C\]
\[\leq 0r.C \equiv \forall r.\neg C\]
Axiom and KB Equivalences

- Lloyd-Topor equivalences

\[ \{ A \sqcup B \sqsubseteq C \} \iff \{ A \sqsubseteq C, B \sqsubseteq C \} \]
\[ \{ A \sqsubseteq B \sqcap C \} \iff \{ A \sqsubseteq B, A \sqsubseteq C \} \]

- Turning GCIs into universally valid concept descriptions

\[ C \sqsubseteq D \iff \top \sqsubseteq \neg C \sqcup D \]

- Internalisation of ABox into TBox

\[ C(a) \iff \{ a \} \sqsubseteq C \]
\[ r(a, b) \iff \{ a \} \sqsubseteq \exists r,\{ b \} \]
\[ \neg r(a, b) \iff \{ a \} \sqsubseteq \neg \exists r,\{ b \} \]
\[ a \approx b \iff \{ a \} \sqsubseteq \{ b \} \]
\[ a \not\approx b \iff \{ a \} \sqsubseteq \neg \{ b \} \]
(Non-)Concept Equivalences

Exercise: Show that the following equivalences are not valid.

\[ \exists r. (C \cap D) \equiv \exists r.C \cap \exists r.D \quad (1) \]
\[ C \cap (D \cup E) \equiv (C \cap D) \cup E \quad (2) \]
\[ \exists r.\{a\} \cap \exists r.\{b\} \equiv \geq 2 r.\{a, b\} \quad (3) \]
\[ \exists r.\top \cap \exists s.\top \equiv \exists r.\exists r^-s.\top \quad (4) \]

Exercise: Show that the following equivalences are valid.

\[ \exists r^-C \sqsubseteq D \equiv C \sqsubseteq \forall r.D \quad (5) \]
\[ C \sqsubseteq \forall r^-D \equiv \exists r.C \sqsubseteq D \quad (6) \]
**Concept Subsumption**

$C \sqsubseteq D$

A concept expression $C$ is **subsumed by** a concept expression $D$ (written: $C \sqsubseteq D$), if for every interpretation $\mathcal{I}$ holds $C^\mathcal{I} \subseteq D^\mathcal{I}$.

Some elementary properties:

- $C \sqsubseteq D \iff C \equiv C \sqcup D$
- $C \equiv D \iff C \sqsubseteq D$ and $D \sqsubseteq C$
- $C \sqsubseteq D$ and $D \sqsubseteq E$ implies $C \sqsubseteq E$ (transitivity)
- $C \sqsubseteq D \iff \neg D \sqsubseteq \neg C$
- $C \sqsubseteq D$ implies $C \sqcap E \sqsubseteq D$
- $C \equiv D$ implies $C \sqcap E \equiv D \sqcap E$
Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors.  
_The Description Logic Handbook: Theory, Implementation and Applications._  

Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph.  
_Foundations of Semantic Web Technologies._  

Sebastian Rudolph.  
Foundations of description logics.  