Chapter 2

A Basic Description Logic

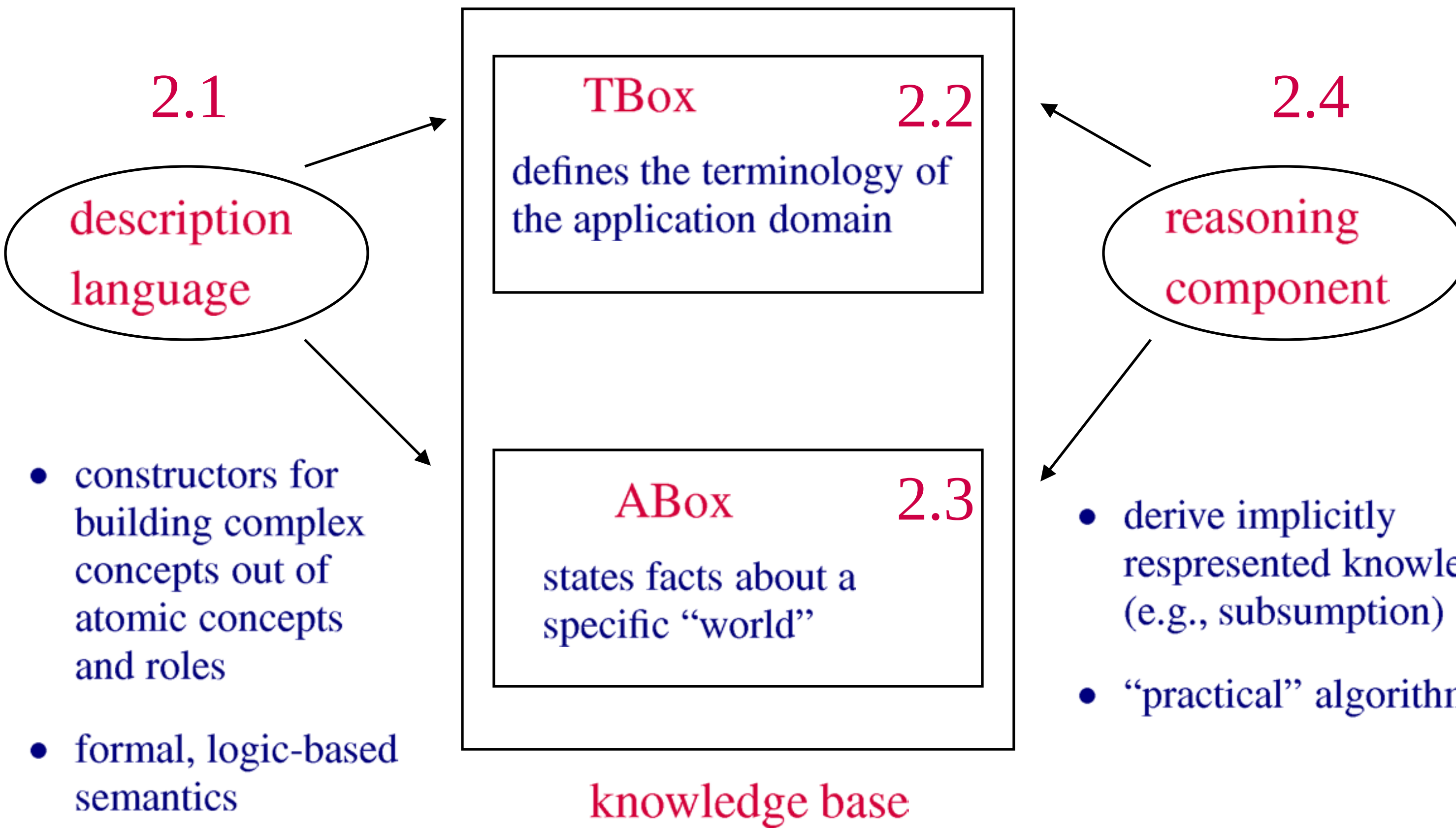
$\mathcal{ALC}$ attributive language with complement

Naming scheme:

- basic language $\mathcal{AL}$
- extended with contructors whose “letter” is added after the $\mathcal{AL}$
- $\mathcal{C}$ stands for complement, i.e., $\mathcal{ALC}$ is obatined from $\mathcal{AL}$ by adding the complement ($\neg$) operator

Description logic system

structure



- constructors for building complex concepts out of atomic concepts and roles
- formal, logic-based semantics

- derive implicitly respresented knowledge (e.g., subsumption)
- “practical” algorithms

Dresden

2.1. The description language

Definition 2.1 (Syntax of $\mathcal{ALC}$)

Let N_C and N_R be disjoint sets of concept names and role names, respectively.

$\mathcal{ALC}$-concept descriptions are defined by induction:

- If $A \in N_C$, then A is an $\mathcal{ALC}$-concept description.
- If C, D are $\mathcal{ALC}$-concept descriptions, and $r \in N_R$, then the following are $\mathcal{ALC}$-concept descriptions:
  - $C \sqcap D$ (conjunction)
  - $C \sqcup D$ (disjunction)
  - $\neg C$ (negation)
  - $\forall r.C$ (value restriction)
  - $\exists r.C$ (existential restriction)

Abbreviations:

- $\top := A \sqcup \neg A$ (top)
- $\bot := A \sqcap \neg A$ (bottom)
- $C \Rightarrow D := \neg C \sqcup D$ (implication)

Dresden

Notation (use and abuse):

- concept names are called atomic
- all other descriptions are called complex
- instead of $\mathcal{ALC}$-concept description we often say $\mathcal{ALC}$-concept or concept description or concept
- A, B often used for concept names, C, D for complex concept descriptions, r, s for role names

Dresden

The description language

examples of $\mathcal{ALC}$-concept descriptions

$\text{Person} \sqcap \text{Female}$

$\text{Participant} \sqcap \exists \text{attends.Talk}$

$\text{Participant} \sqcap \forall \text{attends.}(\text{Talk} \sqcap \neg \text{Boring})$

$\text{Speaker} \sqcap \exists \text{gives.}(\text{Talk} \sqcap \forall \text{topic.DL})$

$\text{Speaker} \sqcap \forall \text{gives.}(\text{Talk} \sqcap \exists \text{topic.}(\text{DL} \sqcup \text{FuzzyLogic}))$

Definition 2.2 (Semantics of $\mathcal{ALC}$)

An interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty domain $\Delta^\mathcal{I}$ and an extension mapping $\cdot^\mathcal{I}$:

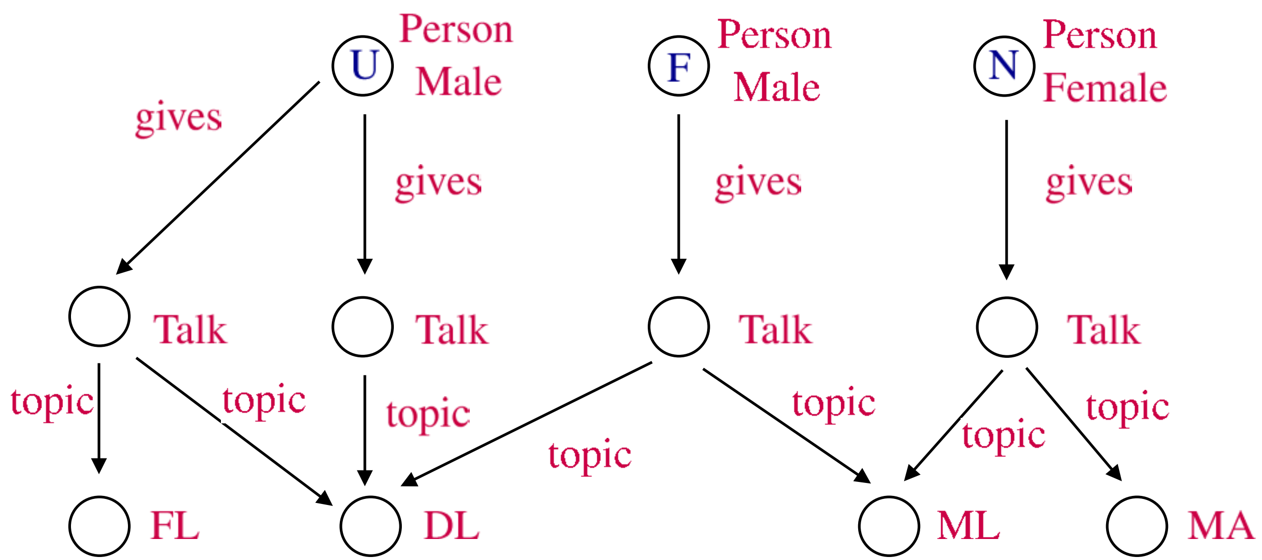
- $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ for all $A \in N_C$, concepts interpreted as sets
- $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ for all $r \in N_R$. roles interpreted as binary relations

The extension mapping is extended to complex $\mathcal{ALC}$-concept descriptions as follows:

- $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$
- $(C \sqcup D)^\mathcal{I} := C^\mathcal{I} \cup D^\mathcal{I}$
- $(\neg C)^\mathcal{I} := \Delta^\mathcal{I} \setminus C^\mathcal{I}$
- $(\forall r.C)^\mathcal{I} := \{d \in \Delta^\mathcal{I} \mid \text{for all } e \in \Delta^\mathcal{I} : (d, e) \in r^\mathcal{I} \text{ implies } e \in C^\mathcal{I}\}$
- $(\exists r.C)^\mathcal{I} := \{d \in \Delta^\mathcal{I} \mid \text{there is } e \in \Delta^\mathcal{I} : (d, e) \in r^\mathcal{I} \text{ and } e \in C^\mathcal{I}\}$

Example

of an interpretation



$$\text{Person} \sqcap \exists \text{gives}.(\text{Talk} \sqcap \forall \text{topic}.\text{DL})$$

$$\text{Person} \sqcap \forall \text{gives}.(\text{Talk} \sqcap \exists \text{topic}.\text{DL})$$

Relationship with First-Order Logic

$\mathcal{ALC}$ can be seen as a fragment of first-order logic:

- Concept names are unary predicates, and role names are binary predicates.
- Interpretations for $\mathcal{ALC}$ can then obviously be viewed as first-order interpretations for this signature.
- Concept descriptions correspond to first-order formulae with one free variable.
- Given such a formula $\phi(x)$ with the free variable x and an interpretation $\mathcal{I}$, the extension of ϕ w.r.t. $\mathcal{I}$ is given by

$$\phi^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \mathcal{I} \models \phi(d)\}$$

- Goal: translate $\mathcal{ALC}$-concepts C into first-order formulae $\tau_x(C)$ such that their extensions coincide.

Relationship with First-Order Logic

Concept description C translated into formula with one free variable $\tau_x(C)$:

- $\tau_x(A) := A(x)$ for $A \in N_C$
- $\tau_x(C \sqcap D) := \tau_x(C) \wedge \tau_x(D)$
- $\tau_x(C \sqcup D) := \tau_x(C) \vee \tau_x(D)$
- $\tau_x(\neg C) := \neg\tau_x(C)$
- $\tau_x(\forall r.C) := \forall y.(r(x,y) \rightarrow \tau_y(C))$
- $\tau_x(\exists r.C) := \exists y.(r(x,y) \wedge \tau_y(C))$

y variable different from x

$$\begin{aligned}\tau_x(\forall r.(A \sqcap \exists r.B)) &= \forall y.(r(x,y) \rightarrow \tau_y(A \sqcap \exists r.B)) \\ &= \forall y.(r(x,y) \rightarrow (A(y) \wedge \exists z.(r(y,z) \wedge B(z))))\end{aligned}$$

Relationship with First-order Logic

Concept description C translated into formula with one free variable $\tau_x(C)$:

- $\tau_x(A) := A(x)$ for $A \in N_C$
- $\tau_x(C \sqcap D) := \tau_x(C) \wedge \tau_x(D)$
- $\tau_x(C \sqcup D) := \tau_x(C) \vee \tau_x(D)$
- $\tau_x(\neg C) := \neg\tau_x(C)$
- $\tau_x(\forall r.C) := \forall y.(r(x,y) \rightarrow \tau_y(C))$
- $\tau_x(\exists r.C) := \exists y.(r(x,y) \wedge \tau_y(C))$

y variable different from x

Lemma 2.3

C and $\tau_x(C)$ have the same extension, i.e.,

$$C^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \mathcal{I} \models \tau_x(C)(d)\}$$

Proof: induction on the structure of C

Relationship with First-Order Logic

$\mathcal{ALC}$ can be seen as a fragment of first-order logic:

- Concept names are unary predicates, and role names are binary predicates.
- Concept descriptions C yield formulae with one free variable $\tau_x(C)$.

These formulae belong to known decidable subclasses of first-order logic:

– two-variable fragment

– guarded fragment

$$\begin{aligned}\tau_x(\forall r.(A \sqcap \exists r.B)) &= \forall y.(r(x,y) \rightarrow \tau_y(A \sqcap \exists r.B)) \\ &= \forall y.(r(x,y) \rightarrow (A(y) \wedge \exists x.(r(y,x) \wedge B(x))))\end{aligned}$$

Relationship with Modal Logic

multimodal K:

several pairs of boxes and diamonds

$\mathcal{ALC}$ is a syntactic variant of the basic modal logic K:

- Concept names are propositional variables, and role names are names for transition relations.
- Concept descriptions C yield modal formulae $\theta(C)$:
  - $\theta(A) := a$ for $A \in N_C$
  - $\theta(C \sqcap D) := \theta(C) \wedge \theta(D)$
  - $\theta(C \sqcup D) := \theta(C) \vee \theta(D)$
  - $\theta(\neg C) := \neg\theta(C)$
  - $\theta(\forall r.C) := \Box_r \theta(C)$
  - $\theta(\exists r.C) := \Diamond_r \theta(C)$

C and $\theta(C)$ have the same semantics: $C^{\mathcal{I}}$ is the set of worlds that make $\theta(C)$ true in the Kripke structure described by $\mathcal{I}$.

Additional constructors

$\mathcal{ALC}$ is only an example of a description logic.

DL researchers have introduced and investigated many additional constructors.

Example letter $\mathcal{Q}$ in the naming scheme

Qualified number restrictions: $(\geq n\, r.C)$, $(\leq n\, r.C)$ with semantics

$$(\geq n\, r.C)^{\mathcal{I}} \;:=\; \{d \in \Delta^{\mathcal{I}} \mid \mathrm{card}(\{e \mid (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}) \geq n\}$$

$$(\leq n\, r.C)^{\mathcal{I}} \;:=\; \{d \in \Delta^{\mathcal{I}} \mid \mathrm{card}(\{e \mid (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}) \leq n\}$$

Persons that attend at most 20 talks, of which at least 3 have the topic DL:

$$\mathsf{Person} \sqcap (\leq 20\, \mathsf{attends}.\mathsf{Talk}) \sqcap (\geq 3\, \mathsf{attends}.(\mathsf{Talk} \sqcap \exists \mathsf{topic}.\mathsf{DL}))$$

Additional constructors

$\mathcal{ALC}$ is only an example of a description logic.

DL researchers have introduced and investigated many additional constructors.

Example

letter $\mathcal{Q}$ in the naming scheme

Qualified number restrictions: $(\geq n\, r.C)$, $(\leq n\, r.C)$ with semantics

$$(\geq n\, r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{card}(\{e \mid (d,e) \in r^{\mathcal{I}} \land e \in C^{\mathcal{I}}\}) \geq n\}$$
$$(\leq n\, r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{card}(\{e \mid (d,e) \in r^{\mathcal{I}} \land e \in C^{\mathcal{I}}\}) \leq n\}$$

Number restrictions: $(\geq n\, r)$, $(\leq n\, r)$ as abbreviation for $(\geq n\, r.\top)$ and $(\leq n\, r.\top)$:

$$(\geq n\, r)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{card}(\{e \mid (d,e) \in r^{\mathcal{I}}\}) \geq n\}$$
$$(\leq n\, r)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{card}(\{e \mid (d,e) \in r^{\mathcal{I}}\}) \leq n\}$$

letter $\mathcal{N}$ in the naming scheme

Additional constructors

In addition to concept constructors, one can also introduce role constructors.

Example letter $\mathcal{I}$ in the naming scheme

Inverse roles: if r is a role, then r^{-1} denotes its inverse

$$(r^{-1})^{\mathcal{I}} := \{(e,d) \mid (d,e) \in r^{\mathcal{I}}\}$$

Inverse roles can be used like role names in value and existential restrictions.

Presenter of a boring talk:

$$\text{Speaker} \sqcap \exists \text{gives}.(\text{Talk} \sqcap \forall \text{attends}^{-1}.(\text{Bored} \sqcup \text{Sleeping}))$$

Description logic system

structure

2.1 description language

- constructors for building complex concepts out of atomic concepts and roles
- formal, logic-based semantics

knowledge base

TBox 2.2

defines the terminology of the application domain

ABox 2.3

states facts about a specific “world”

2.4 reasoning component

- derive implicitly respresented knowledge (e.g., subsumption)
- “practical” algorithms

Dresden

Definition 2.4 (GCIs and TBoxes)

- A general concept inclusion is of the form $C \sqsubseteq D$ where C, D are concept descriptions.
- A TBox is a finite set of GCIs.
- The interpretation $\mathcal{I}$ satisfies the GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- The interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ iff it satisfies all the GCIs in $\mathcal{T}$.

Note: this definition is not specific for $\mathcal{ALC}$.

It applies also to other concept description languages.

Definition 2.4 (GCIs and TBoxes)

- A general concept inclusion is of the form $C \sqsubseteq D$ where C, D are concept descriptions.
- A TBox is a finite set of GCIs.
- The interpretation $\mathcal{I}$ satisfies the GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- The interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ iff it satisfies all the GCIs in $\mathcal{T}$.

$$\mathsf{Talk} \sqcap \forall \mathsf{attends}^{-1}.\mathsf{Sleeping} \sqsubseteq \mathsf{Boring}$$

$$\mathsf{Author} \sqcap \mathsf{PCchair} \sqsubseteq \bot$$

Definition 2.4 (GCIs and TBoxes)

- A general concept inclusion is of the form $C \sqsubseteq D$ where C, D are concept descriptions.
- A TBox is a finite set of GCIs.
- The interpretation $\mathcal{I}$ satisfies the GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- The interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ iff it satisfies all the GCIs in $\mathcal{T}$.

Notation: two TBoxes are called equivalent if they have the same models

Restricted TBoxes

Definition 2.5

A concept definition is of the form $A \equiv C$ where

- A is a concept name;
- C is a concept description.

The interpretation $\mathcal{I}$ satisfies the concept definition $A \equiv C$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$.

abbreviation for the two GCIs
$A \sqsubseteq C$ and $C \sqsubseteq A$

Dresden

Restricted TBoxes

concept definitions and acyclic TBoxes

Definition 2.5 (continued)

An acyclic TBox is a finite set of concept definitions that

- does not contain multiple definitions;
- does not contain cyclic definitions.

multiple definition

$$\begin{array}{lcl} A & \equiv & C \\ A & \equiv & D \end{array} \quad \text{for } C \neq D$$

$$\begin{array}{lcl} A & \equiv & B \sqcap \forall r.P \\ B & \equiv & P \sqcap \forall r.C \\ C & \equiv & \exists r.A \end{array}$$

cyclic definition

No cyclic definitions:
there is no sequence $A_1 \equiv C_1, \ldots A_n \equiv C_n \in \mathcal{T}$ $(n \geq 1)$ such that

- A_{i+1} occurs in C_i $(1 \leq i < n)$
- A_1 occurs in C_n

Dresden

Restricted TBoxes

Definition 2.5 (continued)

An acyclic TBox is a finite set of concept definitions that

- does not contain multiple definitions;
- does not contain cyclic definitions.

The interpretation $\mathcal{I}$ is a model of the acyclic TBox $\mathcal{T}$ iff it satisfies all its concept definitions: $A^{\mathcal{I}} = C^{\mathcal{I}}$ for all $A \equiv C \in \mathcal{T}$

Given an acyclic TBox, we call a concept name A occurring in $\mathcal{T}$ a

- defined concept iff there is C such that $A \equiv C \in \mathcal{T}$;
- primitive concept otherwise.

Example

of an acyclic TBox

$$
\begin{aligned}
\text{Woman} &\equiv \text{Person} \sqcap \text{Female}\\
\text{Man} &\equiv \text{Person} \sqcap \neg\text{Female}\\
\text{Talk} &\equiv \exists\text{topic}.\top\\
\text{Speaker} &\equiv \text{Person} \sqcap \exists\text{gives}.\text{Talk}\\
\text{Participant} &\equiv \text{Person} \sqcap \exists\text{attends}.\text{Talk}\\
\text{BusySpeaker} &\equiv \text{Speaker} \sqcap (\geq 3\ \text{gives}.\text{Talk})\\
\text{BadSpeaker} &\equiv \text{Speaker} \sqcap \forall\text{gives}.(\forall\text{attends}^{-1}.(\text{Bored} \sqcup \text{Sleeping}))
\end{aligned}
$$

Acyclic TBoxes

an important result

Proposition 2.6

For every acyclic TBox $\mathcal{T}$ we can effectively construct an equivalent acyclic TBox $\widehat{\mathcal{T}}$ such that the right-hand sides of concept definitions in $\widehat{\mathcal{T}}$ contain only primitive concepts.

Proof: blackboard

Dresden

Acyclic TBoxes

Proposition 2.6

For every acyclic TBox $\mathcal{T}$ we can effectively construct an equivalent acyclic TBox $\widehat{\mathcal{T}}$ such that the right-hand sides of concept definitions in $\widehat{\mathcal{T}}$ contain only primitive concepts.

We call $\widehat{\mathcal{T}}$ the *expanded version* of $\mathcal{T}$.

Acyclic TBoxes

an important result

Given an acyclic TBox $\mathcal{T}$, a primitive interpretation $\mathcal{J}$ for $\mathcal{T}$ consists of a nonempty set $\Delta^{\mathcal{J}}$ together with an extension mapping $\cdot^{\mathcal{J}}$, that maps

- primitive concepts P to sets $P^{\mathcal{J}} \subseteq \Delta^{\mathcal{J}}$
- role names r to binary relations $r^{\mathcal{J}} \subseteq \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$

The interpretation $\mathcal{I}$ is an extension of the primitive interpretation $\mathcal{J}$ iff $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ and

- $P^{\mathcal{J}} = P^{\mathcal{I}}$ for all primitive concepts P
- $r^{\mathcal{J}} = r^{\mathcal{I}}$ for all role names r

Corollary 2.7

Let $\mathcal{T}$ be an acyclic TBox.
Any primitive interpretation $\mathcal{J}$ has a unique extension to a model of $\mathcal{T}$.

Proof: blackboard

Relationship with First-Order Logic

$\mathcal{ALC}$-TBoxes can be be translated into first-order logic:

$$\tau(\mathcal{T}) := \bigwedge_{C \sqsubseteq D \in \mathcal{T}} \forall x.(\tau_x(C) \rightarrow \tau_x(D))$$

Lemma 2.8

Let $\mathcal{T}$ be a TBox and $\tau(\mathcal{T})$ its translation into first-order logic.
Then $\mathcal{T}$ and $\tau(\mathcal{T})$ have the same models.

Proof: blackboard

Description logic system

structure

2.1 description language

- constructors for building complex concepts out of atomic concepts and roles
- formal, logic-based semantics

knowledge base

TBox 2.2

defines the terminology of the application domain

ABox 2.3

states facts about a specific “world”

2.4 reasoning component

- derive implicitly respresented knowledge (e.g., subsumption)
- “practical” algorithms

2.3. Assertional knowledge

Definition 2.9 (Assertions and ABoxes)

An assertion is of the form

$$C(a) \text{ (concept assertion)} \quad \text{or} \quad r(a,b) \text{ (role assertion)}$$

where C is a concept description, r is a role, and a, b are individual names from a set N_I of such names (disjoint with N_C and N_R).

An ABox is a finite set of assertions.

An interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$ if it satisfies all its assertions:

$$\begin{array}{ll} a^{\mathcal{I}} \in C^{\mathcal{I}} & \text{for all } C(a) \in \mathcal{A} \\ (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}} & \text{for all } r(a,b) \in \mathcal{A} \end{array}$$

$\mathcal{I}$ assigns elements $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ to individual names $a \in N_I$

Dresden

2.3. Assertional knowledge

Definition 2.9 (Assertions and ABoxes)

An assertion is of the form

$C(a)$ (concept assertion) or $r(a, b)$ (role assertion)

where C is a concept description, r is a role, and a, b are individual names from a set N_I of such names (disjoint with N_C and N_R).

An ABox is a finite set of assertions.

Lecturer(FRANZ), teaches(FRANZ, TU03),
Tutorial(TU03), topic(TU03, RinDL),
DL(RinDL)

Relationship with First-Order Logic

$\mathcal{ALC}$-ABoxes can be be translated into first-order logic:

$$\tau(\mathcal{A}) := \bigwedge_{C(a)\in\mathcal{T}} \tau_x(C)(a) \wedge \bigwedge_{r(a,b)\in\mathcal{T}} r(a,b)$$

individual names are viewed as constants

<u>Lemma 2.10</u>

Let $\mathcal{A}$ be a TBox and $\tau(\mathcal{A})$ its translation into first-order logic.
Then $\mathcal{A}$ and $\tau(\mathcal{A})$ have the same models.

Proof: easy

Knowledge Bases

Definition 2.11

A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

The interpretation $\mathcal{I}$ is a model of the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ iff it is a model of $\mathcal{T}$ and a model of $\mathcal{A}$.

First-order translation: $\tau(\mathcal{K}) := \tau(\mathcal{T}) \wedge \tau(\mathcal{A})$

Lemma 2.12

Let $\mathcal{K}$ be a knowledge base and $\tau(\mathcal{K})$ its translation into first-order logic. Then $\mathcal{K}$ and $\tau(\mathcal{K})$ have the same models.

Proof: immediate consequence of Lemma 2.8 and Lemma 2.10

Additional constructors

Individual names can also be used as concept constructors to increase the expressive power of the concept description language.

They yield a singleton set consisting of the extension of the individual name.

Nominals

letter $\mathcal{O}$ in the naming scheme

Nominals: $\{a\}$ for $a \in N_I$ with semantics

$$\{a\}^{\mathcal{I}} \quad := \quad \{a^{\mathcal{I}}\}$$

Nominals can be used to express ABox assertions using GCIs:

$C(a)$ is expressed by $\{a\} \sqsubseteq C$

$r(a, b)$ is expressed by $\{a\} \sqsubseteq \exists r.\{b\}$

Description logic system

structure

2.1 description language

- constructors for building complex concepts out of atomic concepts and roles
- formal, logic-based semantics

knowledge base

TBox 2.2

defines the terminology of the application domain

ABox 2.3

states facts about a specific "world"

2.4 reasoning component

- derive implicitly respresented knowledge (e.g., subsumption)
- "practical" algorithms

Dresden

2.4. Reasoning Problems and Services

make implicitly represented knowledge explicit

Definition 2.13 (terminological reasoning)

Let $\mathcal{T}$ be a TBox.

Satisfiability:

C is satisfiable w.r.t. $\mathcal{T}$ iff $C^{\mathcal{I}} \neq \emptyset$ for some model $\mathcal{I}$ of $\mathcal{T}$.

Subsumption:

C is subsumed by D w.r.t. $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) iff

$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of the TBox $\mathcal{T}$.

Equivalence:

C is equivalent to D w.r.t. $\mathcal{T}$ ($C \equiv_{\mathcal{T}} D$) iff

$C^{\mathcal{I}} = D^{\mathcal{I}}$ for all models $\mathcal{I}$ of the TBox $\mathcal{T}$.

Terminological Reasoning

Note:

If $\mathcal{T} = \emptyset$, then satisfiability/subsumption/equivalence w.r.t. $\mathcal{T}$ is simply called satisfiability/subsumption/equivalence and we write $\sqsubseteq$ and $\equiv$.

Examples:

- $A \sqcap \neg A$ and $\forall r.A \sqcap \exists r.\neg A$ are not satisfiable (unsatisfiable)
- $A \sqcap \neg A$ and $\forall r.A \sqcap \exists r.\neg A$ are equivalent
- $A \sqcap B$ is subsumed by A and by B.
- $\exists r.(A \sqcap B)$ is subsumed by $\exists r.A$ and by $\exists r.B$
- $\forall r.(A \sqcap B)$ is equivalent to $\forall r.A \sqcap \forall r.B$
- $\exists r.A \sqcap \forall r.B$ is subsumed by $\exists r.(A \sqcap B)$

Properties of Subsumption

Lemma 2.14

- The subsumption relation $\sqsubseteq_\mathcal{T}$ is a pre-order on concept descriptions, i.e.,
  - $C \sqsubseteq_\mathcal{T} C$ (reflexive)
  - $C \sqsubseteq_\mathcal{T} D \wedge D \sqsubseteq_\mathcal{T} E \rightarrow C \sqsubseteq_\mathcal{T} E$ (transitive)

  It is not a partial order since it is not antisymmetric:

  - $C \sqsubseteq_\mathcal{T} D \wedge D \sqsubseteq_\mathcal{T} C \not\rightarrow C = D$

- The constructors existential restriction and value restriction are monotonic w.r.t. subsumption, i.e.,
  - $C \sqsubseteq_\mathcal{T} D \rightarrow \exists r.C \sqsubseteq_\mathcal{T} \exists r.D \wedge \forall r.C \sqsubseteq_\mathcal{T} \forall r.D$

Proof: blackboard

Assertional Reasoning

Definition 2.15 (assertional reasoning)

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base.

Consistency:

$\mathcal{K}$ is consistent iff there exists a model of $\mathcal{K}$.

Instance:

a is an instance of C w.r.t. $\mathcal{K}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$.

Lemma 2.16

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base.

If a is an instance of C w.r.t. $\mathcal{K}$ and $C \sqsubseteq_{\mathcal{T}} D$,

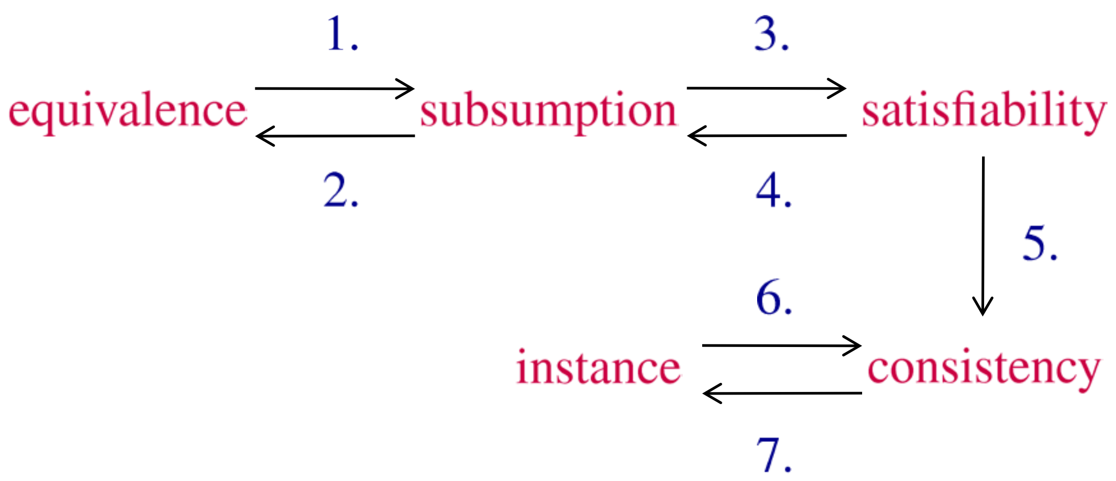
then a is an instance of D w.r.t. $\mathcal{K}$.

Proof: exercise

Reductions

between reasoning problems

There are the following polynomomial time reductions between the introduced reasoning problems:



This holds not only for $\mathcal{ALC}$, but for all DLs that have the constructors conjunction and negation.

Theorem 2.17

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, C, D concept descriptions, and $a \in N_I$.

1. $C \equiv_{\mathcal{T}} D$ iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$
2. $C \sqsubseteq_{\mathcal{T}} D$ iff $C \equiv_{\mathcal{T}} C \sqcap D$
3. $C \sqsubseteq_{\mathcal{T}} D$ iff $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$
4. C is satisfiable w.r.t. $\mathcal{T}$ iff $C \not\sqsubseteq_{\mathcal{T}} \bot$
5. C is satisfiable w.r.t. $\mathcal{T}$ iff $(\mathcal{T}, \{C(a)\})$ is consistent
6. a is an instance of C w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{\neg C(a)\})$ is inconsistent
7. $\mathcal{K}$ is consistent iff a is not an instance of $\bot$ w.r.t. $\mathcal{K}$

Proof: blackboard

Reduction

getting rid of acyclic TBoxes

Expansion of concepts and ABoxes:

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, where $\mathcal{T}$ is acyclic,
and C a concept description.

The expanded versions $\widehat{C}$ and $\widehat{\mathcal{A}}$ of C and $\mathcal{A}$ w.r.t. $\mathcal{T}$ are obtained as follows:

- replace all defined concepts occurring in C and $\mathcal{A}$ by their definitions in the expanded version $\widehat{\mathcal{T}}$ of $\mathcal{T}$.

$\mathcal{T}$

$$\begin{array}{rcl} \text{Woman} & \equiv & \text{Person} \sqcap \text{Female} \\ \text{Talk} & \equiv & \exists\text{topic}.\top \\ \text{Speaker} & \equiv & \text{Person} \sqcap \exists\text{gives}.\text{Talk} \end{array}$$

$C = \text{Woman} \sqcap \text{Speaker}$ expands to

$$\widehat{C} = \text{Person} \sqcap \text{Female} \sqcap \text{Person} \sqcap \exists\text{gives}.(\exists\text{topic}.\top)$$

Dresden

Reduction

getting rid of acyclic TBoxes

Expansion of concepts and ABoxes:

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, where $\mathcal{T}$ is acyclic, and C a concept description.

The expanded versions $\widehat{C}$ and $\widehat{\mathcal{A}}$ of C and $\mathcal{A}$ w.r.t. $\mathcal{T}$ are obtained as follows:

- replace all defined concepts occurring in C and $\mathcal{A}$ by their definitions in the expanded version $\widehat{\mathcal{T}}$ of $\mathcal{T}$.

Proposition 2.18

1. C is satisfiable w.r.t. $\mathcal{T}$ iff $\widehat{C}$ is satisfiable
2. $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent iff $(\emptyset, \widehat{\mathcal{A}})$ is consistent

Proof: blackboard

Dresden

Reduction

getting rid of acyclic TBoxes

Expansion of concepts and ABoxes:

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, where $\mathcal{T}$ is acyclic,
and C a concept description.

The expanded versions $\widehat{C}$ and $\widehat{\mathcal{A}}$ of C and $\mathcal{A}$ w.r.t. $\mathcal{T}$ are obtained as follows:

- replace all defined concepts occurring in C and $\mathcal{A}$ by their definitions in the expanded version $\widehat{\mathcal{T}}$ of $\mathcal{T}$.

Proposition 2.18

1. C is satisfiable w.r.t. $\mathcal{T}$ iff $\widehat{C}$ is satisfiable
2. $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent iff $(\emptyset, \widehat{\mathcal{A}})$ is consistent

Similar reductions exist for the other reasoning problems.

Reduction

getting rid of the TBox

This reduction is in general not polynomial,

since the expanded versions may be exponential in the size of $\mathcal{T}$.

$$\begin{array}{rcl} A_0 & \equiv & \forall r.A_1 \sqcap \forall s.A_1 \\ A_1 & \equiv & \forall r.A_2 \sqcap \forall s.A_2 \\ & \vdots & \\ A_{n-1} & \equiv & \forall r.A_n \sqcap \forall s.A_n \end{array}$$

The size of $\mathcal{T}$ is linear in n,

but the expansion version $\widehat{A}_0$ of A_0 contains A_n 2^n times.

Proof: induction on n

Dresden

Relationship with First-Order Logic

Reasoning in $\mathcal{ALC}$ can be translated into reasoning in first-order logic:

Lemma 2.19

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, C, D be $\mathcal{ALC}$-concept descriptions, and a an individual name.

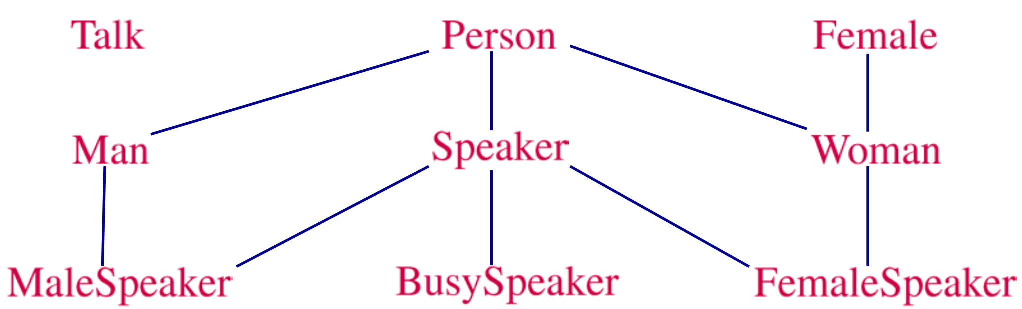
1. $C \sqsubseteq_{\mathcal{T}} D$ iff $\tau(\mathcal{T}) \models \forall x.(\tau_x(C)(x) \rightarrow \tau_x(D)(x))$
2. $\mathcal{K}$ is consistent iff $\tau(\mathcal{K})$ is consistent
3. a is an instance of C w.r.t. $\mathcal{K}$ iff $\tau(\mathcal{K}) \models \tau_x(C)(a)$

Proof: blackboard

Classification

Computing the subsumption hierarchy of all concept names occurring in the TBox.

$$
\begin{aligned}
\text{Man} &\equiv \text{Person} \sqcap \neg\text{Female} \\
\text{Woman} &\equiv \text{Person} \sqcap \text{Female} \\
\text{MaleSpeaker} &\equiv \text{Man} \sqcap \exists\text{gives.Talk} \\
\text{FemaleSpeaker} &\equiv \text{Woman} \sqcap \exists\text{gives.Talk} \\
\text{Speaker} &\equiv \text{FemaleSpeaker} \sqcup \text{MaleSpeaker} \\
\text{BusySpeaker} &\equiv \text{Speaker} \sqcap (\geq 3\ \text{gives.Talks})
\end{aligned}
$$



Realization

Computing the most specific concept names in the TBox to which an ABox individual belongs.

$$\begin{array}{rcl} \text{Man} & \equiv & \text{Person} \sqcap \neg\text{Female} \\ \text{Woman} & \equiv & \text{Person} \sqcap \text{Female} \\ \text{MaleSpeaker} & \equiv & \text{Man} \sqcap \exists \text{gives.Talk} \\ \text{FemaleSpeaker} & \equiv & \text{Woman} \sqcap \exists \text{gives.Talk} \\ \text{Speaker} & \equiv & \text{FemaleSpeaker} \sqcup \text{MaleSpeaker} \\ \text{BusySpeaker} & \equiv & \text{Speaker} \sqcap (\geq 3\, \text{gives.Talks}) \end{array}$$

$\text{Man}(\text{FRANZ}), \quad \text{gives}(\text{FRANZ}, \text{T1}),$
$\text{Talk}(\text{T1})$

FRANZ is an instance of Man, Speaker, <u>MaleSpeaker</u>.

most specific