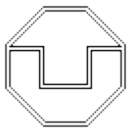$\mathcal{ALC}$ attributive language with complement
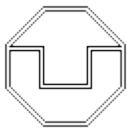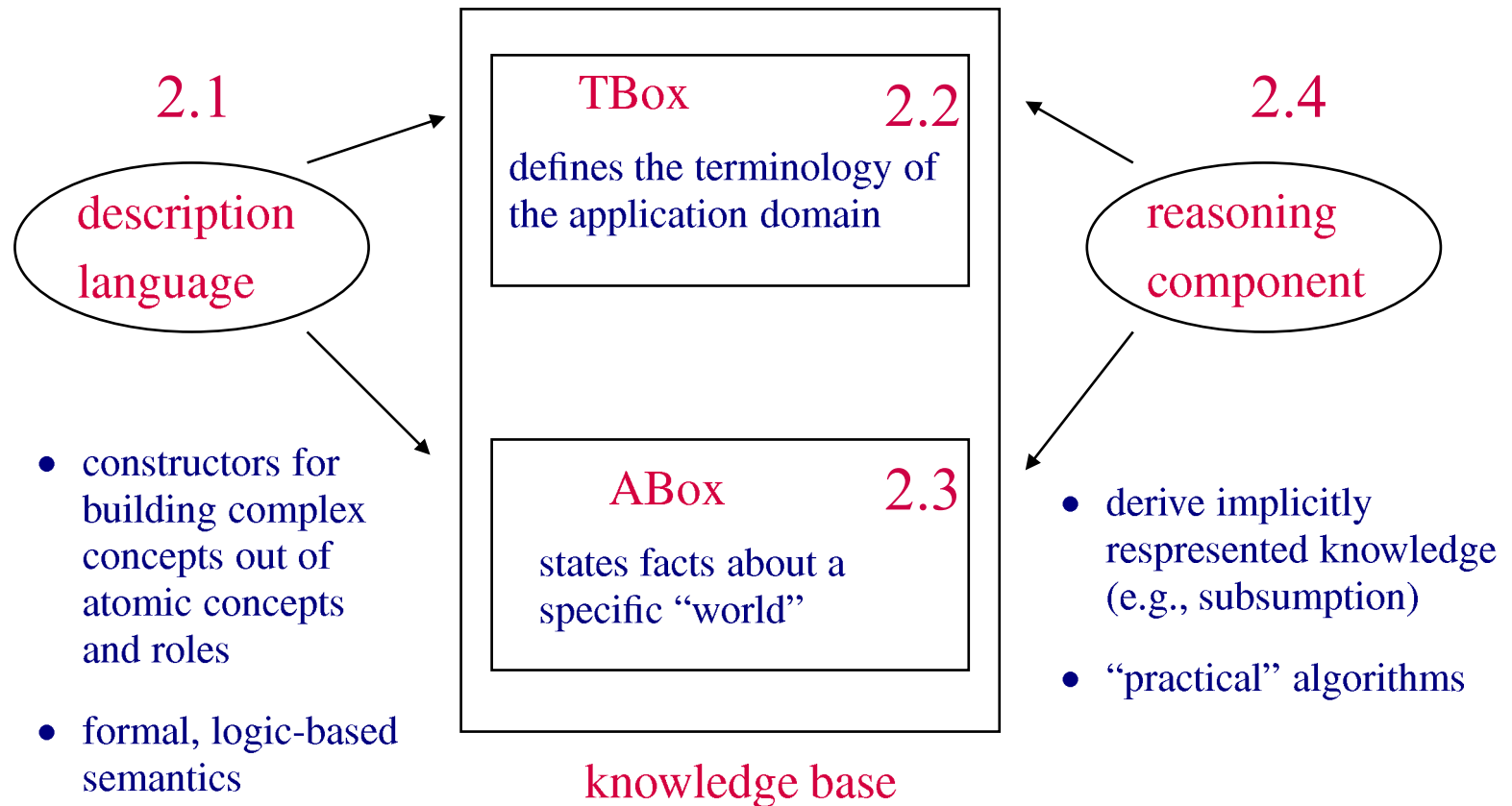[Schmidt-Schauß&Smolka, 1991]

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

Dresden

Description logic system

structure

2.1

description language

TBox **2.2**

defines the terminology of the application domain

2.4

reasoning component

- constructors for building complex concepts out of atomic concepts and roles

- formal, logic-based semantics

ABox **2.3**

states facts about a specific "world"

knowledge base

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

- "practical" algorithms

Dresden

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

Let **C** and **R** be disjoint sets of concept names and role names, respectively.

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

- If $A \in \mathbf{C}$, then A is an $\mathcal{ALC}$-concept description.

- If C, D are $\mathcal{ALC}$-concept descriptions, and $r \in \mathbf{R}$,
  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
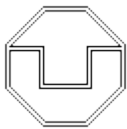
The description language
examples of $\mathcal{ALC}$-concept descriptions

Person ⊓ Female

Participant ⊓ ∃attends.Talk

Participant ⊓ ∀attends.(Talk ⊓ ¬Boring)

Speaker ⊓ ∃gives.(Talk ⊓ ∀topic.DL)

Speaker ⊓ ∀gives.(Talk ⊓ ∃topic.(DL ⊔ FuzzyLogic))

Dresden

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 $A \in \mathbf{C}$,   concepts interpreted as sets

- $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for all $r \in \mathbf{R}$.   roles interpreted as binary relations
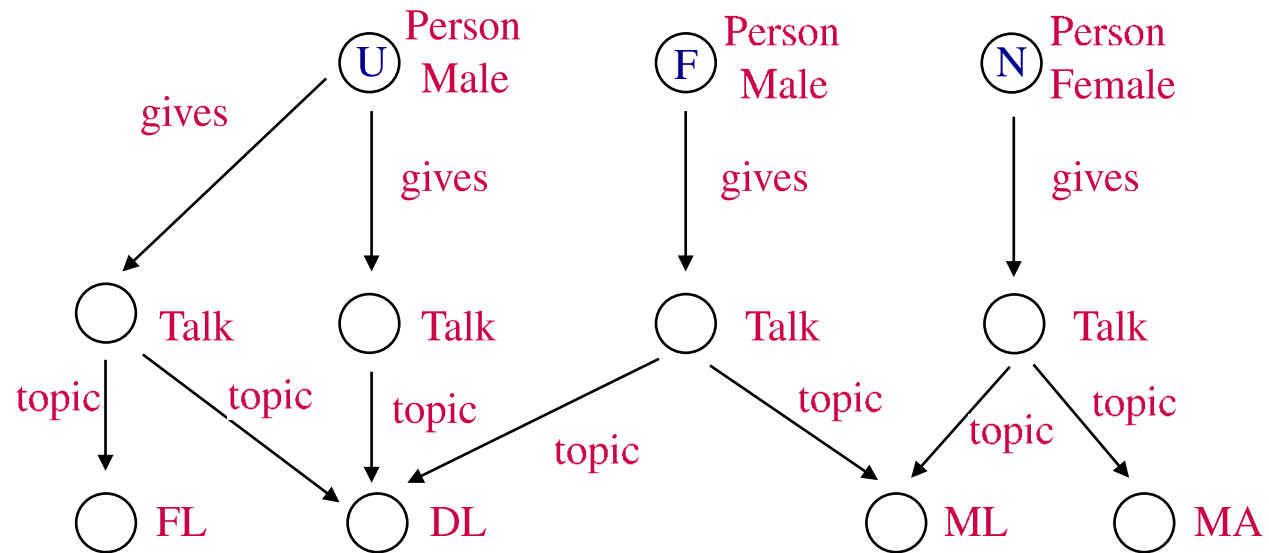
The extension mapping is extended to complex $\mathcal{ALC}$-concept descriptions as follows:

- $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$

- $(C \sqcup D)^{\mathcal{I}} := C^{\mathcal{I}} \cup D^{\mathcal{I}}$

- $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

- $(\forall r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{for all } e \in \Delta^{\mathcal{I}} : (d,e) \in r^{\mathcal{I}} \text{ implies } e \in C^{\mathcal{I}}\}$

- $(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{there is } e \in \Delta^{\mathcal{I}} : (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}}\}$
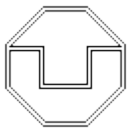
Dresden
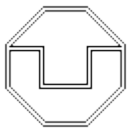
Example of an interpretation

Person ⊓ ∃gives.(Talk ⊓ ∀topic.DL)

Person ⊓ ∀gives.(Talk ⊓ ∃topic.DL)

Dresden

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.
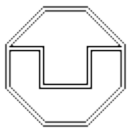
Dresden

Relationship with First-Order Logic

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

- $\pi_x(A) := A(x)$ for $A \in \mathbf{C}$

- $\pi_x(C \sqcap D) := \pi_x(C) \wedge \pi_x(D)$

- $\pi_x(C \sqcup D) := \pi_x(C) \vee \pi_x(D)$

- $\pi_x(\neg C) := \neg \pi_x(C)$

- $\pi_x(\forall r.C) := \forall y.(r(x,y) \rightarrow \pi_y(C))$

  y variable different from x

- $\pi_x(\exists r.C) := \exists y.(r(x,y) \wedge \pi_y(C))$

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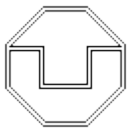\end{aligned}
$$

Dresden

Relationship with First-Order Logic

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

- $\pi_x(A) := A(x)$ for $A \in \mathbf{C}$

- $\pi_x(C \sqcap D) := \pi_x(C) \wedge \pi_x(D)$

- $\pi_x(C \sqcup D) := \pi_x(C) \vee \pi_x(D)$

- $\pi_x(\neg C) := \neg\pi_x(C)$

- $\pi_x(\forall r.C) := \forall y.(r(x,y) \rightarrow \pi_y(C))$

  y variable different from x

- $\pi_x(\exists r.C) := \exists y.(r(x,y) \wedge \pi_y(C))$

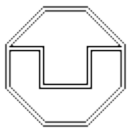**Lemma 2.3**

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

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

Proof: induction on the structure of C

Dresden

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

$$\pi_x(\forall r.(A \sqcap \exists r.B)) = \forall y.(r(x,y) \rightarrow \pi_y(A \sqcap \exists r.B))$$
$$= \forall y.(r(x,y) \rightarrow (A(y) \wedge \exists z.(r(y,z) \wedge B(z))))$$

$$\pi_x(\forall r.(A \sqcap \exists r.B)) = \forall y.(r(x,y) \rightarrow \pi_y(A \sqcap \exists r.B))$$
$$= \forall y.(r(x,y) \rightarrow (A(y) \wedge \exists x.(r(y,x) \wedge B(x))))$$

Dresden

Relationship with Modal Logic

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

multimodal K:

several pairs of
boxes and diamonds

- Concept names are propositional variables,

  and role names are names for transition relations.

- Concept descriptions C yield modal formulae $\pi(C)$:

  - $\pi(A) := a$ for $A \in \mathbf{C}$

  - $\pi(C \sqcap D) := \pi(C) \wedge \pi(D)$

  - $\pi(C \sqcup D) := \pi(C) \vee \pi(D)$

  - $\pi(\neg C) := \neg \pi(C)$

  - $\pi(\forall r.C) := [r]\pi(C)$

  - $\pi(\exists r.C) := \langle r \rangle \pi(C)$

C and $\pi(C)$ have the same semantics: $C^{\mathcal{I}}$ is the set of worlds that make $\pi(C)$

true in the Kripke structure described by $\mathcal{I}$.

Dresden

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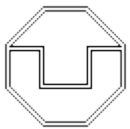
Qualified number restrictions: $(\geq n\,r.C), (\leq n\,r.C)$ with semantics

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

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

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

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

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

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

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

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

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

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

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

Dresden

Additional constructors

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

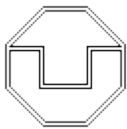### Example
letter $\mathcal{I}$ in the naming scheme

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

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

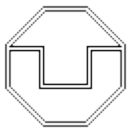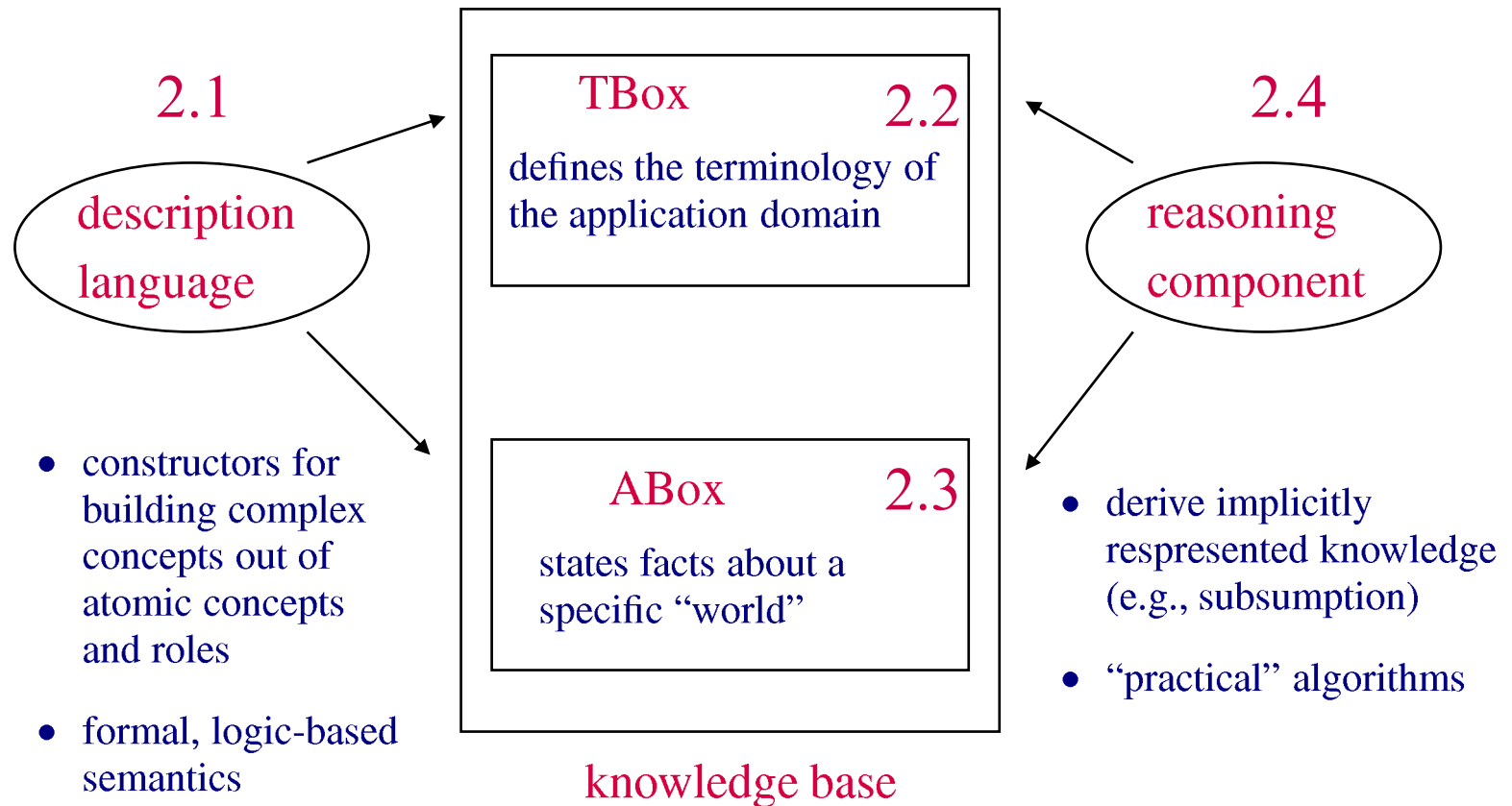Inverse roles can be used like role names in value and existential restrictions.

Presenter of a boring talk:

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

Dresden

Description logic system   structure

2.1

description language

- constructors for building complex concepts out of atomic concepts and roles

- formal, logic-based semantics

TBox   2.2

defines the terminology of the application domain

ABox   2.3

states facts about a specific "world"

knowledge base

2.4

reasoning component

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

- "practical" algorithms

Dresden

2.2. Terminological knowledge

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.
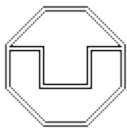
Dresden

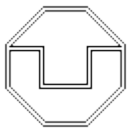Definition 2.4 (GCIs and TBoxes)

- 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 $\mathcal{T}$.

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

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

Dresden

2.2. Terminological knowledge

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
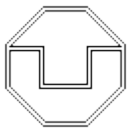 it satisfies all the GCIs in $\mathcal{T}$.

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

Dresden
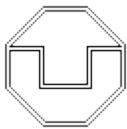
More GCIs $\implies$ less models

Lemma 2.5

If $\mathcal{T} \subseteq \mathcal{T}'$ for two TBoxes $\mathcal{T}$, $\mathcal{T}'$, then each model of $\mathcal{T}'$ is also a model of $\mathcal{T}$.
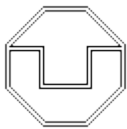
Dresden

Definition 2.6

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

Definition 2.6 (continued)

An acyclic TBox is a finite set of concept definitions that

- does not contain multiple definitions;

- does not contain cyclic definitions.

multiple definition

$$A \equiv C$$
$$A \equiv D$$
$$\text{for } C \neq D$$

$$A \equiv B \sqcap \forall r.P$$
$$B \equiv P \sqcap \forall r.C$$
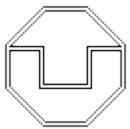$$C \equiv \exists r.A$$

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

Case $n = 1$?

Dresden

concept definitions and acyclic TBoxes

Definition 2.6 (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.

Dresden
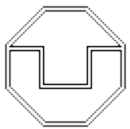
Example   of an acyclic TBox

Woman	≡	Person ⊓ Female
Man	≡	Person ⊓ ¬Female
Talk	≡	$\exists$topic.$\top$
Speaker	≡	Person ⊓ $\exists$gives.Talk
Participant	≡	Person ⊓ $\exists$attends.Talk
BusySpeaker	≡	Speaker ⊓ (≥ 3 gives.Talk)
BadSpeaker	≡	Speaker ⊓ $\forall$gives.($\forall$attends^{-1}.(Bored ⊔ Sleeping))
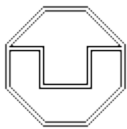
Dresden

© Franz Baader

Proposition 2.7

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
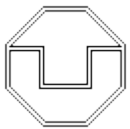
Proposition 2.7

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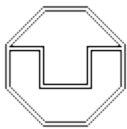
We call $\widehat{\mathcal{T}}$ the *expanded version* of $\mathcal{T}$.

Acyclic TBoxes

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 2.8

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

Proof: blackboard

Dresden
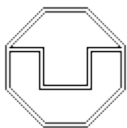
Relationship with First-Order Logic

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

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

Lemma 2.9

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

Proof: blackboard
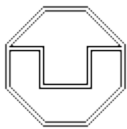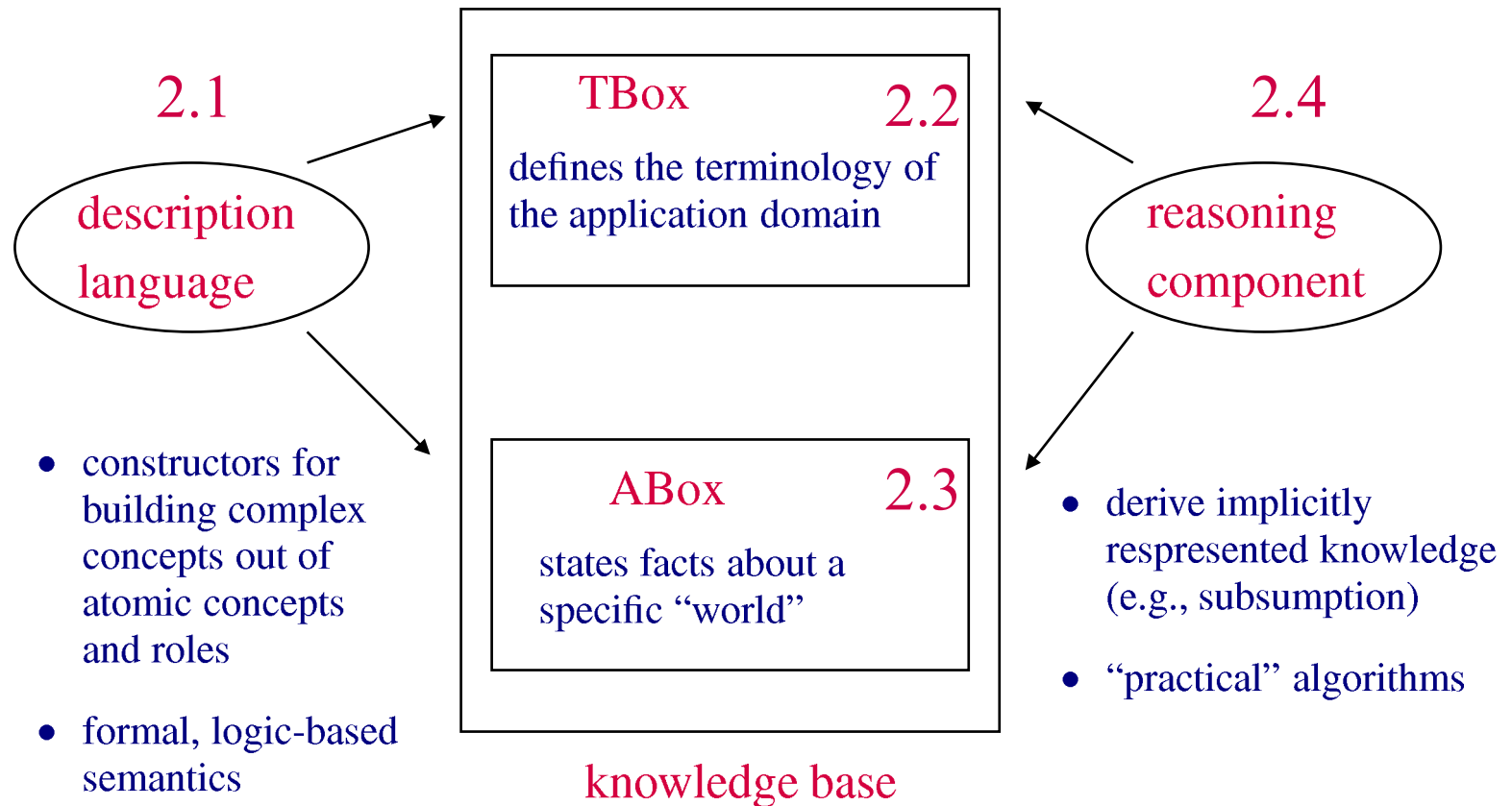
Dresden

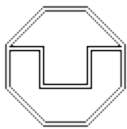# Description logic system

structure

2.1

description language

- constructors for building complex concepts out of atomic concepts and roles

- formal, logic-based semantics

TBox 2.2

defines the terminology of the application domain

ABox 2.3

states facts about a specific "world"

knowledge base

2.4

reasoning component

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

- "practical" algorithms

Dresden

2.3. Assertional knowledge

Definition 2.10 (Assertions and ABoxes)

An assertion is of the form

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

where C is a concept description, r is a role, and a, b are individual names from a set **I** of such names (disjoint with **C** and **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:

$$a^{\mathcal{I}} \in C^{\mathcal{I}} \quad \text{for all } a : C \in \mathcal{A}$$
$$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}} \quad \text{for all } (a, b) : r \in \mathcal{A}$$

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

Dresden

2.3. Assertional knowledge

Definition 2.10 (Assertions and ABoxes)

An assertion is of the form

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

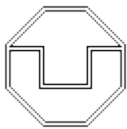where C is a concept description, r is a role, and a, b are individual names from a set **I** of such names (disjoint with **C** and **R**).

An ABox is a finite set of assertions.

$\text{FRANZ} : \text{Lecturer},\quad (\text{FRANZ}, \text{TU03}) : \text{teaches},$
$\text{TU03} : \text{Tutorial},\quad (\text{TU03}, \text{RinDL}) : \text{topic},$
$\text{RinDL} : \text{DL}$
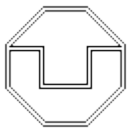
Relationship with First-Order Logic

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

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

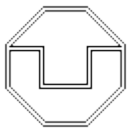individual names are
viewed as constants

Lemma 2.11

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

Proof: easy

Knowledge Bases

Definition 2.12

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: $\pi(\mathcal{K}) := \pi(\mathcal{T}) \wedge \pi(\mathcal{A})$

Lemma 2.13

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

Proof: immediate consequence of Lemma 2.9 and Lemma 2.11
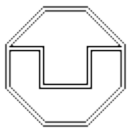
Additional constructors

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 $a \in \mathbf{I}$ with semantics

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

Nominals can be used to express ABox assertions using GCIs:

$a : C$ is expressed by $\{a\} \sqsubseteq C$

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

Description logic system

2.1

description language

- constructors for building complex concepts out of atomic concepts and roles

- formal, logic-based semantics

TBox   2.2

defines the terminology of the application domain

ABox   2.3

states facts about a specific "world"

knowledge base

2.4

reasoning component

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

- "practical" algorithms
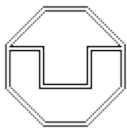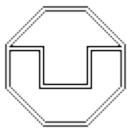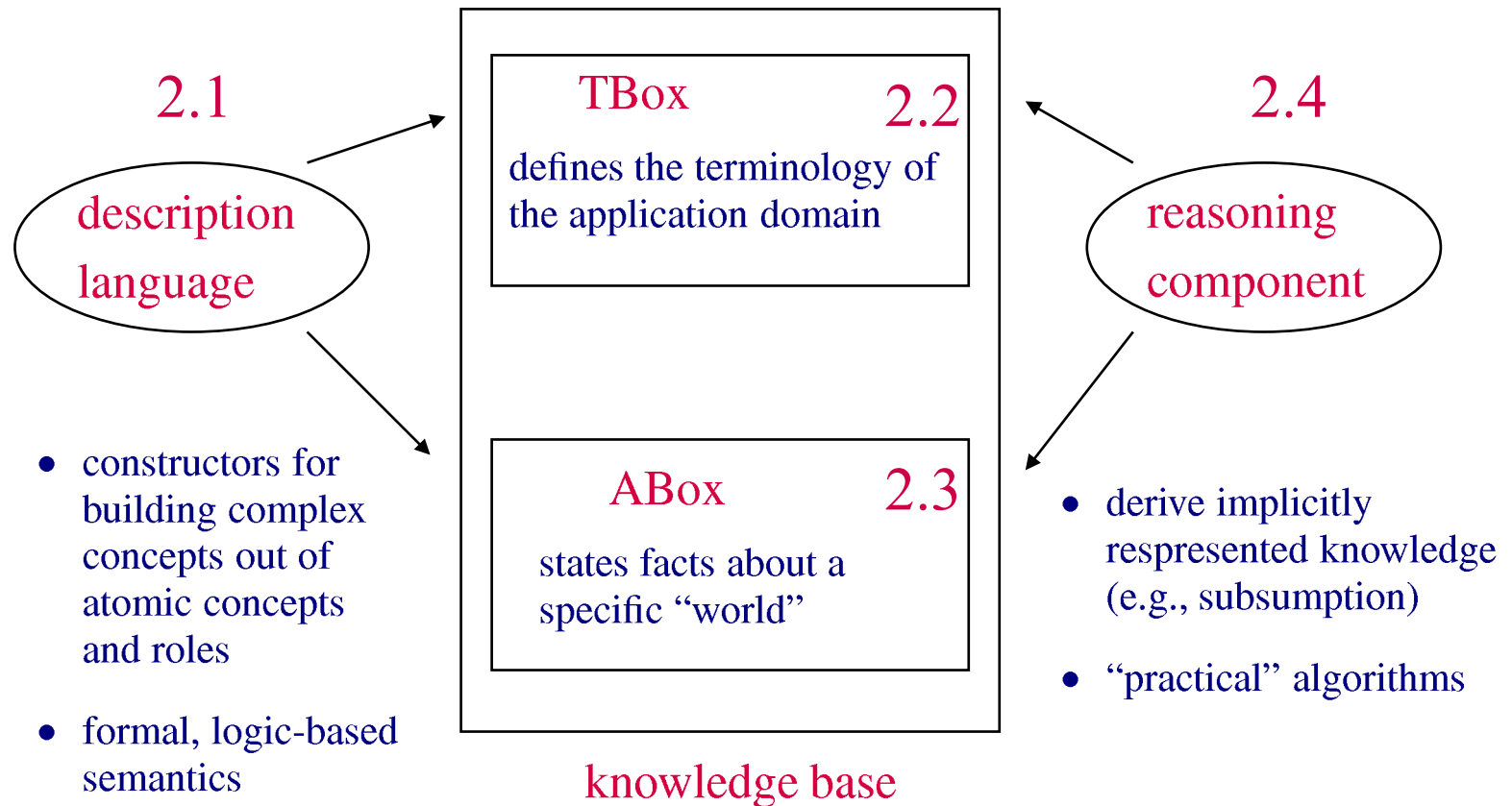
Dresden

2.4. Reasoning Problems and Services

Definition 2.14 (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 $\mathcal{T}$.
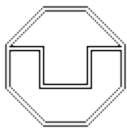
Equivalence:

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 $\mathcal{T}$.

Dresden

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 $\exists r.(A \sqcap B)$

Dresden

Properties of Subsumption

Lemma 2.15

- 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.,

  - $C \sqsubseteq_{\mathcal{T}} D \rightarrow \exists r.C \sqsubseteq_{\mathcal{T}} \exists r.D \wedge \forall r.C \sqsubseteq_{\mathcal{T}} \forall r.D$

- Subsumption reasoning is monotonic, i.e., if $\mathcal{T} \subseteq \mathcal{T}'$, then

  - $C \sqsubseteq_{\mathcal{T}} D \rightarrow C \sqsubseteq_{\mathcal{T}'} D$

Proof: blackboard

Dresden
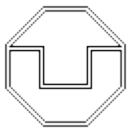
Basic equivalences

Lemma 2.16

- $\neg\top \equiv \bot$

- $C \sqcup D \equiv \neg(\neg C \sqcap D)$

- $\forall r.C \equiv \neg\exists r.\neg C$

- $\exists r.\neg\bot \equiv \bot$

- $\neg(\geq n\, r.C) \equiv (\leq n-1\, r.C)$ if $n \geq 1$

- $(\geq 0\, r.C) \equiv \top$

- $(\leq 0\, r.C) \equiv \forall r.\neg C$

Proof: blackboard

Dresden

Assertional Reasoning

Definition 2.17 (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 2.18

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

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

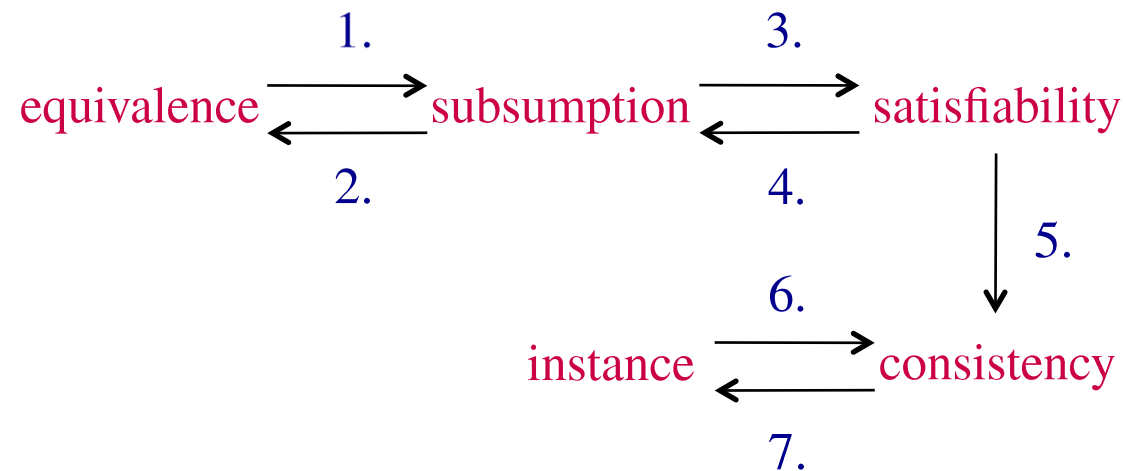then a is an instance of D w.r.t. $\mathcal{K}$. *Proof: exercise*

Reductions between reasoning problems

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

$$\text{equivalence} \underset{2.}{\overset{1.}{\rightleftharpoons}} \text{subsumption} \underset{4.}{\overset{3.}{\rightleftharpoons}} \text{satisfiability}$$

$$\Big\downarrow 5.$$

$$\text{instance} \underset{7.}{\overset{6.}{\rightleftharpoons}} \text{consistency}$$

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

Dresden
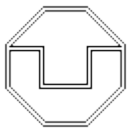
Theorem 2.19

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

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 $(\mathcal{T}, \{a : C\})$ is consistent

6. a is an instance of C w.r.t. $\mathcal{K}$ iff $(\mathcal{T}, \mathcal{A} \cup \{a : \neg C\})$ is inconsistent

7. $\mathcal{K}$ is consistent iff a is not an instance of $\bot$ w.r.t. $\mathcal{K}$

Proof: blackboard

Dresden

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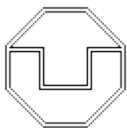
- replace all defined concepts occurring in C and $\mathcal{A}$ by their definitions in the expanded version $\widehat{\mathcal{T}}$ of $\mathcal{T}$.

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

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

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

Dresden

Reduction
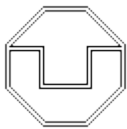getting rid of acyclic TBoxes

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 2.20

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

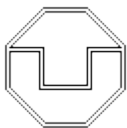Expansion of concepts and ABoxes:

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 2.20

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.*

Dresden

Reduction

getting rid of the TBox

This reduction is in general not polynomial,

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

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

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.
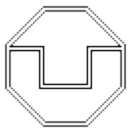
Proof: induction on n

Dresden
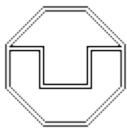
Relationship with First-Order Logic

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

Lemma 2.21

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 $\pi(\mathcal{T}) \models \forall x.(\pi_x(C)(x) \to \pi_x(D)(x))$

2. $\mathcal{K}$ is consistent iff $\pi(\mathcal{K})$ is consistent

3. a is an instance of C w.r.t. $\mathcal{K}$ iff $\pi(\mathcal{K}) \models \pi_x(C)(a)$

Proof: blackboard

Dresden

Classification

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

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

Talk   Person   Female

Man   Speaker   Woman

MaleSpeaker   BusySpeaker   FemaleSpeaker
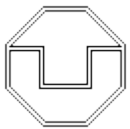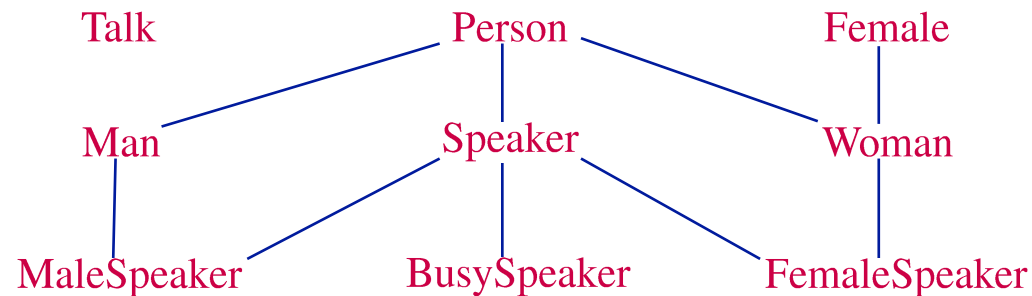
Realization

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

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

$$
\begin{aligned}
&\text{FRANZ} : \text{Man}, \quad (\text{FRANZ, T1}) : \text{gives}, \\
&\text{T1} : \text{Talk}
\end{aligned}
$$

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

most specific

Dresden