JÖNKÖPING UNIVERSITY
School of Engineering

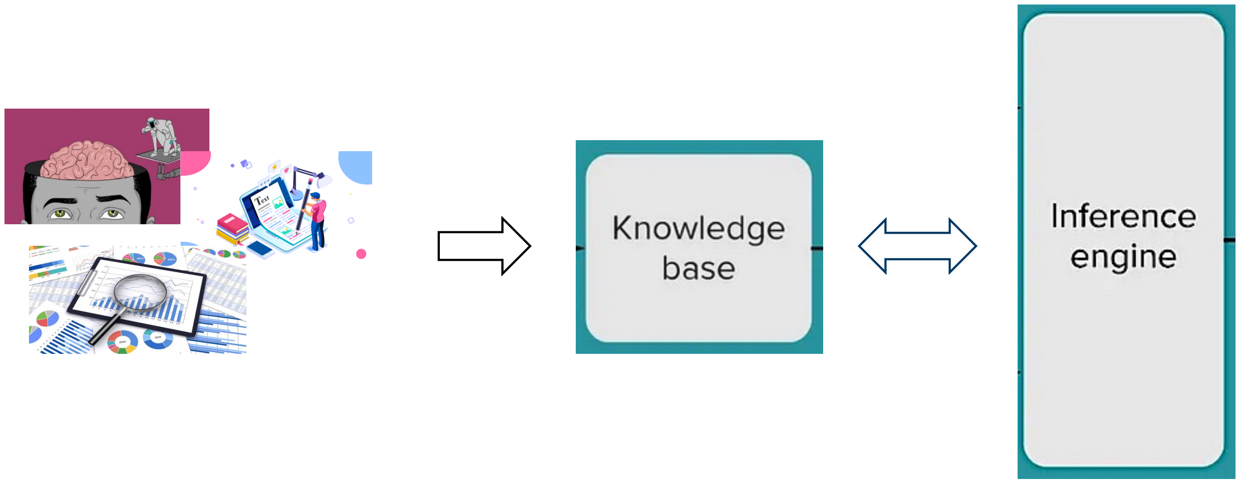
Description Logics

TKRR25 Knowledge Representation and Reasoning

He Tan, Autumn 2025

Knowledge Representation (KR)

- KR are subareas of AI concerned with understanding, designing, and implementing ways of **representing knowledge** so that programs (agents) can use this knowledge, to derive information (reasoning) that is implied by the information.



Logics and KR

- Formal (mathematical) logic provides the foundation of knowledge representation and reasoning, including
  - A well-defined **syntax**
  - A well-defined **semantics**
  - A well-defined proof theory for **reasoning**

- Fundamental problem of designing a logic-based knowledge representation language is find the trade-off between (1) **expressiveness**, i.e., enough expressive power to represent the important knowledge in a problem domain, and (2) **reasoning capability**, i.e., reasoning in a reasonable amount of time.

Logics introduced in the AI course

Propositional logic (PL)

- decidable, but not very expressive

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
& \mid & \neg\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \land \mathit{Sentence} \\
& \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Leftrightarrow \mathit{Sentence} \\
\\
\text{OPERATOR PRECEDENCE} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

- First order logic (FOL)
  - powerful expressiveness, but undecidable

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{Predicate} \mid \mathit{Predicate}(\mathit{Term}, \ldots) \mid \mathit{Term} = \mathit{Term} \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
& \mid & \neg\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \land \mathit{Sentence} \\
& \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Leftrightarrow \mathit{Sentence} \\
& \mid & \mathit{Quantifier}\ \mathit{Variable}, \ldots\ \mathit{Sentence} \\
\\
\mathit{Term} & \rightarrow & \mathit{Function}(\mathit{Term}, \ldots) \\
& \mid & \mathit{Constant} \\
& \mid & \mathit{Variable} \\
\\
\mathit{Quantifier} & \rightarrow & \forall \mid \exists \\
\mathit{Constant} & \rightarrow & A \mid X_1 \mid \mathit{John} \mid \cdots \\
\mathit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\mathit{Predicate} & \rightarrow & \mathit{True} \mid \mathit{False} \mid \mathit{After} \mid \mathit{Loves} \mid \mathit{Raining} \mid \cdots \\
\mathit{Function} & \rightarrow & \mathit{Mother} \mid \mathit{LeftLeg} \mid \cdots \\
\text{OPERATOR PRECEDENCE} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Description Logics (DL)

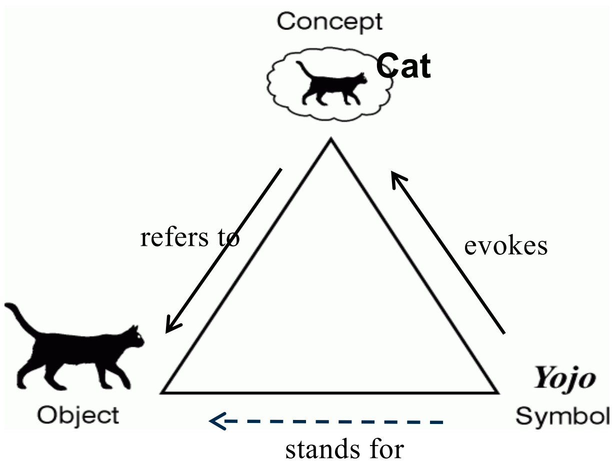
- In 1980's, description logics (DL) was introduced in KR.
- DL is **a family of logics**, each DL with different expressiveness and reasoning capabilities.
  - Each DL is a **fragment of FOL**.
  - Depending on the expressiveness of the DL, reasoning tasks can be **decidable** or **undecidable**.
  - Many DLs are designed to allow for sound and complete reasoning algorithms.

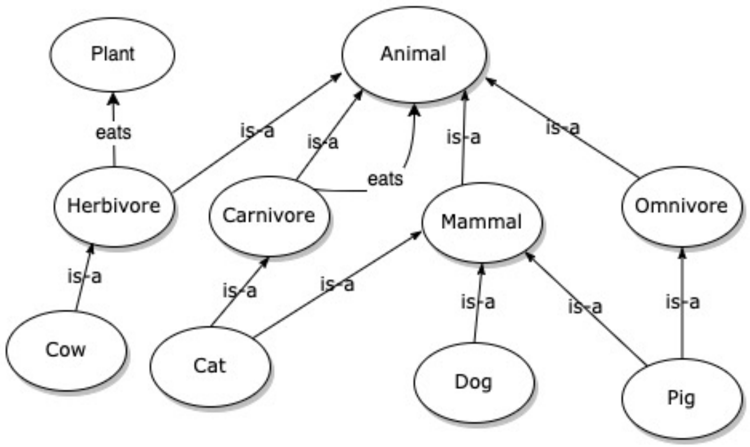
Description

- In DLs, the basic unit of semantic significance is the description, i.e., "*we are describing sets of individuals*".
- DL is **a concept language**, where concepts denote *sets of individuals*.
- **Ontology** is the theory encoded in DLs.

Human Intelligence

- Conceptualization and Categorization
  - We group things into concepts and categories to make sense of the world.





Ontology

- It was called "*first philosophy*" by Aristotle in Book IV of his Metaphysics.
- Questions are addressed by ontology:
  - How to categorize things?
  - What the categories should be?
  - What should belong to the categories?
- Ontologies, when communicated to others, foster a shared understanding of the nature of things.

Family Example

- John is a father, John is married to Mary, Alice is John's child
- ...

individuals

- A man is a male person, a woman is a female person
- A husband is married to a woman, a wife is married to a man
- A parent is a person who has a child who is also a person
- A father is a man and a parent, a mother is a woman and a parent
- A grandparent is a person who has a child who is also a parent
- Every person is a grandchild
- A happy father has at most two children
- A happy mother has at least one daughter

ontology

DL Syntax

- The basic building blocks:
  - *Concepts* (aka. unary predicates in FOL)
    - e.g., Person, Father, Child, Student, Course
  - *Roles* (aka. binary predicates in FOL)
    - e.g., hasChild, teachesCourse
  - *Individuals* (aka. constants in FOL)
    - Mary, Alice, John
- *Constructors*
  - More complex concepts and relations are built from atomic ones using constructors, like $\sqcap$, $\sqcup$, $\exists$, $\forall$, etc.

DL Semantics

- What these symbols and expressions mean in mathematical structure?

  `e.g., Person, hasParent, Person(John)`

- The semantics of mathematical logic is defined in a **model-theoretic** way.

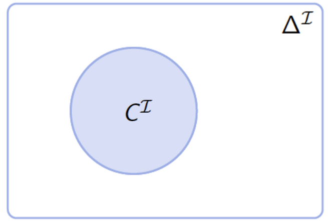
DL Semantics (Cont.')

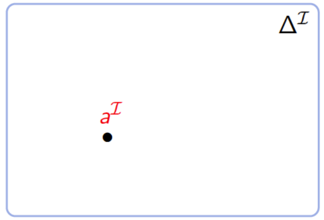
- An *interpretation I* is a pair $(\Delta^I, .^I)$ *(pronounced "delta-i and i"),* where
  - a non-empty set Δ^I , called the domain (also universe of discourse)
  - $.^I$ is an interpretation function mapping the vocabulary elements (i.e., concepts, roles and individuals) to the domain Δ^I :

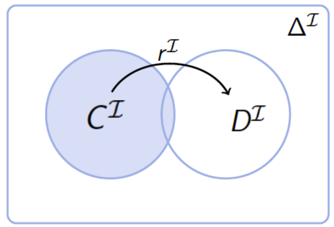
Interpretation – Basic Building Blocks

- An atomic **concept** $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
- An **Individual** $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- An atomic **role** $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

(cartesian product*)







Example

Expression: $Parent \sqsubseteq Person$

Domain $\Delta^I = \{mary, john, alice, yoyo\}$

$Person^I = \{mary, john, alice\}$
$Parent^I = \{mary, john\}$

Then:

$\mathcal{I} \models Parent \sqsubseteq Person$, because $Parent^I \subseteq Person^I$.

A Family of Languages

- The expressiveness of a DL is determined by the contructors that it has.
- Add or remove certain constructors means the expressiveness increases or reduces.
- Higher expressiveness implies higher complexity.

- $\mathcal{AL}$ (*Attributive Language with Complements*) is the base DL and other DLs have additional constructors.

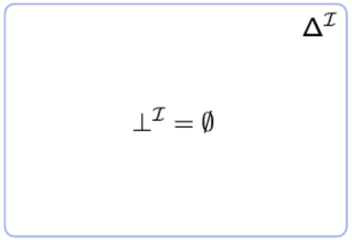
$\mathcal{AL}$: Attributive Language with Complements

Syntax **Semantics**

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg A)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R.\top)^{\mathcal{I}} & = & \{a \in \Delta^{\mathcal{I}} \mid \exists b.\ (a,b) \in R^{\mathcal{I}}\}.
\end{array}
$$

Top and Bottom Concept

- The "top" concept is everything is in Δ^I , also called Thing.
- The bottom concept is also called Nothing.



Example

- We assume that `Person` and `Female` are atomic concepts, `hasChild` is an atomic role.

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

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

$$\text{Person} \sqcap \exists\text{hasChild}.\top$$

$$\text{Person} \sqcap \forall\text{hasChild}.\text{Female}$$

More Constructors

- The letter $\mathcal{U}$: *union* of concpets: $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- The letter $\mathcal{E}$: *full existential quantification*:

$$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b.\, (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$$

- The letter $\mathcal{N}$: for *number restrictions*:

$$(\geqslant n\, R)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \;\middle|\; |\{b \mid (a, b) \in R^{\mathcal{I}}\}| \geq n\right\}$$

$$(\leqslant n\, R)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \;\middle|\; |\{b \mid (a, b) \in R^{\mathcal{I}}\}| \leq n\right\}$$

Example

- We assume that Person, Cat , Dog, Old are atomic concepts, owns , hasPet are atomic roles.

$$\text{Person} \sqcap \exists \text{owns}.(\text{Cat} \sqcap \text{Old})$$

$$\text{Person} \sqcap \forall \text{owns}.(\text{Cat} \sqcup \text{Dog})$$

$$\text{Person} \sqcap \leq 2\ \text{hasPet}.\text{Cat}$$

Exercise

- Formalize these statements, given atomic concepts `Person, Happy, Pet, Cat, Old, Fish` and atomic roles `owns`

- happy person
- happy pet owner
- person who owns at least two cats
- unhappy pet owners who own an old cat
- pet owners who only own cats or fish

Person ⊓ Happy

Person ⊓ Happy ⊓ ∃owns.Pet

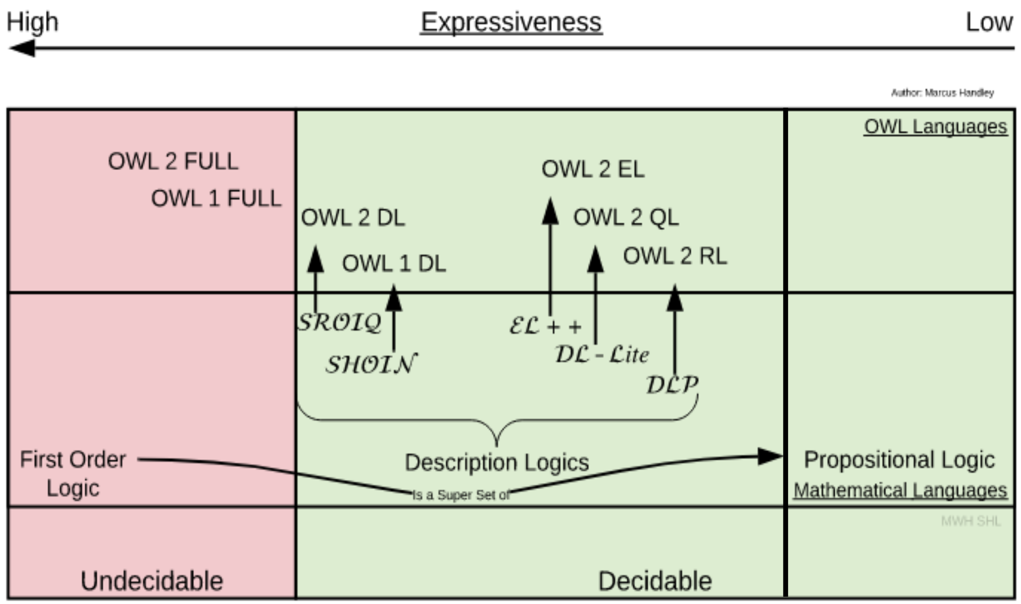
Person ⊓ >= 2 ∃owns.Cat

Person ⊓ (¬Happy) ⊓ ∃owns.(Cat ⊓ Old)

Person ⊓ ∃owns.Pet ⊓ ∀owns.(Cat ⊔ Fish)

$\mathcal{SROIQ}$

- It is a powerful and expressive DL.
- It serves as the basis for ontology language today.



	Syntex	Semantics
$\mathcal{SROIQ}$ constructors		
universal concept	$\top$	Δ^I
bottom concept	$\bot$	$\bot$
atomic concept	A	A^I
intersection	$C \sqcap D$	$C^I \cap D^I$
union	$C \sqcup D$	$C^I \cup D^I$
complement	$\neg C$	$\Delta^I \backslash C^I$
universal restriction	$\forall R.C$	$\{a \in \Delta^I \mid \forall b.(a,b) \in R^I \to b \in C^I\}$
existential restriction	$\exists R.C$	$\{a \in \Delta^I \mid \exists b.(a,b) \in R^I \to b \in C^I\}$
at-least restriction	$\geq n\ \exists R.C$	$\{a \in \Delta^I \mid \#\{b \mid (a,b) \in R^I \to b \in C^I\} \geq n\}$
at-most restriction	$\leq n\ \exists R.C$	$\{a \in \Delta^I \mid \#\{b \mid (a,b) \in R^I \to b \in C^I\} \leq n\}$
local reflexivity	$\exists R.Self$	$\{a \mid (a,a) \in R^I\}$
nominal	$\{a\}$	$\{a^I\}$
atomic role	R	R^I
inverse role	R	R^-
universal role	U	$\Delta^I \times \Delta^I$
individual name	a	a^I

Person ⊑ ∃ loves.Self

DaysOfWeek ≡ { Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday }

hasTeacher ≡ teaches⁻

$\mathcal{SROIQ}$

- $\mathcal{S}$ stands for $\mathcal{ALC}$ + 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

- $\mathcal{N}$ ($\mathcal{Q}$) stands for arbitray (qualified) cardinality restrictions
- $\mathcal{R}$ stands for role box with all kinds of role axioms plus self concepts

Note:

- $\mathcal{S}$ subsumes $\mathcal{ALC}$, $\mathcal{SR}$ subsumes $(\mathcal{ALC} \mid \mathcal{S})[\mathcal{H}]$ $\mathcal{ALCH}$
- $\mathcal{SROIQ}$ subsumes all the other description logics in this scheme.

Tbox Axioms

- Two forms of terminological axioms: ***inclusions*** and ***equivalences***.

$$C \sqsubseteq D \quad (R \sqsubseteq S) \qquad \text{or} \qquad C \equiv D \quad (R \equiv S),$$

- Semantics:

$$C \sqsubseteq D \text{ if } C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$$

$$C \equiv D \text{ if } C^{\mathcal{I}} = D^{\mathcal{I}}$$

Definition and Description

- Definitions are used to introduce ***symbolic names*** for complex descriptions (complex concepts/roles, also called defined concepts/roles).

definition (symbolic name) description

HappyPerson ≡ Person ⊓ Happy

definition (symbolic name) description

Cat ⊑ Animial ⊓ ∃eats.Animal

- No symbolic name is defined more than once.
- Defined concepts/roles (name symbols, N_T) occur on the left-hand side of axioms.
- Primitive concepts/roles (base symbols, B_T) **only** occur on the right-hand sides.

Example Definitions and Descriptions

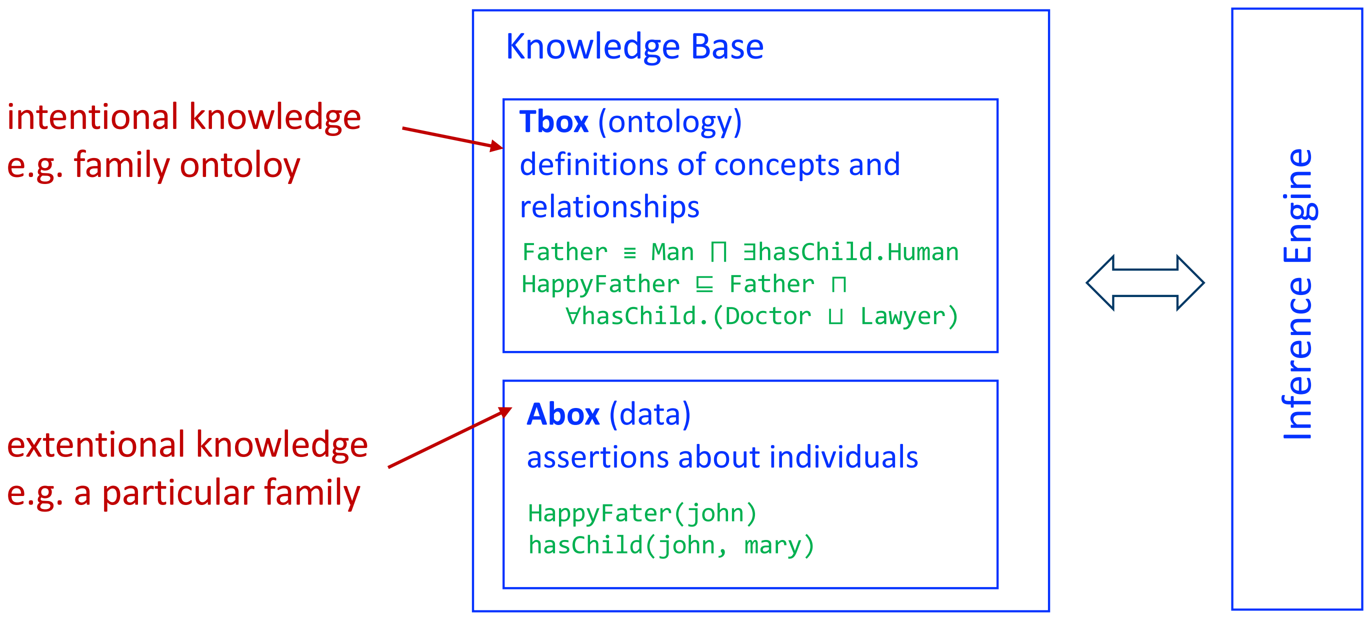
- A vocabulary about family.

$$B_C = \{\text{Person}, \text{Female}\}$$
$$B_R = \{\text{hasChild}, \text{hasHusband}\}$$

Tbox

$$\begin{aligned}
\text{Woman} &\equiv \text{Person} \sqcap \text{Female}\\
\text{Man} &\equiv \text{Person} \sqcap \neg\text{Woman}\\
\text{Mother} &\equiv \text{Woman} \sqcap \exists\text{hasChild}.\text{Person}\\
\text{Father} &\equiv \text{Man} \sqcap \exists\text{hasChild}.\text{Person}\\
\text{Parent} &\equiv \text{Father} \sqcup \text{Mother}\\
\text{Grandmother} &\equiv \text{Mother} \sqcap \exists\text{hasChild}.\text{Parent}\\
\text{MotherWithManyChildren} &\equiv \text{Mother} \sqcap \geqslant 3\ \text{hasChild}\\
\text{MotherWithoutDaughter} &\equiv \text{Mother} \sqcap \forall\text{hasChild}.\neg\text{Woman}\\
\text{Wife} &\equiv \text{Woman} \sqcap \exists\text{hasHusband}.\text{Man}
\end{aligned}$$

DL Architecture



World Descriptions (Abox)

- Two basic forms of individual axioms
  - ***C(a):*** concept assertions, state that ***a*** belongs to (the interpretation of) ***C***
  - ***R(b,c):*** role assertions, state that ***c*** is the filler of the role of ***R*** for ***b***.

MotherWithoutDaughter(MARY) Father(PETER)
hasChild(MARY, PETER) hasChild(PETER, HARRY)
hasChild(MARY, PAUL)

- **Open World Assumptions**: we don't assume that the knowledge in KB is complete.

$\mathcal{SROIQ}$ Axioms

Marry ≉ Peter

hasDaughter ⊑ hasChild

hasSibling ≡ hasBrother ⊔ hasSister

hasFather ∘ hasBrother ⊑ hasUncle

Disjoint (:hasParent :hasSpouse)

$\mathcal{SROIQ}$ **axioms**		
concept assertion	$C(a)$	$a^I \in C^I$
role assertion	$R(a,b)$	$(a^I, b^I) \in R^I$
individual equality	$a \approx b$	$a^I = b^I$
individual inequality	$a \neq b$	$a^I \neq b^I$
concept inclusion	$C \sqsubseteq D$	$C^I \subseteq D^I$
concept eqivalence	$C \equiv D$	$C^I = D^I$
role inclusion	$R \sqsubseteq S$	$R^I \subseteq S^I$
role equivalence	$R \equiv S$	$R^I = S^I$
complex role inclusion	$R_1 \circ R_2 \sqsubseteq S$	$R_1 \circ R_2 \subseteq S$
role disjointness	$Disjoint(R,S)$	$R_1 \cap R_2 = \emptyset$

Exercise

- Given $B_C = \{Woman, Man, Pet, Cat\}$ and $B_R = \{hasChild, hasPet\}$
  Build a knowledge base to capture the following statements. Please introduce necessary definitions in the terminology and choose the right terminological axioms to form the complex descriptions.

```
1. Parent with only (a) male child
2. GrandParent
3. GrandParent with only male grandchildren
4. A woman that has only (a) daughter who has at least one son
5. CatLady (a woman with many cats but no children)
6. teresa is a CatLady
7. andy is teresa's cat.
```

```
Person ≡ Woman ⊔ Man
Parent ≡ Person ⊓ ∃hasChild.Person
ParentWithOneChild ≡ Parent ⊓
        >=1.hasChild.Man ⊓
            <=1.hasChild.Man
```

```
GrandParent ≡ Person ⊓ ∃hasChild.Parent
GPWithOnlyMaleGrandChildren ≡ GrandParent ⊓ ∀hasChild.Man

WDS ≡ Woman ⊓ ∀hasChild.(Woman ⊓ ≥1.hasChild.Man)

CatLady ≡ Woman ⊓ ¬ (∃hasChild.Person) ⊓ ≥2 hasPet.Cat
```

```
CatLady(teresa).
Cat(andy).
hasPet(teresa, andy).
```

Reasoning in Tbox

- Concept Satisfiability: A concept $\mathcal{C}$ is satisfiable with respect to $\mathcal{T}$ if there exists an interpretation $\mathcal{I}$ of $\mathcal{T}$ such that $\mathcal{C}^{\mathcal{I}}$ is nonempty, otherwise it is called unsatisfiable, inconsistent, or contradictory.

- It is supported in $\mathcal{SROIQ}$

Reduction to Unsatisfiability

Proposition 2.13 (Reduction to Unsatisfiability) *For concepts C, D we have*

(i) C *is subsumed by* $D \Leftrightarrow C \sqcap \neg D$ *is unsatisfiable;*
(ii) C *and* D *are equivalent* $\Leftrightarrow$ *both* $(C \sqcap \neg D)$ *and* $(\neg C \sqcap D)$ *are unsatisfiable;*
(iii) C *and* D *are disjoint* $\Leftrightarrow C \sqcap D$ *is unsatisfiable.*

The statements also hold with respect to a TBox.

- The most recent generation of DL systems are based on **satisfiability checking**, and a considerable amount of research work is spent on the development of efficient implementation techniques for this approach.

Reasoning

- Reasoning over a knowledge base to derive information that is implied by it.

Tbox

$$
\begin{aligned}
\mathsf{Woman} &\equiv \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} &\equiv \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasChild.Person} \\
\mathsf{Father} &\equiv \mathsf{Man} \sqcap \exists\mathsf{hasChild.Person} \\
\mathsf{Parent} &\equiv \mathsf{Father} \sqcup \mathsf{Mother} \\
\mathsf{Grandmother} &\equiv \mathsf{Mother} \sqcap \exists\mathsf{hasChild.Parent} \\
\mathsf{MotherWithManyChildren} &\equiv \mathsf{Mother} \sqcap \geqslant 3\ \mathsf{hasChild} \\
\mathsf{MotherWithoutDaughter} &\equiv \mathsf{Mother} \sqcap \forall\mathsf{hasChild.}\neg\mathsf{Woman} \\
\mathsf{Wife} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasHusband.Man}
\end{aligned}
$$

$\mathtt{Man} \sqsubseteq \neg\ \mathtt{Woman}$

$\mathtt{Mother} \sqsubseteq \mathtt{Woman}$

$\mathtt{Mother} \sqsubseteq \mathtt{Parent}$

$\mathtt{Man} \sqcap \mathtt{Woman} \equiv \bot$

Reasoning in Abox

- **Consistency checking**: the assertions in the ABox do not contradict each other and are consistent with the ontology or schema they are based on.

$$\mathcal{A} \textit{ is consistent w.r.t. } \mathcal{T}$$

- **Instance checking**: whether an assertion is entailed by an Abox.

$$\mathcal{A} \models C(a)$$

- Instance checking can be reduced to consistency problem, and [Schaerf, 1994b] has shown that ABox consistency can be reduced to concept satisfiability in languages with the “set” and the “fills” constructors.

Reasoning

- Reasoning over a knowledge representation system to derive information that is implied by it.

Tbox

$$
\begin{aligned}
\mathsf{Woman} &\equiv \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} &\equiv \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Father} &\equiv \mathsf{Man} \sqcap \exists\mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Parent} &\equiv \mathsf{Father} \sqcup \mathsf{Mother} \\
\mathsf{Grandmother} &\equiv \mathsf{Mother} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent} \\
\mathsf{MotherWithManyChildren} &\equiv \mathsf{Mother} \sqcap \geqslant 3\,\mathsf{hasChild} \\
\mathsf{MotherWithoutDaughter} &\equiv \mathsf{Mother} \sqcap \forall\mathsf{hasChild}.\neg\mathsf{Woman} \\
\mathsf{Wife} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasHusband}.\mathsf{Man}
\end{aligned}
$$

MotherWithoutDaughter(MARY)	Father(PETER)
hasChild(MARY, PETER)	hasChild(PETER, HARRY)
hasChild(MARY, PAUL)	

Abox

`Grandmother(Mary)`

Literature

- Chapter 1 and 2 in Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., & Nardi, D. (Eds.). (2007). *The description logic handbook: Theory, implementation and applications*. Cambridge university press. The book is available in JU library in electronic format.

JÖNKÖPING UNIVERSITY

School of Engineering