Basic Reasoning Services

(i) Given a TBox $\mathcal{T}$ and a concept C, check whether C is *satisfiable* with respect to $\mathcal{T}$.

(ii) Given a TBox $\mathcal{T}$ and two concepts C and D, check whether C is *subsumed by* D with respect to $\mathcal{T}$.

(iii) Given a TBox $\mathcal{T}$ and two concepts C and D, check whether C and D are *equivalent* with respect to $\mathcal{T}$.

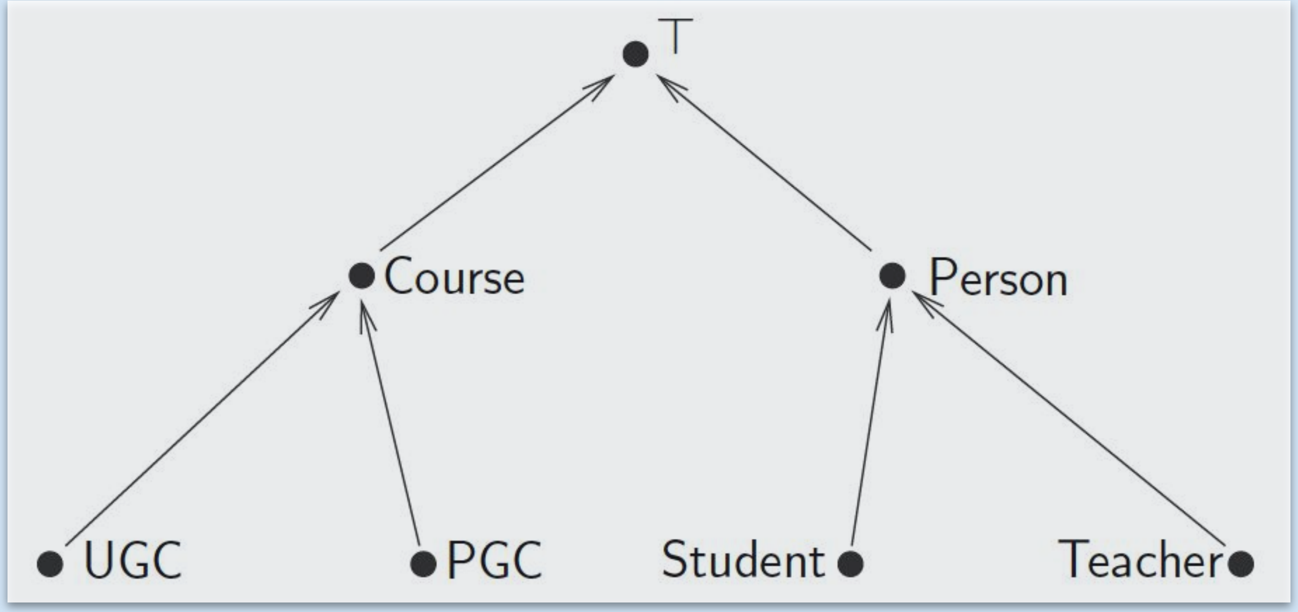
(iv) Given a knowledge base $(\mathcal{T}, \mathcal{A})$, check whether $(\mathcal{T}, \mathcal{A})$ is *consistent*.

(v) Given a knowledge base $(\mathcal{T}, \mathcal{A})$, an individual name a, and a concept C, check whether a is an *instance of* C with respect to $(\mathcal{T}, \mathcal{A})$.

More Sophisticated Reasoning Services

- Classification of a TBox $\mathcal{T}$
    - Compute the ***subsumption hierarchy*** among all the concept names in $\mathcal{T}$ wrt $\mathcal{T}$
    - Concept names - both atomic as well as defined to be considered
- Checking the satisfiability of concepts in $\mathcal{T}$
    - Are all the concept names ***satisfiable*** wrt $\mathcal{T}$?
    - If not - usually some modelling error might be there
- Instance Retrieval
    - Given a KB $\mathcal{K}$ and a concept C (possibly compound)
        - List all individual names in $\mathcal{K}$ that are instances of C wrt $\mathcal{K}$
- Realization of an individual name
    - Given a KB $\mathcal{K}$ and an individual b,
        - List all the concept names A in $\mathcal{K}$ such that b is instance of A

Subsumption Hierarchy of $\mathcal{T}_{ex}$



Notes:
$\sqsubseteq_{\mathcal{T}}$ is a reflexive, transitive relation (a pre-order)

$\sqsubset_{\mathcal{T}}$ is an irreflexive, transitive relation (a strict partial order)

Hasse Diagram of this partial order is shown as subsumption hierarchy.

Domain Ontology Development

- Identify the entity types of interest
  - Introduce these names as primitive / defined concepts
- Identify the required relationships between entities of the domain
  - Introduce roles to capture these
- Identify various constraints that exist in the domain - define scope...
  - Capture these as
    - Inclusion axioms, Subsumptions, Definitions etc
    - What constructs of DL are sufficient for the application at hand? Choice of DL.
- Choose an ontology editor and create an ontology
  - Use built-in reasoners to check:
    - Consistency
    - Are all the concepts satisfiable?
    - Debug the ontology - ensure that only intuitively intended models are allowed.

Lazy student - modeling !

$\mathcal{T}_{ex} = \{\text{Course}$	$\sqsubseteq$	$\neg\text{Person},$	$(\mathcal{T}_{ex}.1)$
UGC	$\sqsubseteq$	$\text{Course},$	$(\mathcal{T}_{ex}.2)$
PGC	$\sqsubseteq$	$\text{Course},$	$(\mathcal{T}_{ex}.3)$
Teacher	$\equiv$	$\text{Person} \sqcap \exists \mathit{teaches}.\text{Course},$	$(\mathcal{T}_{ex}.4)$
$\exists \mathit{teaches}.\top$	$\sqsubseteq$	$\text{Person},$	$(\mathcal{T}_{ex}.5)$
Student	$\equiv$	$\text{Person} \sqcap \exists \mathit{attends}.\text{Course},$	$(\mathcal{T}_{ex}.6)$
$\exists \mathit{attends}.\top$	$\sqsubseteq$	$\text{Person}\ \}$	$(\mathcal{T}_{ex}.7)$

Can we add ***attendsActively*** role?

Current DL - $\mathcal{ALC}$ - not enough !

$\mathcal{T}_1 = \mathcal{T}_{ex} \cup \{$ LazyStudent $\sqsubseteq \forall$attends.$\neg$Course $\}$

(lazy students do not attend any courses at all)

Is $\mathcal{T}_1$ consistent with our intuition?

$\mathcal{T}_1$ does not entail LazyStudent $\sqsubseteq$ Student … We don't expect this…

Suppose we modify it: LazyStudent $\equiv$ Student $\sqcap$ $\forall$attends.$\neg$Course

Now, LazyStudent satisfiable?

No…

Extensions of $\mathcal{ALC}$

- $\mathcal{ALC}$
  - A basic description logic
  - Not enough to capture some constraints in the domain
- Extensions
  - When certain domain knowledge can not be expressed in $\mathcal{ALC}$, we like to extend it
  - Bring in new features
- We now discuss
  - Situations that can't be modeled in $\mathcal{ALC}$ and
  - How to extend it by adding a new construct and
  - What should the interpretation function do regarding the new construct

Extending $\mathcal{ALC}$ - Inverse Roles

$$
\begin{array}{rcll}
\mathcal{T}_{ex} = \{\text{Course} & \sqsubseteq & \neg\text{Person}, & (\mathcal{T}_{ex}.1)\\
\text{UGC} & \sqsubseteq & \text{Course}, & (\mathcal{T}_{ex}.2)\\
\text{PGC} & \sqsubseteq & \text{Course}, & (\mathcal{T}_{ex}.3)\\
\text{Teacher} & \equiv & \text{Person} \sqcap \exists \mathit{teaches}.\text{Course}, & (\mathcal{T}_{ex}.4)\\
\exists \mathit{teaches}.\top & \sqsubseteq & \text{Person}, & (\mathcal{T}_{ex}.5)\\
\text{Student} & \equiv & \text{Person} \sqcap \exists \mathit{attends}.\text{Course}, & (\mathcal{T}_{ex}.6)\\
\exists \mathit{attends}.\top & \sqsubseteq & \text{Person}\ \} & (\mathcal{T}_{ex}.7)
\end{array}
$$

Intuition: Professor

-- unsatisfiable wrt $\mathcal{T}_1$

p $\in$ Professor

$\Rightarrow$ p $\in$ Teacher

$\Rightarrow$ (p,c) $\in$ teaches for some c

$\Rightarrow$ (c,p) $\in$ taughtBy

$\Rightarrow$ p $\in$ $\neg$Professor

A contradiction !!

$\mathcal{T}_1 = \mathcal{T}_{ex} \cup \{$ Professor $\sqsubseteq$ Teacher

Course $\sqsubseteq$ $\forall$taughtBy.$\neg$Professor $\}$

No axiom in TBox states that: (p,c) $\in$ teaches $\Leftrightarrow$ (c,p) $\in$ taughtBy !!

“teaches” and “taughtBy” can be interpreted independently in an $\mathcal{I}$!!

Extending $\mathcal{ALC}$ - Inverse Roles

$$
\begin{aligned}
\mathcal{T}_{ex} = \{\text{Course} &\sqsubseteq \neg\text{Person}, && (\mathcal{T}_{ex}.1)\\
\text{UGC} &\sqsubseteq \text{Course}, && (\mathcal{T}_{ex}.2)\\
\text{PGC} &\sqsubseteq \text{Course}, && (\mathcal{T}_{ex}.3)\\
\text{Teacher} &\equiv \text{Person} \sqcap \exists \mathit{teaches}.\text{Course}, && (\mathcal{T}_{ex}.4)\\
\exists \mathit{teaches}.\top &\sqsubseteq \text{Person}, && (\mathcal{T}_{ex}.5)\\
\text{Student} &\equiv \text{Person} \sqcap \exists \mathit{attends}.\text{Course}, && (\mathcal{T}_{ex}.6)\\
\exists \mathit{attends}.\top &\sqsubseteq \text{Person}\ \} && (\mathcal{T}_{ex}.7)
\end{aligned}
$$

$\mathcal{T}_1 = \mathcal{T}_{ex} \cup \{$ Professor $\sqsubseteq$ Teacher

Course $\sqsubseteq$ $\forall$taughtBy.$\neg$Professor $\}$

Extend a model $\mathcal{I}$ of $\mathcal{T}_{ex}$ by adding taughtBy$^{\mathcal{I}} = \Phi$

It would be a model of $\mathcal{T}_1$

Intuition: Professor

-- unsatisfiable wrt $\mathcal{T}_1$

p $\in$ Professor

$\Rightarrow$ p $\in$ Teacher

$\Rightarrow$ (p,c) $\in$ teaches for some c

$\Rightarrow$ (c,p) $\in$ taughtBy

$\Rightarrow$ p $\in$ $\neg$Professor

A contradiction.

ALC has to be extended to include *inverse* roles to get the desired effect

Inverse Roles

For r a role name, r^{-} is an *inverse* role.
Set of roles: $R \cup \{ r^{-} \mid r \in R \}$

r^{-} : also written as r^{-1}

$\mathcal{ALC}$ extended with inverse roles: $\mathcal{ALCI}$

An interpretation maps inverse roles to binary relations as:

$$(r^{-})^{\mathcal{I}} = \{ (y,x) \mid (x,y) \in r^{\mathcal{I}} \}$$

In $\mathcal{ALCI}$, we can use both r, r^{-} whereeever role names are needed.

New roles names for inverse roles can be introduced for ease of use.

Eg., taughtBy ≡ teaches^{-}

Extending $\mathcal{ALC}$ - Inverse Roles

$$
\begin{array}{rrcll}
\mathcal{T}_{ex} = \{ & \mathsf{Course} & \sqsubseteq & \neg\mathsf{Person}, & (\mathcal{T}_{ex}.1) \\
& \mathsf{UGC} & \sqsubseteq & \mathsf{Course}, & (\mathcal{T}_{ex}.2) \\
& \mathsf{PGC} & \sqsubseteq & \mathsf{Course}, & (\mathcal{T}_{ex}.3) \\
& \mathsf{Teacher} & \equiv & \mathsf{Person} \sqcap \exists \mathit{teaches}.\mathsf{Course}, & (\mathcal{T}_{ex}.4) \\
& \exists \mathit{teaches}.\top & \sqsubseteq & \mathsf{Person}, & (\mathcal{T}_{ex}.5) \\
& \mathsf{Student} & \equiv & \mathsf{Person} \sqcap \exists \mathit{attends}.\mathsf{Course}, & (\mathcal{T}_{ex}.6) \\
& \exists \mathit{attends}.\top & \sqsubseteq & \mathsf{Person}\ \} & (\mathcal{T}_{ex}.7)
\end{array}
$$

Professor

-- unsatisfiable wrt $\mathcal{T}_1$

p $\in$ Professor

$\Rightarrow$ p $\in$ Teacher

$\Rightarrow$ (p,c) $\in$ teaches for some c

$\Rightarrow$ (c,p) $\in$ teaches^{-}

$\Rightarrow$ p $\in$ $\neg$Professor

A contradiction.

$\mathcal{T}_1 = \mathcal{T}_{ex} \cup \{$ Professor $\sqsubseteq$ Teacher

Course $\sqsubseteq$ $\forall$teaches^{-}.$\neg$Professor $\}$

Now, (p,c) $\in$ teaches $\Leftrightarrow$ (c,p) $\in$ teaches^{-}

Professor is indeed unsatisfiable wrt $\mathcal{T}_1$.

Number Restrictions

How to impose:

A student must attend at least 3 courses?

A student can attend at most 6 courses?

$$\begin{aligned}\text{Student} \sqsubseteq\ & \exists \text{attends}.(\text{Course} \sqcap A) \sqcap \\ & \exists \text{attends}.(\text{Course} \sqcap \neg A \sqcap B) \sqcap \\ & \exists \text{attends}.(\text{Course} \sqcap \neg A \sqcap \neg B).\end{aligned}$$

Not possible to achieve “at most” restriction using similar trick.

ALC needs to be extended with new operators...

Number Restrictions and Qualified NRs

$\mathcal{L}$ - a description logic

n - a non-negative number

r - a role and

C - a possibly compound $\mathcal{L}$ concept description

A ***number restriction*** - a concept description of the form (≤n r) or (≥n r)

A ***qualified number restriction***

- a concept description of the form (≤n r.C) or (≥n r.C)

A description logic $\mathcal{L}$ that additionally has number restrictions - $\mathcal{LN}$

A description logic $\mathcal{L}$ that additionally has qualified number restrictions - $\mathcal{LQ}$

Interpreting NRs and QNRs

$\#\{ \dots \} = |\{ .. \}|$

To interpret NRs and QNRs,

an interpretation function maps the new constructs as follows:

$(\leq n\ r)^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \#\{ e \mid (d,e) \in r^{\mathcal{I}} \} \leq n \}$

$(\geq n\ r)^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \#\{ e \mid (d,e) \in r^{\mathcal{I}} \} \geq n \}$

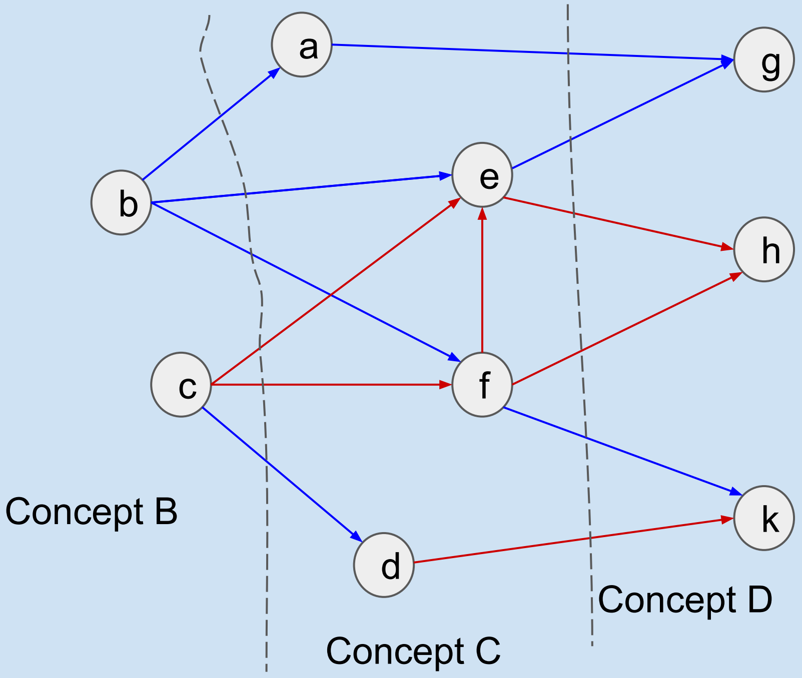
$(\leq n\ r.C)^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \#\{ e \mid (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}} \} \leq n \}$

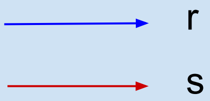
$(\geq n\ r.C)^{\mathcal{I}} = \{ d \in \Delta^{\mathcal{I}} \mid \#\{ e \mid (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}} \} \geq n \}$

$(=n\ r) \equiv (\leq n\ r) \sqcap (\geq n\ r)$

$(=n\ r.C) \equiv (\leq n\ r.C) \sqcap (\geq n\ r.C)$

Examples - NRs and QNRs





$\geq 1r = \{a,b,c,e,f\}$

$\geq 2s = \{c, f\}$

$\leq 1s = \{c, f\}^c$

$\leq 0s = \{a,b,g,h,k\}$

$\geq 1r.C = \{b, c\}$

$\geq 2s.C = \{c\}$

$\geq 1s.D = \{d, e, f\}$

Nominals

Individual names - used in the ABox only, so far.

Suppose Mary is an individual name. And let's define a concept “courses of Mary”:

CourseOfMary ≡ Course ⊓ ∃ taughtBy.Mary

Error: ‘Mary’ - used as a concept name

Concept names ∩ Individual names = Φ as per our definitions.

Again, a new construct is needed..

For any individual name b, { b } is called a ***nominal***

It is a new concept constructor

An interpretation $\mathcal{I}$ maps it as : $(\{ b \})^{\mathcal{I}} = \{ b^{\mathcal{I}} \}$

CourseOfMary ≡ Course ⊓ ∃ taughtBy.{ Mary } -- works!

$\mathcal{L}$ - a DL
If $\mathcal{L}$ additionally allows nominals,
We get a DL $\mathcal{LO}$

Role Hierarchies - back to LazyStudents

Lazy students dont attend anything actively. Extend $\mathcal{T}_{ex}$ like this...

$$\mathcal{T}_2 = \mathcal{T}_{ex} \cup \{ \text{ LazyStudent} \equiv \text{Student} \sqcap \forall \text{attendsActively}.\bot \ \}$$

Let $\mathcal{K} = (\mathcal{T}_2$, { (Bob, CS600): attendsActively })

$\mathcal{K}$ does not entail (Bob: Student) yet !!

We need to say that (x, attendsActively, y) → (x, attends, y) !!

Role Inclusion Axiom (RIA)s : Axioms of the form $r \sqsubseteq s$ where r, s are role names.

Interpretation $\mathcal{I}$ is a model of an RIA $r \sqsubseteq s$ if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.

$\mathcal{T}_3 = \mathcal{T}_{ex} \cup$ {attendsActively $\sqsubseteq$ attends, LazyStudent ≡ Student $\sqcap \forall$ attendsActively.$\bot$ }

Now, ($\mathcal{T}_3$, { (Bob, CS600): attendsActively }) $\models$ Bob: Student

LazyStudent is satisfiable wrt $\mathcal{T}_3$.

Transitive Roles - the need..

$$
\begin{aligned}
\mathsf{Course} &\sqsubseteq \exists hasPart.\mathsf{Section} \sqcap \forall hasPart.\mathsf{Section},\\
\mathsf{Section} &\sqsubseteq \forall hasPart.\mathsf{Section},\\
\mathsf{TeachableCourse} &\equiv \mathsf{Course} \sqcap \forall hasPart.\mathsf{Ready}.
\end{aligned}
$$

Note: A part can be ready even though its sub-part is not ready!

$$
\begin{aligned}
\Delta^{\mathcal{I}} &= \{c, s_1, s_2, s_3, \ldots\},\\
\mathsf{Section}^{\mathcal{I}} &= \{s_1, s_2, s_3\},\\
\mathsf{Ready}^{\mathcal{I}} &= \{s_1, s_2\},\\
\mathsf{Course}^{\mathcal{I}} &= \{c\},\\
hasPart^{\mathcal{I}} &= \{(c, s_1), (c, s_2), (s_1, s_3)\}.
\end{aligned}
$$

Now, c $\in$ TeachableCourse^{I}

Intuitively, we dont expect this..

c hasPart s_1,

s_1 hasPart s_3 and

s_3 is not in Ready

Our intention: a course is teachable

if all *direct* sections and *indirect* sections are ready.

We like hasPart to be treated as *transitive*.

Role Transitivity Axioms

In a description logic $\mathcal{L}$,

A role transitivity axiom is an axiom of the form

Trans(r)

where r is a role name

An interpretation $\mathcal{I}$ is a model of Trans(r) if $r^{\mathcal{I}}$ is transitive.

The extension of $\mathcal{ALC}$ with role transitivity axioms is usually called $\mathcal{S}$

Names for the new DLs

$\mathcal{L}$ - a description logic

$\mathcal{ALC}$ extended with role inverses - $\mathcal{ALCI}$ (In general, $\mathcal{LI}$)

$\mathcal{ALC}$ extended with number restrictions - $\mathcal{ALCN}$ (In general, $\mathcal{LN}$)

$\mathcal{ALC}$ extended with qualified number restrictions - $\mathcal{ALCQ}$ (In general, $\mathcal{LQ}$)

$\mathcal{ALC}$ extended with nominals - $\mathcal{ALCO}$ (In general, $\mathcal{LO}$)

Multiple extension are also possible

-- $\mathcal{ALCIN}$ $\mathcal{ALCIQ}$ $\mathcal{ALCN}$ $\mathcal{ALCQ}$ $\mathcal{ALCO}$ $\mathcal{ALCIOQ}$ ….

If role hierarchy axioms are allowed in a DL, we add $\mathcal{H}$ to its name.

$\mathcal{SHIQ}$ - ALC with role hierarchy axioms, inverses, Trans axioms and qualified number restrictions.