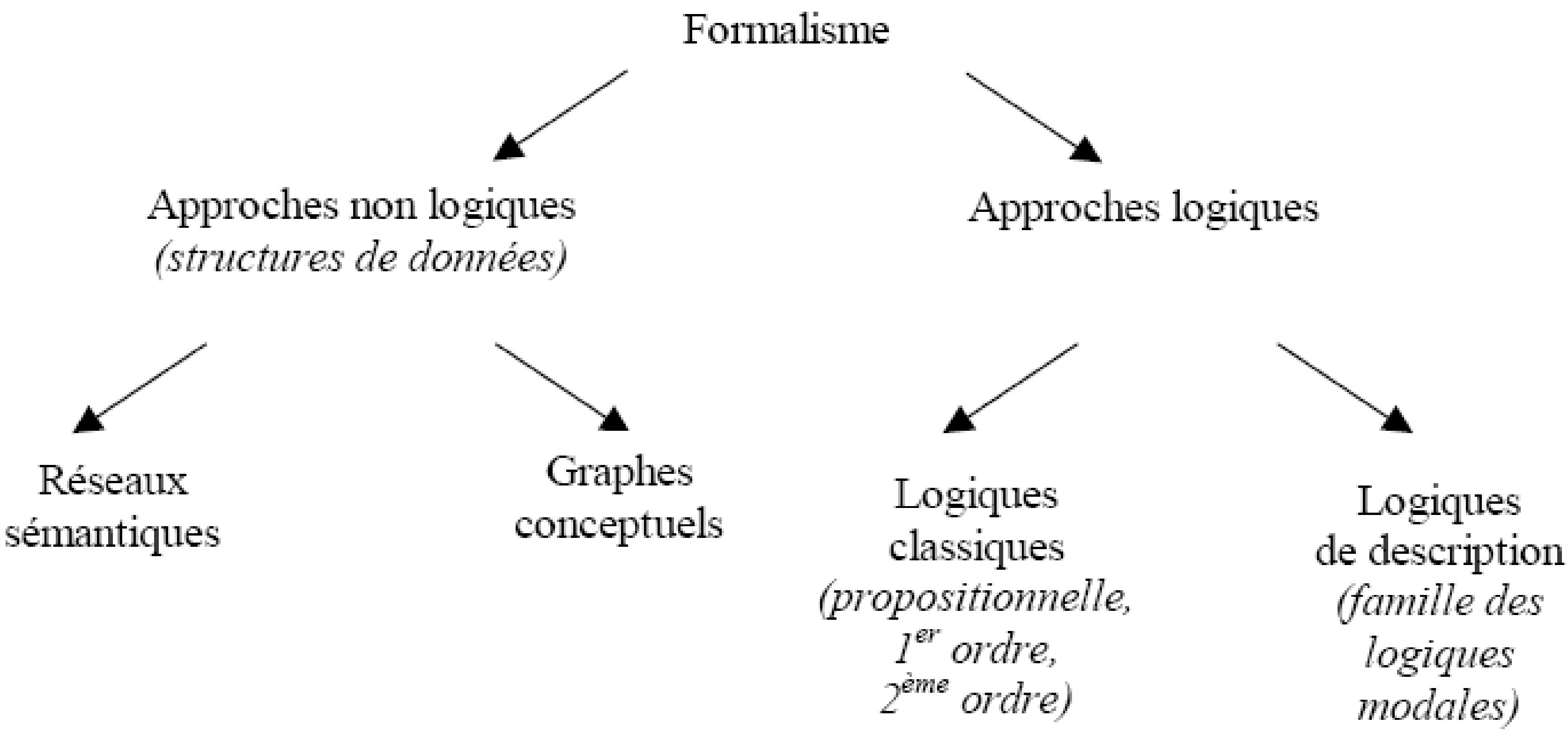
Ontologies et Web sémantique

Cours 5: Logique et ontologie

Dr. TA Tuan Anh
ttanh@ciid.vast.vn

Une classification (rappel)



Logique propositionnelle

- Une logique classique et considérée comme une logique de base (Aristotle, 300 BC)
- Vue d’ensemble
  - formules de la logique propositionnelle
  - tables de vérité
  - interprétation, satisfiabilité, tautologies
  - inférence

Propositions

- Ce qui est vrai ou faux
  - “il pleut”, “il y a un bon a la télévision”,…
- Les symboles du calcul propositionnel (atomes)
  - p1 = “il pleut”,
  - p2 =“il y a un bon a la télévision”
  - …
- Les connecteurs permettent de former de nouveaux énonces (étant donné deux propositions p et q)
  - $\neg$ (non) : $\neg p$
  - $\wedge$ (et) : $p \wedge q$
  - $\vee$ (ou) : $p \vee q$
  - $\rightarrow$ (implique) : $p \rightarrow q$
  - $\leftrightarrow$ (équivaut) : $p \leftrightarrow q$
- Exemple de propositions bien formées
  - $((p \wedge q) \vee r) \rightarrow (p \wedge r)$

Tables de vérité

$\neg$	V	F
	F	V

P $\neg$P1

$\wedge$	V	F
V	V	F
F	F	F

P1 $\wedge$ P2

$\vee$	V	F
V	V	V
F	V	F

P1 $\vee$ P2

$\rightarrow$	V	F
V	V	F
F	V	V

P1 $\rightarrow$ P2

$\leftrightarrow$	V	F
V	V	F
F	F	V

P1 $\leftrightarrow$ P2

Interprétation

- Une interprétation $\mathcal{I}$ est une fonction de l'ensemble des propositions atomiques dans {*vrai*, *faux*}, telle que $\mathcal{I}(\top)$= *vrai* et $\mathcal{I}(\bot)$= *faux*
- $\mathcal{I}$ satisfait formule P, ou P est valide dans $\mathcal{I}$ (noté $\mathcal{I}$ |=P) si et seulement si l'une des conditions suivante est vérifie :
  - si P est un atome alors $\mathcal{I}$(P) = *vrai*
  - si P = ¬P' alors not ($\mathcal{I}$ |= P')
  - si P = P1 $\vee$ P2 alors soit $\mathcal{I}$ |= P1, soit $\mathcal{I}$ |= P2
  - si P = P1 ^ P2 alors a la fois $\mathcal{I}$ |= P1 et $\mathcal{I}$ |= P2
  - si P = P1 $\rightarrow$ P2 alors soit not ($\mathcal{I}$ |= P1), soit $\mathcal{I}$ |= P2
  - si P = P1 $\leftrightarrow$ P2 alors soit $\mathcal{I}$ |= P1 et $\mathcal{I}$ |= P2, soit $\mathcal{I}$ |= ¬P1 et $\mathcal{I}$ |= ¬P2

Satisfiable, tautologie

- La formule A est satisfiable ssi il existe au moins une interprétation I telle que $\mathcal{I} \models A$
  - E.g., $A = p \vee q$
  - A est satisfiable avec 3 interprétations
    - $\mathcal{I}(p)$ = vrai, $\mathcal{I}(q)$ = faux
    - $\mathcal{I}(p)$ = vrai, $\mathcal{I}(q)$ = vrai
    - $\mathcal{I}(p)$ = faux, $\mathcal{I}(q)$ = vrai

- La formule A est une tautologie ssi toutes les interprétations vérifient $\mathcal{I} \models A$ (A est toujours vraie)

Exemple des tautologies

$\models p \Rightarrow p$ (loi d'identité pour l'implication)

$\models p \Leftrightarrow p$ (loi d'identité pour l'équivalence)

$\models p \vee \neg p$ (loi du tiers exclu)

$\models \neg (p \wedge \neg p)$ (loi de non-contradiction)

$\models \neg (p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$

$\models \neg (p \vee q) \Leftrightarrow (\neg p \wedge \neg q)$ (lois de De Morgan)

Noter que les lois de De Morgan peuvent aussi bien s'écrire:

$\neg (p \wedge q) \equiv (\neg p \vee \neg q)$

$\neg (p \vee q) \equiv (\neg p \wedge \neg q)$

Inférence

- Soit A1, A2, ..., An des formules propositionnelles. Une proposition B sera dite être une conséquence logique de A1, A2, ... An, note {A1 , ... , An} |= B

- si et seulement si: pour toute interprétation $\mathcal{I}$ telle que $\mathcal{I}$ |= A1, $\mathcal{I}$ |= A2,..., $\mathcal{I}$ |= An, on a $\mathcal{I}$ |= B

- Exemple
  - $\{p , (p \Rightarrow q)\} \models q$

Exercice

- Démontrer les règles d'inférence suivants

$$\{A \Rightarrow B \, , B \Rightarrow C\} \models A \Rightarrow C$$

$$\{A \Rightarrow B\} \models (B \Rightarrow C) \Rightarrow (A \Rightarrow C)$$

$$\{A \, , B\} \models A \wedge B$$

$$\{A \wedge B\} \models A$$

$$\{ A \Rightarrow B, \; C \Rightarrow D\} \models (A \wedge C) \Rightarrow (B \wedge D)$$

$$\{A\} \models A \vee B$$

$$\{A \Rightarrow B \, , C \Rightarrow D\} \models (A \vee C) \Longrightarrow (B \vee D)$$

Moteurs d'inférence

- Deux types de moteurs d'inférence dans les systèmes d'expert
- Chaînage avant
  - Règle d'inférence : $\{A , (A \Rightarrow B)\} \models B$
  - On part des faits déjà admis puis on "avance" en appliquant un nombre indéterminé de fois la règle du détachement
- Chaînage arrière
  - Règle d'inférence : $\{(B \Rightarrow A), \neg A\} \models \neg B$
  - On part de la négation et on "remonte" jusqu'à ce qu'on ait trouvé la négation d'un fait antérieurement admis

Logique des prédicats

- Logique classique du premier ordre
- Logique de base pour
  - Programmation logique
  - Logiques de description
- Une logique pour décrire des objets, fonctions et relations. Le langage comporte
  - un ensemble symboles de constante, notés a, b,...
  - un ensemble de symboles de variables, notés x, y,...
  - un ensemble de symboles de fonction, notés f, g,...
  - un ensemble de symboles de relation, notés P, Q,...
  - un ensemble fini de connecteurs : ceux de la logique des propositions, auxquels on ajoute deux connecteurs, les quantificateurs $\forall$ et $\exists$

Termes

- Désignent les objets sur lesquels portent les relations
- L'ensemble des termes est le plus petit ensemble tel que
  - toute variable et constante est un terme
  - si f est une fonction d'arité n et $t_1, . . ., t_n$ sont des termes, alors $f(t_1, . . ., t_n)$ est un terme
- Exemple:
  - marie, john, a
  - x, y, z
  - père(marie), f(x), f(g(a))

Atomes

- Les plus petits éléments auxquels on puisse assigner la valeur vrai ou faux
- Si R est un symbole de relation et $t_1, \ldots, t_n$ sont des termes, $R(t_1, \ldots, t_n)$ est une atome
- Si t_1, t_2 sont deux termes, $t_1 = t_2$ est une atome
- Exemple:
  - épouse(marie, john)
  - épouse(père(john), mère(john))
  - père(marie) = john
  - P(x, y), p(f(x), f(y))
  - x = f(y)

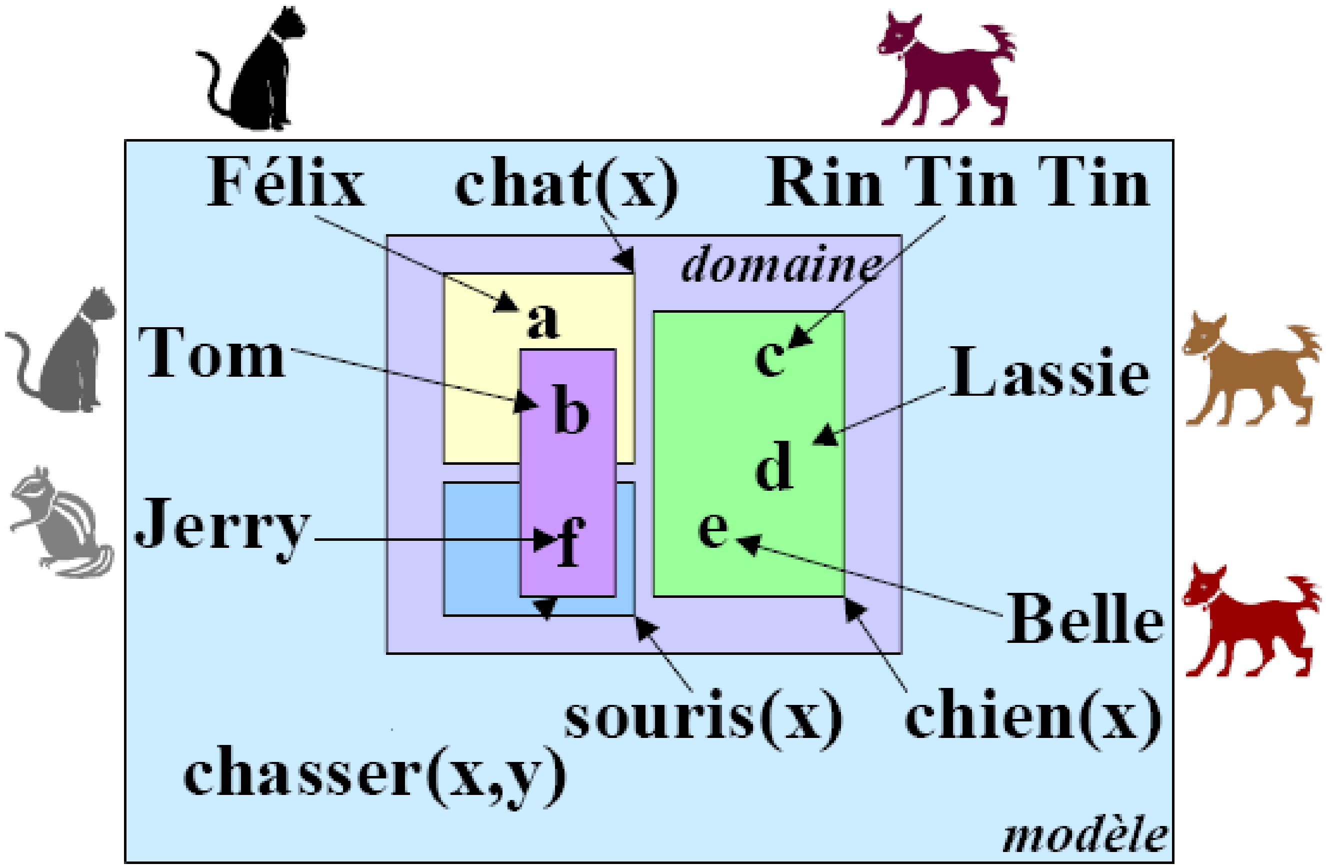
Formules

- L'ensemble des formules bien formées est le plus petit ensemble tel que
  - tout atome est une formule
  - si F, F1 et F2 sont des formules, $\neg$F, F1 $\wedge$ F2, F1 $\vee$ F2, F1 $\rightarrow$ F2, F1 $\leftrightarrow$ F2 sont des formules
  - si F est une formule, $\forall x_1, x_2, \ldots, x_n$ F et $\exists x_1, x_2, \ldots, x_n$ F sont des formules
- Exemple
  - $\forall$x épouse(père(x), mère(x))
  - $\forall$x avoirFils(x, y) $\rightarrow \exists$ y (x = père(y) $\vee$ x = mère(y))

Interprétation

- The meaning of a First-Order formula is assigned using an *interpretation*
- An interpretation $\mathcal{I}$ consists of:
  - Domain Δ: a set of objects
  - A set of relations R: $\Delta \times ... \times \Delta$
  - A set of functions F: $\Delta \times ... \times \Delta^n \mapsto \Delta$
  - A mapping function $\cdot$ which:
    - Maps constants to objects: $c^{\mathcal{I}} \in \Delta$
    - Maps predicate symbols to relations: $p^{\mathcal{I}} \subseteq \Delta^n$
    - Maps function symbols to functions: $f^{\mathcal{I}} \subseteq \Delta^n \longrightarrow \Delta$
- An interpretation is a *model* of a formula A if it makes the formula *true*:
  - $\mathcal{I} \models A$

Exemple



Félix
chat(x)
Rin Tin Tin
domaine
a
Tom
b
c
Lassie
d
Jerry
f
e
Belle
souris(x)
chien(x)
chasser(x,y)
modèle

Satisfiable (1)

$p(t_1, ..., t_n)$ (atomic formula)	is true iff	$\langle t_1^{\mathcal{I}}, ..., t_n^{\mathcal{I}} \rangle \in p^{\mathcal{I}}$
$\neg A$	is true iff	$A^{\mathcal{I}}$ is *not* true
$A \wedge B$	is true iff	$A^{\mathcal{I}}$ *and* $B^{\mathcal{I}}$ are true
$A \vee B$	is true iff	$A^{\mathcal{I}}$ *or* $B^{\mathcal{I}}$ is true (or both)
$A \rightarrow B$	is true iff	in every case where $A^{\mathcal{I}}$ is true, $B^{\mathcal{I}}$ is true

Satisfiable (2)

- We have not discussed semantics of variables
- Variables *have no semantics*
- What to do with variables?
- *Assign* values to variables using an assignment B
  - e.g., $\{x \mapsto a, y \mapsto john\}$
- An interpretation $\mathcal{I}$ makes a formula A *true* under a variable assignment B:
  - $\mathcal{I} \models_B A$
- Quantifiers:
  - $\exists x.A$: there exists an assignment for x which makes A true
  - $\forall x.A$: for all possible assignments of x, A is true

Exemples

- Langage
  - Constants : bill, hillary, tom
  - Relations : man, person, animal
  - Formule F = $\forall x.man(x) \rightarrow person(x)$
- $\triangle$ = {a, b, c}
- Example 1: Interpretation $\mathcal{I}$
  - $bill^{\mathcal{I}} = a$, $hillary^{\mathcal{I}} = b$, $tom^{\mathcal{I}} = c$
  - $person^{\mathcal{I}} = \{a, b\}$, $man^{\mathcal{I}} = \{a\}$, $animal = \{c\}$
  - $\mathcal{I} \models F$
- Example 2: Interpretation $\mathcal{I}$
  - $bill^{\mathcal{I}} = a$, $hillary^{\mathcal{I}} = b$, $tom^{\mathcal{I}} = c$
  - $person^{\mathcal{I}} = \{a, b\}$, $man^{\mathcal{I}} = \{a, c\}$, $animal = \{c\}$
  - not ($\mathcal{I} \models F$)

Description Logics

- Based on concepts and roles (frame paradigm)
  - Concepts (classes) are interpreted as sets of objects
  - Roles (properties) are interpreted as binary relations on objects
- DLs allow building complex concepts and roles from simpler ones using
  - Conjunction, disjunction, negation
  - Restricted forms of quantification
- Most DLs similar to 2-variable fragment of FOL
  - Classes correspond to unary predicates
  - Properties correspond to binary predicates
  - No function symbols

DL Basics

- Concepts (classes)
  - E.g., Person, Doctor, Parent
- Roles (properties)
  - E.g., hasChild, hasName, hasAncestor
- Individuals (objects)
  - E.g., John, Mary, Italy
- Constructors for forming concepts, e.g., subsumption, disjunction, conjunction,…
  - Man $\sqsubseteq$ Person
  - Woman $\equiv$ Person $\sqcap$ $\neg$Man
- Constructors for forming roles, e.g., inverse, transitive, symmetric, …
  - isChildOf $\equiv$ hasChild^{-}

$\mathcal{ALC}$

- The simplest Description Logic
  - Concepts constructed using $\sqcap$, $\sqcup$, $\neg$, $\exists$, $\forall$
  - Only atomic roles (i.e., no inverse, transitive, …)
- Example: Person $\sqcap$ $\forall$hasChild.Doctor
  - Person all of whose children are doctors

$\mathcal{ALC}$ Syntax

$A \mid$	(atomic concept)
$\top \mid$	(universal concept)
$\bot \mid$	(bottom concept)
$C \sqcap D \mid$	(intersection)
$C \sqcup D \mid$	(disjunction)
$\neg C \mid$	(negation)
$\forall R.C \mid$	(value restriction)
$\exists R.C \mid$	(existential quantification)

Individual assertions in $\mathcal{ALC}$:

$a \in C$
$\langle a, b \rangle \in R$

Axioms in $\mathcal{ALC}$:

$C \sqsubseteq D$
$C \equiv D$

Ontology example

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

DL Knowledge Base

- Axioms in the TBox to define terminology
  - Subsumption: $C \sqsubseteq D$
  - Equivalence: $C \equiv D$
- Assertions in the ABox
  - $C(a)$ where C is a concept and a is an individual
  - $R(a, b)$ where C is a relation
- Knowledge base Σ = <TBox, ABox>
  - Set of TBox statements
  - Set of ABox statements

KB example

TBox

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

ABox

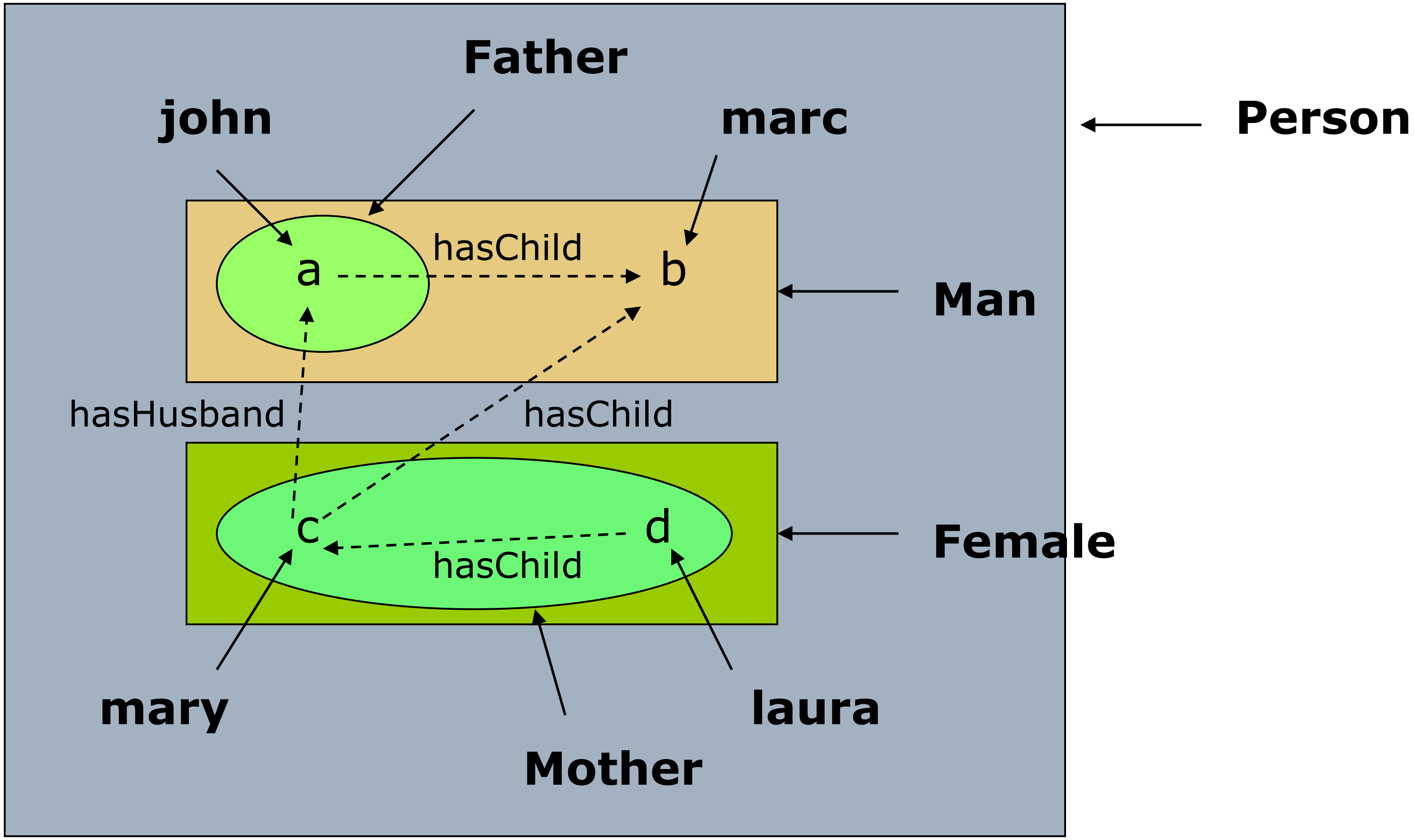
Man(john) Man(marc)
Woman(mary) Woman(laura)
hasHushband(mary, john)
hasChild(john, marc)
hasChild(mary, marc)
hasChild(laura, mary)

Interpretation of $\mathcal{ALC}$

Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	primitive concept
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	primitive role
$\top$	$\Delta^{\mathcal{I}}$	top
$\bot$	$\emptyset$	bottom
$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	complement
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	conjunction
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	disjunction
$\forall R.C$	$\{x \mid \forall y.\, R^{\mathcal{I}}(x, y) \rightarrow C^{\mathcal{I}}(y)\}$	universal quant.
$\exists R.C$	$\{x \mid \exists y.\, R^{\mathcal{I}}(x, y) \wedge C^{\mathcal{I}}(y)\}$	existential quant.

Interpretation example



Semantics of $\mathcal{ALC}$

An interpretation $\mathcal{I}$ *satisfies*:

concept definition	$C \equiv D$	iff	$C^{\mathcal{I}} = D^{\mathcal{I}}$
axiom	$C \sqsubseteq D$	iff	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
TBox	$\mathcal{T}$	iff	$\mathcal{I}$ *satisfies* all axioms in $\mathcal{T}$
			$\mathcal{I}$ is a *model* of $\mathcal{T}$
concept assertion	$a : C$	iff	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$\langle a, b \rangle : R$	iff	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
ABox	$\mathcal{A}$	iff	$\mathcal{I}$ *satisfies* all assertions in $\mathcal{A}$
			$\mathcal{I}$ is a *model* of $\mathcal{A}$

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is said to be a *model* of a knowledge base Σ if every axiom of Σ is satisfied by $\mathcal{I}$.

A knowledge base Σ is said to be *satisfiable* if it admits a model.

Reasoning in $\mathcal{ALC}$

- Σ is a knowledge base
- S is a sentence: axiom or assertion
- S is entailed by Σ, denoted by $\Sigma \models$ S, iff for every model $\mathcal{I}$ for knowledge base Σ, $\mathcal{I}$ satisfies S
- Example
  - Let Σ be the knowledge base given in the previous have
    - $\Sigma \models$ Wife(mary)
    - $\Sigma \models$ Grandmother(laura)
    - $\Sigma \models$ Grandmother $\sqsubseteq$ Mother

Types of reasoning

- **Concept Satisfiability**

$\Sigma \not\models C \equiv \bot$ $\quad\quad$ $\texttt{Student} \sqcap \neg\texttt{Person}$

the problem of checking whether C is satisfiable w.r.t. Σ, i.e. whether there exists a model $\mathcal{I}$ of Σ such that $C^{\mathcal{I}} \neq \emptyset$

- **Subsumption**

$\Sigma \models C \sqsubseteq D$ $\quad\quad$ $\texttt{Student} \sqsubseteq \texttt{Person}$

the problem of checking whether C is subsumed by D w.r.t. Σ, i.e. whether $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in every model $\mathcal{I}$ of Σ

- **Satisfiability**

$\Sigma \not\models$ $\quad\quad$ $\texttt{Student} \doteq \neg\texttt{Person}$

the problem of checking whether Σ is satisfiable, i.e. whether it has a model

- **Instance Checking**

$\Sigma \models C(a)$ $\quad\quad$ $\texttt{Professor(john)}$

Reduction to satisfiability

- Concept Satisfiability

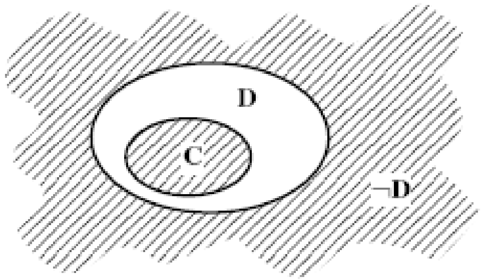
  $\Sigma \not\models C \equiv \bot \quad \leftrightarrow$

  exists x s.t. $\Sigma \cup \{C(x)\}$ has a model

- Subsumption

  $\Sigma \models C \sqsubseteq D \quad \leftrightarrow$

  $\Sigma \cup \{(C \sqcap \neg D)(x)\}$ has no models



- Instance Checking

  $\Sigma \models C(a) \quad \leftrightarrow$

  $\Sigma \cup \{\neg C(a)\}$ has no models

Example of concept satisfiability

$parent \equiv person \sqcap \exists has\_child.person$
$woman \equiv person \sqcap female$
$mother \equiv person \sqcap \exists has\_child.person \sqcap female$

- How about :woman υ mother?

$$\neg woman \sqcap mother \equiv$$
$$\neg(female \sqcap person) \sqcap female \sqcap parent \equiv$$
$$(\neg female \sqcup \neg\ person) \sqcap female \sqcap parent \equiv$$
$$(\neg female \sqcup \neg\ person) \sqcap female \sqcap parent \equiv$$
$$\neg person \sqcap female \sqcap parent \equiv$$
$$\neg person \sqcap female \sqcap person \sqcap \exists has\_child.person \equiv$$
$$\neg person \sqcap female \sqcap person \sqcap \exists has\_child.person$$

↯

- So, no mother is not a women

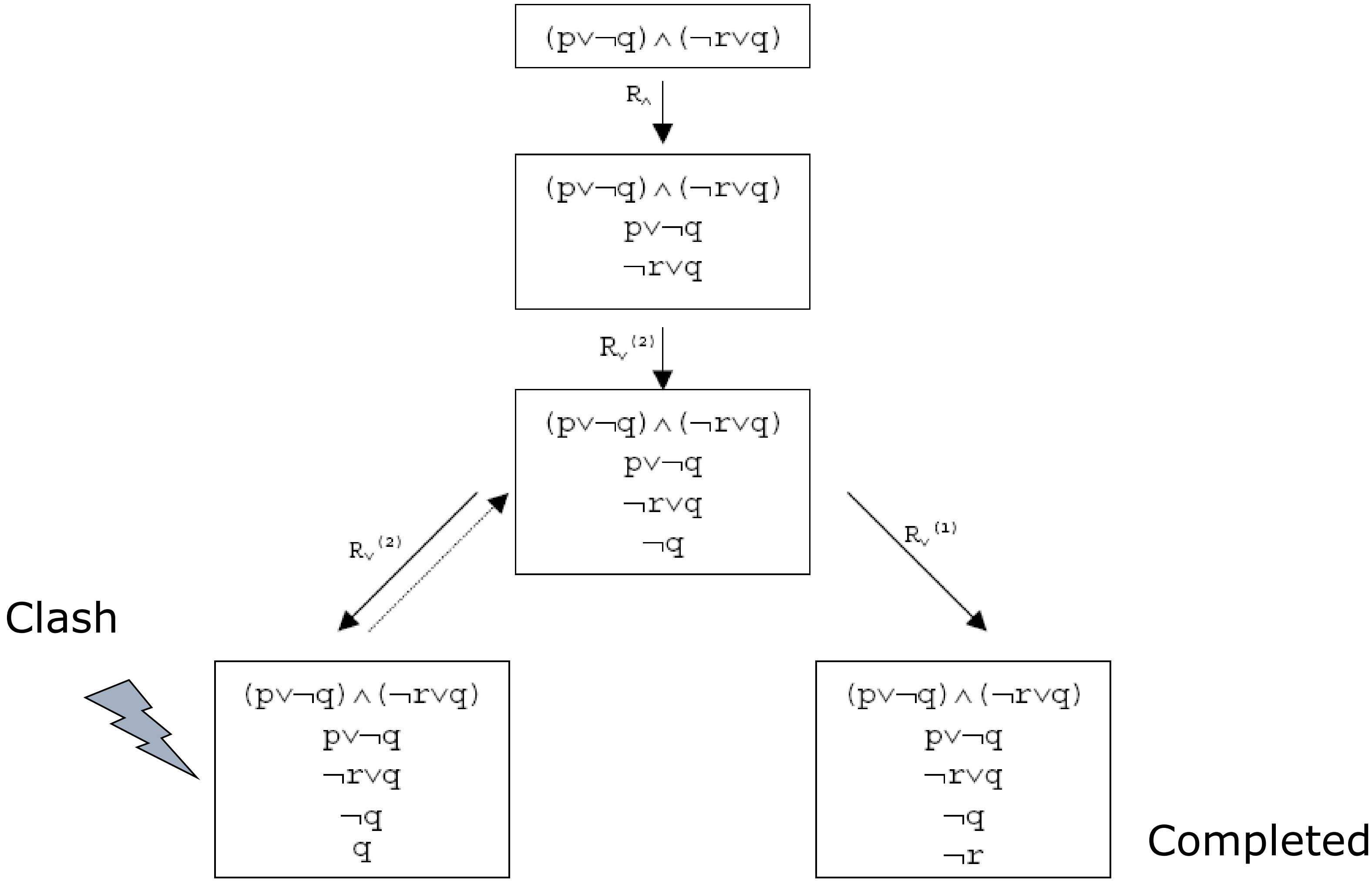
Reasoning procedures

- Terminating, efficient and complete algorithms for deciding satisfiability - and all the other reasoning services - are available for $\mathcal{ALC}$
- Algorithms are based on tableaux-calculus techniques
- Completeness is important for the usability of description logics in real applications
- Such algorithms are efficient for both average and real knowledge bases, even if the problem in the corresponding logic is in PSPACE or EXPTIME

Tableaux Calculus

- The basic idea is to incrementally build the model by looking at the formula, by decomposing it in a top/down fashion. The procedure exhaustively looks at all the possibilities, so that it can eventually prove that no model could be found for unsatisfiable the formula
  - Syntactically transform the formula to a *Constraint System* S, also called *tableaux*
  - Add constraints to S, applying specific *completion rules* (eventually yield several possible alternative branches)
  - Apply the completion rules until either a contradiction (a *clash*) is generated in every branch, or there is a *completed* branch where no more rule is applicable
  - A model corresponds to a particular completed branch of the tableaux

Example



(p∨¬q) ∧ (¬r∨q)
R∧
(p∨¬q) ∧ (¬r∨q)
p∨¬q
¬r∨q
R∨(2)
(p∨¬q) ∧ (¬r∨q)
p∨¬q
¬r∨q
¬q
R∨(2)
R∨(1)
Clash
(p∨¬q) ∧ (¬r∨q)
p∨¬q
¬r∨q
¬q
q
(p∨¬q) ∧ (¬r∨q)
p∨¬q
¬r∨q
¬q
¬r
Completed

Tableaux algorithm for $\mathcal{ALC}$

- Given a concept C
- Proof process
  - Break down C syntactically in conjunction form $\{C_1, C_2, \ldots\}$
  - Work only on concepts in negation normal form
    - Using Morgan's rules, e.g., $\neg\exists R.C \equiv \forall R.\neg C$
  - Decompose concepts using tableau rules (corresponding to constructors in logic)
  - Stop when clash occurs, i.e., $\{C_1, \neg C_1, \ldots\}$, or when no more rules are applicable
  - Detect cycles to guarantee termination
- C is unsastisfiable iff a clash is generated in every tableaux branches

Negation Normal Form

- We can transform any $\mathcal{ALC}$ formula into an equivalent one in Negation Normal Form, so that negation appears only in front of atomic concepts:
  - $\neg(C \sqcap D) \iff \neg C \sqcup \neg D$
  - $\neg(C \sqcup D) \iff \neg C \sqcap \neg D$
  - $\neg(\forall R.C) \iff \exists R.\neg C$
  - $\neg(\exists R.C) \iff \forall R.\neg C$

Tableaux rules for $\mathcal{ALC}$

$x \bullet \{C_1 \sqcap C_2, \ldots\}$	$\rightarrow_\sqcap$	$x \bullet \{C_1 \sqcap C_2, C_1, C_2, \ldots\}$
$x \bullet \{C_1 \sqcup C_2, \ldots\}$	$\rightarrow_\sqcup$	$x \bullet \{C_1 \sqcup C_2, C, \ldots\}$ for $C \in \{C_1, C_2\}$
$x \bullet \{\exists R.C, \ldots\}$	$\rightarrow_\exists$	$x \bullet \{\exists R.C, \ldots\}$ R $y \bullet \{C\}$
$x \bullet \{\forall R.C, \ldots\}$ R $y \bullet \{\ldots\}$	$\rightarrow_\forall$	$x \bullet \{\forall R.C, \ldots\}$ R $y \bullet \{C, \ldots\}$

Tableaux example

Satisfiability of the concept:

$$\boxed{\big((\forall \mathtt{CHILD.Male}) \sqcap (\exists \mathtt{CHILD}.\neg \mathtt{Male})\big)}$$

$\big((\forall \mathtt{CHILD.Male}) \sqcap (\exists \mathtt{CHILD}.\neg \mathtt{Male})\big)(\mathtt{x})$	
$(\forall \mathtt{CHILD.Male})(\mathtt{x})$	$\sqcap$-*rule*
$(\exists \mathtt{CHILD}.\neg \mathtt{Male})(\mathtt{x})$	"
$\mathtt{CHILD}(x, y)$	$\exists$-*rule*
$\neg \mathtt{Male}(y)$	"
$\mathtt{Male}(y)$	$\forall$-*rule*
$\langle CLASH \rangle$	

Tableaux with individuals

Check the satisfiability of the ABox:

$(\mathtt{Parent} \sqcap \forall\mathtt{CHILD.Male})(\mathtt{john})$

$\neg\mathtt{Male}(\mathtt{mary})$

$\mathtt{CHILD}(\mathtt{john}, \mathtt{mary})$

$\mathtt{john} : \mathtt{Parent} \sqcap \forall\mathtt{CHILD.Male}$	
$\mathtt{mary} : \neg\mathtt{Male}$	
$\mathtt{john}\ \mathtt{CHILD}\ \mathtt{mary}$	
$\mathtt{john} : \mathtt{Parent}$	$\sqcap$-*rule*
$\mathtt{john} : \forall\mathtt{CHILD.Male}$	"
$\mathtt{mary} : \mathtt{Male}$	$\forall$-*rule*
$\langle CLASH \rangle$	

The knowledge base is inconsistent.

Exercise

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

- Check satisfiability of $C \equiv \text{Grandmother} \sqcap \neg\text{Mother}$

Extensions of $\mathcal{ALC}$

- $\mathcal{S}$ often used for $\mathcal{ALC}$ with transitive roles (R_+)
- $\mathcal{H}$ for role hierarchy (e.g., hasDaughter $\sqsubseteq$ hasChild)
- $\mathcal{O}$ for nominals/singleton classes (e.g., {Italy})
- $\mathcal{I}$ for inverse roles (e.g., isChildOf $\equiv$ hasChild$^-$)
- $\mathcal{N}$ for number restrictions (e.g., >2hasChild)
- $\mathcal{Q}$ for qualified number restrictions (e.g., >2hasChild.Doctor)
- $\mathcal{F}$ for functional selections. If a role is functional, we write: $\exists$f.C $\equiv$ f : C (e.g., hasMother : Woman)

Examples

- $\text{ParentWithManySons} \equiv \text{Parent} \sqcap {>}2\ \text{hasChild.Man}$
- $\text{parentOf} \equiv \text{hasChild}^{-}$
- $\text{parentOf} \sqsubseteq \text{ancestorOf}$
- $\text{ancestorOf} \sqsubseteq \text{ancestorOf}^{+}$
- $\text{Italian} \equiv \text{Person} \sqcap \exists\ \text{nationality.}\{\text{Italy}\}$

Extension semantics

Constructor	Syntax	Semantics
concept name	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction ($\mathcal{U}$)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation ($\mathcal{C}$)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
universal	$\forall R.C$	$\{x \mid \forall y : R^{\mathcal{I}}(x,y) \rightarrow C^{\mathcal{I}}(y)\}$
existential ($\mathcal{E}$)	$\exists R.C$	$\{x \mid \exists y : R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\}$
cardinality ($\mathcal{N}$)	$\geqslant n\ R$	$\{x \mid \sharp\{y \mid R^{\mathcal{I}}(x,y)\} \geq n\}$
	$\leqslant n\ R$	$\{x \mid \sharp\{y \mid R^{\mathcal{I}}(x,y)\} \leq n\}$
qual. cardinality ($\mathcal{Q}$)	$\geqslant n R.C$	$\{x \mid \sharp\{y \mid R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\} \geq n\}$
	$\leqslant n R.C$	$\{x \mid \sharp\{y \mid R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\} \leq n\}$
enumeration ($\mathcal{O}$)	$\{a_1 \ldots a_n\}$	$\{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$
selection ($\mathcal{F}$)	$f : C$	$\{x \in \mathrm{Dom}(f^{\mathcal{I}}) \mid C^{\mathcal{I}}(f^{\mathcal{I}}(x))\}$

Role semantics

Constructor	Syntax	Semantics
role name	P	$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
conjunction	$R \sqcap S$	$R^{\mathcal{I}} \cap S^{\mathcal{I}}$
disjunction	$R \sqcup S$	$R^{\mathcal{I}} \cup S^{\mathcal{I}}$
negation	$\neg R$	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}$
inverse	R^{-}	$\{(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (y, x) \in R^{\mathcal{I}}\}$
composition	$R \circ S$	$\{(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists z.\ (x, z) \in R^{\mathcal{I}} \wedge (z, y) \in S^{\mathcal{I}}\}$
range	$R\vert_C$	$\{(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
product	$C \times D$	$\{(x, y) \in C^{\mathcal{I}} \times D^{\mathcal{I}}\}$

Mapping $\mathcal{ALC}$ to FOL

A (atomic concept)	$A(x)$
$\top$	$\top$
$\bot$	$\bot$
$C \sqcap D$	$tr(C) \wedge tr(D)$
$C \sqcup D$	$tr(C) \vee tr(C)$
$\neg C$	$\neg tr(C)$
$\forall R.C$	$\forall y : R(x, y) \rightarrow tr(C, y)$
$\exists R.C$	$\exists y : R(x, y) \wedge tr(C, y)$

$a \in A$	$A(a)$
$\langle a, b \rangle \in R$	$R(a, b)$

$C \sqsubseteq D$	$\forall x . tr(C, x) \rightarrow tr(D, x)$
$C \equiv D$	$\forall x . tr(C, x) \leftrightarrow tr(D, x)$

Extending $\mathcal{ALC}$ to $\mathcal{SHOIN}$

Concept descriptions in $\mathcal{SHOIN}$:

$C, D \longrightarrow \{o_1, \ldots, o_n\} \mid$	(enumeration)
$\exists R.\{o\} \mid$	(hasValue)
$\geqslant nR \mid$	(minimal cardinality)
$\leqslant nR \mid$	(maximal cardinality)

Axioms in $\mathcal{SHOIN}$:

$Q \sqsubseteq R$	(role hierarchy)
$R \equiv Q^-$	(inverse roles)
$R^+ \sqsubseteq R$	(transitive roles)

Mapping example

- In DL

  $\text{associateProfessor} \sqsubseteq \text{academicStaffMember}$
  $\text{fullProfessor} \sqsubseteq \text{academicStaffMember}$
  $\text{fullProfessor} \sqsubseteq \neg\ \text{associateProfessor}$
  $\text{facultyMember} \equiv \text{academicStaffMember}$

- In FOL

  $\forall x.\text{associateProfessor}(x) \rightarrow \text{academicStaffMember}(x)$
  $\forall x.\text{fullProfessor}(x) \rightarrow \text{academicStaffMember}(x)$
  $\forall x.\text{fullProfessor}(x) \rightarrow \neg\ \text{associateProfessor}(x)$
  $\forall x.\text{facultyMember}() \leftrightarrow \text{academicStaffMember}(x)$

Mapping $\mathcal{SHOIN}$ to FOL

$\{o_1, \dots, o_n\}$	$x = o_1 \vee \dots \vee x = o_n$
$\exists R.\{o\}$	$R(x, o)$
$\geqslant nR$	$\exists y_1, \dots, y_n : \bigwedge R(X, y_i) \wedge \bigwedge y_i \neq y_j$
$\leqslant nR$	$\forall y_1, \dots, y_{n+1} : \bigwedge R(X, y_i) \rightarrow \bigvee y_i = y_j$

$Q \sqsubseteq R$	$\forall x, y : Q(x, y) \longrightarrow R(x, y)$
$R \equiv Q^-$	$\forall x, y : R(x, y) \leftrightarrow Q(y, x)$
$R^+ \sqsubseteq R$	$\forall x, y, z : R(x, y) \wedge R(y, z) \longrightarrow R(x, z)$

Example

In DL syntax:

$firstYearCourse \sqsubseteq \forall isTaughtBy.Professor$

$mathCourse \sqsubseteq \exists isTaughtBy.\{949352\}$

$academicStaffMember \sqsubseteq \exists teaches.undergraduateCourse$

$course \sqsubseteq \geq 1 isTaughtBy$

$department \sqsubseteq \geq 10 hasMember \sqcap \leq 30 hasMember$

FOL equivalent:

$\forall x.firstYearCourse(x) \rightarrow (\forall y.isTaughtBy(x, y) \rightarrow Professor(y))$

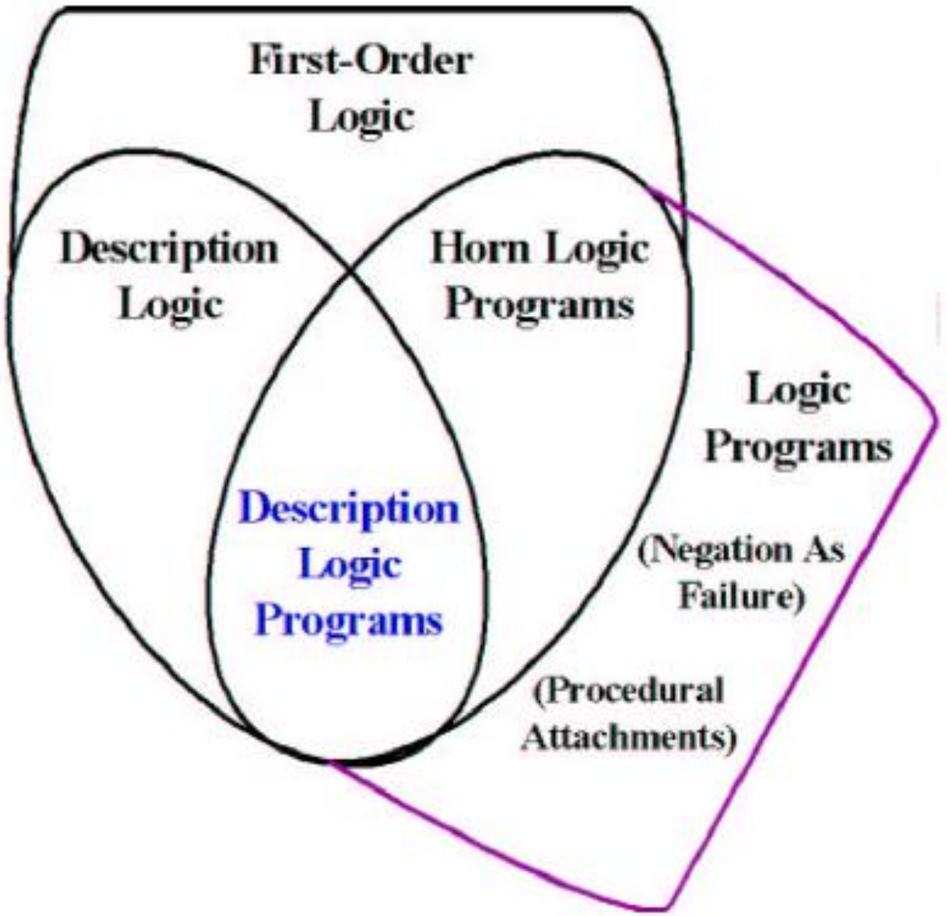
$\forall x.mathCourse(x) \rightarrow isTaughtBy(x, 949352)$

$\forall x.academicStaffMember(x) \rightarrow (\exists y.teaches(x, y) \wedge undergraduateCourse(y))$

$\forall x.course(x) \rightarrow (\exists y.isTaughtBy(x, y))$

$\forall x.department(x) \rightarrow$
$(\exists y_1, ..., y_{10}.hasMember(x, y_1) \wedge ... \wedge hasMember(x, y_{10}) \wedge y_1 \neq y_2 \wedge ... \wedge y_1 \neq$
$y_{10} \wedge ... \wedge y_9 \neq y_{10}) \wedge (\forall y_1, ..., y_{31}.hasMember(x, y_1) \wedge ... \wedge hasMember(x, y_{31}) \rightarrow$
$y_1 = y_2 \vee ... \vee y_1 = y_{31} \vee ... \vee y_{30} = y_{31})$

Expressivity overlaps



Further reading

- This course bases on the following materials:
  - Description Logic Handbook, edited by F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Cambridge University Press.
  - Online course on Description Logics of Enrico Franconi: http://www.inf.unibz.it/~franconi/dl/course/
  - Cours "Semantic Web" de Jos de Bruijn, http://www.debruijn.net/teaching/swt/