

Logique et représentation des connaissances

Motivations

Représentation en logique des prédicats

Projet CYC

Logique descriptive

Motivation - interprétation des questions

Question : « Quelle est la taille moyenne des suisses ? »

- ceux qui ont un passeport suisse ou ceux qui vivent en Suisse ?
- seulement les hommes ?
- y compris les enfants ?
- une estimation ou la moyenne sur tous les individus ?
- avec quelle précision ?

Un terme --> plusieurs définitions possibles

suyvant le contexte, le point de vue, le domaine d'activité

Motivation : développement des SI

Comment interroger correctement une base de données ?

- Dans quelle collection de données chercher, quels critères de sélection utiliser ?

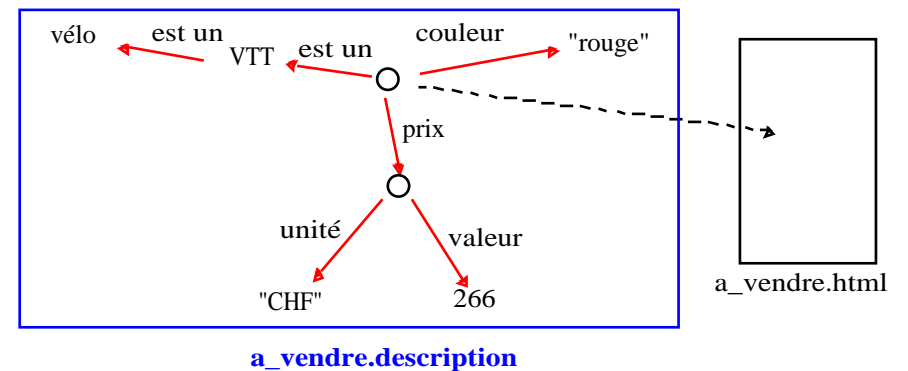
Comment s'assurer que les traitements correspondent aux intentions des utilisateurs ?

- « Prélever 0,01% de taxe sur toutes les transactions boursières à caractère spéculatif »
- « Imprimer une liste des **anciens employés** »

Motivation : Web sémantique

But : traitement automatique des pages Web

=> Décrire formellement le contenu des pages



⚠ Il faut se mettre d'accord sur la signification des étiquettes

Concept

Représentation mentale générale et abstraite d'un objet [Le Petit Robert]

Extension d'un concept : tous les objets qui appartiennent à ce concept

l'ensemble de toutes les pommes, {rouge, jaune, bleu}, {2, 3, 5, 7, 11, 13, 17, ...}

Compréhension d'un concept : les propriétés qui caractérisent les objets appartenant à ce concept

fruit, comestible, avec des pépins, rond, ...

véhicule, à moteur, avec des roues, peut transporter un ou plusieurs passagers, ...

On utilise un **terme** pour désigner un concept

pomme, poire, moteur à explosion, impôt sur la fortune

Utilisation de la logique des prédicats

But : définir des concepts

Méthode : donner des conditions nécessaires et suffisantes pour appartenir au concept.

Un concept --> un prédicat unaire

Une propriété --> un prédicat binaire

$\forall x. \text{chaise}(x) \Leftrightarrow \text{meuble}(x) \wedge$

$(\exists y. \text{dossier}(y) \wedge \text{partie}(x, y)) \wedge$

$(\exists z. \text{pied}(z) \wedge \text{partie}(x, z)) \wedge$

$\text{usage}(x, \text{"s'asseoir"}) \wedge$

$(\forall u. \text{partie}(x, u) \Rightarrow (\text{dossier}(u) \vee \text{pied}(u)))$

Un autre meuble

$\forall x. \text{bergère}(x) \Leftrightarrow$

$\text{fauteuil}(x) \wedge \text{dimension}(x, \text{"large"}) \wedge \text{dimension}(x, \text{"profond"}) \wedge$

$(\exists! y. \text{dossier}(y) \wedge \text{partie}(x, y) \wedge \text{dimension}(y, \text{"haut"})) \wedge$

$(\exists b1. \text{bras}(b1) \wedge \text{partie}(x, b1)) \wedge (\exists b2. \text{bras}(b2) \wedge \text{partie}(x, b2)) \wedge b1 \neq b2$

$\wedge (\exists j1. \text{joue}(j1) \wedge \text{partie}(b1, j1) \wedge \text{forme}(j1, \text{"plein"})) \wedge$

$(\exists j2. \text{joue}(j2) \wedge \text{partie}(b2, j2) \wedge \text{forme}(j2, \text{"plein"})) \wedge$

$(\exists c. \text{coussin}(c) \wedge \text{partie}(x, c))$

Questions

Etant donné des formules qui définissent des concepts

Etant donné une interprétation des prédicats de propriété

- Déterminer si un objet fait partie d'un concept
- Trouver tous les objets qui font partie d'un concept
- Tester si un concept est plus général qu'un autre (subsomption)
- Tester si un concept est satisfaisable

Exemple

Base de connaissance :

```
{<<définition de chaise>>, meuble(m1), meuble(m2),  
dossier(d1), dossier(d2),  
pied(p1), pied(p2), pied(p3), pied(p4),  
partie(m1, d1), partie(m1, p2), usage(m2, "s'asseoir"),  
usage(m1, "lire"), usage(m1, "s'asseoir")}
```

Peut-on en déduire

chaise(m1) ?

chaise(m2) ?

Projet CYC (www.cyc.com)

Représenter le « sens commun »

= toutes les connaissances que chacun possède

- les oiseaux volent (mais pas tous)
- on ne peut pas traverser un mur
- les automobiles ont un moteur

Utiliser ce sens commun pour développer des applications / systèmes plus « intelligents »

Représentation basée sur la logique des prédicats

- micro-théories, basées sur la théorie des ensembles (concepts)
- langage : plusieurs milliers de prédicats, fonctions, constantes
- millions d'axiomes

Sommet de la hiérarchie

43 groupes de concepts

Fundamentals	Composition of	Devices
Top Level	Substances	Construction
Time and Dates	Agents	Financial
Types of Predicates	Organizations	Food
Spatial Relations	Actors	Clothing
Quantities	Roles	Weather
Mathematics	Professions	Geography
Contexts	Emotion	Transportation
Groups	Propositional Attitudes	Information
"Doing"	Social	Perception
Transformations	Biology	Agreements
Changes Of State	Chemistry	Linguistic Terms
Transfer Of Possession	Physiology	Documentation
Movement	General Medicine	
Parts of Objects	Materials	
	Waves	

#\$Thing

#\$Thing is the universal set: the collection of everything!

Every Cyc constant in the Knowledge Base is a member of this collection; in the prefix notation of the language CycL,

we express that fact as (#\$isa CONST #\$Thing).

Thus, too, every collection in the Knowledge Base is a subset of the collection #\$Thing; in CycL,

we express that fact as (#\$genls COL #\$Thing).

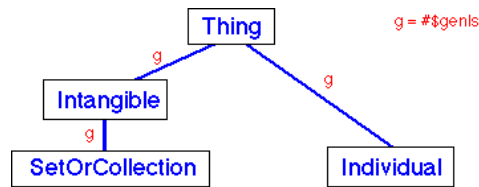
See #\$isa and #\$genls for further explanation of those relationships.

isa: #\$Collection

some subsets: #\$Intangible #\$Individual #\$ComputationalObject
#\$Microtheory (plus 1488 more public subsets, 13568 unpublished subsets)

Hiérarchie de concepts

Relation # \$genls entre ensembles



Le prédicat # \$genls exprime l'inclusion des extensions (ensembles des individus) des concepts concernés.

\$Intangible

The collection of things that are not physical -- are not made of, or encoded in, matter.

Every # \$Collection is an # \$Intangible (even if its instances are tangible), and so are some # \$Individuals.

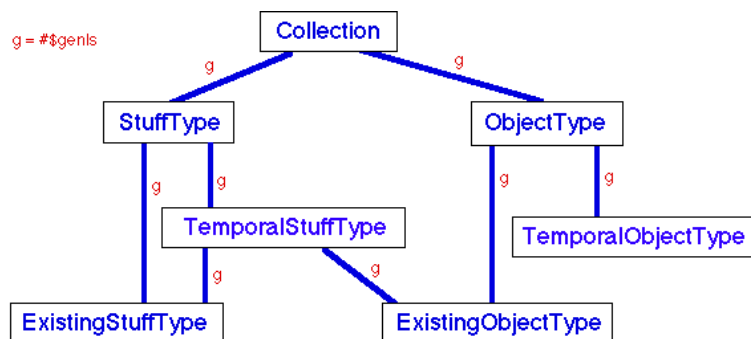
Caution: do not confuse 'tangibility' with 'perceivability' -- humans can perceive light even though it's intangible--at least in a sense.

For more on this issue, see the relevant # \$cyclistNotes.

isa: # \$Collection

genls: # \$Thing

Un bout de hiérarchie



Arithmetic Functions

\$FunctionFromQuantitiesToQuantities

isa: # \$RelationType

genls: # \$EvaluatableFunction # \$NonPredicateFunction

\$InverseFunc : <# \$FunctionFromQuantitiesToQuantities>
<# \$FunctionFromQuantitiesToQuantities>

isa: # \$BinaryPredicate # \$RelationshipPredicate # \$FunctionalSlot

\$PlusFn

isa: # \$FunctionFromQuantitiesToQuantities # \$CommutativeRelation
\$VariableArityRelation

argsIsa: # \$ScalarInterval

resultIsa: # \$ScalarInterval

Movement

#\$MovementEvent	#\$Translation-Complete
#\$MovementProcess	#\$Translation-Flow
#\$Movement-Rotation	#\$Translation-LocationChange
#\$Movement-TranslationEvent	#\$Translation-NoLocationChange
#\$Movement-Periodic	#\$Path-Generic
i#\$Movement-NonPeriodic	#\$Rotation-Periodic
#\$Movement-TranslationProcess	#\$Rotation-NonPeriodic
#\$Translation-SinglePath	
#\$Translation-MultiPath	
#\$Translation-Periodic	
#\$Translation-NonPeriodic	

Movement predicates

#\$SubjectMoving : <#\$MovementEvent> <#\$PartiallyTangible>
#\$movesInDirection-Throughout : <#\$PartiallyTangible>
<#\$UnitVectorInterval>
#\$movesInDirection-Average : <#\$PartiallyTangible> <#\$UnitVectorInterval>
#\$directionOfTranslation-Avg : <#\$Movement-TranslationEvent>
<#\$UnitVectorInterval>
#\$toLocation : <#\$Translocation> <#\$PartiallyTangible>
#\$fromLocation : <#\$Translocation> <#\$PartiallyTangible>
#\$pathway-Complete : <#\$Movement-TranslationEvent> <#\$Path-Generic>
#\$distanceTranslated : <#\$Movement-TranslationEvent> <#\$Distance>
#\$toOrientation : <#\$Movement-Rotation> <#\$OrientationAttribute>
#\$fromOrientation : <#\$Movement-Rotation> <#\$OrientationAttribute>

Inférence dans CyC

Système d'inférence sophistiqué pour prouver qu'une formule est une conséquence des axiomes et d'autres formules.

Rappel: aucun algorithme ne peut décider dans tous les cas si $F \models f$

Basé sur le **principe de résolution**

- donc peut s'engager sur la "mauvaise voie" et ne jamais finir

Contrôle de la profondeur et de la largeur des recherches

- l'utilisateur peut limiter l'espace de recherche (taille et temps)

Logiques descriptives (description logics)

Centrée sur la définition de concepts

Plus simple, utilisable/compréhensible par des non spécialistes

Moins expressif que la logique des prédicats

Système d'inférence plus simple / complet

Construction de bases de données terminologiques

Notions utilisées

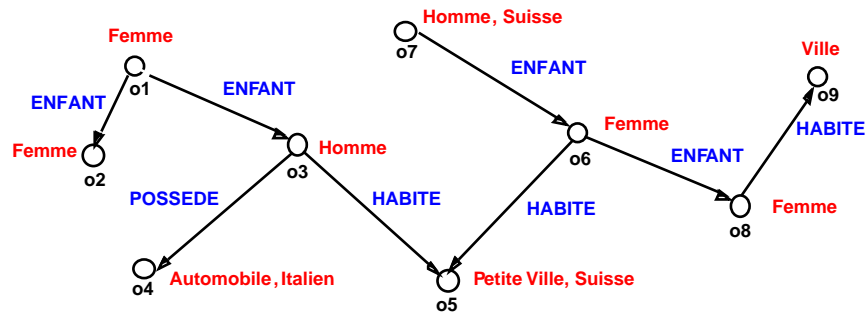
- définition de concepts et concepts prédéfinis
- rôles (caractères définitoires)
- individus

Niveau A (Assertion-Box)

On représente le monde par des objets et des liens.

Les objets appartiennent à des **concepts**

Les liens appartiennent à des **rôles**



Formules

Femme(o1), Femme(o2), Femme(o6), Femme(o8),

Homme(o3), Homme(o7),

Ville(o9), Petite Ville(o5), Automobile(o4),

Suisse(o5), Suisse(o7), Italien(o4),

ENFANT(o1, o3), ENFANT(o1, o2), ENFANT(o7, o6), ENFANT(o6, o8),

HABITE(o3, o5), HABITE(o6, o5), HABITE(o8, o9),

POSSEDE(o3, o4)

Définit l'interprétation des concepts et rôles de base

$Femme^I = \{o1, o2, o6, o8\}$, $Homme^I = \{o3, o7\}$, ...

$ENFANT^I = \{(o1, o3), (o1, o2), (o7, o6), (o6, o8)\}$,

$HABITE^I = \dots$

Niveau T (Terminological Box)

Définition des concepts, par des connecteurs et contraintes

Exemples de définition de concepts

Personne	la classe de toutes les personnes
Suisse \sqcap Français	les objets qui sont suisses et français
\exists POSSEDE . Bateau	ceux qui possèdent un bateau
\forall ENFANT . Homme	ceux dont tous les enfants sont des hommes
≤ 4 POSSEDE	ceux qui possèdent moins de 5 objets

Syntaxe et Interprétation

On se donne un domaine d'interprétation Δ^I

Interprétation d'un concept C : sous-ensemble C^I de Δ^I

Interprétation d'un rôle R : relation binaire R^I sur $\Delta^I \times \Delta^I$

Constructions ensemblistes

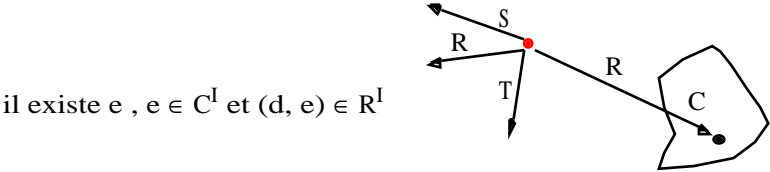
T	le tout	$T^I = \Delta^I$
\perp	rien	$\perp^I = \emptyset$
$C \sqcap D$	intersection	$(C \sqcap D)^I = C^I \cap D^I$
$C \sqcup D$	union	$(C \sqcup D)^I = C^I \cup D^I$
..		

Personne = Femme \sqcup Homme

Spécification existentielle

$\exists R . C$

Interprétation :
ensemble des individus $d \in \Delta^I$ tels que



$\exists \text{ ENFANT} . \text{Femme}$

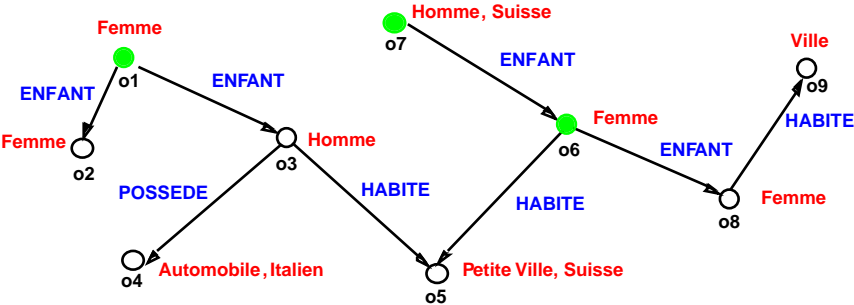
$\exists \text{ POSSEDE} . \text{Suisse}$

$\exists \text{ POSSEDE} . (\text{Automobile} \sqcap \text{Italien})$

$\exists \text{ ENFANT} . \text{Femme} \sqcap \exists \text{ ENFANT} . \text{Homme}$

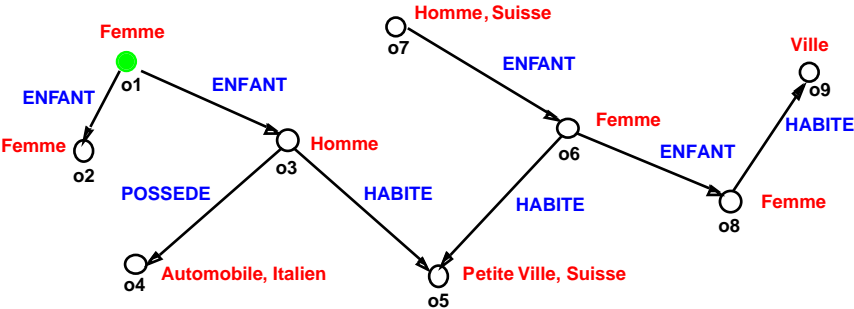
Exemples

$I(\exists \text{ ENFANT} . \text{Femme}) = \{o1, o7, o6\}$



Exemple

$I(\exists \text{ ENFANT} . \text{Femme} \sqcap \exists \text{ ENFANT} . \text{Homme})$



Contraintes de cardinalité

$$\leq n \mathbf{R}$$

Interprétation :

ensemble des individus $d \in \Delta^I$ tels que

$$\text{card}\{ e \in \Delta^I \mid (d, e) \in R^I \} \leq n$$

$$\leq 2 \text{ POSSEDE}$$

$$\geq 2 \text{ ENFANT}$$

Contrainte plus précise (dans certains langages)

$$\leq n \mathbf{R} . \mathbf{C}$$

$$\text{card}\{ e \in C^I \mid (d, e) \in R^I \} \leq n$$

Spécification universelle

$$\forall \mathbf{R} . \mathbf{C}$$

Interprétation :

ensemble des individus $d \in \Delta^I$ tels que

pour tout $e \in \Delta^I$ si $(d, e) \in R^I$ alors $e \in C^I$

$$\forall \text{ ENFANT} . \text{Femme}$$

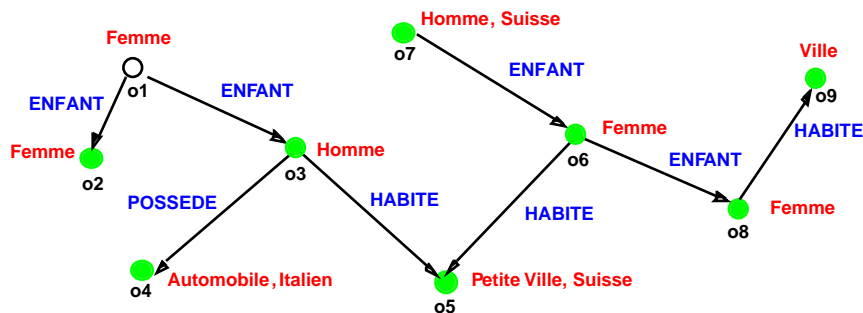
$$\forall \text{ ENFANT} . \text{Femme} \sqcap \exists \text{ ENFANT} . \text{Femme}$$

$$\exists \text{ ENFANT} . (\forall \text{ ENFANT} . \text{Femme})$$

$$\forall \text{ ENFANT} (\exists \text{ ENFANT} . \text{Femme})$$

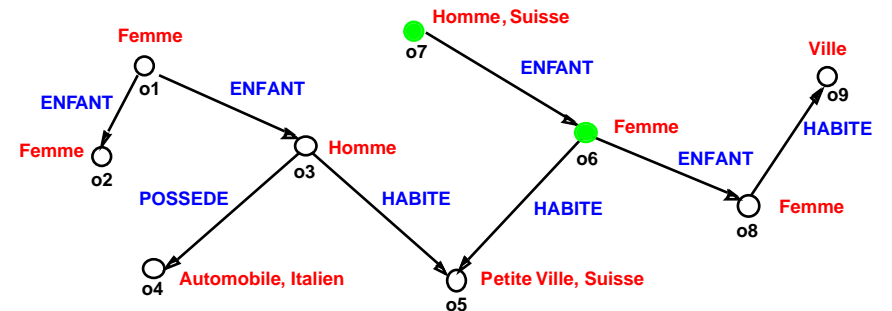
Exemple

$I(\forall \text{ ENFANT} . \text{Femme})$



Exemple

$\forall \text{ ENFANT} . \text{Femme} \sqcap \exists \text{ ENFANT} . \text{Femme}$



Le retour de la bergère

Bergère := FAUTEUIL $\sqcap \exists$ DIMENSION . Large $\sqcap \exists$ DIMENSION . Profond

$\sqcap \exists$ PARTIE . (DOSSIER $\sqcap \exists$ DIMENSION . Haut)

$\sqcap \leq 2$ PARTIE. BrasAJouePleine $\sqcap \geq 2$ PARTIE. BrasAJouePleine

BrasAJouePleine := BRAS $\sqcap \exists$ PARTIE . (JOUE $\sqcap \exists$ FORME . Plein)

Retour au Web sémantique

Langage de description de concepts : OWL

Logique descriptive, mais syntaxes différentes (XML et "abstraite")

Les noms sont des URI (uniques sur le Web)

OWL - classes

```
Class(pp:animal partial
  restriction(pp:eats someValuesFrom(owl:Thing)))
```

```
Class(pp:person partial pp:animal)
```

```
Class(pp:man complete
  intersectionOf(pp:person pp:male pp:adult))
```

```
Class(pp:animal+lover complete
  intersectionOf(pp:person
    restriction(pp:has_pet minCardinality(3))))
```

Restrictions (Existe et Pour tout)

```
Class(pp:vegetarian complete
  intersectionOf(pp:animal
    restriction(pp:eats
      allValuesFrom(complementOf(pp:animal))))
  restriction(pp:eats
    allValuesFrom(
      complementOf(restriction(pp:part_of
        someValuesFrom(pp:animal))))))
```

```
DisjointClasses(pp:young pp:adult)
```

(suite)

```
Class(pp:old+lady complete
  intersectionOf(pp:elderly pp:female pp:person))

Class(pp:old+lady partial
  intersectionOf(
    restriction(pp:has_pet allValuesFrom(pp:cat))
    restriction(pp:has_pet someValuesFrom(pp:animal))))
```