ASWS

Logiques de description & ontologies

Jean-Gabriel Ganascia
LIP6 – University Pierre et Marie Curie
4, place Jussieu, 75252 Paris, Cedex 05
Jean-Gabriel.Ganascia@lip6.fr

Références bibliographiques

- *Description Logics*, Franz Baader, Ian Horrocks, Ulrike Sattler, in « Handbook of Knowlefge Representation », editeurs Frank Van Harmelen, Vladimir Lifschitz, Bruce Porter, chapitre 3, pp. 135-179
- *The Description Logic Handbook: Theory, Implementation and Applications*, F. Baader, E. Franconi, B. Hollunder, B. Neble, H.-J. Profitlich, Cambridge University Press, 2003

Logiques terminologiques
Logiques de description…

Formalismes inspirés des représentations sémantiques

(réseaux sémantiques, frame, graphes conceptuels, …)

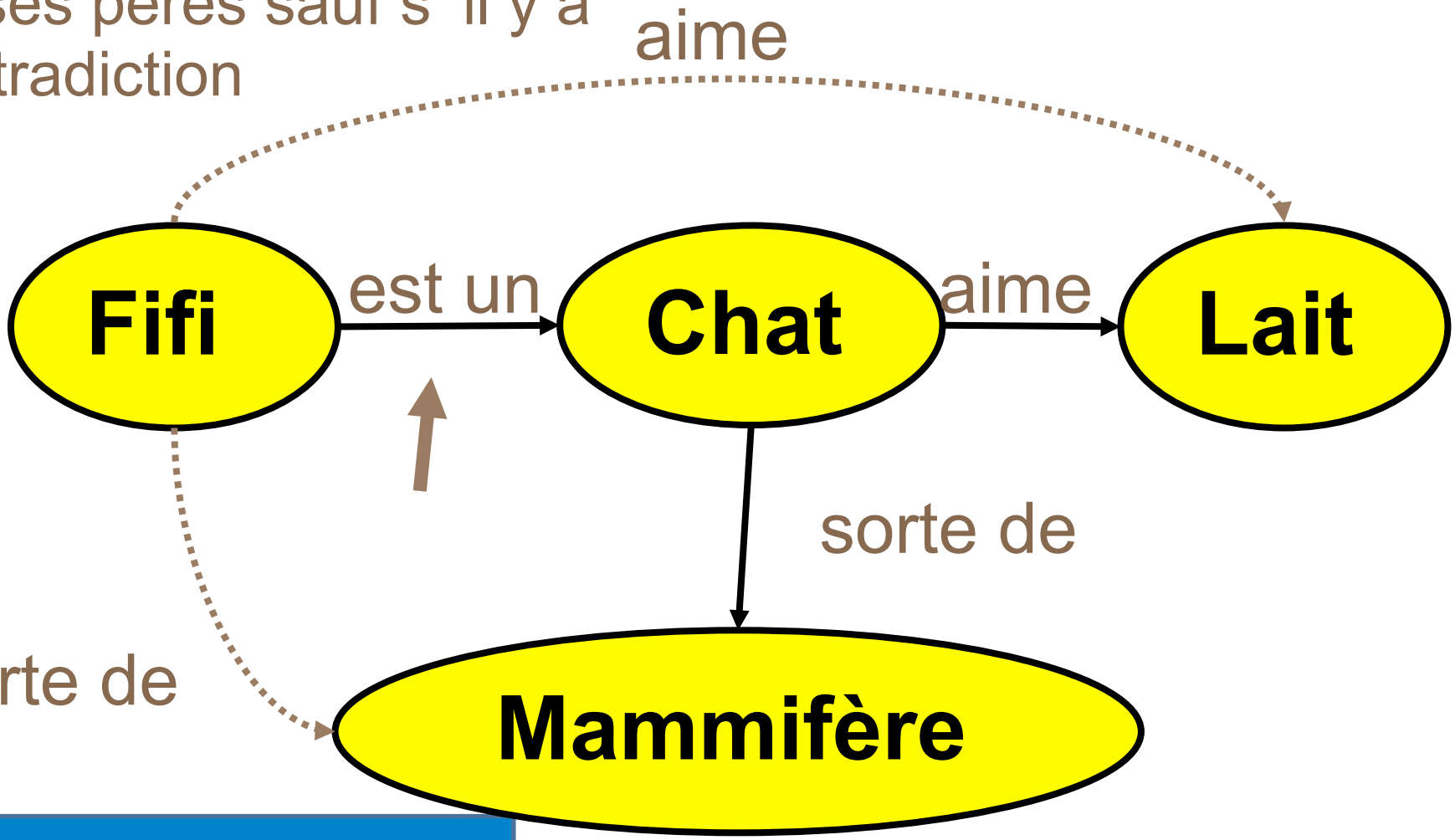
TKRS: Terminological Knowledge Representation Systems

Deux composants

- Classes générales d'individus – *T-Box*
  - Propriétés générales des classes
  - Relations entre les classes
- Instanciation de ces schémas – *A-Box*
  - Assertions relatives à des individus

Inférences: liens ‘est un’ ou ‘sorte de’

Un nœud hérite des propriétés de ses pères sauf s’ il y a contradiction



Représentation en logique des prédicats du premier ordre

- Prédicats d'arité 1 ou2 (unaires ou binaires)
  - **Concepts**: arité 1
  - **Relations**: arité 2
- Des règles peuvent être utilisées pour traduire l'héritage et quelques contraintes
- Mère EST-UN Parent

  $\forall X$ Mère(X) $\rightarrow$Parent(X)
- Tous les enfants de Parent sont des Personne

  $\forall X,Y$ Parent(X) $\wedge$ aEnfant(X, Y) $\rightarrow$ Personne(Y)
- On s'attend à ce que certaines inférences soient faites
  - **Héritage**: toutes les propriétés d'une superclasse devraient aussi être des propriétés de ses sous-classes.
  - Ainsi, tous les enfants de Mère doit être des Personne

    $\forall X,Y$ Mère(X) $\wedge$ aEnfant(X, Y) $\rightarrow$ Personne(Y)

Logiques terminologiques
Logiques de description

Formalismes inspirés des représentations sémantiques
(réseaux sémantiques, frame, graphes conceptuels, …)

Se distinguent des représentations sémantiques par une sémantique formelle

- Fragments décidables de la logique du premier ordre
- Procédures de décision pour la résolution de problèmes clefs:
  - Satisfiabilité – cohérence
  - Subsomption
- Systèmes implémentés – très efficaces

Logique de description

- Termes (noms de concepts)
  - Concepts: prédicats unaires
  - Rôles: relations binaires

TBox

Terminologie

- Contraintes (propriétés et opérateurs)
  - Négation, intersection, union,
  - Quantification universelle, existentielle
  - Contraintes sur les cardinalités des ensembles (nbre côtes $\succeq 8$)
  - …

- Assertions (énumérations)
  - Constantes
  - Instances des concepts et des rôles sur ces constantes

Assertions

ABox

Pas de variables…

Homme ⊓¬Femme ⊓ (∃ marié.Médecin) ⊓ (∀ aEnfant.(Médecin ⊔ Avocat))

- Symbole union (⊔) et intersection (⊓) de concepts
- Quantificateur existentiel: ∃ marié.Médecin
  - Ensemble des individus mariés à au moins un médecin
- Quantificateur universel: (∀ aEnfant.(Médecin ⊔ Avocat))
  - Ensemble des individus dont tous les enfants sont soit médecin, soit avocats
- Axiomes: ∃ aEnfant.Humain ⊑ Humain (subsomption)
  - Seuls les êtres humains peuvent avoir des enfants humains
- Axiomes: Père ≡ Homme ⊓ ∃ aEnfant.T

Remarque sur la notation

Usuellement

- Les concepts sont des chaînes de caractères qui commencent par une majuscule: Homme, Femme, Personne, Femelle, Mère, Parent…
- Les rôles sont des chaînes de caractères qui commencent par une minuscule: age, enfant, aEnfant, aFemmeFamille…

La « *famille* » des logiques de description

- Une logique de description donnée et définie par des concepts, des rôles et des opérateurs
- La logique $\mathcal{AL}$ (Attribute Language) contient uniquement la négation atomique et la quantification existentielle limitée
  - Les concepts sont construits en utilisant ⊓, ¬, ∃ et ∀
- La plus petite logique de description contenant la logique propositionnelle est $\mathcal{ALC}$ (équivalent à la logique multimodale $K_{(m)}$) – cela signifie $\mathcal{AL}$ et complémentation $\mathcal{C}$
  - Les concepts sont construits en utilisant ⊓, ⊔, ¬, ∃ et ∀
- $\mathcal{FL}^-$ correspond à $\mathcal{AL}$ sans la négation atomique
- $\mathcal{FL}_0$ correspond à $\mathcal{FL}^-$ sans la quantification existentielle limitée

$\mathcal{FL}_o$: la plus simple logique de description

Syntaxe

Alphabet

- concepts atomiques A, B, C, D…
- Rôles atomiques r, s, u, v, …
- Symboles {⊓, ∀, .}

Grammaire

concept ::= <concept atomique> |

<concept> ⊓ <concept> |

∀<role atomic>.<concept>

$\mathcal{FL}^-$ Syntaxe

Alphabet

- concepts atomiques A, B, C, D…
- Rôles atomiques r, s, u, v, …
- Symboles {⊓, ∃, ∀, .}

Grammaire

concept ::= <concept atomique> |
<concept> ⊓ <concept> |
∃<role atomique> |
∀<role atomic>.<concept>

$\mathcal{FL}$ Syntaxe

Alphabet

- concepts atomiques A, B, C, D…
- Rôles atomiques r, s, u, v, …
- Symboles {⊓, ∃, .}

Grammaire

concept ::= <concept atomique> |
<concept> ⊓ <concept> |
∃<role atomique>.<concept>

$\mathcal{FL}^-$: base de connaissance Σ=<TBox, ABox>

- TBox: axiomes terminologiques C ⊑ D, C ≡ D
  - Définitions

  Parent≡∃aENFANT ⊓ Personne
  - Subsomptions

  Homme ⊑ Personne *(⊑: subsomption)*
- ABox: assertions a:C, <a, b>:r
  - Assertions de concepts

  Jean:Parent

  Jean:Personne ⊓∃aENFANT

  Albert:personne
  - Assertions de rôles

  <Jean, Thomas>:aENFANT

$\mathcal{FL}^-$: Sémantique intuitive

concept ::= <concept atomique> |
<concept> ⊓ <concept> |
∃<role atomique> |
∀<role atomic>.<concept>

- **Concepts**: classes, ensemble d'individus
- **Rôles**: relations entres paires d'individus
- **Concepts atomiques**: concepts primitifs
- **∃r: existence d'un élément couvert par le rôle r**

exemple: ∃aENFANT : concept des choses qui ont des enfants

- **C ⊓ D: le concept conjoint C et D**

exemple: ∃aENFANT ⊓ personne ≡ parents

- **∀r.C : restriction d'un rôle par un concepts**

exemple: ∀aENFANT.Medecin : concept des choses dont tous les enfants sont médecins…

$\mathcal{AL}$ Syntaxe

Alphabet

- concepts atomiques A, B, C, D…
- Rôles atomiques r, s, u, v, …
- Symboles {⊓, ∃, ∀, ¬, .}

Grammaire

concept ::= <concept atomique> |
<concept> ⊓ <concept> |
∃<role atomique> |
¬<concept atomique> |
∀<role atomic>.<concept>

$\mathcal{ALC}$: la plus simple des logiques de description propositionnelles

Alphabet

- Ensemble de concepts atomiques A, B, C, D…
- Ensemble de rôles atomiques R, S, U, V, …
- Symboles {⊓,⊔,∃,∀,¬,⊤, ⊥ ,.}

Grammaire

- **⊥ et ⊤ sont des concepts**
- **Si C et D sont des concepts:**
  - **¬C est un concept** *(et pas uniquement un concept atomique)*
  - **C ⊔ D et C ⊓D sont des concepts**
- **Si r est un rôle et C un concept**
  - **∀r.C et ∃r.C sont des concepts**

$\mathcal{ALCN}$: ajout de contraintes sur les cardinalités à $\mathcal{ALC}$

Alphabet

- Ensemble de concepts atomiques A, B, C, D…
- Ensemble de rôles atomiques R, S, U, V, …
- Symboles {⊓,⊔,∃,∀,¬,⊤,⊥,.⇔≤\⇔≥\}

Grammaire

- **⊥ et ⊤ sont des concepts**
- **Si C et D sont des concepts:**
  - **¬C est un concept** *(et pas uniquement un concept atomique)*
  - **C ⊔ D et C ⊓D sont des concepts**
- **Si r est un rôle et C un concept**
  - **∀r.C et ∃r.C sont des concepts**
- **Si r est un rôle**
  - **$\leq^n\nabla$ et $\geq^n$r sont des concepts**

$\mathcal{FL}^-$: Sémantique formelle

Une interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ consiste en

- Un ensemble non vide $\Delta^{\mathcal{I}}$ (le *domaine*)
- Un fonction (la *fonction d'interprétation*) qui associe
  - À tout concept C, un sous-ensemble $C^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$
  - À tout rôle R, un sous-ensemble $R^{\mathcal{I}}$ de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
  - À tout individu i, un élément $i^{\mathcal{I}}$ de $\Delta^{\mathcal{I}}$

$\mathcal{FL}^-$: Fonction d'extension

Une fonction d'interprétation $.^{\mathcal{I}}$ est une *fonction d'extension* ssi

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(\forall R.C)^{\mathcal{I}} = \{x \in \Delta \mid \forall y.\ (x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
- $(\exists R)^{\mathcal{I}} = \{x \in \Delta \mid \exists y.\ (x, y) \in R^{\mathcal{I}}\}$

Remarque: $C^{\mathcal{I}}$ est l'ensemble des individus de l'extension de C dans l'interprétation $\mathcal{I}$

Ainsi, écrire $x \in C^{\mathcal{I}}$ est équivalent à C(x).

De même, $(x, y) \in R^{\mathcal{I}}$ est équivalent à R(x,y)

$\mathcal{ALC}$: la plus simple des logiques de description propositionnelles

Alphabet

- Ensemble de concepts atomiques A, B, C, D…
- Ensemble de rôles atomiques R, S, U, V, …
- Symboles {⊔,⊓,∃,∀,¬, ⊥ ,⊤,.}

Grammaire

- **⊥ et ⊤ sont des concepts**
- **Si C et D sont des concepts:**
  - **¬C est un concept**
  - **C ⊓ D et C ⊔D sont des concepts**
- **Si R est un rôle et C un concept**
  - **∀R.C et ∃R.C sont des concepts**

Sémantique $\mathcal{ALC}$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$
$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$

Extensions $\mathcal{ALC}$

$$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$
$$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.(x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$$

$\mathcal{ALCN}$: ajout de contraintes sur les cardinalités à $\mathcal{ALC}$

Alphabet

- Ensemble de concepts atomiques A, B, C, D…
- Ensemble de rôles atomiques R, S, U, V, …
- Symboles {⊓,⊔,∃,∀,¬,⊤,⊥,.⇔≤\⇔≥\}

Grammaire

- **⊥ et ⊤ sont des concepts**
- **Si C et D sont des concepts:**
  - **¬C est un concept** *(et pas uniquement un concept atomique)*
  - **C ⊔ D et C ⊓D sont des concepts**
- **Si r est un rôle et C un concept**
  - **∀r.C et ∃r.C sont des concepts**
- **Si r est un rôle**
  - **$\leq^n\nabla$ et $\geq^n$r sont des concepts**

Sémantique générale

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
\{x\}^{\mathcal{I}} &= \{x^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \mid \forall y.(x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\} \\
(\leqslant n R)^{\mathcal{I}} &= \{x \mid \#\{y \mid \langle x, y\rangle \in R^{\mathcal{I}}\} \leqslant n\} \\
(\geqslant n R)^{\mathcal{I}} &= \{x \mid \#\{y \mid \langle x, y\rangle \in R^{\mathcal{I}}\} \geqslant n\} \\
(R^{-})^{\mathcal{I}} &= \{(x, y) \mid (y, x) \in R^{\mathcal{I}}\}
\end{aligned}
$$

Tbox définitoire

- Une inclusion générale de concepts est de la forme $C \sqsubseteq D$ où C et D sont des concepts.
- Une interprétation $\mathcal{I}$ est un modèle de si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.
- $\mathcal{I}$ est un modèle de la Tbox $\mathcal{T}$ si c'est un modèle de toutes les inclusions de concepts de $\mathcal{T}$.
- $C \equiv D$ est une abréviation pour $C \sqsubseteq D$ et $D \sqsubseteq C$
- Un axiome de la forme $A \equiv C$ où A est un nom de concept est appelé une définition de A
- Une Tbox $\mathcal{T}$ est dite définitoire si elle ne contient que des définitions avec les restrictions additionnelles suivantes:
  - $\mathcal{T}$ contient au plus une définition pour chaque nom de concept
  - $\mathcal{T}$ est acyclique

Base de connaissance
Knowledge Base

- Une Base de connaissance $\mathcal{K}$ est une paire $\Re\mathcal{T},\mathcal{A}\top$ où
  - $\mathcal{T}$ est un ensemble d'axiomes "terminologiques" (la Tbox)
  - $\mathcal{A}$ est un ensemble d'axiomes "assertionels" (la Abox)
- Les axiomes sont de la forme

  $\mathrm{C} \sqsubseteq \mathrm{D}, \mathrm{C} \equiv \mathrm{D}, \mathrm{r} \sqsubseteq \mathrm{s}, \mathrm{r} \equiv \mathrm{s}$ et $\mathrm{r}^{+} \sqsubseteq \mathrm{r}$

  où C, D sont des concepts, r, s des rôles et r^{+} un ensemble de rôles transitifs

  Remarque: le graphe des définitions doit être acyclique
- Les axiomes de la Abox sont de la forme:

  $\mathrm{x}:\mathrm{D}$

  $<\mathrm{x},\mathrm{y}>:\mathrm{r}$

  où x,y sont des noms d'individus, D un concept et r un rôle

Sémantique formelle: modèle

Si $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ est une interprétation

- a:C est satisfait par $\mathcal{I}$ si $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- <a,b>:r est satisfait par $\mathcal{I}$ si $<a^{\mathcal{I}}, b^{\mathcal{I}}> \in r^{\mathcal{I}}$
- Une interprétation $\mathcal{I}$ est dite être un modèle de la ABox A si toutes les assertions de $\mathcal{A}$ sont satisfaites par $\mathcal{I}$.
- Une interprétation $\mathcal{I}$ est dite être un modèle de la TBox T si tous les axiomes de $\mathcal{T}$ sont satisfaites par $\mathcal{I}$.

Une interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ est dite être un **modèle** de la base de connaissance Σ si tous les axiomes de Σ sont satisfaites par $\mathcal{I}$.

Une base de connaissance Σ est satisfiable si elle admet un modèle

Raisonnement: 4 propriétés

- Satisfiabilité: un concept C est satisfiable si et seulement si il existe une interprétation $\mathcal{I}$ telle que $C^{\mathcal{I}} \neq \emptyset$
- Subsomption: un concept C est subsumé par D si et seulement si $C^{\mathcal{I}} \subset D^{\mathcal{I}}$ pour toute interprétation $\mathcal{I}$

Remarque: la subsomption est décidable en temps polynomial pour $\mathcal{FL}^-$

- Équivalence: un concept C est équivalent à un concept D si et seulement si $C^{\mathcal{I}} = D^{\mathcal{I}}$ pour toute interprétation $\mathcal{I}$
- Incompatibilité: deux concepts C et D sont incompatibles si et seulement si $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ pour toute interprétation $\mathcal{I}$

Requêtes & classification

- x est une instance du concept C si et seulement si, pour toute interprétation $\mathcal{I}$, $x^{\mathcal{I}} \in C^{\mathcal{I}}$
- <x, y> est une instance du rôle R si et seulement si, pour toute interprétation $\mathcal{I}$, $\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$

Classification

Étant donné un concept C et une TBox $\mathcal{T}$, déterminer si D subsume C ou si C subsume D pour tous les concepts D de $\mathcal{T}$

Remarque: La classification est utile pour insérer de nouveaux concepts dans une taxinomie

Quatre problèmes

- Satisfiabilité d'une base de connaissance: est-ce que la ABox et la Tbox sont cohérentes l'une avec l'autre?
- Satisfiabilité d'un concept: étant donné une base de connaissance K et un concept C existe-t-il au moins un modèle de K pour lequel l'extension de C n'est pas vide?
- Subsomption: étant donné une base de connaissance K et deux concepts quelconque C et D de K, est-ce que D est plus général que C (ou est-ce que C est subsumé par D)?
- Vérification d'une instance: étant donné une base de connaissance K et une instance a d'un concept C, est-ce que a est une instance de C dans tout modèle de K?

Réductions à la Subsumption

- C est insatisfiable ssi $C \sqsubseteq \bot$
- C & D sont équivalents ssi $C \sqsubseteq D$ & $D \sqsubseteq C$
- C & D sont incompatibles ssi $C \sqcap D \sqsubseteq \bot$

Réductions à l'insatisfiabilité
Possible si la négation est définie…

- C ⊑ D ssi C⊓¬D insatisfiable
- C & D sont équivalents ssi C⊓¬D & ¬C⊓D sont insatisfiables
- C & D sont incompatibles ssi C⊓D est insatisfiable

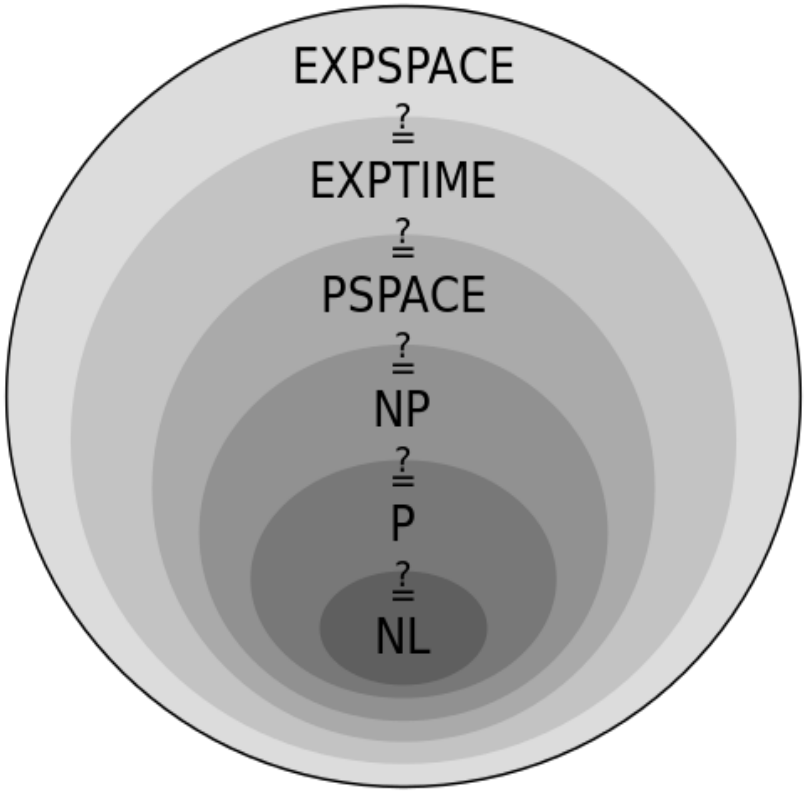
Compromis

- Plus le pouvoir d'expression d'un langage est grand, plus les procédures de démonstration sont complexes
- En général les langages pour lesquels des procédures complètes existent sont insuffisants pour exprimer ce que l'on souhaite

Description logic	Subsumption computation complexity
$\mathcal{AL}$	PTIME
$\mathcal{ALC}$	PSPACE
$\mathcal{SHIF}$	EXPTIME
$\mathcal{SHOIN}$	NEXPTIME

- Questions:
  - Un langage étant donné, existe-t-il une procédure de démonstration complète?
  - Si ce n'est pas le cas, peut-on comparer les procédures incomplètes?
  - Une procédure complète étant donnée, existe-t-il des algorithmes efficaces en temps et en place?

Rappels sur la complexité



- P: temps polynomial, machine déterministe
- NP: temps polynomial, machine non déterministe
- EXPTIME: temps exponentiel, machine déterministe
- NEXPTIME: temps exponentiel, machine non déterministe
- L: espace logarithmique, machine déterministe
- NL: espace logarithmique, machine non déterministe
- PSPACE: espace polynomial, machine déterministe (= NPSPACE)
- EXSPACE: espace exponentiel

Complexity of reasoning in Description Logics

Note: the information here is (always) incomplete and updated often

Base description logic: $\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplements

$\mathcal{ALC} ::= \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$

Concept constructors:

- ☐ $\mathcal{F}$ – functionality[2]: $(\leq 1\ R)$
- ☐ $\mathcal{N}$ – (unqualified) number restrictions: $(\geq n\ R)$, $(\leq n\ R)$
- ☐ $\mathcal{Q}$ – qualified number restrictions: $(\geq n\ R.C)$, $(\leq n\ R.C)$
- ☐ $\mathcal{O}$ – nominals: $\{a\}$ or $\{a_1,\ldots,a_n\}$ ("one-of" constructor)

- ☐ μ – least fixpoint operator: $\mu X.C$
- ☐ $R \sqsubseteq S$ – role-value-maps
- ☐ $f = g$ – agreement of functional role chains ("same-as")

Role constructors: trans reg

- ☐ $\mathcal{I}$ – role inverses: R^-

- ☐ $\cap$ – role intersection[3]: $R \cap S$
- ☐ $\cup$ – role union: $R \cup S$
- ☐ $\neg$ – role complement: full
- ☐ o – role chain (composition): $R \circ S$
- ☐ * – reflexive-transitive closure[4]: R^*
- ☐ id – concept identity: $id(C)$

Forbid complex roles[5] in number restrictions[6]

TBox options:

- ◉ Empty TBox
- ○ Acyclic TBox ($A \equiv C$, A is a concept name; no cycles)
- ○ General TBox ($C \sqsubseteq D$ for arbitrary concepts C and D)

Role axioms (RBox): OWL-Lite OWL-DL OWL 1.1

- ☐ $\mathcal{S}$ – Role transitivity: Trans(R)
- ☐ $\mathcal{H}$ – Role hierarchy: $R \sqsubseteq S$
- ☐ $\mathcal{R}$ – Complex role inclusions: $R \circ S \sqsubseteq R$, $R \circ S \sqsubseteq S$
- ☐ $\mathcal{s}$ – some additional features

Reset You have selected the Description Logic: $\mathcal{ALC}$

Complexity of reasoning problems[7]

Reasoning problem	Complexity[8]	Comments and references
Concept satisfiability	PSpace-complete	• Hardness for $\mathcal{ALC}$: see [73]. • Upper bound for $\mathcal{ALCQ}$: see [77, Theorem 4.6].
ABox consistency	PSpace-complete	• Hardness follows from that for concept satisfiability. • Upper bound for $\mathcal{ALCQ}$: see [7, Appendix A].

Procédures de raisonnement

- Il existe des algorithmes complets et efficaces pour décider de la satisfiabilité, de la subsomption et donc des autres propriétés de certaines logiques de descriptions

- Les techniques de raisonnement sont fondées
  sur la méthode des tableaux (pour la satisfiabilité)
  sur la subsomption structurelle (pour la subsomption)
  sur les automates
  sur la résolution…

Remarque: il existe des liens entre la méthode des tableaux et d'autres techniques (résolution, ASP, etc.)

- La complétude est importante pour l'utilisation des logiques de descriptions dans les applications réelles

« ontologies »

But: donner sens aux informations sur le web

Permettre à des robots de se débrouiller et de « comprendre » suffisamment pour nous aider à retrouver l'information

Solution: utiliser des annotations avec des logiques de description

Supposons que l'on ait cette page…

WWW2002

The eleventh international world wide web conference

Sheraton waikiki hotel

Honolulu, hawaii, USA

7-11 may 2002

1 location 5 days learn interact

Registered participants coming from

australia, canada, chile denmark, france, germany, ghana, hong kong, india, ireland, italy, japan, malta, new zealand, the netherlands, norway, singapore, switzerland, the united kingdom, the united states, vietnam, zaire

Register now

On the 7th May Honolulu will provide the backdrop of the eleventh international world wide web conference. This prestigious event …

Speakers confirmed

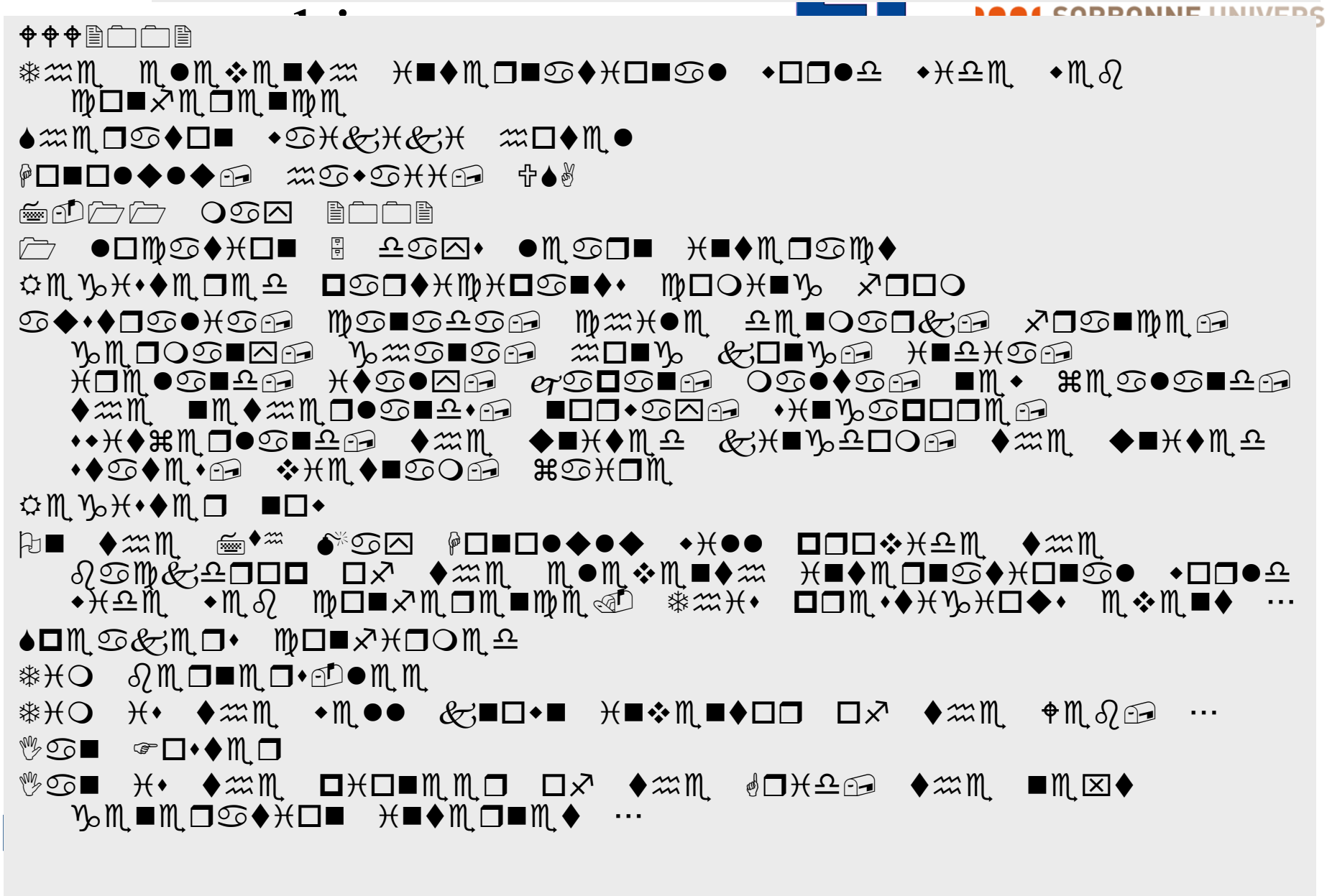
Tim berners-lee

Tim is the well known inventor of the Web, …

Ian Foster

Ian is the pioneer of the Grid, the next generation internet …

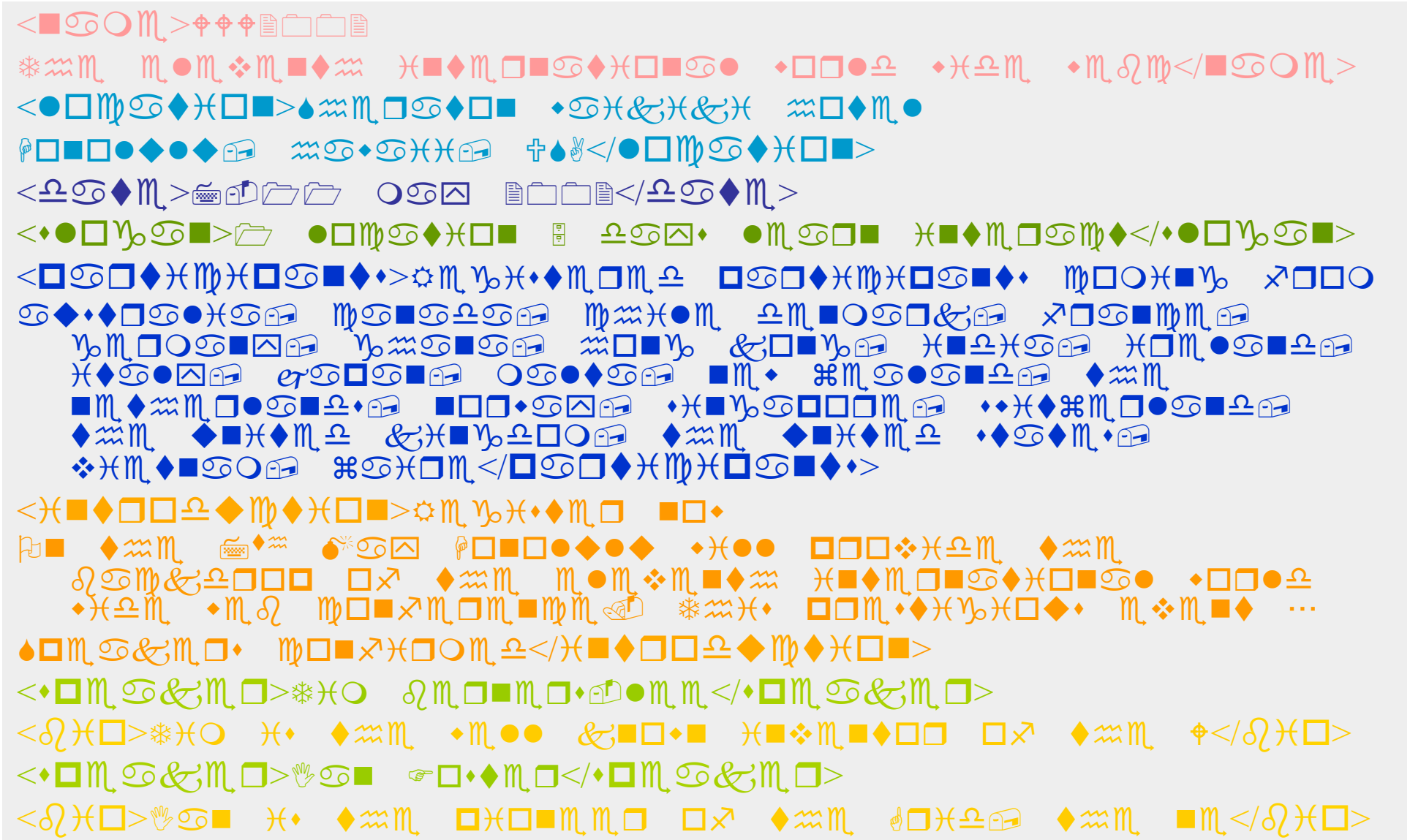
Quelle information voit la



Comment enrichir cette description ?
Mettre des balises <XML>…

<name>⌖⌖⌖🗎🗀🗀🗎
❄♒♏ ♏●♏❖♏■♦♒ ♓■♦♏❒■♋♦♓□■♋● ⬩□❒●♎ ⬩♓♎♏ ⬩♏♌♍□■</name>
<location>💧♒♏❒♋♦□■ ⬩♋♓&♓&♓ ♒□♦♏●
☝□■□●◆●◆🖃 ♒♋⬩♋♓♓🖃 ☥💧✌</location>
<date>📻📠🗁🗁 ❍♋🖂 🗎🗀🗀🗎</date>
<slogan>🗁 ●□♍♋♦♓□■ 🗄 ♎♋🖂⬩ ●♏♋❒■ ♓■♦♏❒♋♍♦</slogan>
<participants>☼♏♑♓⬩♦♏❒♏♎ ◘♋❒♦♓♍♓◘♋■♦⬩ ♍□❍♓■♑ ➚❒□❍
♋◆⬩♦❒♋●♓♋🖃 ♍♋■♋♎♋🖃 ♍♒♓●♏ ♎♏■❍♋❒&🖃 ➚❒♋■♍♏🖃
♑♏□❍♋■🖂🖃 ♑♒♋■♋🖃 ♒□■♑ &□■♑🖃 ♓■♎♓♋🖃 ♓❒♏●♋■♎🖃
♓♦♋●🖂🖃 ♂♋◘♋■🖃 ❍♋●♦♋🖃 ■♏⬩ ⌘♏♋●♋■♎🖃 ♦♒♏
■♏♦♒♏❒●♋■♎⬩🖃 ■□❒⬩♋🖂🖃 ⬩♓■♑♋◘□❒♏🖃 ⬩⬩♓♦⌘♏❒●♋■♎🖃
♦♒♏ ◆■♓♦♏♎ &♓■♑♎□❍🖃 ♦♒♏ ◆■♓♦♏♎ ⬩♦♋♦♏⬩🖃
❖♓♏♦■♋❍🖃 ⌘♋♓❒♏</participants>
<introduction>☼♏♑♓⬩♦♏❒ ■□⬩
⚐■ ♦♒♏ 📻⬩♒ 💣♋🖂 ☝□■□●◆●◆ ⬩♓●● ◘❒□❖♓♎♏ ♦♒♏
♌♋♍&♎❒□◘ □➚ ♦♒♏ ♏●♏❖♏■♦♒ ♓■♦♏❒■♋♦♓□■♋● ⬩□❒●♎
⬩♓♎♏ ⬩♏♌ ♍□■➚♏❒♏■♍♏🖃 ❄♒♓⬩ ◘❒♏⬩♦♓♑♓□◆⬩ ♏❖♏■♦ ⋯
💧□♏♋&♏❒⬩ ♍□■➚♓❒❍♏♎</introduction>
<speaker>❄♓❍ ♌♏❒■♏❒⬩🖃●♏♏</speaker>
<bio>❄♓❍ ♓⬩ ♦♒♏ ⬩♏●● &■□⬩■ ♓■❖♏■♦□❒ □➚ ♦♒♏ ⌖♏♌🖃</bio>…

La machine voit…



… mais que se passe t-il avec

Particularité du web sémantique

Annotation donnant du **sens** aux contenus

- **Métadonnées** sémantiques
- Inférences automatiques

Pas d'analyse directe des contenus

Autres pistes

Extraction du sens à partir des textes

Langage contrôlé

…

Brève histoire du web sémantique

Tim Berners Lee

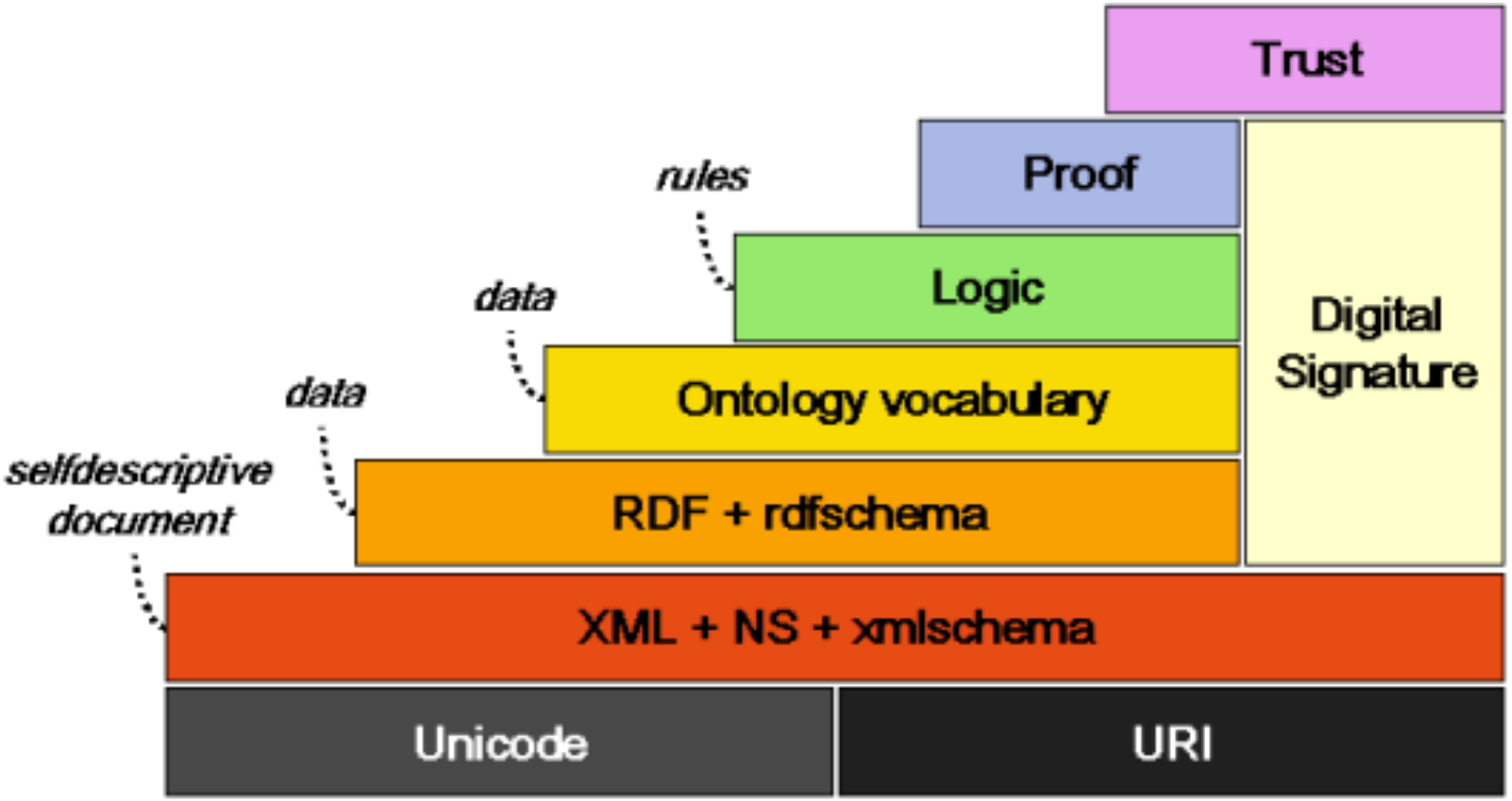
1990 Invention du web

1994 Métadonnées

1999 Schémas RDF
(Ressource Description Framework)

2004 Annotations sémantiques
(RDFS) et Ontologies (OWL)

Le « cake » sémantique



OWL: Ontologie Web Language

- Transformation d'une ontologie dans une formalisation XML

`intersectionOf`	$C_1 \sqcap \ldots \sqcap C_n$
`unionOf`	$C_1 \sqcup \ldots \sqcup C_n$
`complementOf`	$\neg C$
`allValuesOf`	$\forall r.C$
`someValuesOf`	$\exists r.C$
`minCardinality`	$(\geq_n r)$
`inverseOf`	r^{-1}

OWL: Ontologie Web Language

- Human ⊓ Male

```
<owl:Class>
   <owl:intersectionOf
rdf:parseType="collection">
      <owl:Class rdf:about="#Human"/>
      <owl:Class rdf:about="#Male"/>
   </owl:intersectionOf>
</owl:Class>
```

OWL: Ontologie Web Language

- ≥ 2 HasChild.Thing

```
<owl:Restriction>
   <owl:onProperty rdf:resource="#hasChild"/>
   <owl:minCardinality rdf:about="#Human"/>
      rdf:datatype="&xsd:NonNegativeInteger">2
   </owl:minCardinality>
</owl:Restriction>
```

L’ontologie Pizza en logique de description

$$\textsc{Pizza} \sqsubseteq \exists\, \mathsf{hasBase}.\textsc{PizzaBase}$$

$$\textsc{Pizza} \sqcap \textsc{PizzaBase} \equiv \bot$$

$$\textsc{NonVegetarianPizza} \equiv \textsc{Pizza} \sqcap \neg \textsc{VegetarianPizza}$$

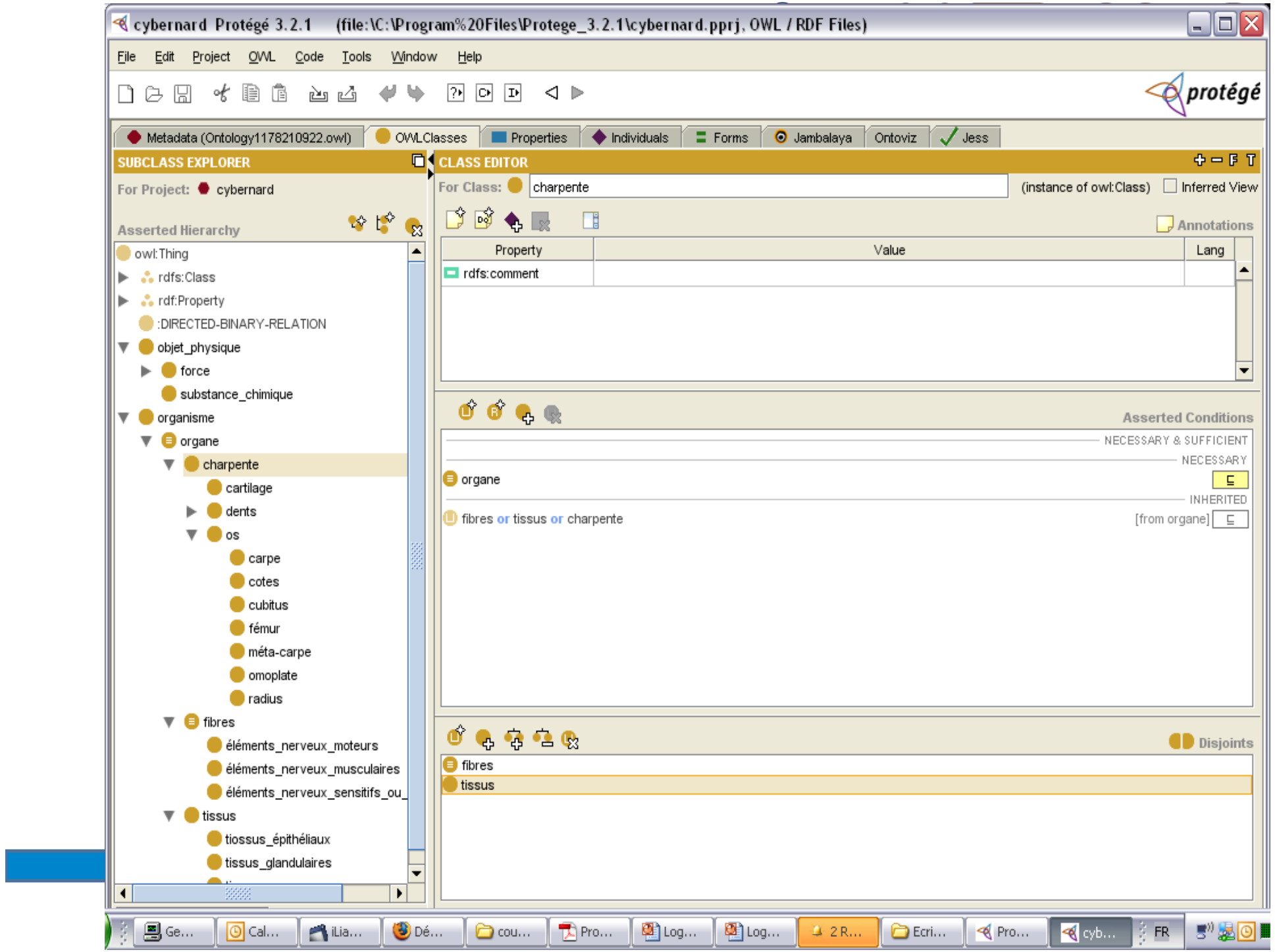
$$Tr(\mathsf{isIngredientOf})$$

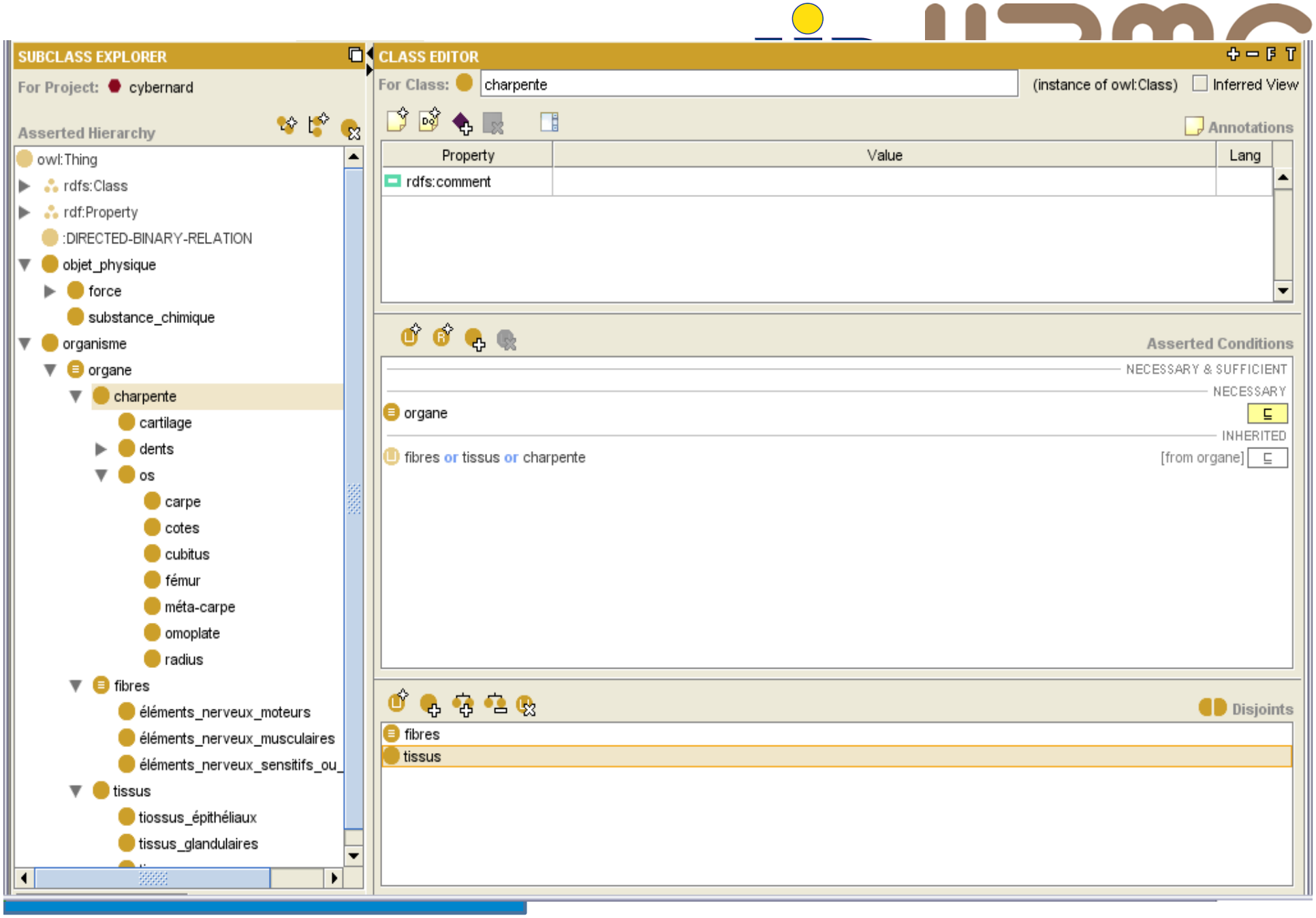
$$\mathsf{isIngredientOf} \equiv \mathsf{hasIngredient}^{-}$$

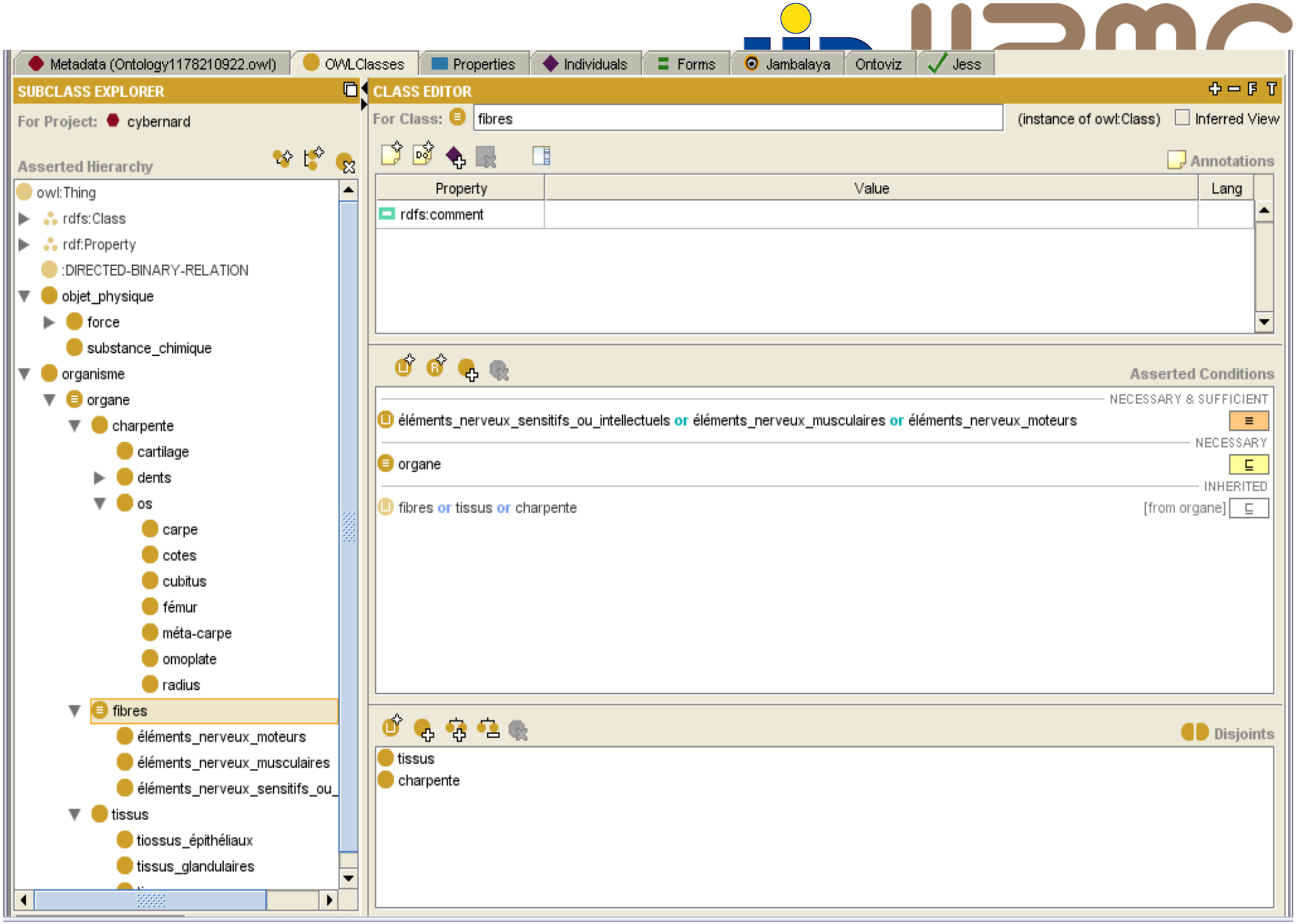
L’ontologie Pizza en OWL

```
<owl:Class rdf:about="#CaperTopping">
    <rdfs:label xml:lang="pt »>CoberturaDeCaper</rdfs:label>
    <rdfs:subClassOf>
        <owl:Class rdf:about="#VegetableTopping"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasSpiciness"/>
            <owl:someValuesFrom rdf:resource="#Mild"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

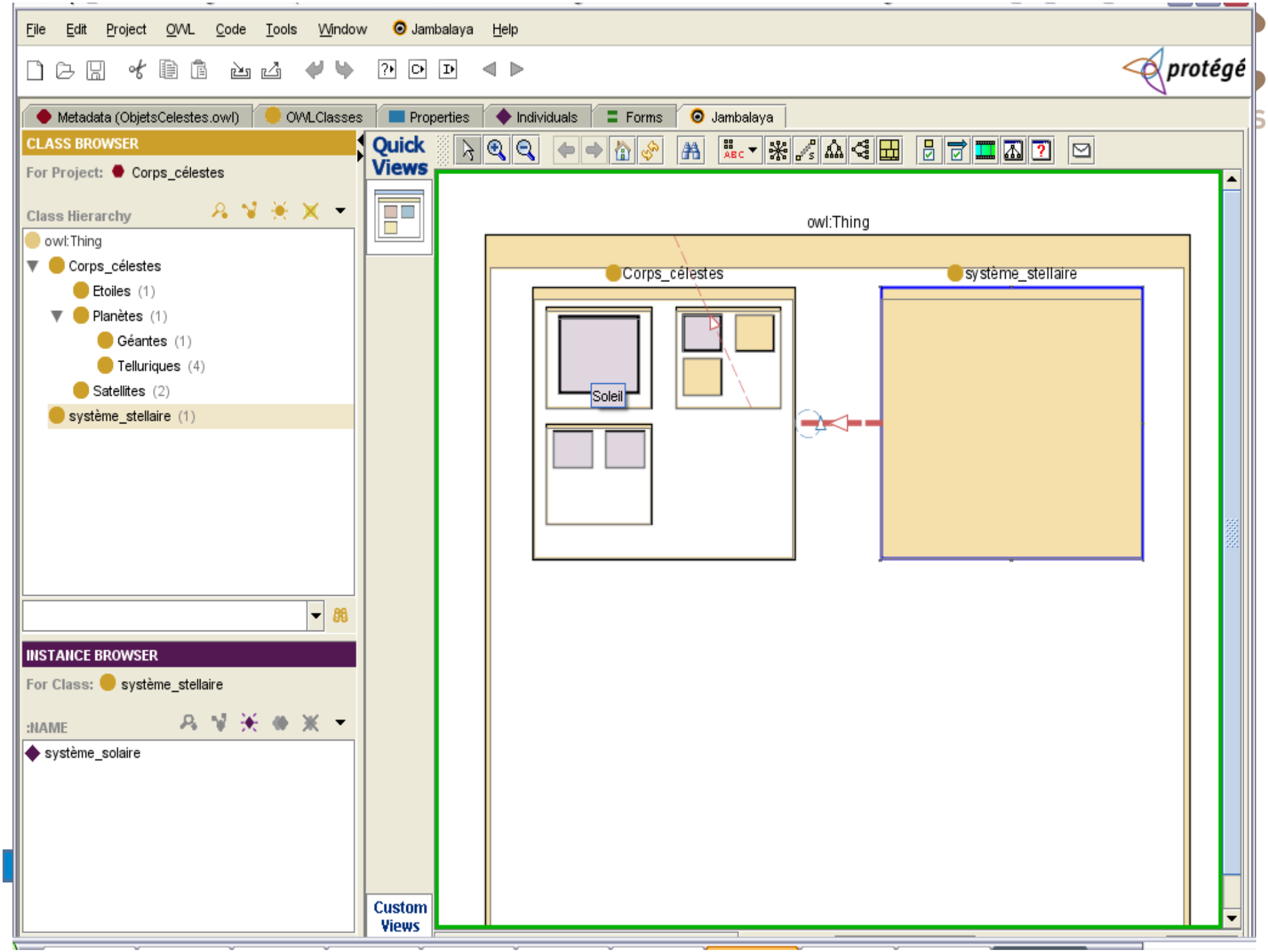
Protégé: un environnement de construction d'ontologies


cybernard Protégé 3.2.1 (file:\C:\Program%20Files\Protege_3.2.1\cybernard.pprj, OWL / RDF Files)
File Edit Project OWL Code Tools Window Help
protégé
Metadata (Ontology1178210922.owl) OWLClasses Properties Individuals Forms Jambalaya Ontoviz Jess
SUBCLASS EXPLORER
For Project: cybernard
Asserted Hierarchy
owl:Thing
rdfs:Class
rdf:Property
:DIRECTED-BINARY-RELATION
objet_physique
force
substance_chimique
organisme
organe
charpente
cartilage
dents
os
carpe
cotes
cubitus
fémur
méta-carpe
omoplate
radius
fibres
éléments_nerveux_moteurs
éléments_nerveux_musculaires
éléments_nerveux_sensitifs_ou_
tissus
tiossus_épithéliaux
tissus_glandulaires
CLASS EDITOR
For Class: charpente
(instance of owl:Class)
Inferred View
Annotations
Property
Value
Lang
rdfs:comment
Asserted Conditions
NECESSARY & SUFFICIENT
NECESSARY
organe
INHERITED
fibres or tissus or charpente
[from organe]
Disjoints
fibres
tissus


SUBCLASS EXPLORER
For Project: cybernard
Asserted Hierarchy
owl:Thing
rdfs:Class
rdf:Property
:DIRECTED-BINARY-RELATION
objet_physique
force
substance_chimique
organisme
organe
charpente
cartilage
dents
os
carpe
cotes
cubitus
fémur
méta-carpe
omoplate
radius
fibres
éléments_nerveux_moteurs
éléments_nerveux_musculaires
éléments_nerveux_sensitifs_ou_
tissus
tiossus_épithéliaux
tissus_glandulaires
CLASS EDITOR
For Class: charpente
(instance of owl:Class)
Inferred View
Annotations
Property
Value
Lang
rdfs:comment
Asserted Conditions
NECESSARY & SUFFICIENT
NECESSARY
organe
INHERITED
fibres or tissus or charpente
[from organe]
Disjoints
fibres
tissus


Metadata (Ontology1178210922.owl)
OWLClasses
Properties
Individuals
Forms
Jambalaya
Ontoviz
Jess
SUBCLASS EXPLORER
For Project: cybernard
Asserted Hierarchy
owl:Thing
rdfs:Class
rdf:Property
:DIRECTED-BINARY-RELATION
objet_physique
force
substance_chimique
organisme
organe
charpente
cartilage
dents
os
carpe
cotes
cubitus
fémur
méta-carpe
omoplate
radius
fibres
éléments_nerveux_moteurs
éléments_nerveux_musculaires
éléments_nerveux_sensitifs_ou_
tissus
tiossus_épithéliaux
tissus_glandulaires
CLASS EDITOR
For Class: fibres
(instance of owl:Class)
Inferred View
Annotations
Property
Value
Lang
rdfs:comment
Asserted Conditions
NECESSARY & SUFFICIENT
éléments_nerveux_sensitifs_ou_intellectuels or éléments_nerveux_musculaires or éléments_nerveux_moteurs
NECESSARY
organe
INHERITED
fibres or tissus or charpente
[from organe]
Disjoints
tissus
charpente

Corps céleste
Sorte de
Sorte de
Sorte de
étoile
planète
satellite
Sorte de
Sorte de
géante
Est un
Est un
Est un
Système solaire
Contient
Soleil
tellurique
Lune
Charon
Est un
Est un
Gravite
Est un
Gravite
Gravite
Est un
Est un
Contient
Gravite
Gravite
Gravite
Contient
Gravite
Mercure
Contient
Contient
Contient
Venus
Mars
Pluton
Nombre satellite
Terre
1
Nombre satellite


File
Edit
Project
OWL
Code
Tools
Window
Jambalaya
Help
protégé
Metadata (ObjetsCelestes.owl)
OWLClasses
Properties
Individuals
Forms
Jambalaya
CLASS BROWSER
For Project: Corps_célestes
Class Hierarchy
owl:Thing
Corps_célestes
Etoiles (1)
Planètes (1)
Géantes (1)
Telluriques (4)
Satellites (2)
système_stellaire (1)
INSTANCE BROWSER
For Class: système_stellaire
:NAME
système_solaire
Quick Views
Custom Views
owl:Thing
Corps_célestes
système_stellaire
Soleil

Le « cake » sémantique

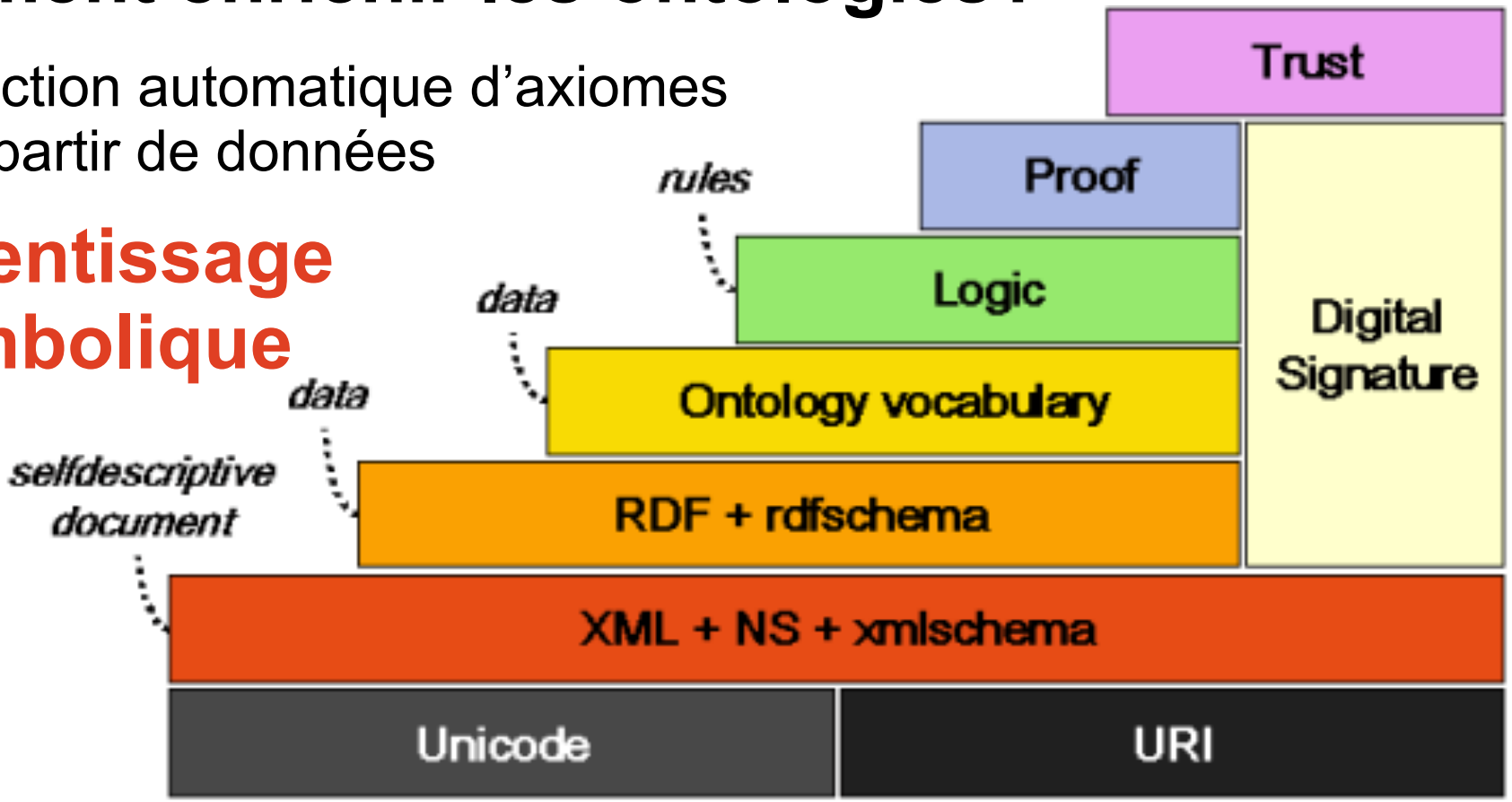
Comment construire ces annotations?

Extraction automatique à partir de textes

Comment enrichir les ontologies?

Induction automatique d'axiomes
à partir de données

Apprentissage symbolique



Exercise

1. Nobody steal.
2. No human does steal.
3. All human are honest
4. At least one human is a king. *(with a king as a role)*
5. At more one human is a king
6. Pierre is a king
7. One cannot be both a man and a woman

king and steal are roles

1. $\neg \exists steal.T$
2. $\exists steal.T \sqcap Humain \sqsubseteq \bot$
3. $Human \sqsubseteq Honest$
4. $Human \sqcap \exists king.T \not\equiv \bot$
   $\exists king^{-1}.Human$
5. $\exists^{\leq 1} king^{-1}.Human$
6. $Pierre:\exists king.T$
   $Pierre:King$
7. $Man \sqcap Woman \sqsubseteq \bot$

Exercise *(following)*

1. A minor is a person less than 18 years old
2. Family whose children are all minor.
3. Man who has more than three daughters
4. A woman whose children are less than 3 years old
5. A woman who is the child of a man who is older than 90

hasAge is a role,
≤ 18 is a concept

1. $\text{Minor} \equiv \text{Person} \sqcap \exists \text{hasAge}.\leq 18$
2. $\text{Family} \sqcap \forall \text{hasChildren}.\text{Minor}$
3. $\text{Man} \sqcap \exists^{\geq 3} \text{hasChildren}.\text{Woman}$
4. $\text{Woman} \sqcap \forall \text{hasChildren}.\exists \text{hasAge}.\leq 3$
5. $\text{Woman} \sqcap \text{hasChildren}^{-1}.(\text{Man} \sqcap \exists \text{hasAge}.\geq 90)$