School of CIT
Social Computing Research Group

Technical
University
of Munich

Natural Language Processing
IN2361

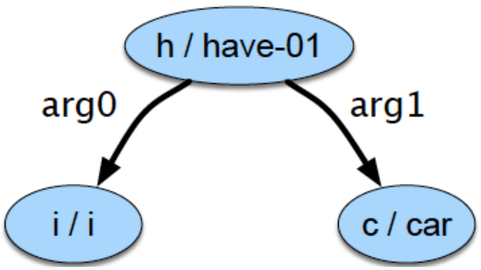
Prof. Dr. Georg Groh

Chapter 15
Logical Representations of Sentence Meaning

- content is based on [1]
- certain elements (e.g. equations or tables) were taken over or taken over in a modified form from [1]
- citations of [1] or from [1] are omitted for legibility
- errors are fully in the responsibility of Georg Groh
- BIG thanks to Dan and James for a great book!

Example

$$\exists e,y\ Having(e) \wedge Haver(e, Speaker) \wedge HadThing(e,y) \wedge Car(y)$$




```
(h / have-01
     arg0: (i / i)
     arg1: (c / car))
```

```
Having:
     Haver:    Speaker
     HadThing:  Car
```

Figure 15.1 A list of symbols, two directed graphs, and a record structure: a sampler of meaning representations for *I have a car.*

Desiderata for Semantic Representations of NLP Entities

- **Verifiablility**: compare state of affairs described by representation to the state of affairs as modeled in a knowledge base

"Does Maharani serve vegetarian food?" →

<representation> ←→

$\exists v\colon Serves(r, v) \wedge Restaurant(r) \wedge VegFood(v) \wedge Name(r, \text{“}Maharani\text{“})$

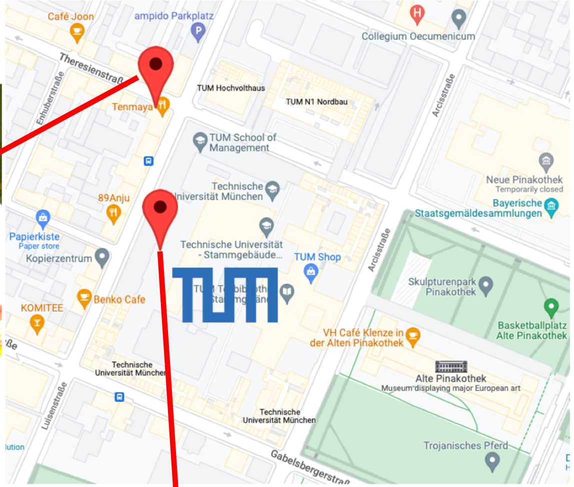
- **Unambiguity:**

"I wanna eat someplace that is close to TUM" :

ambiguous sentence: more than one representation

→ Maybe you are Godzilla :D

<repr1>≘



"I wanna eat Italian food" :

vague sentence: not more than one represenation

pizza?

pasta?

<repr2>≘

Desiderata for Semantic Representations of NLP Entities

- **Canoncial form:** distinct inputs that mean the same thing should have the same meaning representation.

Does Maharani have vegetarian dishes?
Do they have vegetarian food at Maharani?
Are vegetarian dishes served at Maharani?
Does Maharani serve vegetarian fare?

$\xrightarrow{!}$ same <representation>

Ücret.

- Ability to support **inference and variables:**

Can vegetarians eat at Maharani? requires inference: vegetarians eat vegetarian food

I'd like to find a restaurant where I can get vegetarian food requires variables: Search for x $Serves(x, VegetarianFood)$

- Sufficient **Expressiveness**

Model-Theoretic Semantics

- Model: a formal construct that stands for the particular state of affairs in the world
- Expressions in a meaning representation language can be mapped to elements of the model (“interpretation”)
- meaning representation language vocabulary: logical and non-logical
- Each non-logical vocabulary element is uniquely mapped to corresponding denotation in the model (“interpretation”)
- Domain of the model: set of represented objects (concepts, individuals, etc.)
- Properties of objects denote sets of elements of the domain;
  example: red ←→ set of red individuals
- Relations denote sets of tuples of elements of the domain

Extensional approach

Model-Theoretic Semantics

Domain	$\mathcal{D} = \{a,b,c,d,e,f,g,h,i,j\}$	FOL: terms
Matthew, Franco, Katie and Caroline	a,b,c,d	
Frasca, Med, Rio	e,f,g	
Italian, Mexican, Eclectic	h,i,j	
Properties		FOL: unary predicates
Noisy	$Noisy = \{e,f,g\}$	
Frasca, Med, and Rio are noisy		
Relations		FOL: binary predicates
Likes	$Likes = \{\langle a,f\rangle, \langle c,f\rangle, \langle c,g\rangle, \langle b,e\rangle, \langle d,f\rangle, \langle d,g\rangle\}$	
Matthew likes the Med		
Katie likes the Med and Rio		
Franco likes Frasca		
Caroline likes the Med and Rio		
Serves	$Serves = \{\langle f,j\rangle, \langle g,i\rangle, \langle e,h\rangle\}$	
Med serves eclectic		
Rio serves Mexican		
Frasca serves Italian		

Figure 15.2 A model of the restaurant world.

First Order Logic

$$
\begin{array}{rcl}
\textit{Formula} & \rightarrow & \textit{AtomicFormula} \\
& | & \textit{Formula Connective Formula} \\
& | & \textit{Quantifier Variable}, \ldots \textit{ Formula} \\
& | & \neg \textit{ Formula} \\
& | & (\textit{Formula}) \\
\textit{AtomicFormula} & \rightarrow & \textit{Predicate}(\textit{Term}, \ldots) \\
\textit{Term} & \rightarrow & \textit{Function}(\textit{Term}, \ldots) \\
& | & \textit{Constant} \\
& | & \textit{Variable} \\
\textit{Connective} & \rightarrow & \wedge \mid \vee \mid \Longrightarrow \\
\textit{Quantifier} & \rightarrow & \forall \mid \exists \\
\textit{Constant} & \rightarrow & A \mid \textit{VegetarianFood} \mid \textit{Maharani} \cdots \\
\textit{Variable} & \rightarrow & x \mid y \mid \cdots \\
\textit{Predicate} & \rightarrow & \textit{Serves} \mid \textit{Near} \mid \cdots \\
\textit{Function} & \rightarrow & \textit{LocationOf} \mid \textit{CuisineOf} \mid \cdots
\end{array}
$$

Figure 15.3 A context-free grammar specification of the syntax of First-Order Logic representations. Adapted from Russell and Norvig 2002.

Interpretation & Inference

- Interpretation: mapping FOL theory (set of formulas) to a model

- Inference: algorithmically (via system of rules (*calculus*)) deduce new formulas from existing formulas

$$\vdash \quad \frac{\alpha \qquad \alpha \implies \beta}{\beta}$$

Modus ponens

$$\frac{VegetarianRestaurant(Leaf) \qquad \forall x VegetarianRestaurant(x) \implies Serves(x, VegetarianFood)}{Serves(Leaf, VegetarianFood)}$$

T ⊨ F : Every model for T is a model of F

- forward chaining: deduce "everything possible" via forward application of rules
- backward chaining: theory + query: prove as contradiction free via"backward application" of deduction rules

sound
(if T ⊢ F then T ⊨ F)
but not
complete
(if T ⊨ F then T ⊢ F)

Soundness, Completeness, Decidability

- Goal for any calculus: soundness and completeness $T \vdash F \Longleftrightarrow T \vDash F$

- More practical than complete: refutation complete:
  $T \wedge \neg F \vDash \square \Longrightarrow T \wedge \neg F \vdash \square$
  (i.e. prove $T \vDash F$ via $T \wedge \neg F \vDash \square$)

- FOL: decision problem of deciding whether a formula F is satisfiable in a theory T ($T \vDash F$) or a is tautology ($\emptyset \vDash F$) is undecidable although at least tautologies are recursively enumerable.

- Resolution calculus is sound and refutation complete (but not complete (that is not a problem) :

$$\frac{\begin{array}{l} C_1^{(1)} \vee C_2^{(1)} \vee \cdots \vee C_n^{(1)} \vee L \\ C_1^{(2)} \vee C_2^{(2)} \vee \cdots \vee C_m^{(2)} \vee \neg L' \end{array}}{\sigma_{L,L'}(C_1^{(1)} \vee C_2^{(1)} \vee \cdots \vee C_n^{(1)} \vee C_1^{(2)} \vee C_2^{(2)} \vee \cdots \vee C_m^{(2)})}$$

$C_j^{(1)}$, $C_i^{(2)}$, L, L' are literals;
$C_j^{(1)}$, $C_i^{(2)}$ are variable-disjoint
$\sigma_{L,L'}$ is a unifier of L, L‘

Example [2]

Everyone who loves all animals is loved by someone.
Anyone who kills an animal is loved by no one.
Jack loves all animals.
Either Jack or Curiosity killed the cat, who is named Tuna.
Did Curiosity kill the cat?

A. $\forall x \ [\forall y \ \ Animal(y) \Rightarrow Loves(x, y)] \Rightarrow [\exists y \ \ Loves(y, x)]$

B. $\forall x \ [\exists z \ \ Animal(z) \land Kills(x, z)] \Rightarrow [\forall y \ \ \neg Loves(y, x)]$

C. $\forall x \ \ Animal(x) \Rightarrow Loves(Jack, x)$

D. $Kills(Jack, Tuna) \lor Kills(Curiosity, Tuna)$

E. $Cat(Tuna)$

F. $\forall x \ \ Cat(x) \Rightarrow Animal(x)$

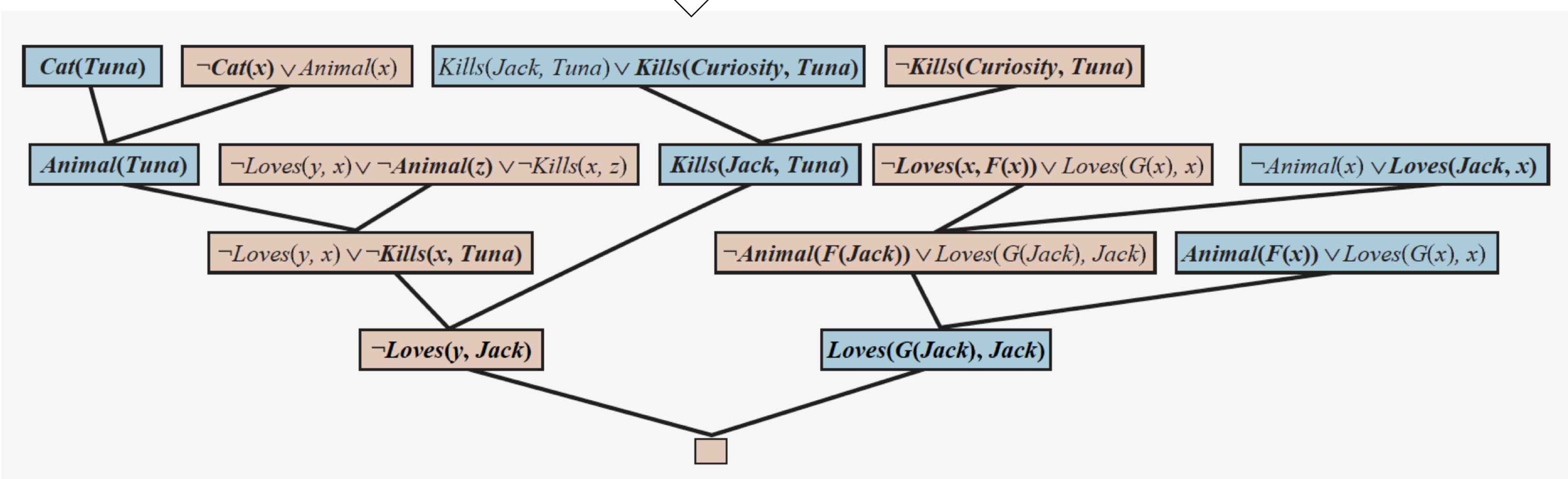
$\neg$G. $\neg Kills(Curiosity, Tuna)$

Eliminate implications, move negations inwards, standadize variables, skolemize, drop universal quantifiers, distribute or over and

Example [2]

A1.	$Animal(F(x)) \lor Loves(G(x), x)$
A2.	$\neg Loves(x, F(x)) \lor Loves(G(x), x)$
B.	$\neg Loves(y, x) \lor \neg Animal(z) \lor \neg Kills(x, z)$
C.	$\neg Animal(x) \lor Loves(Jack, x)$
D.	$Kills(Jack, Tuna) \lor Kills(Curiosity, Tuna)$
E.	$Cat(Tuna)$
F.	$\neg Cat(x) \lor Animal(x)$
$\neg$G.	$\neg Kills(Curiosity, Tuna)$

Refutation-equivalent (Skolem-) CNF



Event and State Representations

- State: holds for longer periods of time. example: $Serves(Leaf, VegetarianFare)$

- Event: more point-like in time. ←→ attributes of predicate in sentence ←→ FOL predicate (fixed arity !) ←?→ verb's arguments in subcategorization frame (may vary in number !):
  example:
  I ate.
  I ate a turkey sandwich.
  I ate a turkey sandwich at my desk.
  I ate at my desk.
  I ate lunch.
  I ate a turkey sandwich for lunch.
  I ate a turkey sandwich for lunch at my desk.

  - solution (neo-Davidsonian event representation): use event variables & property predicates

$$\exists e\ Eating(e) \ \wedge \ Eater(e, Speaker) \wedge Eaten(e, TurkeySandwich)$$

$$\exists e\ Eating(e) \ \wedge \ Eater(e, Speaker) \wedge Eaten(e, TurkeySandwich) \\ \wedge \ Meal(e, Lunch) \wedge Location(e, Desk)$$

Representations of Time

- Temporal logic: events ←→ points or intervals

$$\left.\begin{array}{l}\text{I arrived in New York.}\\ \text{I am arriving in New York.}\\ \text{I will arrive in New York.}\end{array}\right\} \exists e Arriving(e) \wedge Arriver(e, Speaker) \wedge Destination(e, NewYork)$$

$$\begin{aligned}
\exists e, i, n\, Arriving(e) \;&\wedge\; Arriver(e, Speaker) \wedge Destination(e, NewYork) \\
&\wedge\; IntervalOf(e, i) \wedge EndPoint(i, n) \wedge Precedes(n, Now) \\
\exists e, i, n\, Arriving(e) \;&\wedge\; Arriver(e, Speaker) \wedge Destination(e, NewYork) \\
&\wedge\; IntervalOf(e, i) \wedge MemberOf(i, Now) \\
\exists e, i, n\, Arriving(e) \;&\wedge\; Arriver(e, Speaker) \wedge Destination(e, NewYork) \\
&\wedge\; IntervalOf(e, i) \wedge EndPoint(i, n) \wedge Precedes(Now, n)
\end{aligned}$$

Language ←?→ Representations of Time

Ok, we fly from San Francisco to Boston at 10.

Flight 1390 will be at the gate an hour now.

refers to future event

refers to past event

Flight 1902 arrived late.

Flight 1902 had arrived late.

both in past but second has important event(s) between then and now

solution: Reichenbach‘s reference point approach:

When Mary’s flight departed, I ate lunch.

When Mary’s flight departed, I had eaten lunch.

Reichenbach's Reference Point Approach

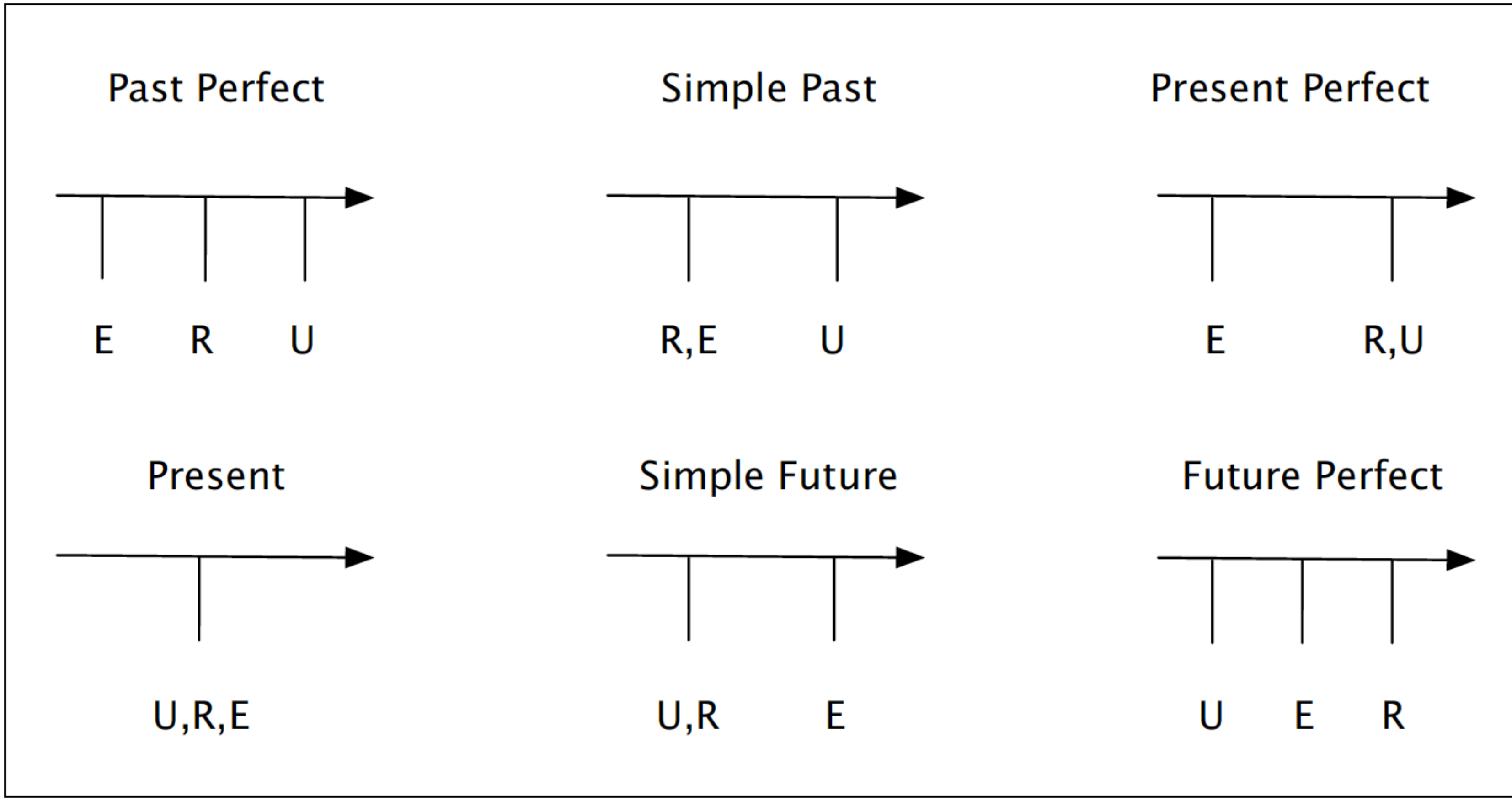


Figure 15.5 Reichenbach's approach applied to various English tenses. In these diagrams, time flows from left to right, **E** denotes the time of the event, **R** denotes the reference time, and **U** denotes the time of the utterance.

Aspect

- Events (involve change) vs states (do not involve change)

- Stative expressions: event participant is in a state at point in time

  I like Flight 840.

  I need the cheapest fare.

  I want to go first class.

- Activity expressions: event participant does activity (event) over interval in time

  She drove a Mazda.

  I live in Brooklyn.

- Accomplishment expressions: interval has definitive end-point, results in state change

  He booked me a reservation.

  United flew me to New York.

- Achievement expressions: point in time, results in state change

  She found her gate.

  I reached New York.

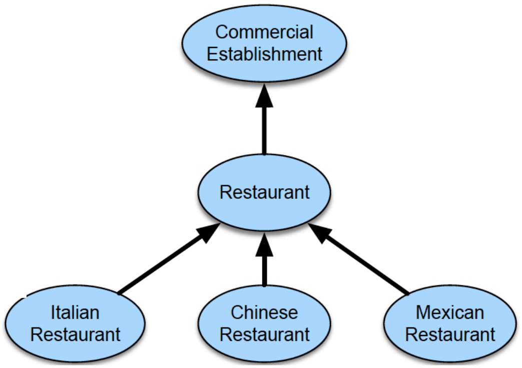
Description Logics → Ontologies

- representation of knowledge about categories, individuals that belong to those categories, the relationships among these individuals → OWL, Semantic Web

- T-Box ("terminology"): category + relationship principal structure
  A-Box ("assertion"): info about individuals

- Ontologies: main relation between categories: subsumption:
  $C \sqsubseteq D$ (C is subsumed by D: every instance of C is also in D)

$$\begin{aligned} \text{Restaurant} &\sqsubseteq \text{CommercialEstablishment} \\ \text{ItalianRestaurant} &\sqsubseteq \text{Restaurant} \\ \text{ChineseRestaurant} &\sqsubseteq \text{Restaurant} \\ \text{MexicanRestaurant} &\sqsubseteq \text{Restaurant} \end{aligned}$$



Description Logics → Ontologies

- Disjointness: $\mathsf{ChineseRestaurant} \sqsubseteq \textbf{not}\ \mathsf{ItalianRestaurant}$

- Coverage: $\mathsf{Restaurant} \sqsubseteq$
  $(\textbf{or}\ \mathsf{ItalianRestaurant}\ \mathsf{ChineseRestaurant}\ \mathsf{MexicanRestaurant})$

- Further relationships between categories: Roles:

$$
\begin{aligned}
\mathsf{MexicanCuisine} &\sqsubseteq \mathsf{Cuisine} \\
\mathsf{ItalianCuisine} &\sqsubseteq \mathsf{Cuisine} \\
\mathsf{ChineseCuisine} &\sqsubseteq \mathsf{Cuisine} \\
\mathsf{VegetarianCuisine} &\sqsubseteq \mathsf{Cuisine}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{ExpensiveRestaurant} &\sqsubseteq \mathsf{Restaurant} \\
\mathsf{ModerateRestaurant} &\sqsubseteq \mathsf{Restaurant} \\
\mathsf{CheapRestaurant} &\sqsubseteq \mathsf{Restaurant}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{ItalianRestaurant} &\equiv \mathsf{Restaurant} \sqcap \exists \mathsf{hasCuisine}.\mathsf{ItalianCuisine} \\
\mathsf{ModerateRestaurant} &\equiv \mathsf{Restaurant} \sqcap \exists \mathsf{hasPriceRange}.\mathsf{ModeratePrices}
\end{aligned}
$$

$$
\begin{aligned}
\forall x\, ItalianRestaurant(x) &\rightarrow Restaurant(x) \\
&\wedge (\exists y\, Serves(x,y) \wedge ItalianCuisine(y))
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{VegetarianRestaurant} \equiv\ & \mathsf{Restaurant} \\
& \sqcap \exists \mathsf{hasCuisine}.\mathsf{VegetarianCuisine} \\
& \sqcap \forall \mathsf{hasCuisine}.\mathsf{VegetarianCuisine}
\end{aligned}
$$

Decription Logics: Inference

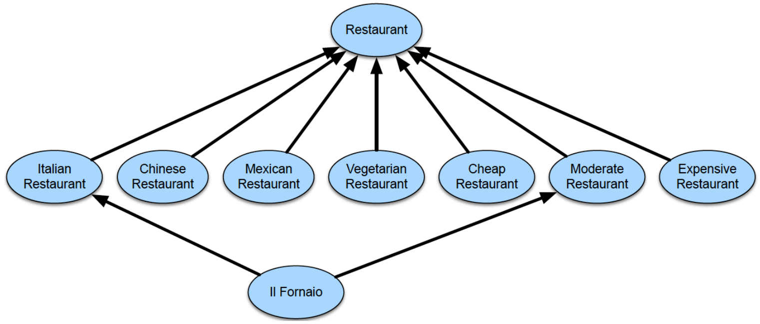
- Subsumption inference

$$\mathsf{IlFornaio} \sqsubseteq \mathsf{ModerateRestaurant} \sqcap \exists \mathsf{hasCuisine.ItalianCuisine}$$

restaurant chain
(→ a category)

$$\mathsf{IlFornaio} \sqsubseteq \mathsf{ItalianRestaurant} \quad ? \rightarrow \text{true}$$

$$\mathsf{IlFornaio} \sqsubseteq \mathsf{VegetarianRestaurant} \quad ? \rightarrow \text{false}$$

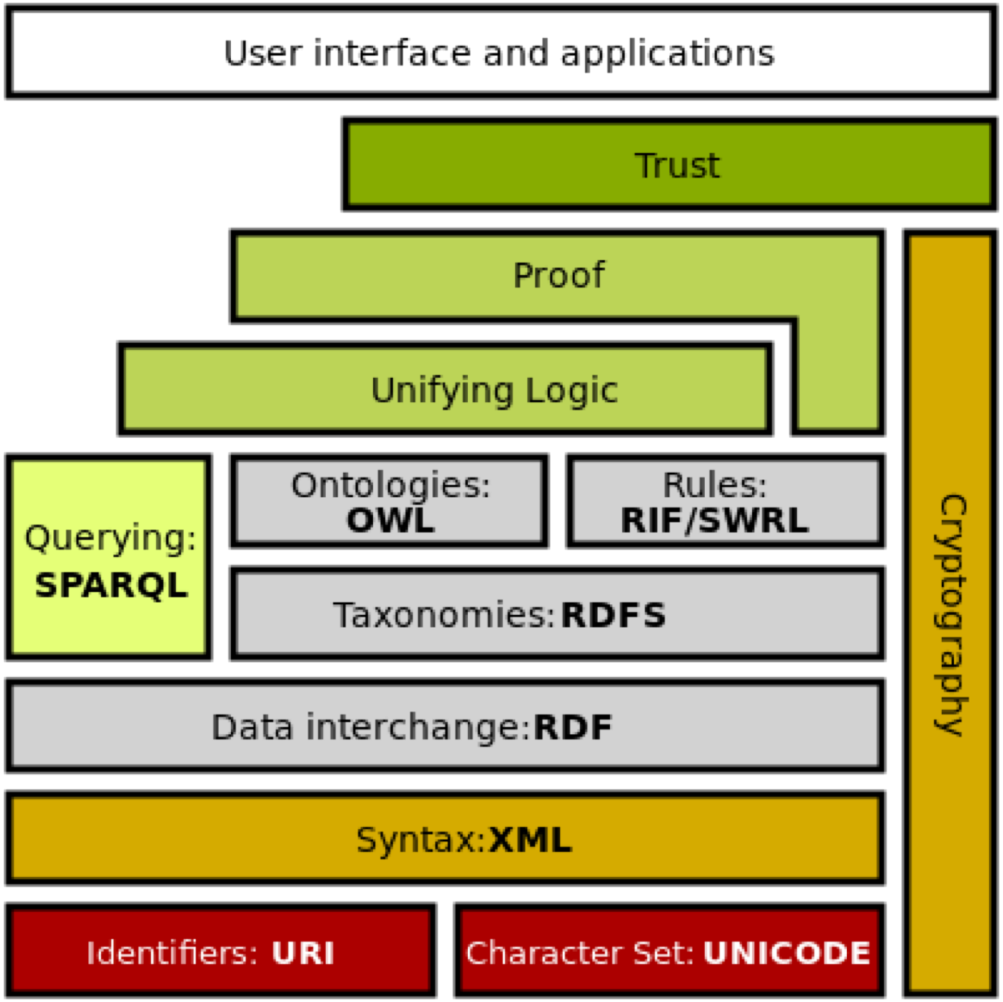


- Instance checking:

$$\mathsf{Restaurant(Gondolier)}$$
$$\mathsf{hasCuisine(Gondolier, ItalianCuisine)}$$

$$\mathsf{ItalianRestaurant(Gondolier)} \ ? \rightarrow \text{true}$$

Semantic Web



Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Jan, 2022); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2022) (this slideset is especially based on chapter 2)

(2) Russel, Norvig: Artifical Intelligence, 3rd edition

Recommendations for Studying

- minimal approach:

  work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

  minimal approach + read the corresponding pages in Jurafsky [1]

- interested students

  standard approach + do a selection of the exercises in Jurafsky [1]