School of CIT
Social Computing Research Group

Technical
University
of Munich

Natural Language Processing
IN2361

Prof. Dr. Georg Groh

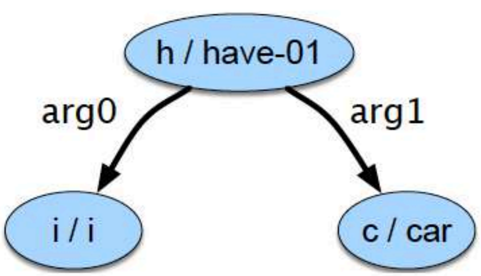
Chapter 19
Logical Representations of Sentence Meaning

- 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!

Logical Representations of Sentence Meaning

text $\xrightarrow{\text{semantic parsing / semantic analysis}}$ meaning representation

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




```
(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

- **Verifiability**: 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:**

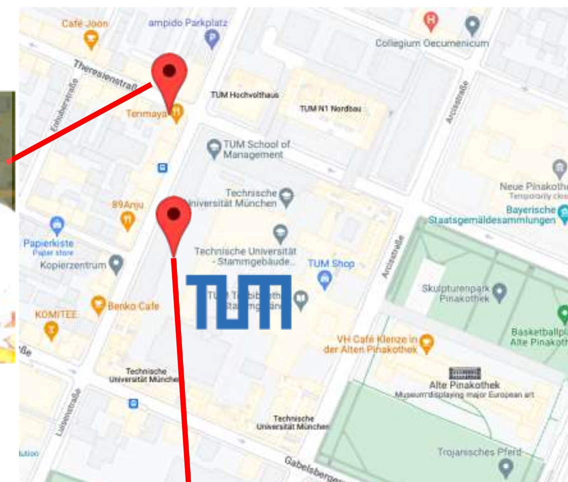
  "I wanna eat someplace that is close to TUM" :

  ambiguous sentence: more than one representation

  "I wanna eat Italian food" :

  vague sentence: not more than one represenation

<repr1>≙



<repr2>≙

Desiderata for Semantic Representations of NLP Entities

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

$$\left.\begin{array}{l}\textit{Does Maharani have vegetarian dishes?}\\ \textit{Do they have vegetarian food at Maharani?}\\ \textit{Are vegetarian dishes served at Maharani?}\\ \textit{Does Maharani serve vegetarian fare?}\end{array}\right\} \xrightarrow{!} \text{same <representation>}$$

- 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

  predicate symbols, function symbols, ...
- 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.

Model-Theoretic Semantics

Domain	$\mathcal{D} = \{a, b, c, d, e, f, g, h, i, j\}$
Matthew, Franco, Katie and Caroline	a, b, c, d
Frasca, Med, Rio	e, f, g
Italian, Mexican, Eclectic	h, i, j

FOL: terms

Properties

Noisy

Frasca, Med, and Ri

licates

Relations

dicates

Likes

$g\rangle\}$

Matthew likes the M

Katie likes the Med

Franco likes Frasca

Caroline likes the M

Serves

Med serves eclectic

Rio serves Mexican

Frasca serves Italian

Figure 15.2 A model of the r

With this alone: the following statements are hard to represent:

Katie likes the Rio and Matthew likes the Med.

Katie and Caroline like the same restaurants.

Franco likes noisy, expensive restaurants.

Not everybody likes Frasca.

→ We need more expressiveness → FOL

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.

First Order Logic – Examples

a restaurant that serves Mexican food near ICSI.

$$\exists x Restaurant(x) \;\wedge\; Serves(x, MexicanFood)$$
$$\wedge\; Near((LocationOf(x), LocationOf(ICSI))$$

All vegetarian restaurants serve vegetarian food.

$$\forall x VegetarianRestaurant(x) \implies Serves(x, VegetarianFood)$$

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

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

Modus ponens

$$\frac{\begin{array}{l} VegetarianRestaurant(Leaf) \\ \forall x VegetarianRestaurant(x) \Longrightarrow Serves(x, VegetarianFood) \end{array}}{Serves(Leaf, VegetarianFood)}$$

„entails" T $\models$ 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

syntatically derived

sound

(if T $\vdash$ F then T $\models$ F)

but not

logically derived

complete

(if T $\models$ F then T $\vdash$ F)

Soundness, Completeness, Decidability

- Goal for any calculus: soundness and completeness T ⊢ F $\Longleftrightarrow$ T ⊨ F

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

- FOL: decision problem of deciding whether a formula F is satisfiable in a theory T (T ⊨ F) or a is tautology (∅ ⊨ 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) :

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

disjunction normal form

$C_j^{(1)}, C_i^{(2)}, L, L'$ 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. $\forall x\ [\exists z\ \ Animal(z) \wedge Kills(x,z)] \Rightarrow [\forall y\ \ \neg Loves(y,x)]$

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

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

E. $Cat(Tuna)$

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

¬G. $\neg Kills(Curiosity, Tuna)$

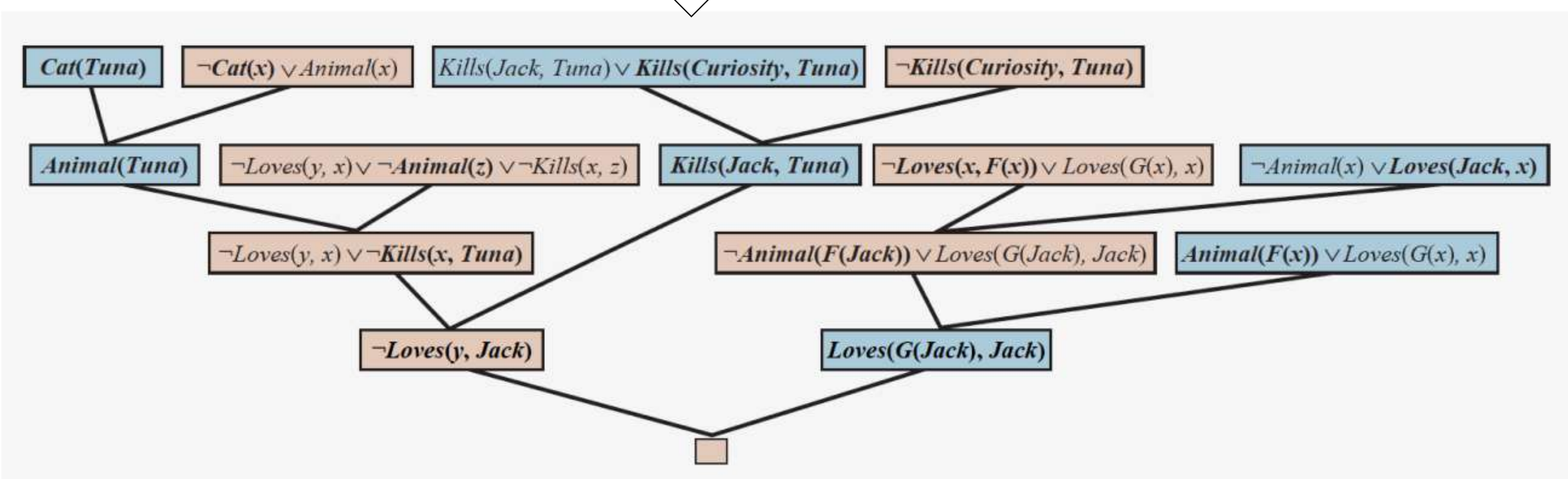
apply negation to prove by contradiction

Eliminate implications, move negations inwards, standardize 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.	$\neg Cat(x) \lor Animal(x)$
¬G.	$\neg Kills(Curiosity, Tuna)$

Refutation-equivalent (Skolem-) CNF



Beyond FOL

- Second order logic: allow for quantifiers over predicates

$$\exists P\ P(b)$$

example: set of all cubes and tetrahedrons [3]

$$\exists P\ \forall x\ (Px\ \leftrightarrow\ (Cube(x)\ \vee\ Tet(x))).$$

the set of all cubes and tetrahedrons does not contain any dodecahedrons [3]

$$\forall P\ (\forall x\ (Px\ \leftrightarrow\ (Cube(x)\ \vee\ Tet(x)))\ \rightarrow\ \neg\,\exists x\ (Px\ \wedge\ Dodec(x))).$$

Beyond FOL

- Modal logic: represent statements about necessity $\Box$ and possibility $\Diamond$

A *relational model* is a tuple $\mathfrak{M} = \langle W, R, V \rangle$ where:

1. W is a set of possible worlds
2. R is a binary relation on W
3. V is a valuation function which assigns a truth value to each pair of an atomic formula and a world, (i.e. $V : W \times F \to \{0, 1\}$ where F is the set of atomic formulae)

[3]

Then we recursively define the truth of a formula at a world w in a model $\mathfrak{M}$:

- $\mathfrak{M}, w \models P$ iff $V(w, P) = 1$
- $\mathfrak{M}, w \models \neg P$ iff $w \not\models P$
- $\mathfrak{M}, w \models (P \wedge Q)$ iff $w \models P$ and $w \models Q$
- $\mathfrak{M}, w \models \Box P$ iff for every element u of W, if wRu then $u \models P$
- $\mathfrak{M}, w \models \Diamond P$ iff for some element u of W, it holds that wRu and $u \models P$

[3]

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

I arrived in New York.
I am arriving in New York.
I will arrive in New York.

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

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

Language ←?→ Representations of Time

Ok, we fly from San Francisco to Boston at 10. → refers to future event

Flight 1390 will be at the gate an hour now. → 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 (see later chapter on temporal reasoning):

When Mary's flight departed, I ate lunch.

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

Reichenbach's Reference Point Approach

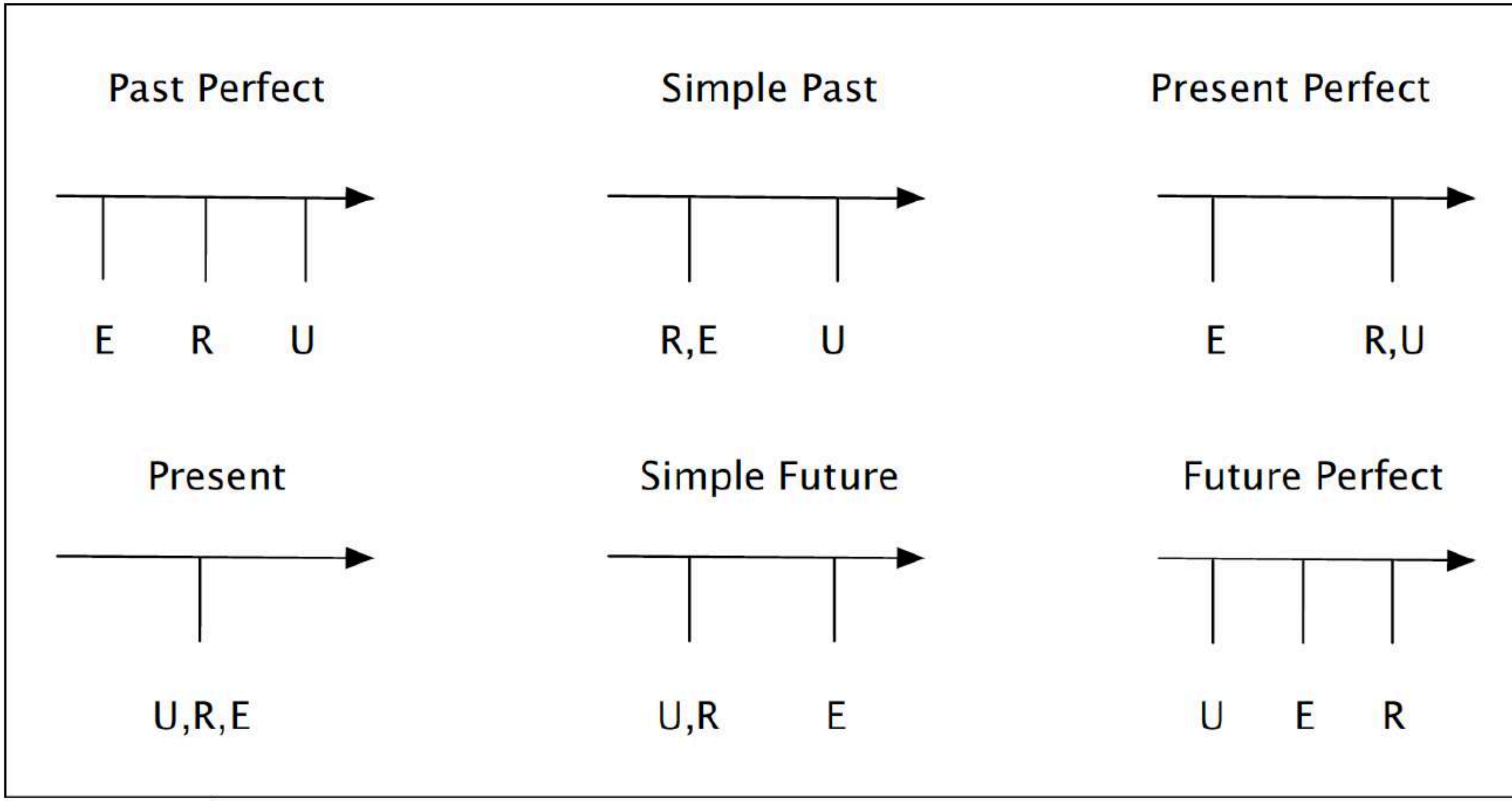


Figure 15.5 Reichenbach's approach applied to various English tenses. In these diagrams, time flows from left to right, **E** denotes the time of the event, **R** denotes the reference time, and **U** denotes the time of the utterance.

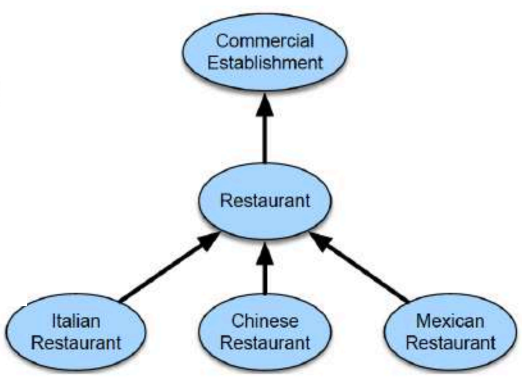
Description Logics → Ontologies

- 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: ChineseRestaurant $\sqsubseteq$ **not** ItalianRestaurant

- Coverage: Restaurant $\sqsubseteq$ (**or** ItalianRestaurant ChineseRestaurant MexicanRestaurant)

- Further relationships between categories: Roles:

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

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

$$
\forall x \, ItalianRestaurant(x) \rightarrow Restaurant(x) \wedge (\exists y Serves(x,y) \wedge ItalianCuisine(y))
$$

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

Description Logics: Inference

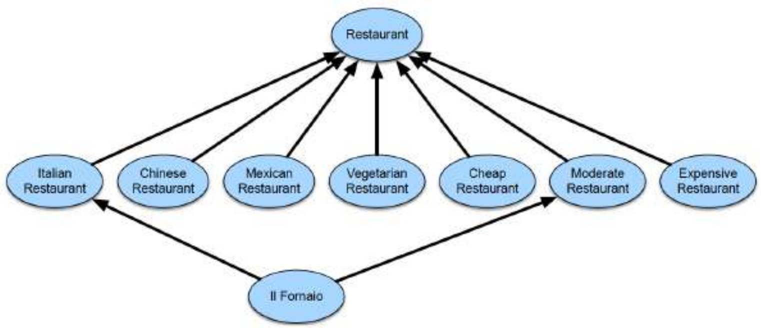
- Subsumption inference

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

restaurant chain
(→ a category)

$\mathsf{IlFornaio} \sqsubseteq \mathsf{ItalianRestaurant}$? → true

$\mathsf{IlFornaio} \sqsubseteq \mathsf{VegetarianRestaurant}$? → false

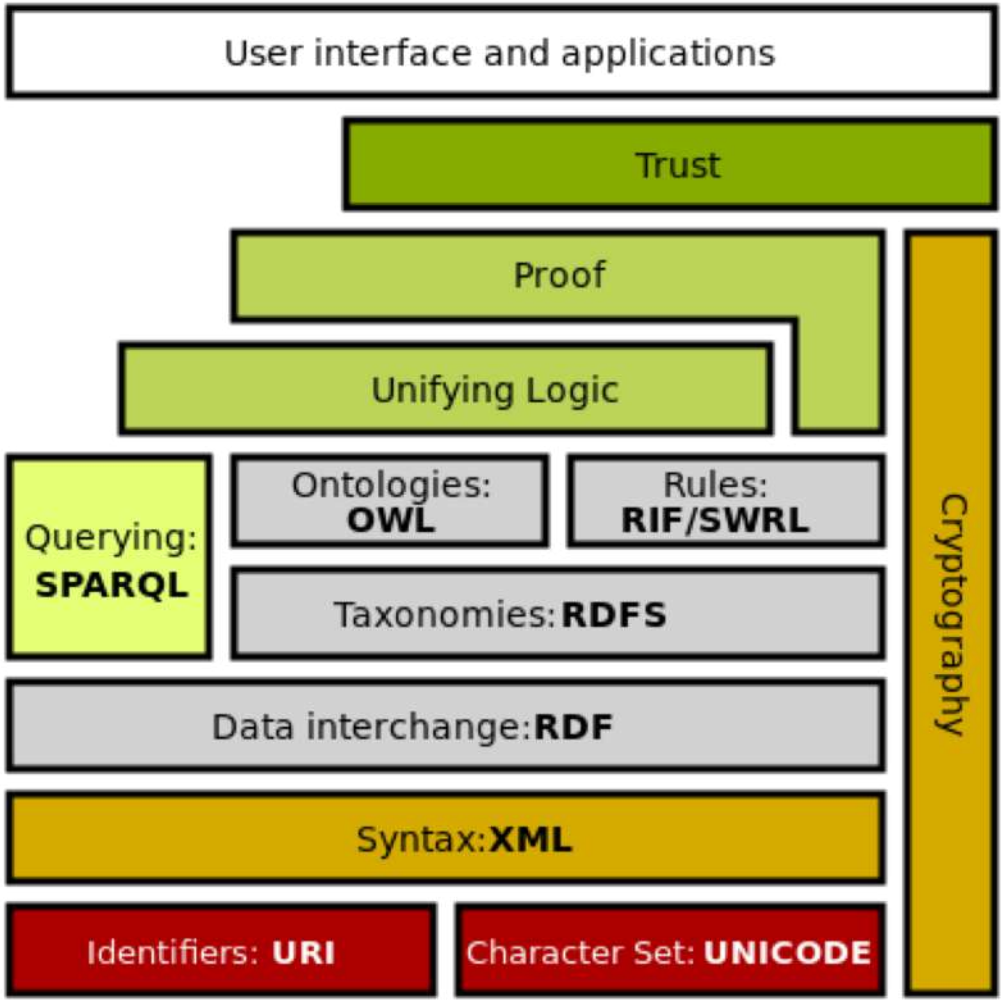


- Instance checking:

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

ItalianRestaurant(Gondolier) ? → true

Semantic Web



Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Jan, 2023); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2023) (this slideset is especially based on chapter 19)

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

(3) Wikipedia articles (Jan 2024) for Second order logic and Modal Logic

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]