

Symbolic AI

Andre Freitas



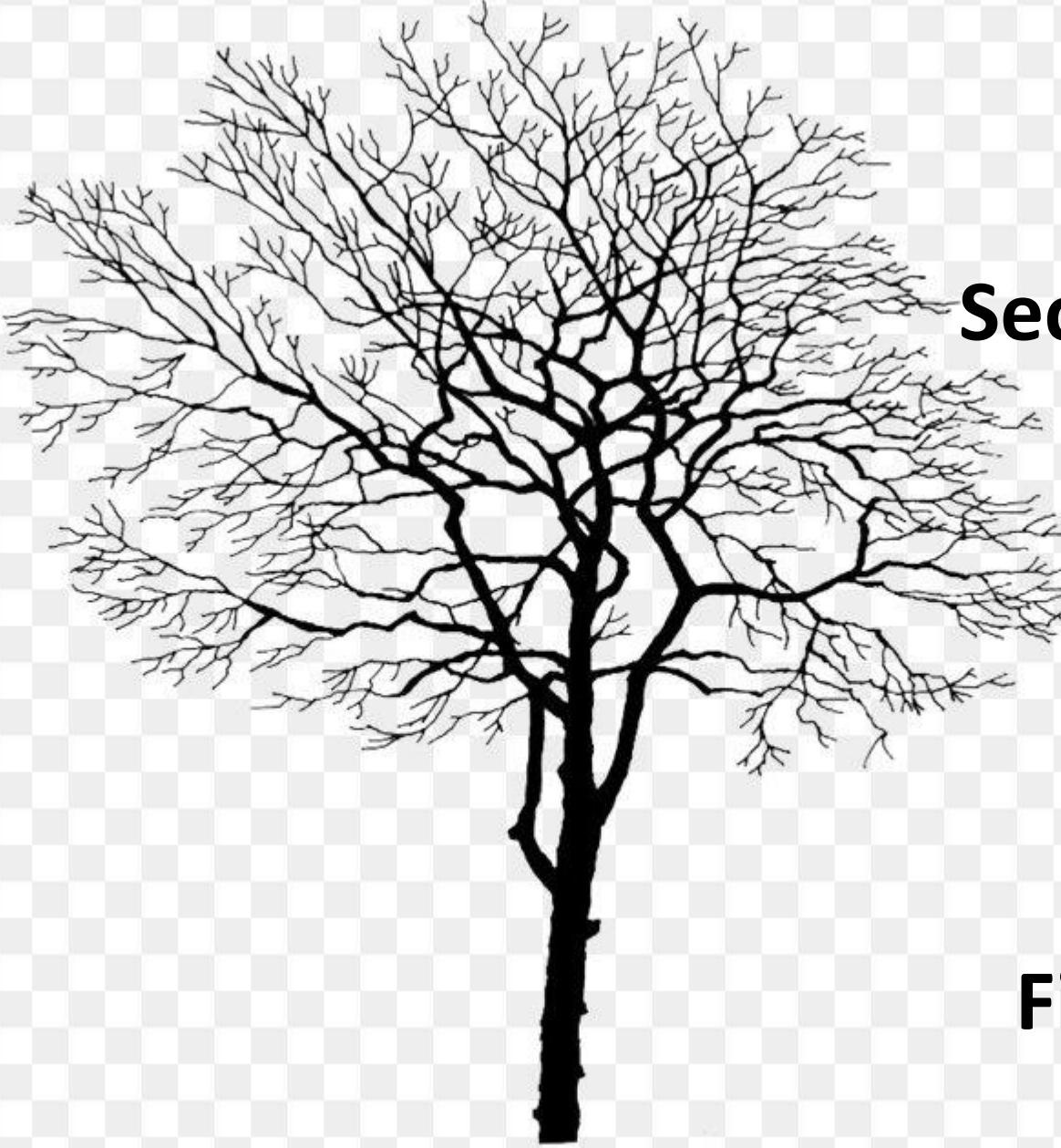
Photo by Vasilyev

A close-up photograph of a person's hand gripping a dark-colored, textured gear shift lever in a car. The background shows the interior of the car, including a wooden dashboard and various control buttons.

**Overview of the second
part of the course**

Pedagogical Take

- Giles provided you with the foundations on logical representation, reasoning and programming.
- We will build and expand on it.
- Comparatively, this part of the course will cover more topics (broader strokes).
- Fundamental to provide you with an end-to-end view on Symbolic AI.
- Mastering complexity.



Second Part

First Part

How to Study

- Be in a position to define and explain:
 - The core concepts and algorithms.
 - Why they are relevant?
 - When you should use/not use them?
- Basic application of the core algorithms.

Employability Skills

- This unit is heavily complementary to Machine Learning and Text Mining.
- Contemporary AI is evolving rapidly in the direction of hybrid **neuro-symbolic models**.
- Larger palette to build AI Systems.

[Web](#)[News](#)[Images](#)[Shopping](#)[Videos](#)[More ▾](#)[Search tools](#)

About 186,000,000 results (0.53 seconds)

Michelle Obama (m. 1992)

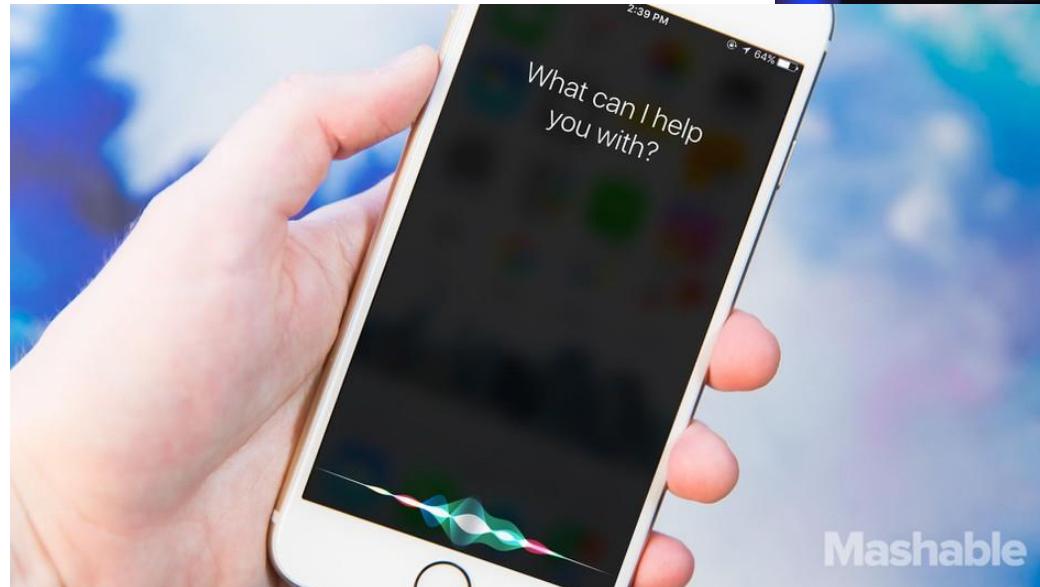
Barack Obama, Spouse



[Michelle Obama - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Michelle_Obama ▾ Wikipedia

Michelle LaVaughn Robinson Obama (born January 17, 1964), an American writer, is the wife of the 44th and current President of the United States, Barack Obama. She is also the First Lady of the United States.

Capers Funnye - Sidley Austin - Hyde Park, Chicago - Jeremiah Wright



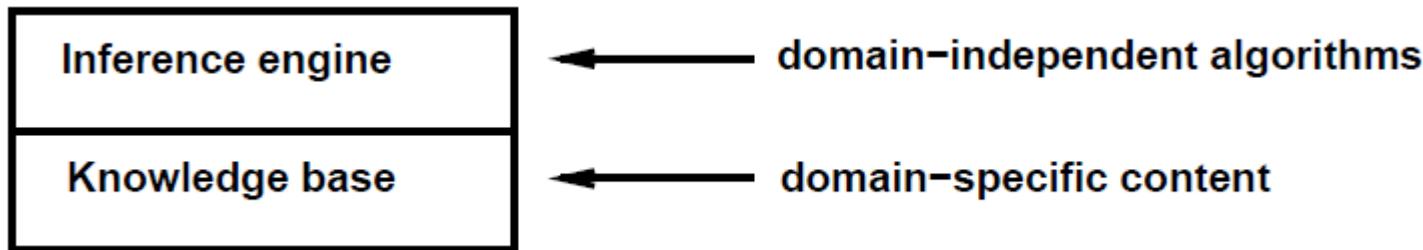
S LARGEST AIRPORT IS NAMED FOR
A WORLD WAR II HERO;
ITS SECOND LARGEST,
FOR A WORLD WAR II BATTLE

What is
Toronto?????

\$36,681

\$

Knowledge Bases



- Knowledge base = set of sentences in a formal language
- Declarative approach to building an agent (or other system):
 - Tell it what it needs to know
- Then it can Ask itself what to do | answers should follow from the KB
- Agents can be viewed at the knowledge level
 - i.e., what they know, regardless of how implemented
- Or at the implementation level
 - i.e., data structures in KB and algorithms that manipulate them

KB Agent

function KB-AGENT(*percept*) **returns** an *action*

static: *KB*, a knowledge base

t, a counter, initially 0, indicating time

 TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept, t*))

action \leftarrow ASK(*KB*, MAKE-ACTION-QUERY(*t*))

 TELL(*KB*, MAKE-ACTION-SENTENCE(*action, t*))

t \leftarrow *t* + 1

return *action*

A simple knowledge-based agent

- The agent must be able to:
 - Represent states, actions, etc.
 - Incorporate new percepts
 - Update internal representations of the world
 - Deduce hidden properties of the world
 - Deduce appropriate actions
- => sound and complete reasoning with partial information states

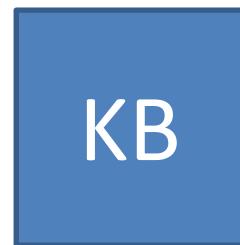
Knowledge in Learning

- Task: to design agents that already know something, and are trying to learn more.
- Agents must have a learning process to gain the background knowledge in the first place
 - Learning taken place afterwards define the agent's incremental/cumulative development
- Agents can start off like normal agents
 - Gain initial knowledge through inductive learning
 - After, uses background knowledge to learn more effectively

Goals of this second part of the course

- Which knowledge representation elements support more expressive, accurate, efficient and general algorithms?
- How to build knowledge bases (under a certain representation)?
- What are the conceptual frameworks and algorithms for knowledge-based learning and inference?

Representation



**Problem
Task**

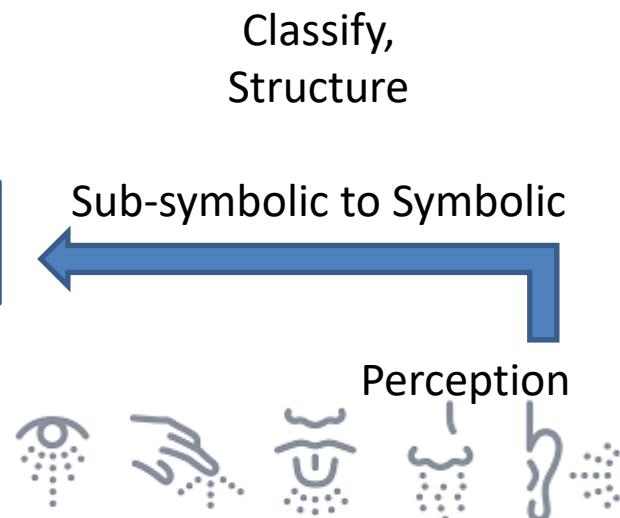


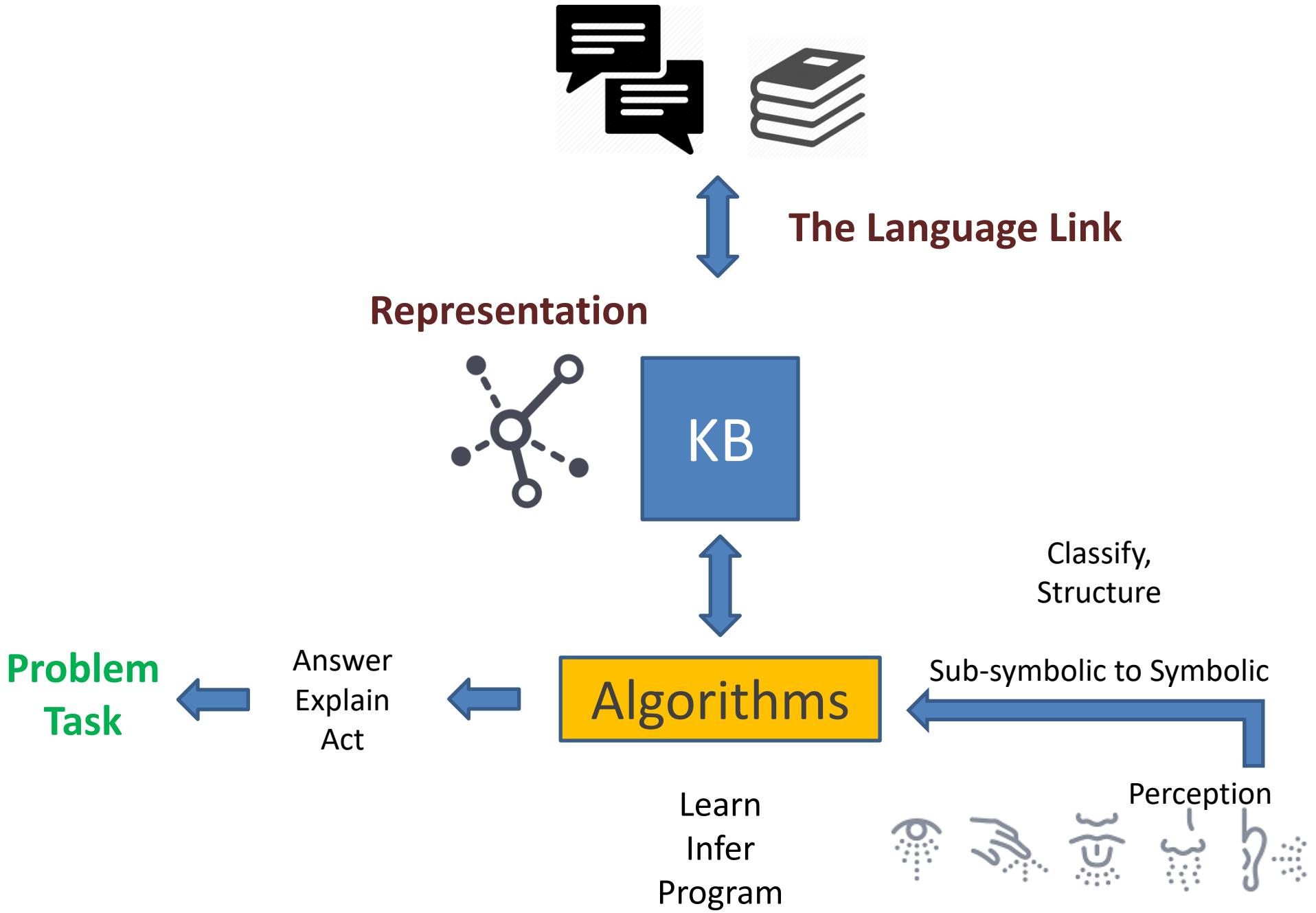
Answer
Explain
Act



Algorithms

Learn
Infer
Program



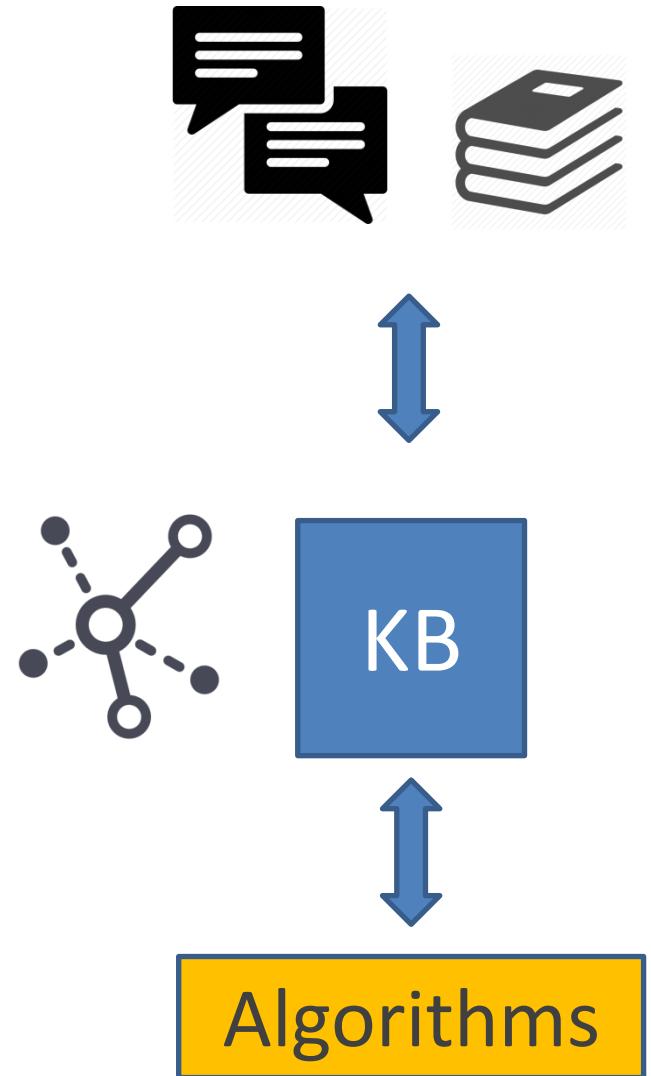


Syllabus

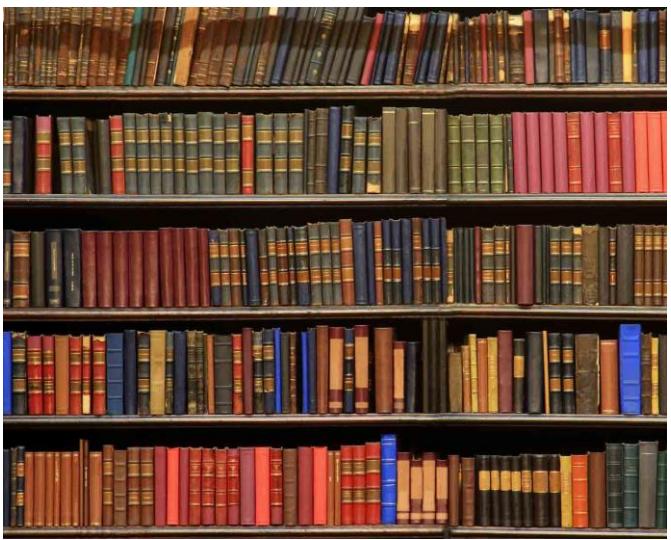
- 2. KB Construction
- 3. Montague Semantics
- 4. Semantic Parsing

- 1. Knowledge Representation

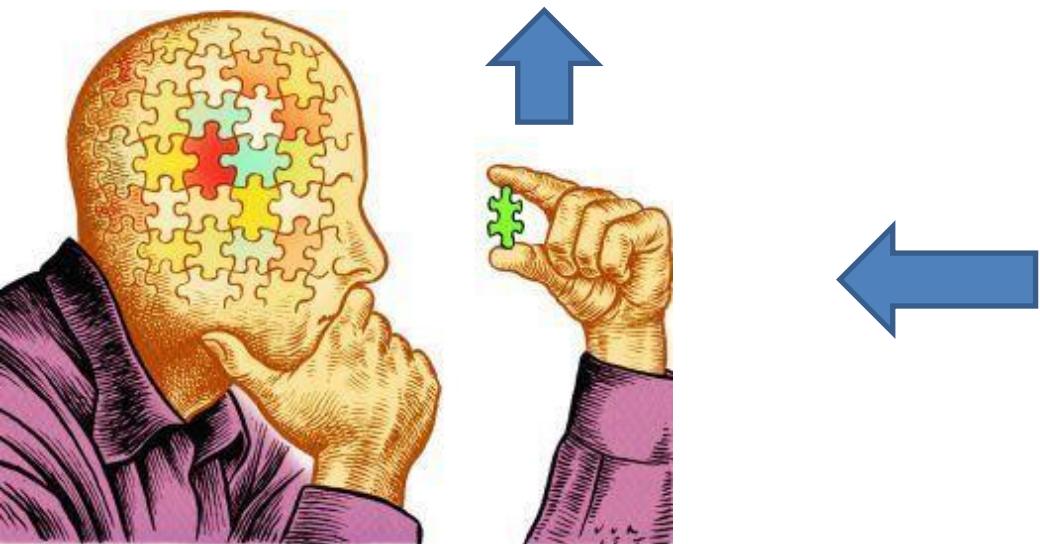
- 5. Explanation-based Learning
- 6. Inductive Logic Programming
- 7. Natural Language Inference
- 8. Neuro-Symbolic Reasoning



Knowledge Representation



Communication of the Representation



Representation
of the Reality

Structure of the
Reality





Contains Ordnance Survey data © Crown copyright and database right 2011





Services

- 1 Altrincham – Bury
- 2 Altrincham – Piccadilly
- 3 Ashton-under-Lyne – Eccles
- 4 Bury – Piccadilly
- 5 East Didsbury – Rochdale Town Centre
- 6 Manchester Airport – Victoria*
- 7 MediaCityUK – Etihad Campus

Manchester Airport – Victoria*

MediaCityUK – Etihad Campus

* Early services operate on a 20 minute frequency between Manchester Airport and Didsbury. Please check journey planning posters or [tfgm.com](#) before travelling.

Key

- Metrolink stop
- Bus interchange
- Rail interchange
- Change with other services
- Park + Ride with number of spaces
- Car park fewer than 50 spaces
- Cycle Hub membership required



Ceci n'est pas une pipe.

“Human knowledge is a process of approximation. In the focus of experience, there is comparative clarity. But the discrimination of this clarity leads into the penumbral background. There are always questions left over. The problem is to discriminate exactly what we know vaguely.”

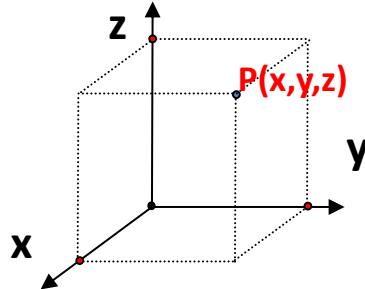
Alfred North Whitehead

KR: Five Roles

- 1. Surrogate
 - That is, a representation
- 2. Expression of ontological commitment
 - of the world
- 3. Theory of intelligent reasoning
 - and our knowledge of it
- 4. Medium of efficient computation
 - that is accessible to programs
- 5. Medium of human expression
 - and usable

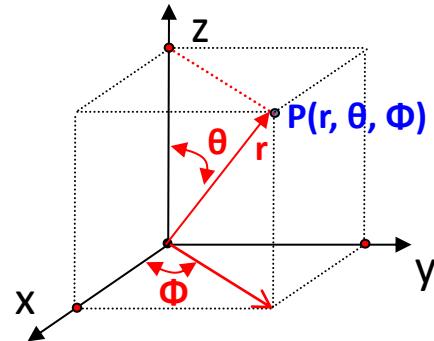
Cartesian Coordinates

$P(x, y, z)$



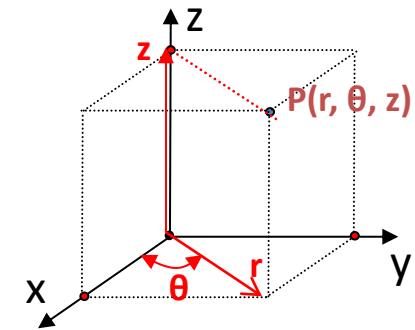
Spherical Coordinates

$P(r, \theta, \Phi)$



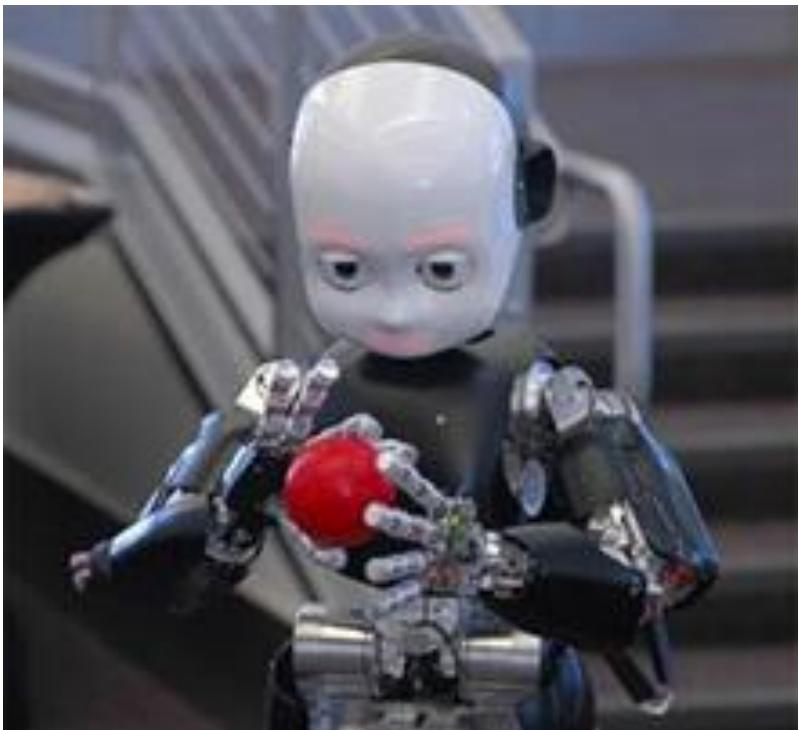
Cylindrical Coordinates

$P(r, \theta, z)$



$$ds^2 = \left(1 - \frac{2m}{r}\right) dt^2 - \frac{1}{\left(1 - \frac{2m}{r}\right)} dr^2 - (r)^2(d\theta^2 + \sin^2(\theta)d\phi^2)$$

Representations deeply impact on learning and inference



Embodied representations

An **apple** is a sweet, edible [fruit](#) produced by an **apple tree** (*Malus pumila*). Apple [trees](#) are [cultivated](#) worldwide and are the most widely grown species in the [genus Malus](#). The tree originated in [Central Asia](#), where its wild ancestor, *Malus sieversii*, is still found today. Apples have been grown for thousands of years in [Asia](#) and [Europe](#) and were brought to North America by [European colonists](#). Apples have religious and [mythological](#) significance in many cultures, including [Norse](#), [Greek](#) and [European Christian traditions](#).

Symbolic representations

“The distinctive feature of brains such as the one we own is their uncanny ability to create maps...

But when brains make maps, they are also creating images, the main currency of our minds. Ultimately consciousness allows us to experience maps as images, to manipulate those images, and to apply reasoning to them.”

Antonio Damasio (2010)

Semantics

=

Formal meaning representation
model (lots of data)

+

inference model

This behaves a lot
like intelligence!

> 2000 years of tradition!

Semantics

=

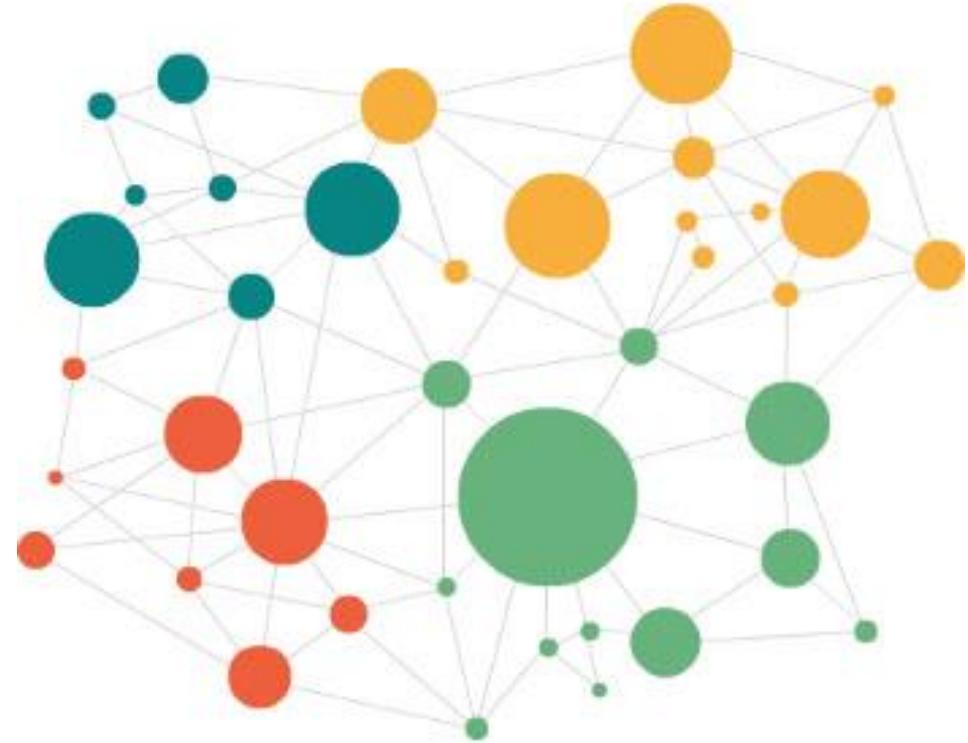
Logics, linguistics, philosophy, cognitive sciences, computer science

Formal meaning representation model (lots of data)

+

inference model

This behaves a lot like intelligence!



Building Knowledge Bases

Data

Intelligence

Structure/Semantics



Unstructured
Data

KB Construction

Structured
Data

Easy to generate

Easy to analyze
(computationally)

Consistent
Comparable
Processable

From Text to Structure

...

Barack Obama went with his daughter Malia to the baseball game.

...

...

Today, during an official visit, Natasha called to her father, the president of the United States.

...

From Text to Structure

...
Barack Obama went with
his daughter **Malia** to the
baseball game.

...
Today, during an official
visit, **Natasha** called to
her father, the president
of the United States.

From Text to Structure

...
Barack Obama went with
his daughter **Malia** to the
baseball game.
...

...
Today, during an official
visit, **Natasha** called to
her father, the **president
of the United States**.
...

Co-reference resolution

From Text to Structure

...
Barack Obama went with
his daughter **Malia** to the
baseball game.
...

...
Today, during an official
visit, **Natasha** called to
her father, the **president**
of the United States.
...

Regularities in Natural Language

Barack Obama went with his daughter Malia to the baseball game.

Barack/NNP

Obama/NNP

went/VBD

with/IN

his/PRP\$

daughter/NN

Malia/NN

to/TO

the/DT

baseball/NN

game/NN

./.

Regularities in Natural Language

Barack Obama went with his daughter Malia to the baseball game.

Barack/NNP

Obama/NNP

went/VBD

with/IN

his/PRP\$

daughter/NN

Malia/NN

to/TO

the/DT

baseball/NN

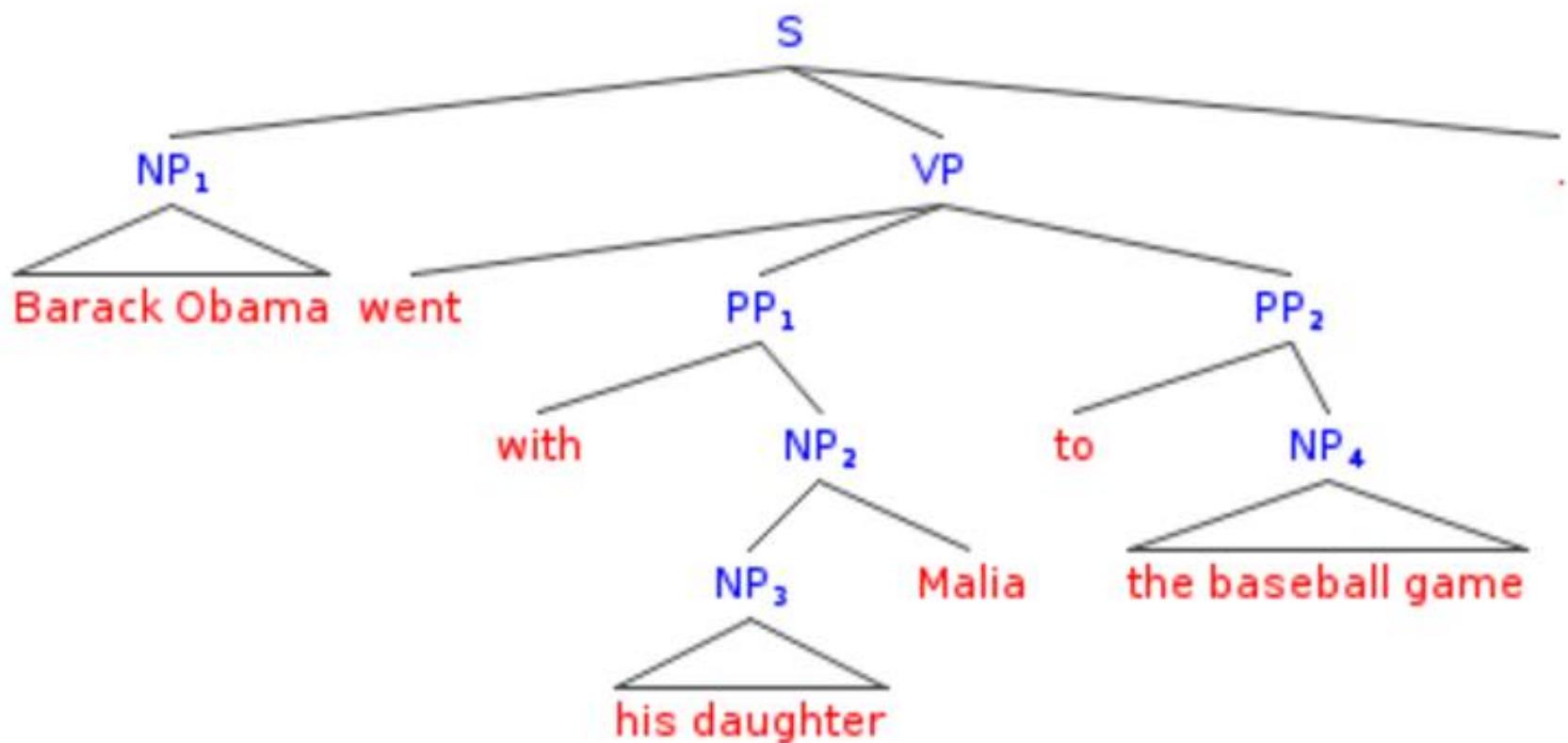
game/NN

./. .

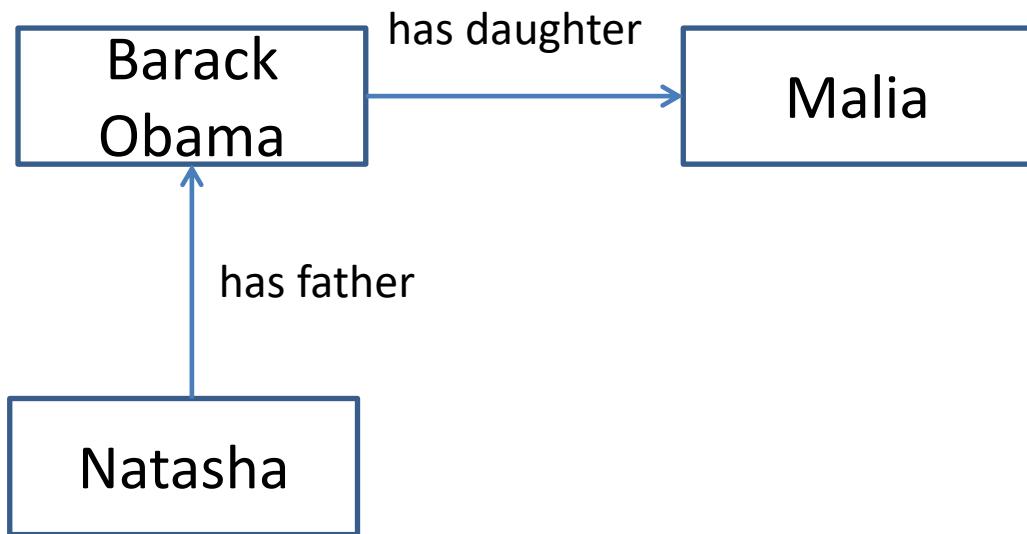


Proper noun

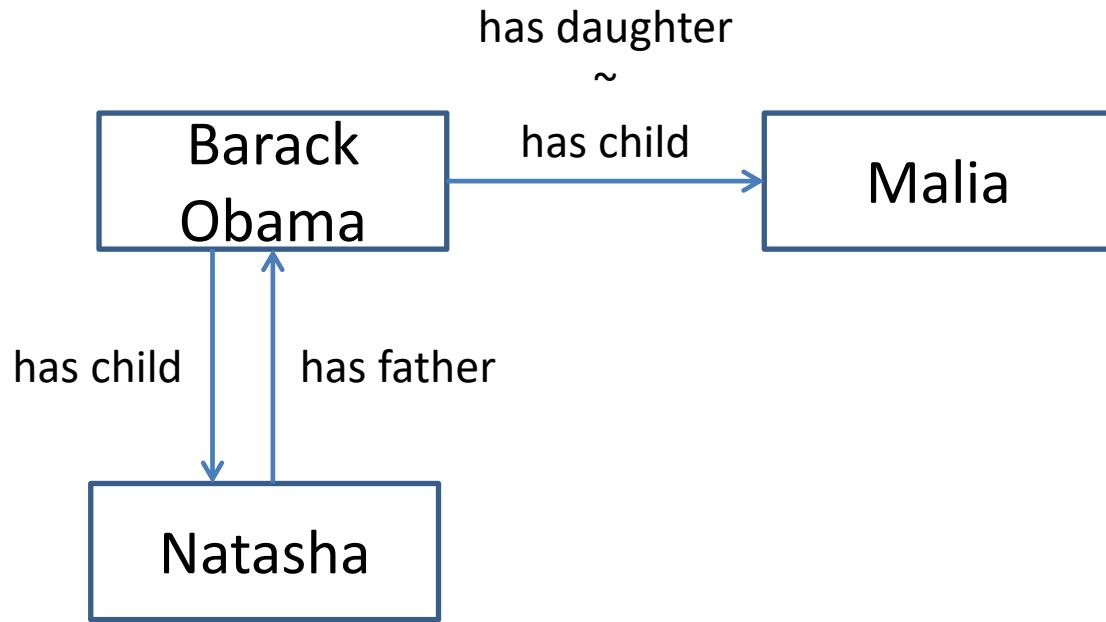
Regularities in Natural Language



Structural/Logical Form



Structural/Logical Form



Applying some logical or corpus-based inference we get

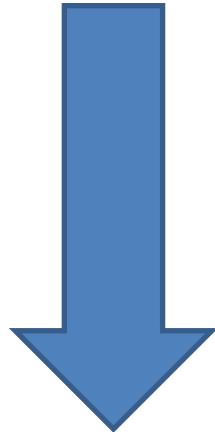
Rephrasing it

- has_child(Barack_Obama, Malia)
- has_child(Barack_Obama, Natasha)
- ...

Now we can answer this query

- *How many children does Barack Obama have?*

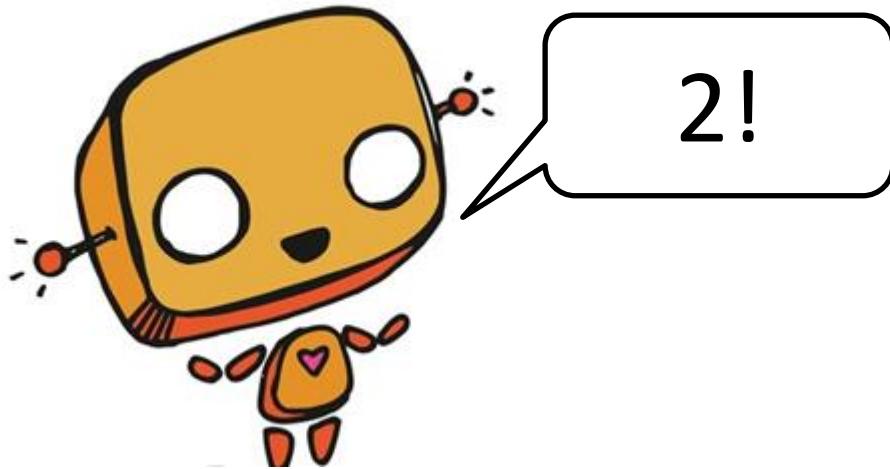
some magic called **semantic parsing** goes on ...



- `count(has_child(Barack_Obama, ?x))`

It computes!

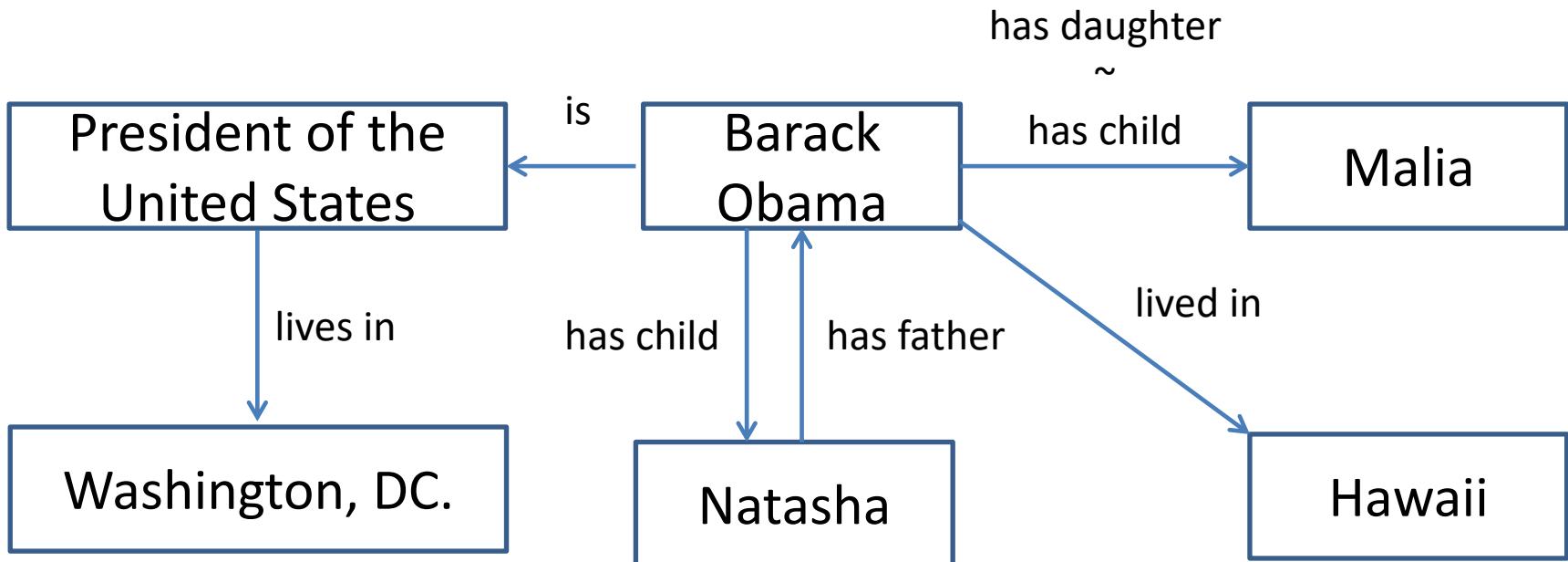
Query: count(has_child(Barack_Obama, ?x))



KB:

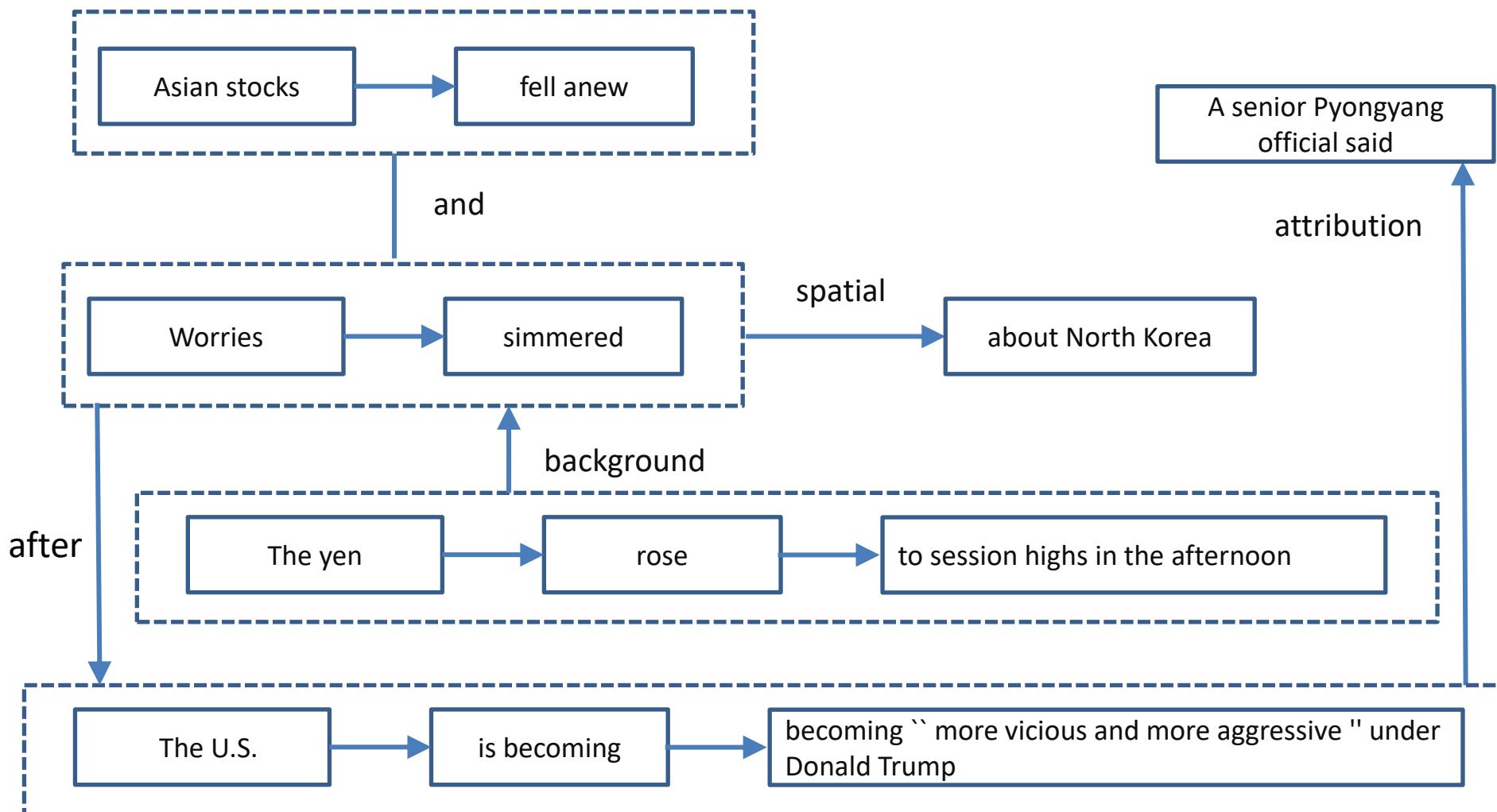
- has_child(Barack_Obama, Malia)
- has_child(Barack_Obama, Natasha)
- ...

Extrapolating



Semantic representation

Asian stocks fell anew and the yen rose to session highs in the afternoon as worries about North Korea simmered, after a senior Pyongyang official said the U.S. is becoming ``more vicious and more aggressive'' under President Donald Trump .



Semantic Parsing using CCGs

$$\frac{\text{show} \quad \text{me} \quad \frac{\text{flights}}{\lambda x. flight(x)} \quad \frac{\text{to}}{\lambda y. \lambda x. to(x, y)} \quad \frac{\text{Boston}}{BOSTON}}{\lambda x. to(x, BOSTON)} \rightarrow$$
$$\frac{\lambda f. \lambda x. f(x) \wedge to(x, BOSTON)}{\lambda x. flight(x) \wedge to(x, BOSTON)} \leftarrow$$
$$\frac{S}{\lambda x. flight(x) \wedge to(x, BOSTON)} \rightarrow$$

Symbol

Tree



Any cognitive representation for long, vertical, usually green with a wood basis

Tree is the name of a set

Tree is a noun

Some Goal



Similarity, discrimination



Representation

Structure of the Reality

Symbol



Any cognitive representation for long, vertical, usually green with a wood basis

Some Goal



Similarity, discrimination

Extension of the set



Representation

Structure of the Reality

Symbol

Operating on the Representation

Tree



Any cognitive representation for long, vertical, usually green with a wood basis

The tallest tree

Some Goal



Similarity, discrimination



Representation

Structure of the Reality

Symbol

Tree



Any cognitive representation for long, vertical, usually green with a wood basis

Operating on the Representation

Definite article: “get me one”

The tallest tree

Some Goal
Similarity, discrimination



Representation

Structure of the Reality

Symbol

Tree



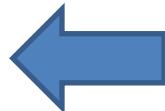
Any cognitive representation for long, vertical, usually green with a wood basis

Operating on the Representation

Superlative adjective : "top most"

The tallest tree

Some Goal



Similarity, discrimination



Structure of the Reality

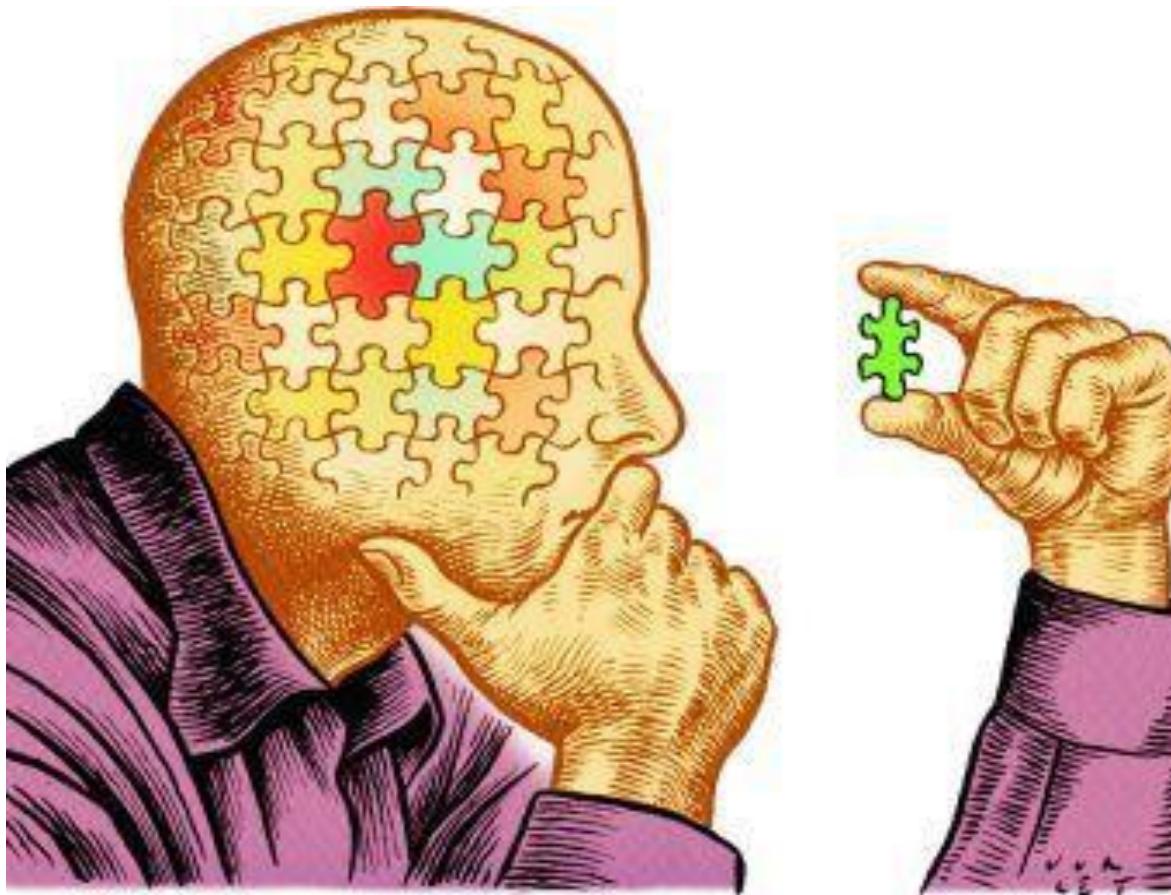
Natural Language Representation

- There is a mapping between natural language and knowledge representation.
- Looking at natural language is looking at the representation (constrained by the communication medium).

Language as a ‘Geological outcrop’ of our cognitive representation.



https://www.geocaching.com/geocache/GC3YMK9_one-day-geology-of-oman-1?guid=c3516272-9eca-4c10-ae14-8dabd9346b98



Knowledge in Learning and Inference

Inductive Logic Programming

- ILP algorithms are constructive induction algorithms
 - Able to create new predicates to facilitate the expression of explanatory hypotheses

- Express Grandparent
 - Empty background
 - Hypotheses are long and complicated

$\text{Grandparent}(x, y) \Leftrightarrow$

$$\begin{aligned} & [\exists z \text{ Mother}(x, z) \wedge \text{Mother}(z, y)] \\ \vee & [\exists z \text{ Mother}(x, z) \wedge \text{Father}(z, y)] \\ \vee & [\exists z \text{ Father}(x, z) \wedge \text{Mother}(z, y)] \\ \vee & [\exists z \text{ Father}(x, z) \wedge \text{Father}(z, y)] \end{aligned}$$

Chu

Inductive Logic Programming

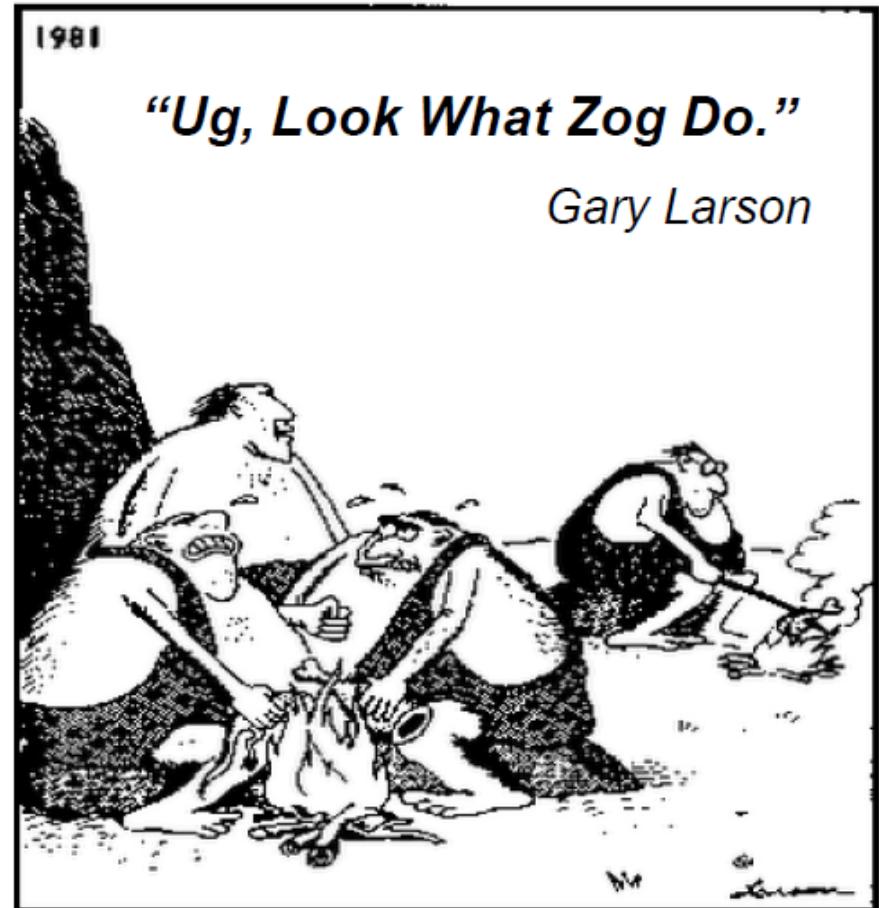
- By creating a new predicate, the definition of Grandparent can be reduced

$$\text{Parent}(x, y) \Leftrightarrow [\text{Mother}(x, y) \vee \text{Father}(x, y)]$$
$$\text{Grandparent}(x, y) \Leftrightarrow [\exists z \text{ Parent}(x, z) \wedge \text{Parent}(z, y)]$$

- Background knowledge can reduce the size of hypotheses required to explain the observations

Explanation-based Learning

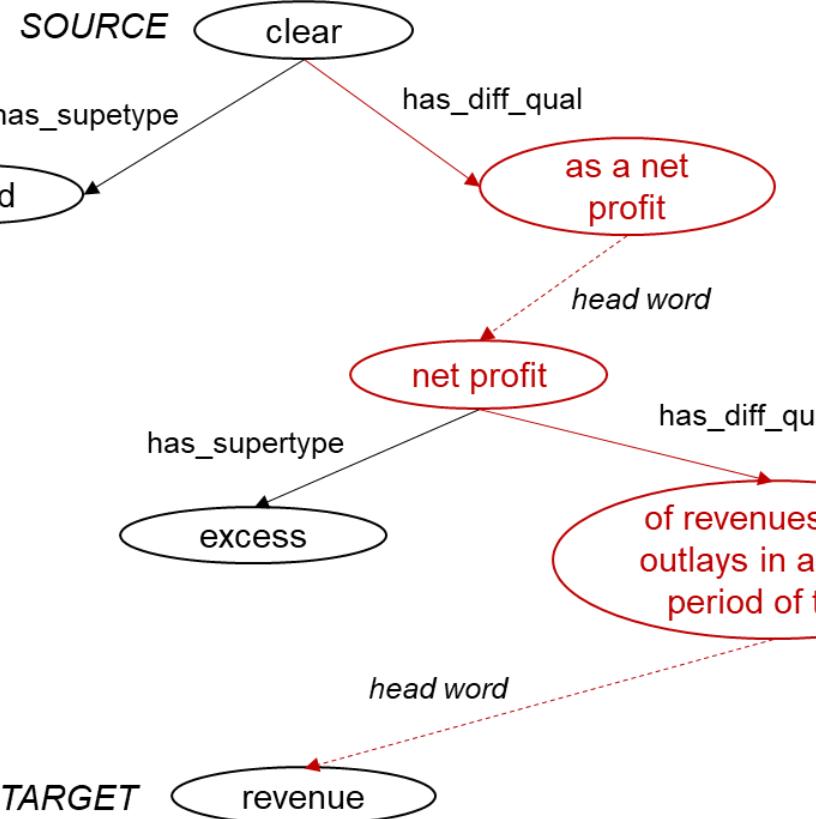
- **Explanation-based Learning (EBL)**
 - Method for extracting rules from individual observations through an explanation.
- **Explanation**
 - Stick holds the food over the fire while keeping hands safe.
- **Generalization**
 - Any long, rigid, sharp object can be used to toast food over the fire.
 - General rule follows logically from the background knowledge of the cavemen's usual cooking process.



Chu

Natural Language Inference

T: IBM **cleared** \$18.2 billion in the first quarter.
H: IBM's **revenue** in the first quarter was \$18.2 billion.



Entailment?

YES

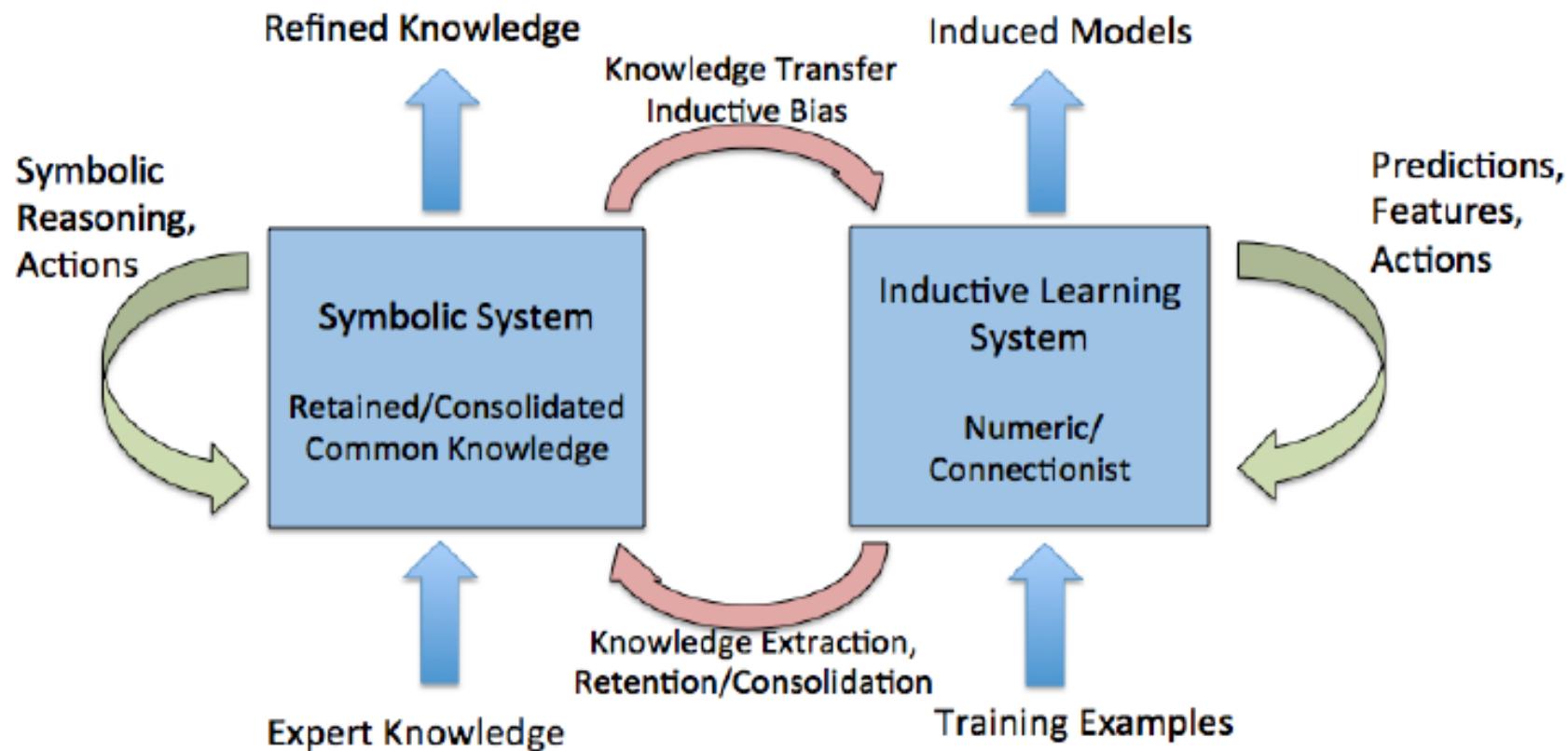
Why?

- To clear is to yield as a net profit
- A net profit is an excess of revenues over outlays in a given period of time

Natural Language Inference

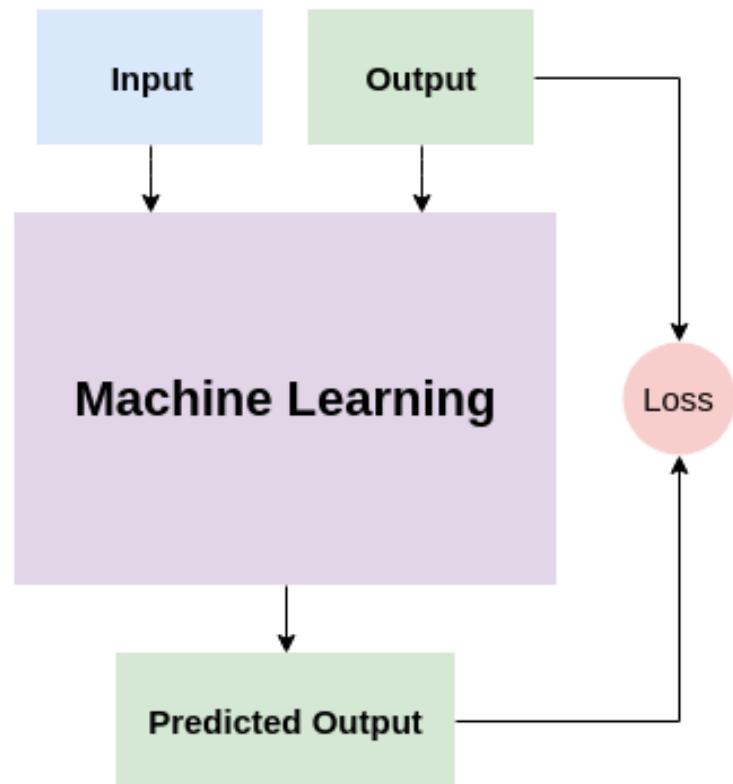
diagram	symbol	name	example
	$x \equiv y$	equivalence	<i>couch</i> \equiv <i>sofa</i>
	$x \sqsubset y$	forward entailment (strict)	<i>crow</i> \sqsubset <i>bird</i>
	$x \sqsupset y$	reverse entailment (strict)	<i>European</i> \sqsupset <i>French</i>
	$x \wedge y$	negation (exhaustive exclusion)	<i>human</i> \wedge <i>nonhuman</i>
	$x \mid y$	alternation (non-exhaustive exclusion)	<i>cat</i> \mid <i>dog</i>
	$x \square y$	cover (exhaustive non-exclusion)	<i>animal</i> \square <i>nonhuman</i>
	$x \# y$	independence	<i>hungry</i> $\#$ <i>hippo</i>

Neuro-Symbolic Models

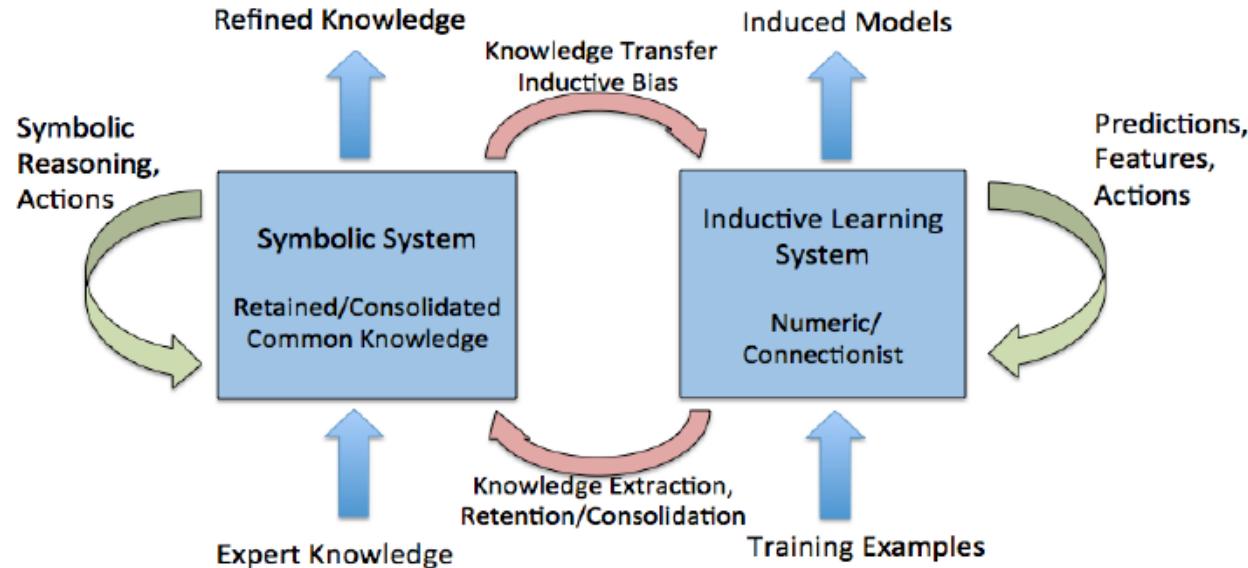


Current Limits of Deep Learning

1. Deep learning thus far is data hungry
2. Deep learning thus far is shallow and has limited capacity for transfer
3. Deep learning thus far has no natural way to deal with hierarchical structure
4. Deep learning thus far is not sufficiently transparent
5. Deep learning thus far has not been well integrated with prior knowledge
6. Deep learning thus far cannot inherently distinguish causation from correlation
7. Deep learning presumes a largely stable world, in ways that may be problematic
8. Deep learning thus far works well as an approximation, but its answers often cannot be fully trusted
9. Deep learning thus far is difficult to engineer with



Statistical vs Symbolic AI Systems



	Statistical	Symbolic
Explainability	Hard	Easy
Generalizing algebraic operations	Hard	Easy
Robustness to noise	Easy	Hard
Robustness to ambiguity	Easy	Hard
Robustness to mislabeling	Easy	Hard

∂ ILP

It is possible for systems to combine statistical perceptual with conceptual interpretable reasoning!

	Deep Learning	Symbolic Program Synthesis	∂ ILP
Robust to noise	YES	NO	YES
Can learn from non-symbolic data	YES	NO	YES
Data efficient	NO	YES	YES
Interpretable	NO	YES	YES

Gated Graph Neural Networks

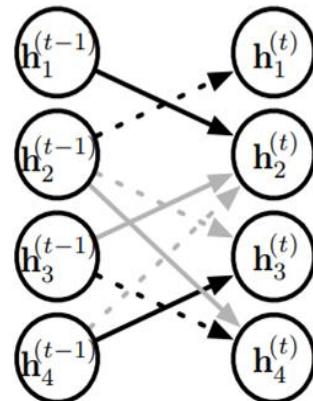
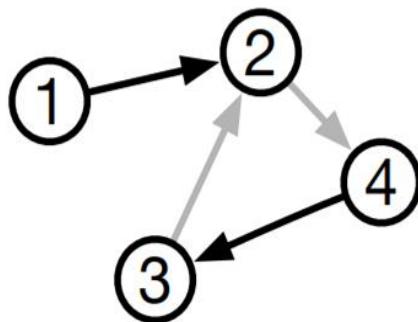
GGNNs are a neural network architecture defined according to a graph structure $G = (V, E)$

Nodes $v \in V$ take unique values from $1, \dots, |V|$, and edges are pairs $e = (v, v_0) \in V \times V$

GGNNs map graphs to outputs via two steps. First, there is a **propagation step** that computes node representations for each node; second, an **output model** $ov = g(hv, lv)$ maps from node representations and corresponding labels to an output ov for each $v \in V$

The propagation model is similar to an LSTM. Each node in the graph v has a hidden state representation $h(t)v$ that is updated at every time step t . The computation starts at $t = 0$ with initial hidden states xv that depends on the problem.

The structure of the graph, encoded in a matrix A serves to retrieve the hidden states of adjacent nodes based on the edge types between them. The hidden states are then updated by a gated update module.



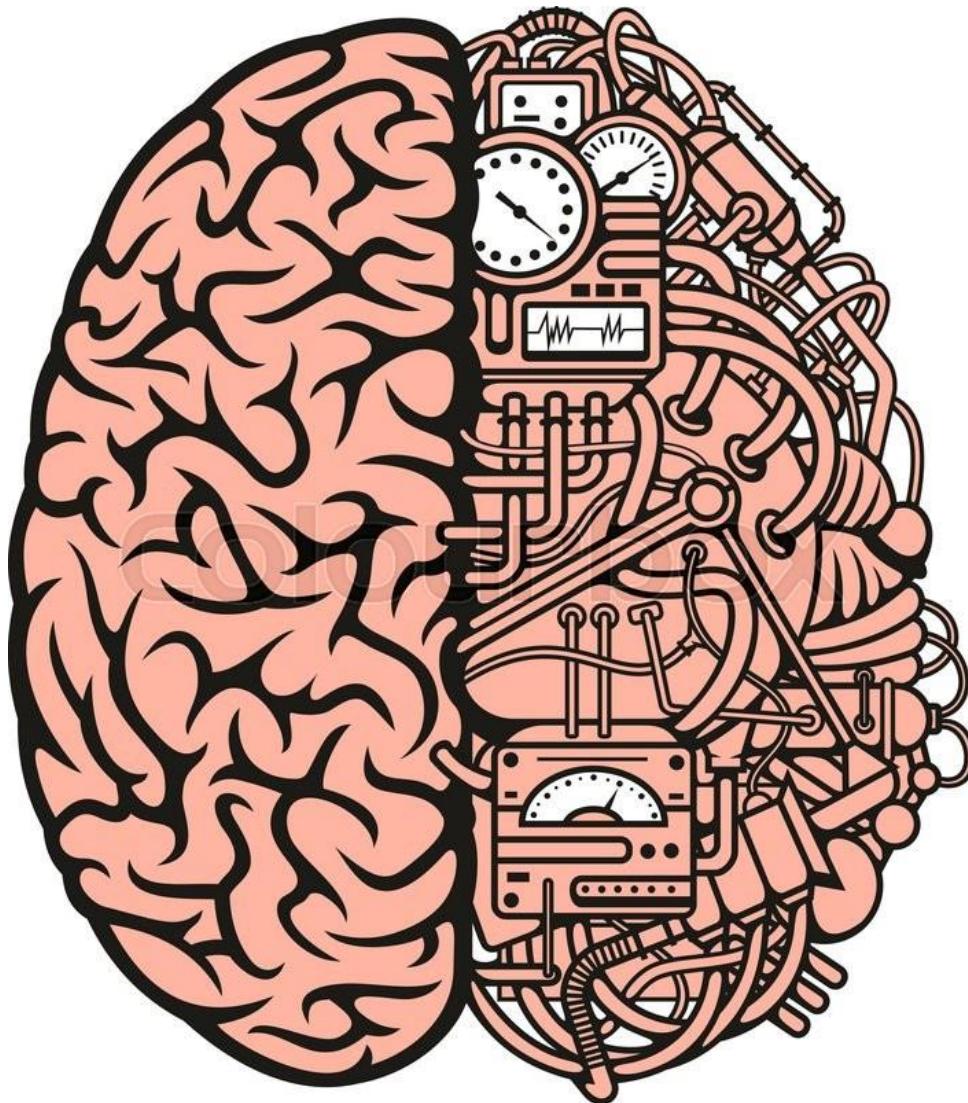
	Outgoing Edges				Incoming Edges			
	1	2	3	4	1	2	3	4
1	B							
2		C	B'				C'	
3	C							B'
4		B			C'			

(a)

(b)

(c) $\mathbf{A} = [\mathbf{A}^{(\text{out})}, \mathbf{A}^{(\text{in})}]$

Building Neuro-Symbolic Systems



Explainable QA

Q: The Pollution Prevention Act of 1990 expanded a publicly available or private database?

A: publicly available

Explanation:

Paragraph A: Pollution Prevention Act of 1990

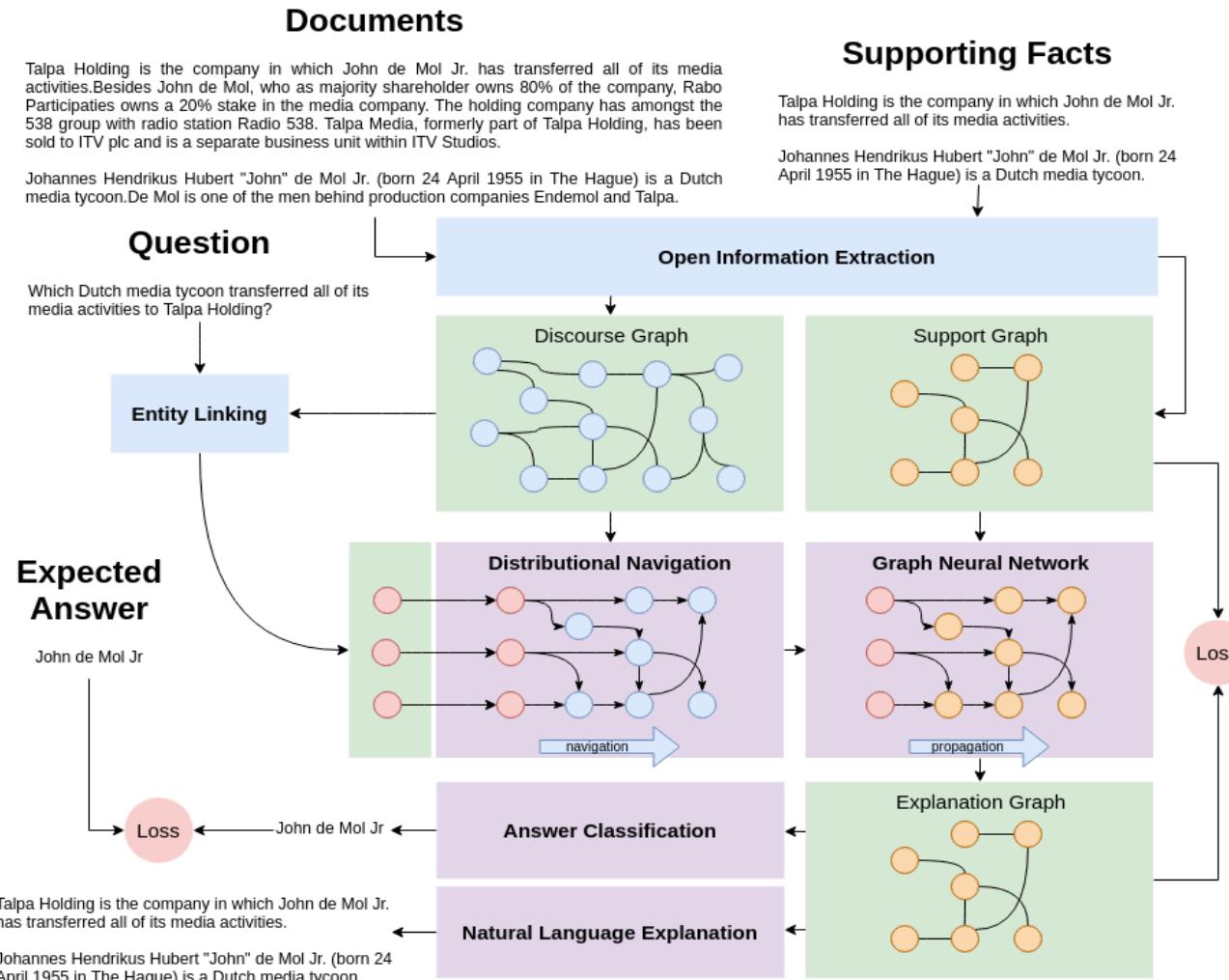
The Pollution Prevention Act of 1990 (PPA) in the United States created a national policy to have pollution prevented or reduced at the source wherever possible. It also expanded the Toxics Release Inventory.

Paragraph B: Toxics Release Inventory

The Toxics Release Inventory (TRI) is a publicly available database containing information on toxic chemical releases and other waste management activities in the United States.

Explainable QA

Neuro-Symbolic =
Knowledge Graphs +
Gated Graph Neural
Networks (GGNN)



Explainable Science QA

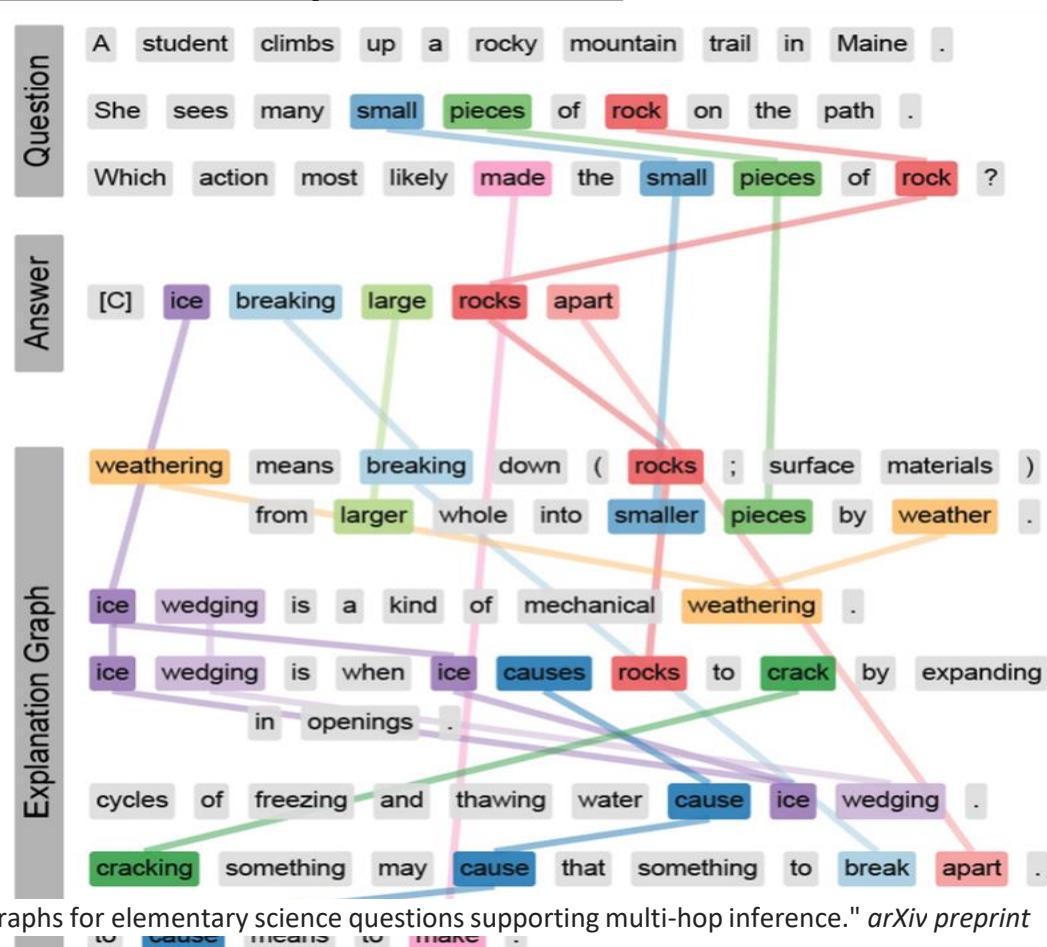
Q: A student climbs up a rocky mountain trail in Maine. She sees many small pieces of rock on the path. Which action most likely made the small pieces of rock?

- [0]: sand blowing into cracks
- [1]: leaves pressing down tightly
- [2]: ice breaking large rocks apart
- [3]: shells and bones sticking together

A: ice breaking large rocks apart

Explanation:

- weathering means breaking down (rocks) by weather
- ice wedging is a kind of mechanical weathering
- ice wedging is when ice causes rocks to crack
- cycles of freezing and thawing water cause ice wedging
- cracking something may cause that something to break
- to cause means to make





Powered by Thomson Reuters

[News Feed](#) [Connection](#) [Portfolio](#)

[Settings](#) [All](#) [Pinned](#)

3 number of pinned articles

[BRIEF-Avio receives EUR 40 mln financing from European Investment Bank <SPA2.MI>](#)

Companies: Avio SPA 80%, Avio SPA 80%, Avio SPA 80%

Topics: Business Finance, Contracts / Business Deals Events: ContactDetails

Industry: Spacecraft Manufacturing

Publication date: Oct 6, 2017 10:24:02 AM

Airbus SE

BAE Systems PLC

Boeing Co

[BRIEF-Airtelis orders three H215 Airbus Helicopters<AIR.PA><AIRG.DE>](#)

Companies: Airbus Helicopters SAS 50%, Airbus Helicopters SAS 50%, Airbus Helicopters SAS 50%

Topics: N/A Events: BusinessRelation

Industry: Aerospace & Defense - NEC, Aircraft Parts Manufacturing - NEC

Publication date: Oct 4, 2017 10:46:00 AM

Raytheon Co

BAE Systems PLC

Boeing Co

[BRIEF-British Airline Pilots' Association - pilots union calls for investigation into collapse of Monarch Airlines<MONA.UL>](#)

Companies: Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%

Topics: Other Events: N/A

Industry: Airlines - NEC

Publication date: Oct 9, 2017 1:55:13 PM

Raytheon Co

BAE Systems PLC

Boeing Co



THOMSON REUTERS

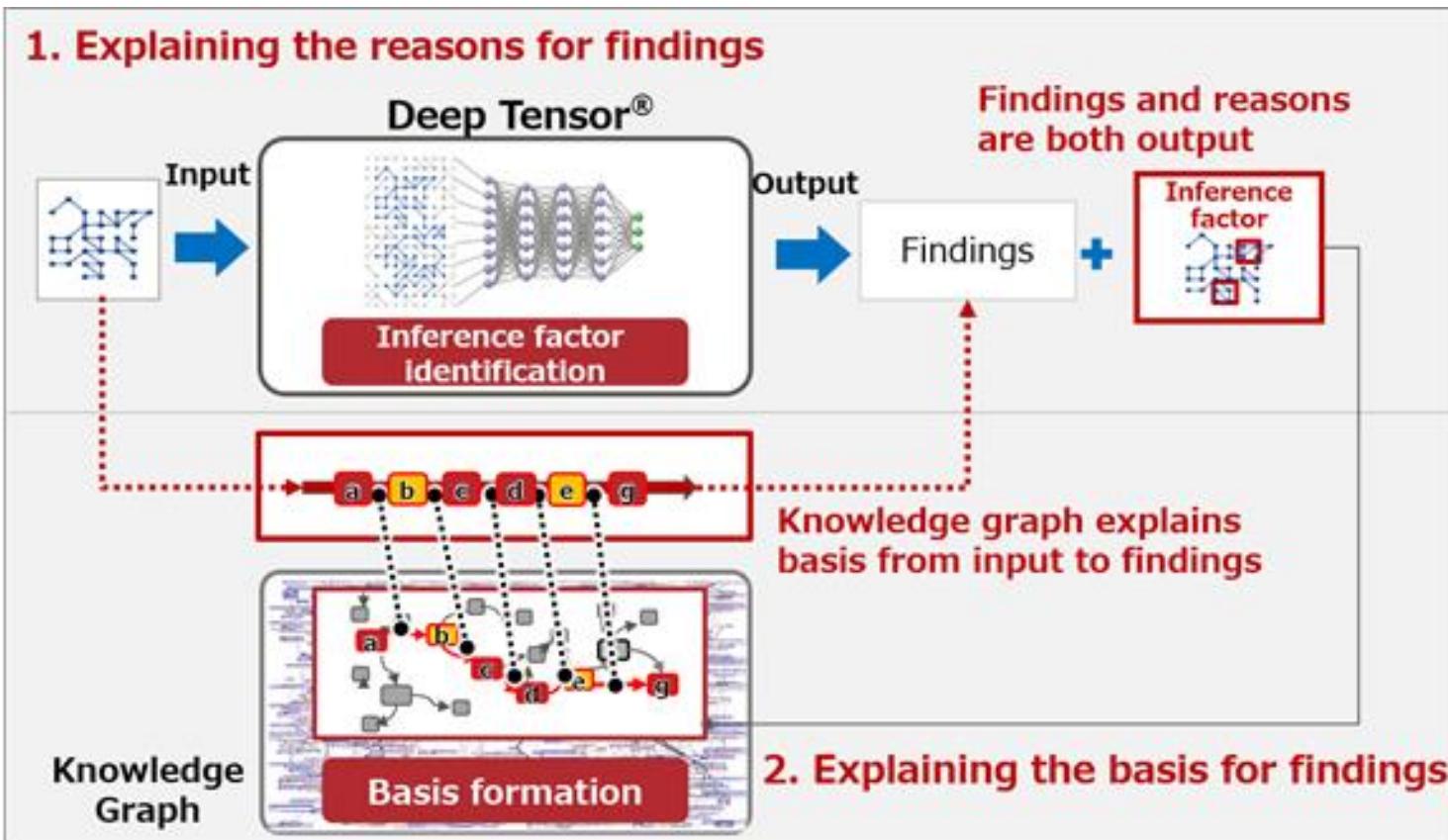
BRIEF-Avio receives EUR 40 mn financing from European Investment Bank <SPA2.MI>

Oct 6 (Reuters) - AVIO SPA <SPA2.MI> * SAYS SIGNED WITH EUROPEAN INVESTMENT BANK CONTRACT FOR EUR 40 MILLION FINANCING Source text for Eikon: [ID:nBIA5D9Ntk] Further company coverage: [SPA2.MI] (Gdynia Newsroom) (gdynia.newsroom@thomsonreuters.com; +48 58 772 0920 ;)



THOMSON REUTERS

Explainable Findings From Tensor Inferences Back to KGs



Summary of Today

- End-to-end overview of this part of the course and its underlying motivation.
- Representation, semantics, learning/inference.
- Representation and NL.
- Dialogue between Statistical and Symbolic approaches.

Next Class

- We will jump directly into Knowledge Representation (beyond FOL).
- Frames, Prototypes, Ontologies.
- Representing more complex NL discourse.

Recommended Reading

Deep Learning: A Critical Appraisal

Gary Marcus¹
New York University

<https://arxiv.org/ftp/arxiv/papers/1801/1801.00631.pdf>

Symbolic AI

Andre Freitas



Photo by Vasilyev Alexandr

Today

- Knowledge Bases for supporting AI Systems.
- Knowledge Representation paradigms for KBs.



Powered by Thomson Reuters

[News Feed](#) [Connection](#) [Portfolio](#)

[Settings](#) [All](#) [Pinned](#)

3 number of pinned articles

[BRIEF-Avio receives EUR 40 mln financing from European Investment Bank <SPA2.MI>](#)

Companies: Avio SPA 80%, Avio SPA 80%, Avio SPA 80%

Topics: Business Finance, Contracts / Business Deals Events: ContactDetails

Industry: Spacecraft Manufacturing

Publication date: Oct 6, 2017 10:24:02 AM

Airbus SE

BAE Systems PLC

Boeing Co

[BRIEF-Airtelis orders three H215 Airbus Helicopters<AIR.PA><AIRG.DE>](#)

Companies: Airbus Helicopters SAS 50%, Airbus Helicopters SAS 50%, Airbus Helicopters SAS 50%

Topics: N/A Events: BusinessRelation

Industry: Aerospace & Defense - NEC, Aircraft Parts Manufacturing - NEC

Publication date: Oct 4, 2017 10:46:00 AM

Raytheon Co

BAE Systems PLC

Boeing Co

[BRIEF-British Airline Pilots' Association - pilots union calls for investigation into collapse of Monarch Airlines<MONA.UL>](#)

Companies: Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%

Topics: Other Events: N/A

Industry: Airlines - NEC

Publication date: Oct 9, 2017 1:55:13 PM

Raytheon Co

BAE Systems PLC

Boeing Co



THOMSON REUTERS

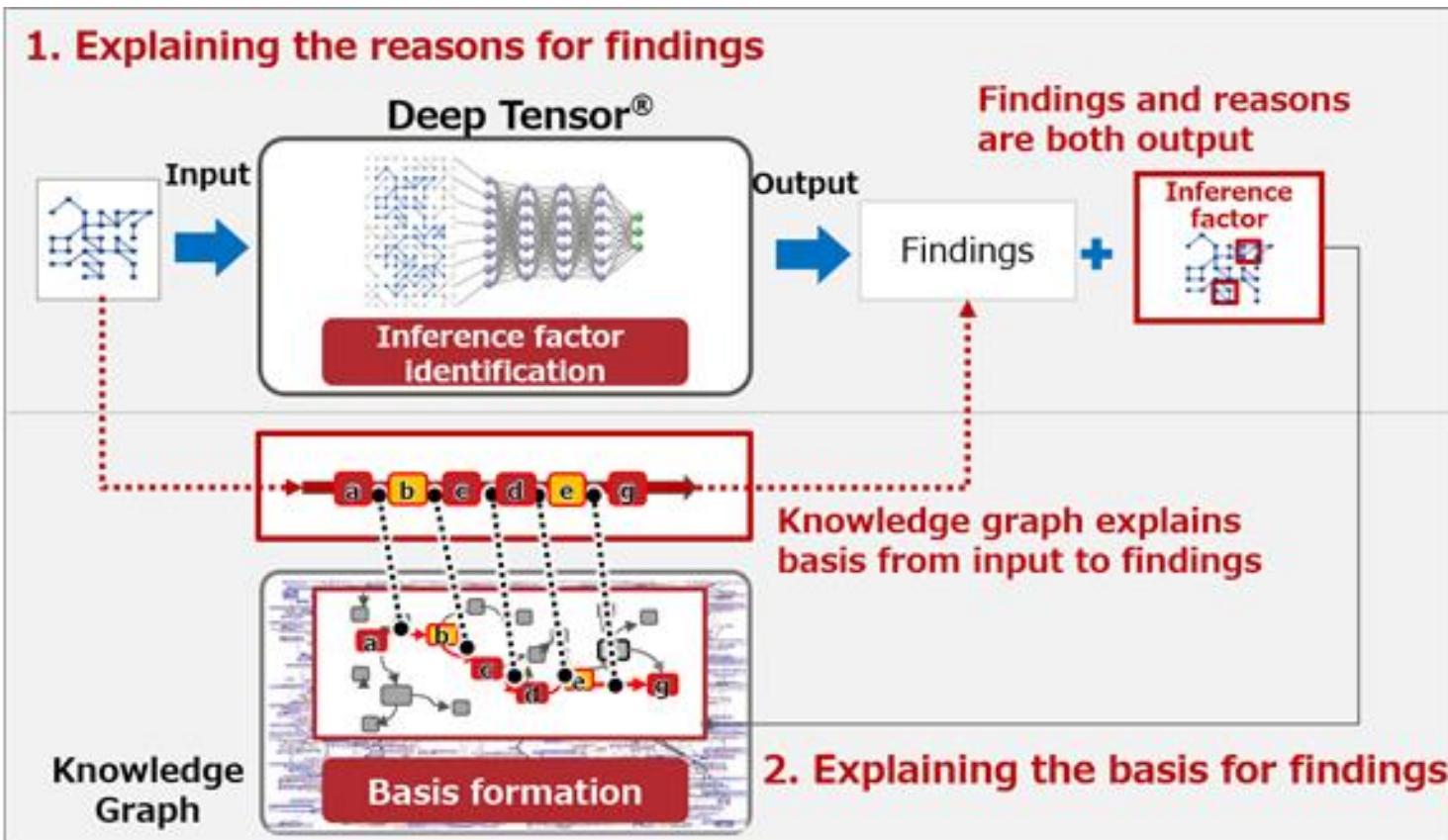
BRIEF-Avio receives EUR 40 mn financing from European Investment Bank <SPA2.MI>

Oct 6 (Reuters) - AVIO SPA <SPA2.MI> * SAYS SIGNED WITH EUROPEAN INVESTMENT BANK CONTRACT FOR EUR 40 MILLION FINANCING Source text for Eikon: [ID:nBIA5D9Ntk] Further company coverage: [SPA2.MI] (Gdynia Newsroom) (gdynia.newsroom@thomsonreuters.com; +48 58 772 0920 ;)



THOMSON REUTERS

Explainable Findings From Tensor Inferences Back to KGs



Machine Knowledge for Answer Engines

Weikum, 2019

Precise and concise answers
for advanced information needs:



properties of entity

- ★ Nobel laureate whose daughter also won a Nobel prize?



sets of entities

- ★ Pop singers who are also poets?

relationships between entities

- ★ Commonalities & relationships among:
Alan Turing, Paul Bocuse, Steve Jobs, Katherine Goble?



Machine Knowledge for Answer Engines

Weikum, 2019

**Precise and concise answers
for advanced information needs:**

real applications

- ★ Proteins that bind to the Zika virus?
- Polymer materials for super-capacitors?
- European politicians mentioned in Panama Leaks?

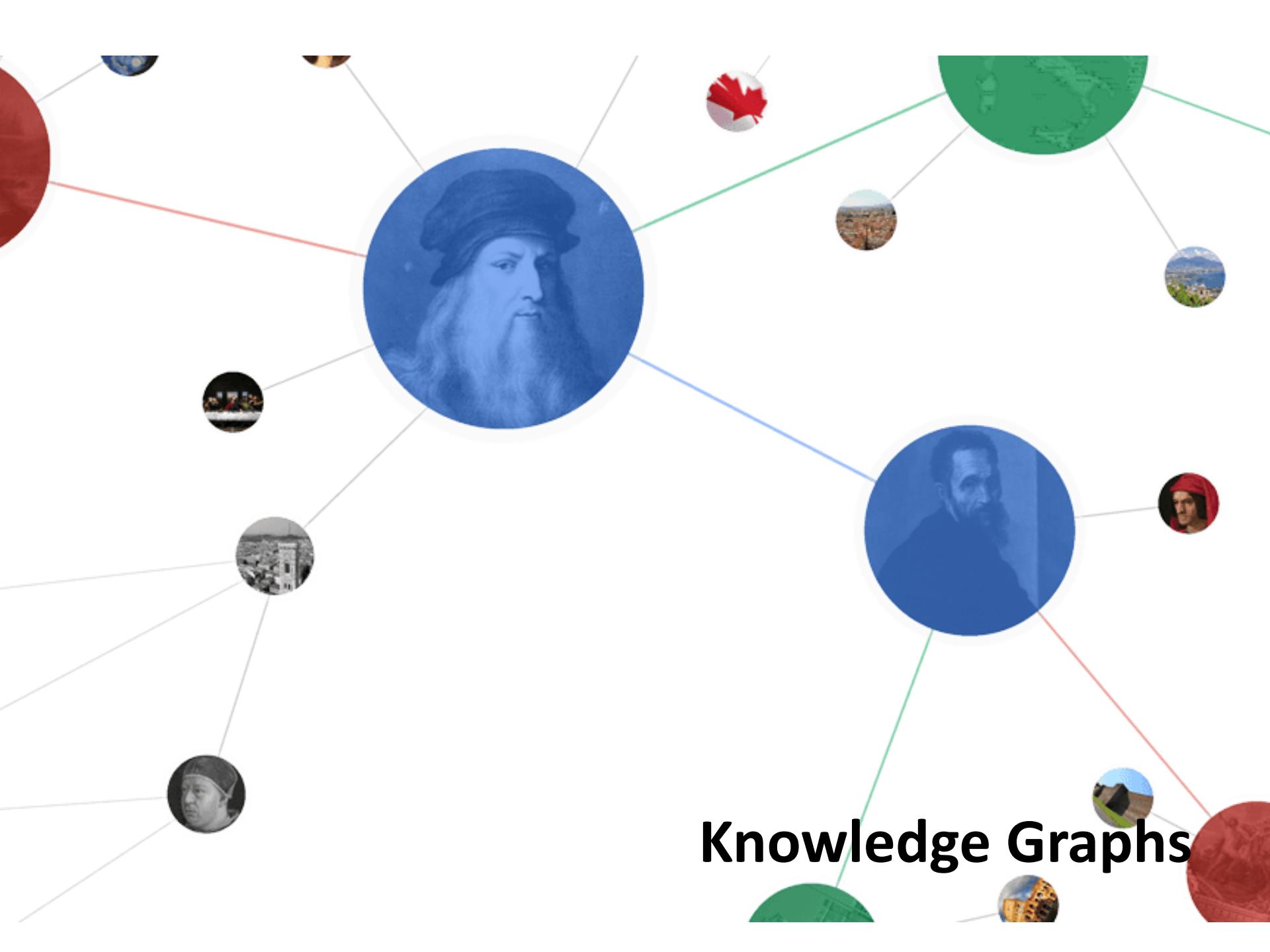
Representation

- **Representation**: organization of a perceptual/symbolic space into an abstraction
 - Attention/selection
 - Intent/goal
- Maximization of Inferential Locality
- Abstraction for Purpose
- Correctness/Completeness for Purpose

Good Knowledge Representation Languages

- Combines the best of natural and formal languages:
 - expressive
 - concise
 - unambiguous
 - independent of context
 - what you say today will still be interpretable tomorrow
 - efficient
 - the knowledge can be represented in a format that is suitable for computers
 - operational
 - there is an inference procedure which can act on it to make new sentences

Knowledge Graphs



Brief History of Knowledge Bases

Weikum, 2019



Vannevar Bush



Cyc



WordNet



Wikipedia



Cyc



WordNet



Terminology

- Ontology
 - provides more complete definitions for concepts
 - Graphical conceptual model
- Thesaurus
 - simple relationships between words
 - synonyms, homonyms, antonyms, etc.
 - often combined with a taxonomy
- Taxonomy
 - hierarchical arrangement of concepts
 - often used as a “backbone” for an ontology
- Lexicon
 - provides natural language descriptions of words and their meanings

Word Senses & Relations

Reminder: lemma and wordform

- A lemma or citation form
 - Same stem, part of speech, rough semantics
- A word form
 - The “inflected” word as it appears in text

Word form	Lemma
banks	bank
sung	sing
duermes	dormir

Lemmas have senses

- One lemma “bank” can have many meanings:

- Sense 1: • ...a **bank** can hold the investments in a custodial account...
 1
- Sense 2: • “...as agriculture burgeons on the east **bank** the river will shrink
 even more”
 2

- Sense (or word sense)
 - A discrete representation
of an aspect of a word’s meaning.
- The lemma **bank** here has two senses

Polysemy

- 1. The **bank** was constructed in 1875 out of local red brick.
- 2. I withdrew the money from the **bank**.
- Are those the same sense?
 - Sense 2: “A financial institution”
 - Sense 1: “The building belonging to a financial institution”
- A polysemous word has **related** meanings
 - Most non-rare words have multiple meanings.

How do we know when a word has more than one sense?

- The “zeugma” test: Two senses of **serve**?
 - Which flights **serve** breakfast?
 - Does Lufthansa **serve** Philadelphia?
 - ?Does Lufthansa serve breakfast and San Jose?
- Since this conjunction sounds weird,
 - we say that these are **two different senses of “serve”**

Synonyms

- Word that have the same meaning in some or all contexts:
 - couch / sofa
 - big / large
 - automobile / car
 - vomit / throw up
 - Water / H₂O
- Two lexemes are synonyms:
 - if they can be substituted for each other in all situations.
 - If so they have the same **propositional meaning.**

Synonymy is a relation between senses rather than words

- Consider the words *big* and *large*.
- Are they synonyms?
 - How **big** is that plane?
 - Would I be flying on a **large** or small plane?
- How about here:
 - Miss Nelson became a kind of **big** sister to Benjamin.
 - ?Miss Nelson became a kind of **large** sister to Benjamin.
- Why?
 - *big* has a sense that means being older, or grown up.
 - *large* lacks this sense.

Antonyms

- Senses that are opposites with respect to one feature of meaning.
- Otherwise, they are very similar!

dark/light short/long fast/slow rise/fall

hot/cold up/down in/out

Hyponymy and Hypernymy

- One sense is a hyponym of another if the first sense is more specific, denoting a subclass of the other:
 - *car* is a hyponym of *vehicle*
 - *mango* is a hyponym of *fruit*
- Conversely hypernym/superordinate (“hyper is super”):
 - *vehicle* is a hypernym of *car*
 - *fruit* is a hypernym of *mango*

Superordinate/hyper	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

Hyponymy more formally

- Extensional:
 - The class denoted by the superordinate extensionally includes the class denoted by the hyponym.
- Hyponymy is usually transitive
 - (A hypo B and B hypo C entails A hypo C).
- Another name: the IS-A hierarchy:
 - A IS-A B (or A ISA B)
 - B **subsumes** A

Hyponyms and Instances

- WordNet has both classes and instances.
- An instance is an individual, a proper noun that is a unique entity:
 - San Francisco is an **instance** of city
 - But city is a class
 - city is a **hyponym** of municipality...location...

WordNet & Other Online Thesauri

Senses of “bass” in Wordnet

Noun

- S: (n) **bass** (the lowest part of the musical range)
- S: (n) **bass**, **bass part** (the lowest part in polyphonic music)
- S: (n) **bass**, **basso** (an adult male singer with the lowest voice)
- S: (n) **sea bass**, **bass** (the lean flesh of a saltwater fish of the family Serranidae)
- S: (n) **freshwater bass**, **bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- S: (n) **bass**, **bass voice**, **basso** (the lowest adult male singing voice)
- S: (n) **bass** (the member with the lowest range of a family of musical instruments)
- S: (n) **bass** (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Adjective

- S: (adj) **bass**, **deep** (having or denoting a low vocal or instrumental range) "a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"

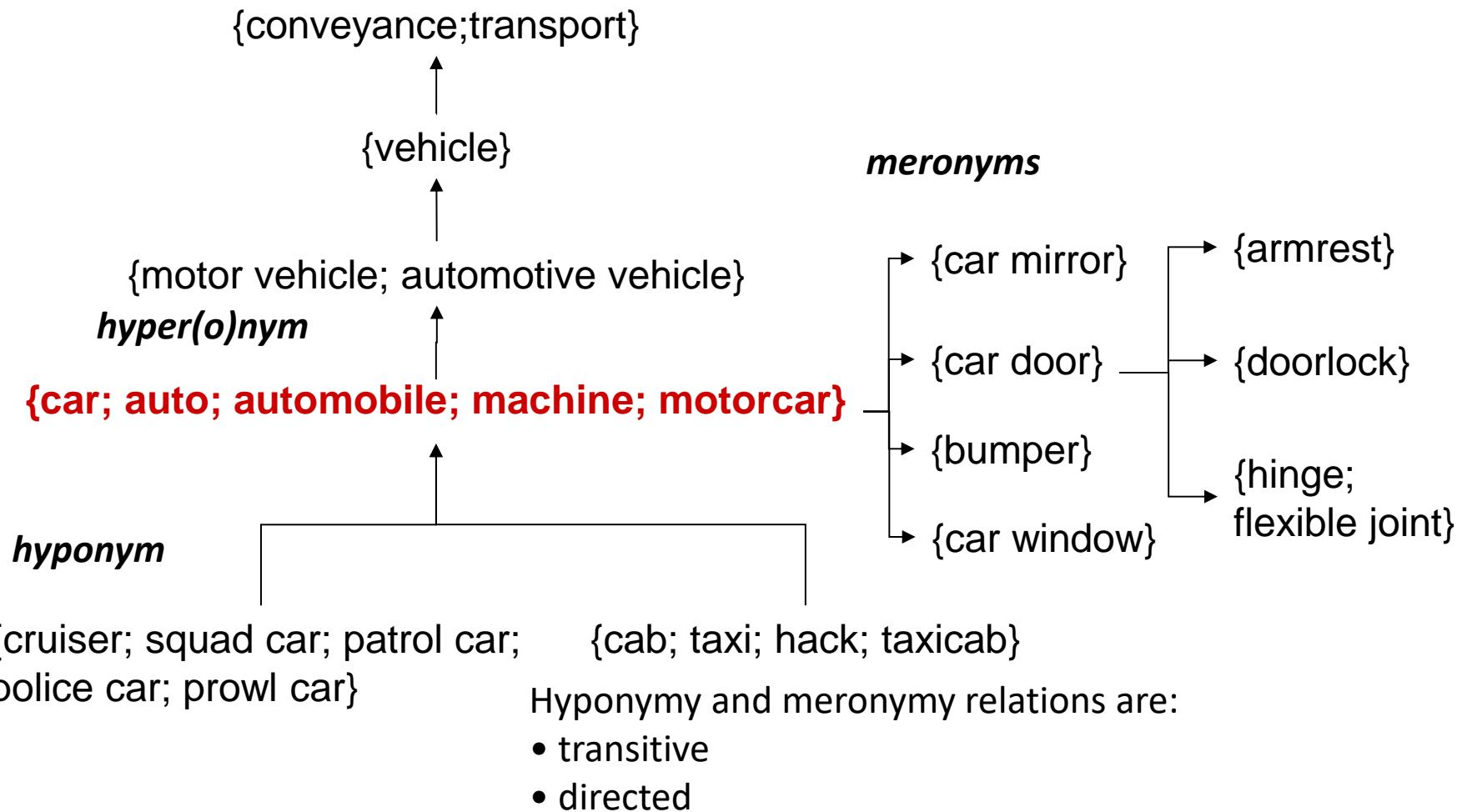
How is “sense” defined in WordNet?

- The synset (synonym set), the set of near-synonyms, instantiates a sense or concept, with a gloss.
- Example: chump as a noun with the gloss:
“a person who is gullible and easy to take advantage of”
- This sense of “chump” is shared by 9 words:
chump¹, fool², gull¹, mark⁹, patsy¹, fall guy¹, sucker¹, soft touch¹, mug²

WordNet Hypernym Hierarchy for “bass”

- S: (n) bass, basso (an adult male singer with the lowest voice)
 - direct hypernym / inherited hypernym / sister term
 - S: (n) singer, vocalist, vocalizer, vocaliser (a person who sings)
 - S: (n) musician, instrumentalist, player (someone who plays a musical instrument (as a profession))
 - S: (n) performer, performing artist (an entertainer who performs a dramatic or musical work for an audience)
 - S: (n) entertainer (a person who tries to please or amuse)
 - S: (n) person, individual, someone, somebody, mortal, soul (a human being) "there was too much for one person to do"
 - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
 - S: (n) living thing, animate thing (a living (or once living) entity)
 - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) "how big is that part compared to the whole?"; "the team is a unit"
 - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) "it was full of rackets, balls and other objects"
 - S: (n) physical entity (an entity that has physical existence)
 - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Wordnet: a network of semantically related words



Wordnet Semantic Relations

WN 1.5 starting point

The ‘synset’ as a weak notion of synonymy:

“two expressions are synonymous in a linguistic context C if the substitution of one for the other in C does not alter the truth value.” (Miller et al. 1993)

Relations between synsets:

HYPONYMY	noun-to-noun	car/ vehicle
	verb-to-verb	walk/ move
MERONYMY	noun-to-noun	head/ nose
ANTONYMY	adjective-to-adjective	good/bad
	verb-to-verb	open/ close
ENTAILMENT	verb-to-verb	buy/ pay
CAUSE	verb-to-verb	kill/ die

Some observations on Wordnet

- Synsets are more compact representations for concepts than word meanings in traditional lexicons.
- Synonyms and hypernyms are substitutional variants:
 - begin – commence
 - I once had a **canary**. The **bird** got sick. The poor **animal** died.
- Hyponymy and meronymy chains are important transitive relations for predicting properties and explaining textual properties:
object -> artifact -> vehicle -> 4-wheeled vehicle -> car
- Strict separation of part of speech (PoS) although concepts are closely related (**bed – sleep**) and are similar (**dead – death**).

PoS (Part-of-Speech)

The Penn Treebank POS tagset.

1. CC	Coordinating conjunction	25. TO	<i>to</i>
2. CD	Cardinal number	26. UH	Interjection
3. DT	Determiner	27. VB	Verb, base form
4. EX	Existential <i>there</i>	28. VBD	Verb, past tense
5. FW	Foreign word	29. VBG	Verb, gerund/present participle
6. IN	Preposition/subordinating conjunction	30. VBN	Verb, past participle
7. JJ	Adjective	31. VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective, comparative	32. VBZ	Verb, 3rd ps. sing. present
9. JJS	Adjective, superlative	33. WDT	<i>wh</i> -determiner
10. LS	List item marker	34. WP	<i>wh</i> -pronoun
11. MD	Modal	35. WP\$	Possessive <i>wh</i> -pronoun
12. NN	Noun, singular or mass	36. WRB	<i>wh</i> -adverb
13. NNS	Noun, plural	37. #	Pound sign
14. NNP	Proper noun, singular	38. \$	Dollar sign
15. NNPS	Proper noun, plural	39. .	Sentence-final punctuation
16. PDT	Predeterminer	40. ,	Comma
17. POS	Possessive ending	41. :	Colon, semi-colon
18. PRP	Personal pronoun	42. (Left bracket character
19. PP\$	Possessive pronoun	43.)	Right bracket character
20. RB	Adverb	44. "	Straight double quote
21. RBR	Adverb, comparative	45. '	Left open single quote
22. RBS	Adverb, superlative	46. "	Left open double quote
23. RP	Particle	47. '	Right close single quote
24. SYM	Symbol (mathematical or scientific)	48. "	Right close double quote

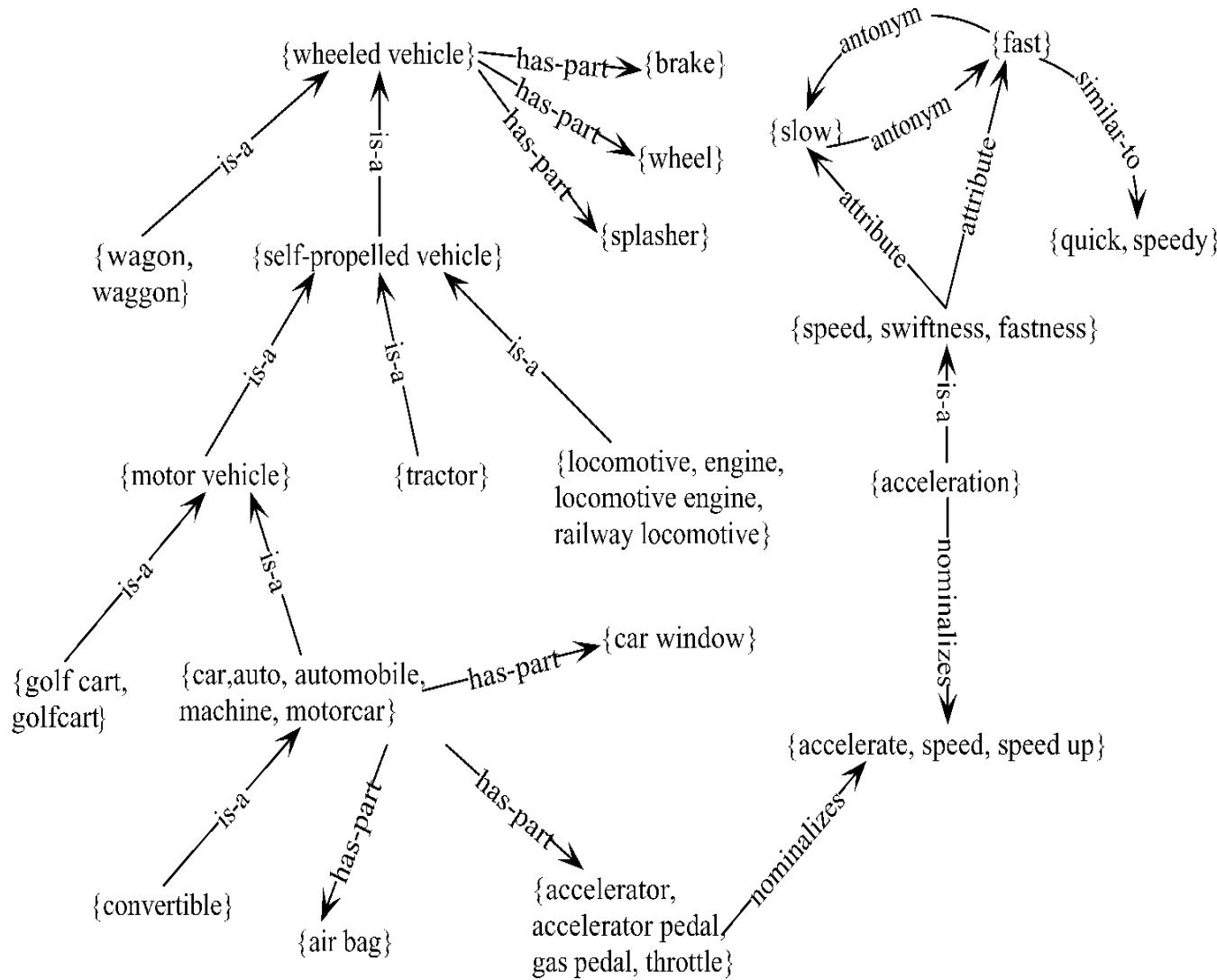
WordNet Noun Relations (Reference)

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	<i>breakfast</i> ¹ → <i>meal</i> ¹
Hyponym	Subordinate	From concepts to subtypes	<i>meal</i> ¹ → <i>lunch</i> ¹
Instance Hypernym	Instance	From instances to their concepts	<i>Austen</i> ¹ → <i>author</i> ¹
Instance Hyponym	Has-Instance	From concepts to concept instances	<i>composer</i> ¹ → <i>Bach</i> ¹
Member Meronym	Has-Member	From groups to their members	<i>faculty</i> ² → <i>professor</i> ¹
Member Holonym	Member-Of	From members to their groups	<i>copilot</i> ¹ → <i>crew</i> ¹
Part Meronym	Has-Part	From wholes to parts	<i>table</i> ² → <i>leg</i> ³
Part Holonym	Part-Of	From parts to wholes	<i>course</i> ⁷ → <i>meal</i> ¹
Substance Meronym		From substances to their subparts	<i>water</i> ¹ → <i>oxygen</i> ¹
Substance Holonym		From parts of substances to wholes	<i>gin</i> ¹ → <i>martini</i> ¹
Antonym		Semantic opposition between lemmas	<i>leader</i> ¹ ⇔ <i>follower</i> ¹
Derivationally		Lemmas w/same morphological root	<i>destruction</i> ¹ ⇔ <i>destroy</i> ¹
Related Form			

WordNet Verb Relations (Reference)

Relation	Definition	Example
Hypernym	From events to superordinate events	<i>fly</i> ⁹ → <i>travel</i> ⁵
Troponym	From events to subordinate event (often via specific manner)	<i>walk</i> ¹ → <i>stroll</i> ¹
Entails	From verbs (events) to the verbs (events) they entail	<i>snore</i> ¹ → <i>sleep</i> ¹
Antonym	Semantic opposition between lemmas	<i>increase</i> ¹ ⇔ <i>decrease</i> ¹
Derivationally	Lemmas with same morphological root	<i>destroy</i> ¹ ⇔ <i>destruction</i> ¹
Related Form		

WordNet: Viewed as a Graph



WordNet 3.0

- Where it is:
 - <http://wordnetweb.princeton.edu/perl/webwn>
- Libraries
 - Python: WordNet from NLTK
 - <http://www.nltk.org/Home>
 - Java:
 - JWNL, extJWNL on sourceforge

Extended WordNet (XWN)

- WordNet with syntactic and semantic annotations over its glosses.
- Contains logical forms and disambiguated glosses.
- XWN 2.0-1 is based on WordNet 2.0.

1. **excellent**, first-class, fantabulous -- (**of the highest quality**; "made an excellent speech"; "the school has excellent teachers"; "a first-class mind")

(TOP (S (NP (JJ **excellent**))
 (VP (VBZ **is**)
 (NP (NP (NN **something**))
 (PP (IN **of**)
 (NP (DT **the**) (JJS **highest**) (NN **quality**))))
 (.)))

excellent:JJ(x1) -> **of**:IN(x1, x2) **highest**:JJ(x1)
quality:NN(x1)

```
<wf pos="IN" >of</wf>
<wf pos="DT" >the</wf>
<wf pos="JJS" lemma="high" quality="silver" wnsn="1"
>highest</wf>
<wf pos="NN" lemma="quality" quality="normal" wnsn="2"
>quality</wf>
```

Accessing WordNet from Prolog

```
? - substance_of(water,X).
```

```
X = [tear|_G407]
```

```
? - has_substance(water,X).
```

```
X = [h2o|_G407]
```

```
?- part_of(leg,X).
```

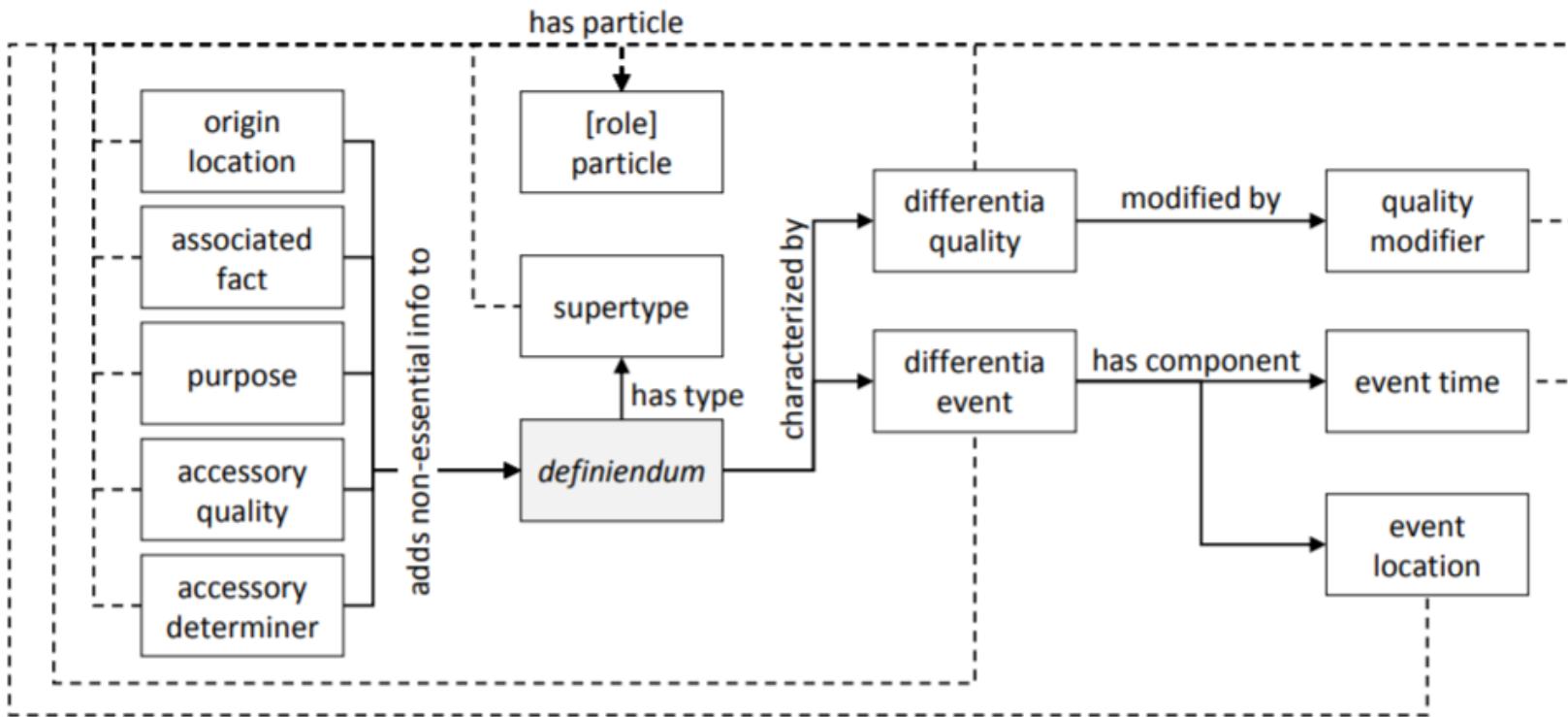
```
X = [table|_G407]
```

```
?- has_part(leg,X).
```

```
X = [knee|_G407]
```

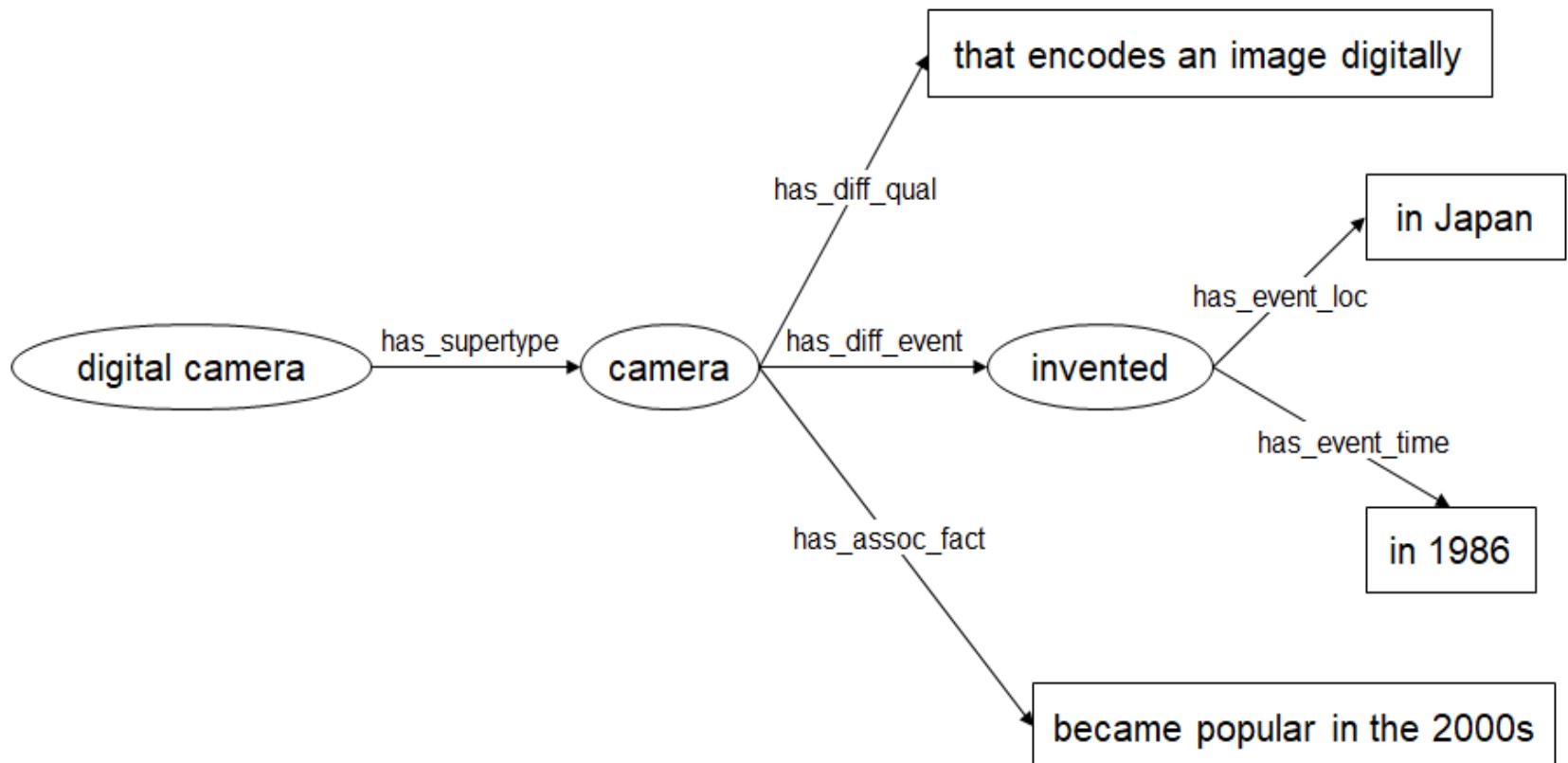
Semantic Roles for Lexical Definitions

Aristotle's classic theory of definition introduced important aspects such as the **genus-differentia definition pattern** and the **essential/non-essential property differentiation**.



Definition Graphs

digital camera: a camera invented in Japan in 1986 that encodes an image digitally and became popular in the 2000s



MeSH: Medical Subject Headings thesaurus from the National Library of Medicine

- **MeSH (Medical Subject Headings)**
 - 177,000 entry terms that correspond to 26,142 biomedical “headings”

Synset

- **Hemoglobins**

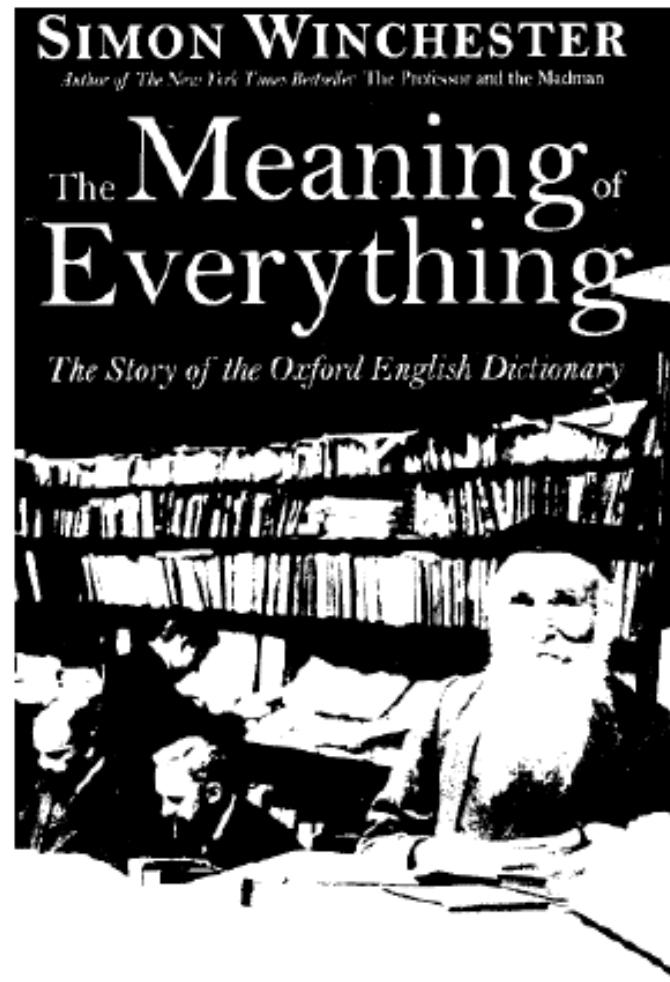
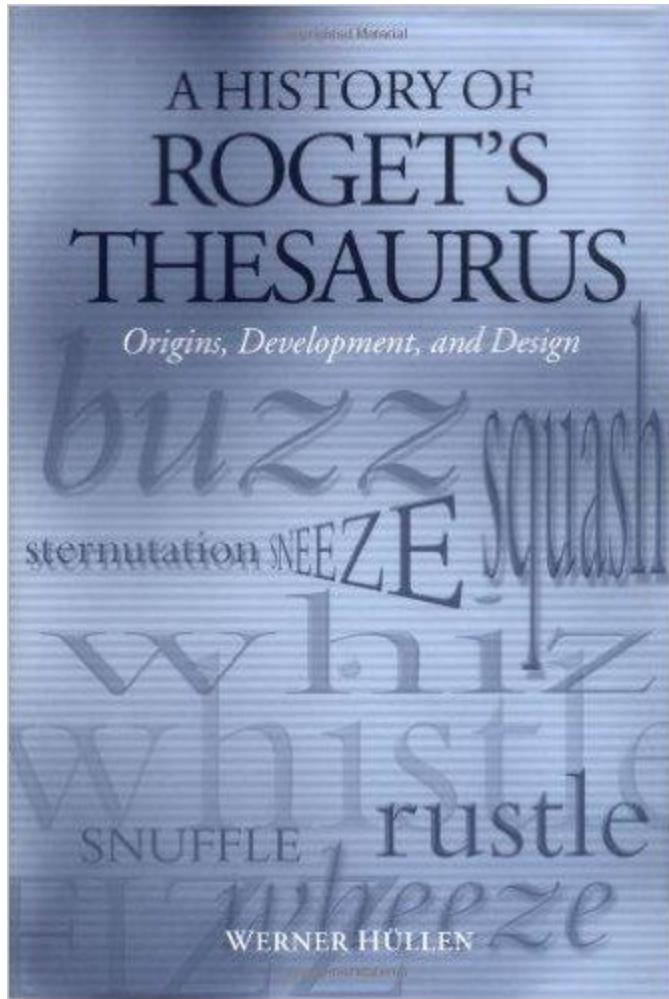
Entry Terms: Eryhem, Ferrous Hemoglobin, Hemoglobin

Definition: The oxygen-carrying proteins of ERYTHROCYTES. They are found in all vertebrates and some invertebrates. The number of globin subunits in the hemoglobin quaternary structure differs between species. Structures range from monomeric to a variety of multimeric arrangements

The MeSH Hierarchy

1. + Anatomy [A]
 2. + Organisms [B]
 3. + Diseases [C]
 4. - Chemicals and Drugs [D]
 - 1 o [Inorganic Chemicals \[D01\]](#) +
 - 2 o [Organic Chemicals \[D02\]](#) +
 - 3 o [Heterocyclic Compounds \[D03\]](#) +
 - 4 o [Polycyclic Compounds \[D04\]](#) +
 - 5 o [Macromolecular Substances \[D05\]](#) +
 - 6 o [Hormones, Hormone Substitutes, and](#)
 - 7 o [Enzymes and Coenzymes \[D08\]](#) +
 - 8 o [Carbohydrates \[D09\]](#) +
 - 9 o [Lipids \[D10\]](#) +
 - 10 o [Amino Acids, Peptides, and Proteins \[D11\]](#)
 - 11 o [Nucleic Acids, Nucleotides, and Nucleosides \[D12\]](#)
 - 12 o [Complex Mixtures \[D20\]](#) +
 - 13 o [Biological Factors \[D23\]](#) +
 - 14 o [Biomedical and Dental Materials \[D25\]](#)
 - 15 o [Pharmaceutical Preparations \[D26\]](#) +
 - 16 o [Chemical Actions and Uses \[D27\]](#) +
 5. + Analytical, Diagnostic and Therapeutic Techniques and Equipment [E]
 6. + Psychiatry and Psychology [F]
 7. + Phenomena and Processes [G]
- [Amino Acids, Peptides, and Proteins \[D12\]](#)
[Proteins \[D12.776\]](#)
[Blood Proteins \[D12.776.124\]](#)
[Acute-Phase Proteins \[D12.776.124.050\]](#) +
[Anion Exchange Protein 1, Erythrocyte \[D12.776.124.078\]](#)
[Ankyrins \[D12.776.124.080\]](#)
[beta 2-Glycoprotein I \[D12.776.124.117\]](#)
[Blood Coagulation Factors \[D12.776.124.125\]](#) +
[Cholesterol Ester Transfer Proteins \[D12.776.124.197\]](#)
[Fibrin \[D12.776.124.270\]](#) +
[Glycophorin \[D12.776.124.300\]](#)
[Hemocyanin \[D12.776.124.337\]](#)
► [Hemoglobins \[D12.776.124.400\]](#)
[Carboxyhemoglobin \[D12.776.124.400.141\]](#)
[Erythrocytins \[D12.776.124.400.220\]](#)

Curating Definitions: A Tour de Force



The theoretical distinction between dictionaries and encyclopaedias has traditionally been an issue of central importance for **lexicologists** (linguists who study word meaning) and **lexicographers** (dictionary writers).

The Dictionary View

- The dictionary view treats knowledge of word meaning as distinct from cultural knowledge, social knowledge and physical knowledge.
- Componential analysis or semantic decomposition approach:
 - word meaning is modelled in terms of semantic features or primitives.

bachelor is represented
as [MALE,ADULT,MARRIED]

Is the pope a *bachelor*?

Prototypes

“best example” of a category: e.g. *blackbird* vs. *penguin* for the category ‘bird’. But notice that the prototype may be abstract.



- not necessarily incompatible with feature theories
- fuzzy boundaries
- family resemblance



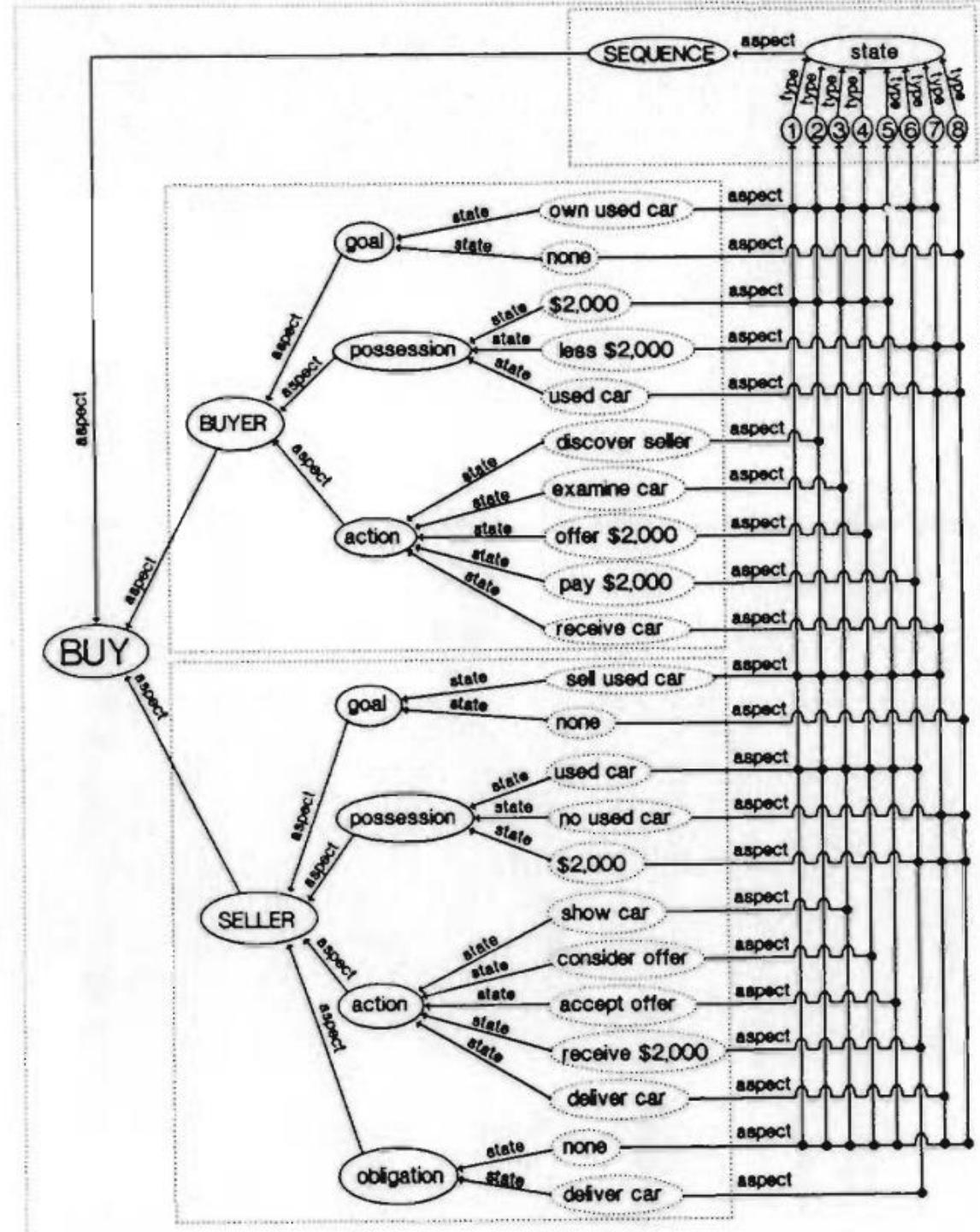
rank	category				
	BIRD	FRUIT	VEHICLE	FURNITURE	WEAPON
top eight					
1	robin	orange	automobile	chair	gun
2	sparrow	apple	station wagon	sofa	pistol
3	bluejay	banana	truck	couch	revolver
4	bluebird	peach	car	table	machine gun
5	canary	pear	bus	easy chair	rifle
6	blackbird	apricot	taxi	dresser	switchblade
7	dove	tangerine	jeep	rocking chair	knife
8	lark	plum	ambulance	coffee table	dagger
...
middle ranks					
26*	hawk	tangelo	subway	lamp	whip
27	raven	papaya	trailer	stool	ice pick
28	goldfinch	honeydew	cart	hassock	slingshot
29	parrot	fig	wheelchair	drawers	fists
30	sandpiper	mango	yacht	piano	axe
...
last five					
51*	ostrich	nut	ski	picture	foot
52	titmouse	gourd	skateboard	closet	car
53	emu	olive	wheelbarrow	vase	glass
54	penguin	pickle	surfboard	fan	screwdriver
55	bat	squash	elevator	telephone	shoes

* Since the total number of listed items varied between 50 and 60, the numbers of middle and bottom ranks are not identical with the original ranks for all categories.

Figure 1.3 A selection of examples from Rosch's goodness-of-example rating tests (Rosch 1975)

Frame Semantics

<https://framenet.icsi.berkeley.edu/fndrupal/frameIndex>



Commonsense Data (ConceptNet)

<http://conceptnet5.media.mit.edu/>

knife

knife – CapableOf → spread butter

knife can spread butter

knife – CapableOf → spread peanut butter

A knife can spread peanut butter

knife – UsedFor → stab

knife is for stabbing

knife – AtLocation → in kitchen

Something you might find in a kitchen is a knife.

knife – UsedFor → cut food

knife may be used to cut food.

cut – RelatedTo → knife

cut is related to knife

knife – AtLocation → kitchen drawer

You are likely to find a knife in a kitchen drawer

knife – MadeOf → steel

a knife can be made from steel.

knife – UsedFor → butter

a knife is used for butter

knife – IsA → tool

A knife is a type of tool

knife – AtLocation → kitchen

*Something you find in the kitchen is knife

knife – UsedFor → cut

When you want to cut, you will use knife.

knife – AtLocation → drawer

*Something you find in a drawer is a knife

knife – IsA → weapon

Kinds of weapons : knife

knife – UsedFor → eat

When you want to eat, you will use a knife.

machete – IsA → knife

a machete is a kind of a knife.

knife – AtLocation → store

*Something you find at a store is knives

blade – PartOf → knife

The blade is part of a knife

knife – CapableOf → butter bread

a knife can butter bread

in kitchen – AtLocation → knife

Something you find a knife is in the kitchen.

$$A \sqsubseteq B$$
$$A \sqsubseteq \neg B$$
$$A \sqcap B$$
$$A \sqcup B$$
$$A \equiv B$$
$$\top \sqsubseteq \forall P.A$$
$$\exists / \forall P.A$$

**Ontologies &
Description Logics**

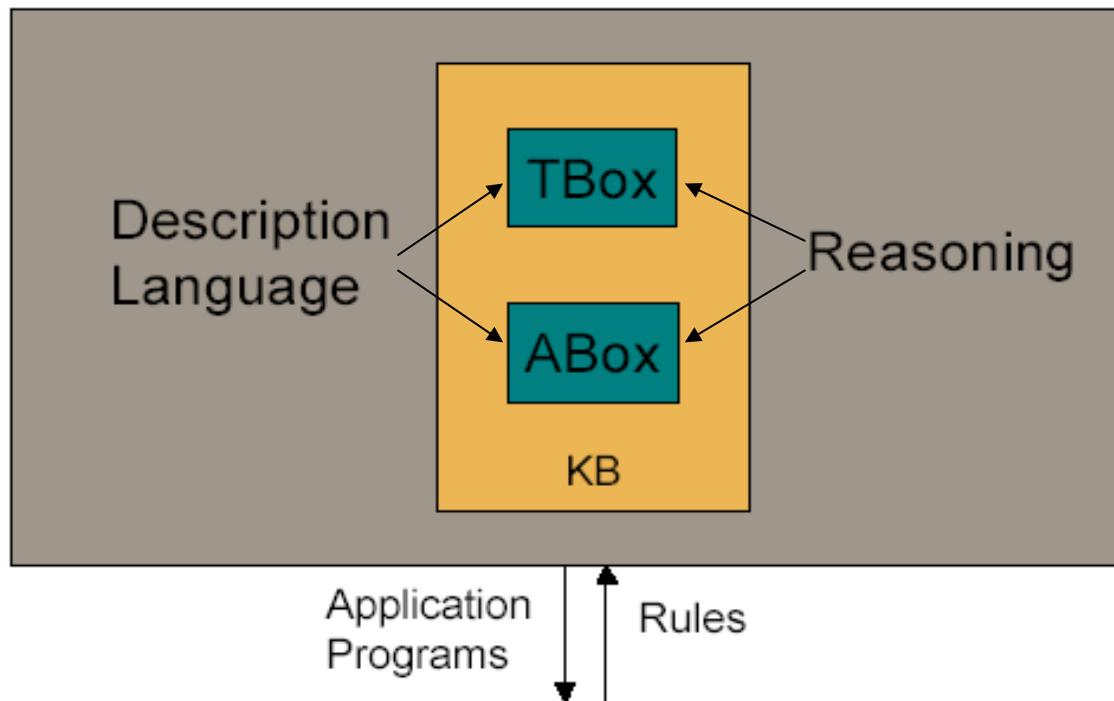
Description Logic

- Description Logics
 - Overcome the ambiguities of early semantic networks and frames
- Well-studied and decidable (most DL languages)
- Tight coupling between theory and practice

TBox and ABox

- TBox: terminology
 - The vocabulary of an application domain:
 - Concepts: sets of individuals
 - Roles: binary relationships between individuals.
 - Examples:
 - Concepts: Person, Female, Mother
 - Role: hasChild, meaning that some person is the child of some other
- ABox: assertions
 - About named individuals in terms of this vocabulary
 - Example
 - Elizabeth and Charles are Persons. We write this as Person(Elizabeth), and Person(Charles).
 - Individuals, like “myCar”, have attributes, like “color”, and those attributes have values, like “red”. When this happens we say that red is the colorOf attribute of myCar.
We write this as colorOf(myCar, red).

Architecture of a DL System

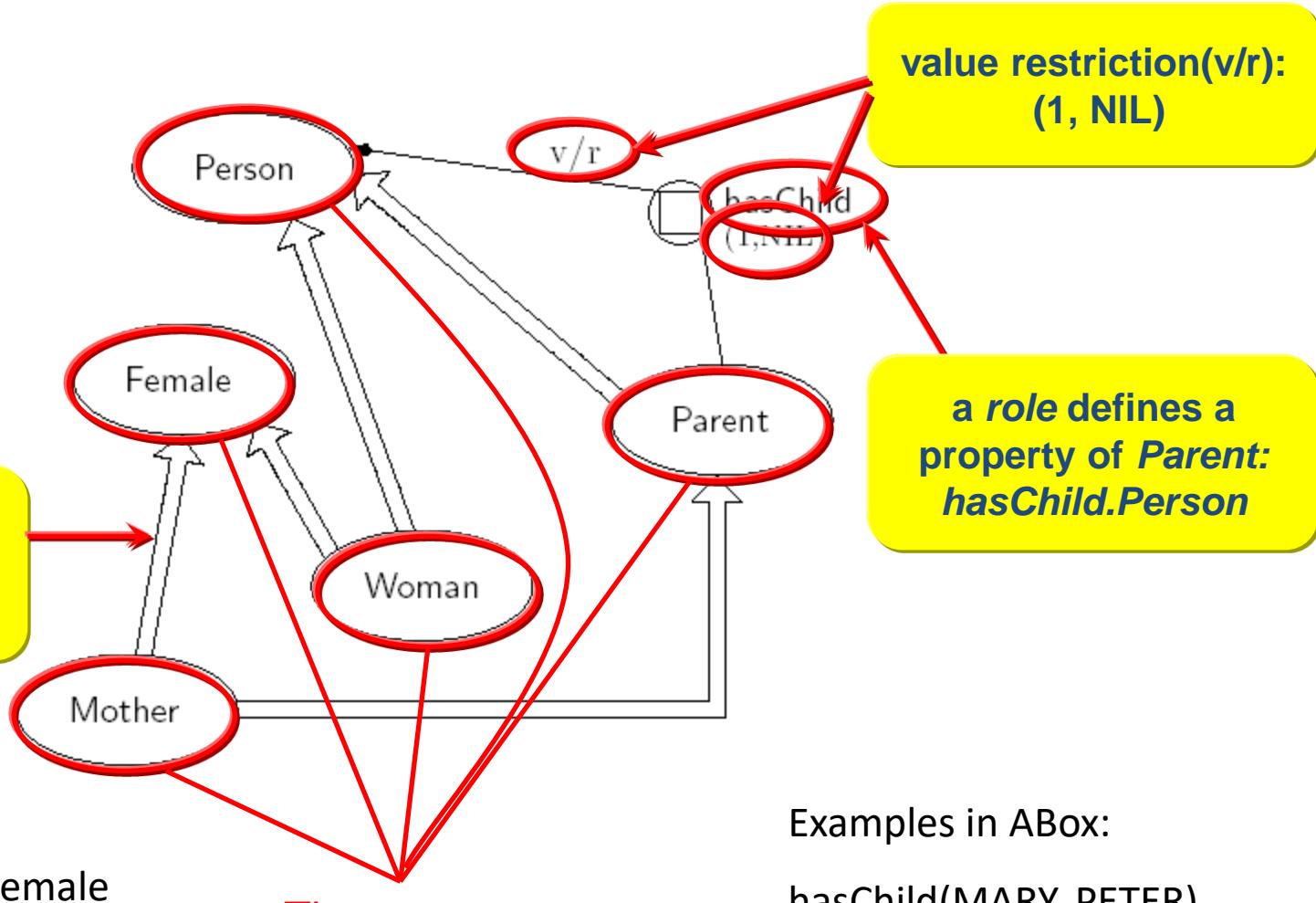


Formulas

- Building blocks that allow complex descriptions of concepts and roles.
 - Example (we'll look at the syntax in more detail soon.)
 - A Woman is a Female Person
 - Woman = Person \sqcup Female
 - A Mother is a Woman and she has a child
 - Mother = Woman \sqcup hasChild.T
- The TBox can be used to assign names to complex descriptions.

*We will use the terms *description* and *concept* interchangeably.*

An Example about Family Relationships



Examples in TBox:

Woman ⊑ Person \sqcup Female

Mother ⊑ Woman \sqcup hasChild.Person

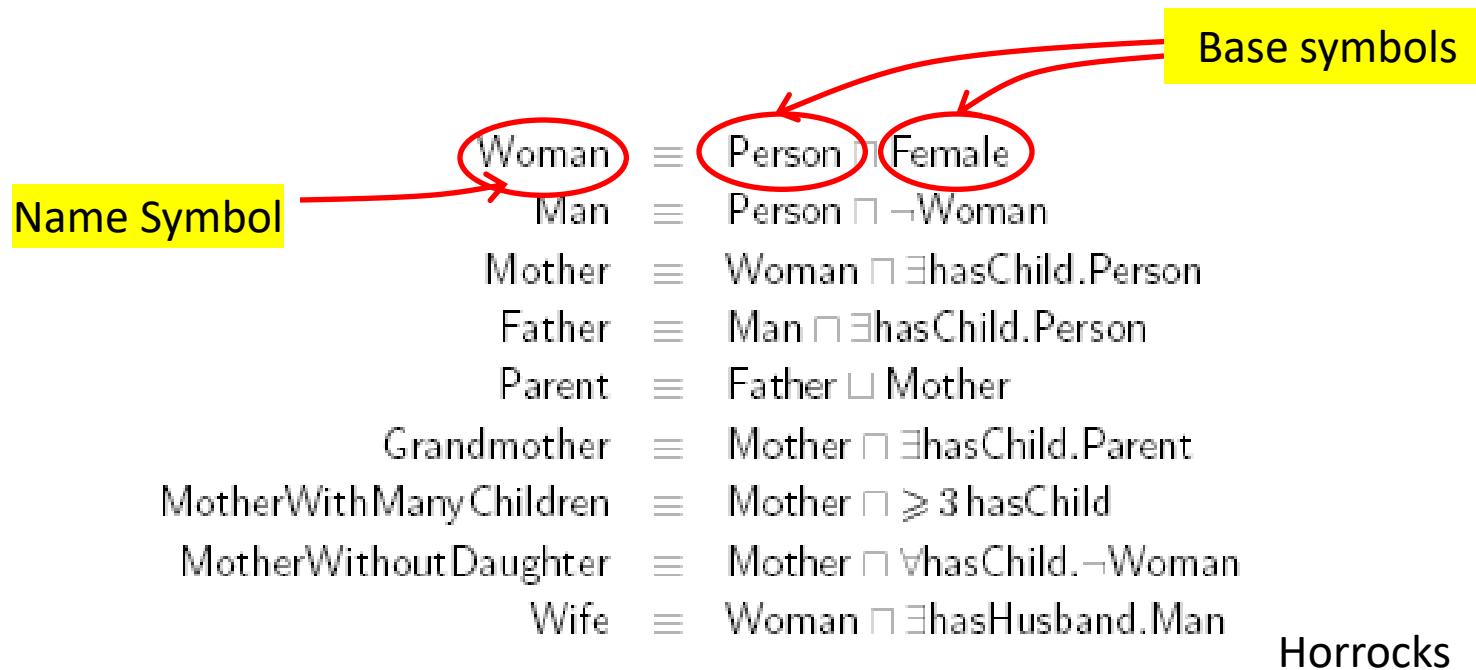
Examples in ABox:

hasChild(MARY, PETER)

Father(PETER)

Name Symbols vs. Base Symbols

- Atomic concepts occurring in a TBox \mathbf{T} can be divided into two sets, name symbols $N_{\mathbf{T}}$ (or defined concepts) and base symbols $B_{\mathbf{T}}$ (or primitive concepts, occur only on the right-hand side)
- A base interpretation for \mathbf{T} only interprets the base symbols.



DL for the Semantic Web

- Web Ontology Language (OWL): W3C Recommendation on 10 Feb 2004
- builds on RDF and RDF Schema and adds more vocabulary for describing properties and classesExtends existing Web standards
- has three increasingly-expressive sublanguages:
 - OWL Lite (based on DL SHIF (D)) ,
 - OWL DL (based on DL SHOIN(D)),
 - and OWL Full (OWL DL + RDF)
- benefits from many years of DL research
 - Well defined **semantics**
 - **Formal properties** well understood (complexity, decidability)
 - Known **reasoning algorithms**
 - Implemented systems (highly optimised)

OWL Class Constructor

Constructor	DL Syntax	Example	Modal Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male	$C_1 \wedge \dots \wedge C_n$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer	$C_1 \vee \dots \vee C_n$
complementOf	$\neg C$	\neg Male	$\neg C$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} \sqcup {mary}	$x_1 \vee \dots \vee x_n$
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor	$[P]C$
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer	$\langle P \rangle C$
maxCardinality	$\leq n P$	≤ 1 hasChild	$[P]_{n+1}$
minCardinality	$\geq n P$	≥ 2 hasChild	$\langle P \rangle_n$

OWL Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} \equiv {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg \{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
inverseOf	$P_1 \equiv P_2^-$	hasChild \equiv hasParent $^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor $^+$ \sqsubseteq ancestor
functionalProperty	$T \sqsubseteq \leqslant 1P$	T $\sqsubseteq \leqslant 1$ hasMother
inverseFunctionalProperty	$T \sqsubseteq \leqslant 1P^-$	T $\sqsubseteq \leqslant 1$ hasSSN $^-$

Ontology Editors



pizza.owl Protégé 3.2 beta (file:\C:\Nick\Applications\Protege_3.2_b235\examples\pizza\pizza.owl.pprj, OWL / RDF...)

File Edit Project OWL Code Tools Window Help

OWLClasses Properties Forms Individuals Metadata

SUBCLASS EXPLORER For Project: pizza.owl

CLASS EDITOR For Class: RealItalianPizza (instance of owl:Class) Inferred View

Annotations

Property	Value	Lang
rdfs:comment	This defined class has conditions that are part of the definition: ie any Pizza that has the country of origin, Italy is a RealItalianPizza. It also has conditions that merely describe the members - that all RealItalianPizzas must only have ThinAndCrispy bases.	en
rdfs:label	PizzaitalianaReal	pt

Asserted Conditions

- Pizza \exists hasCountryOfOrigin **has** Italy NECESSARY & SUFFICIENT
- hasBase **only** ThinAndCrispyBase NECESSARY
- hasBase **some** PizzaBase INHERITED [from Pizza]

Disjoints

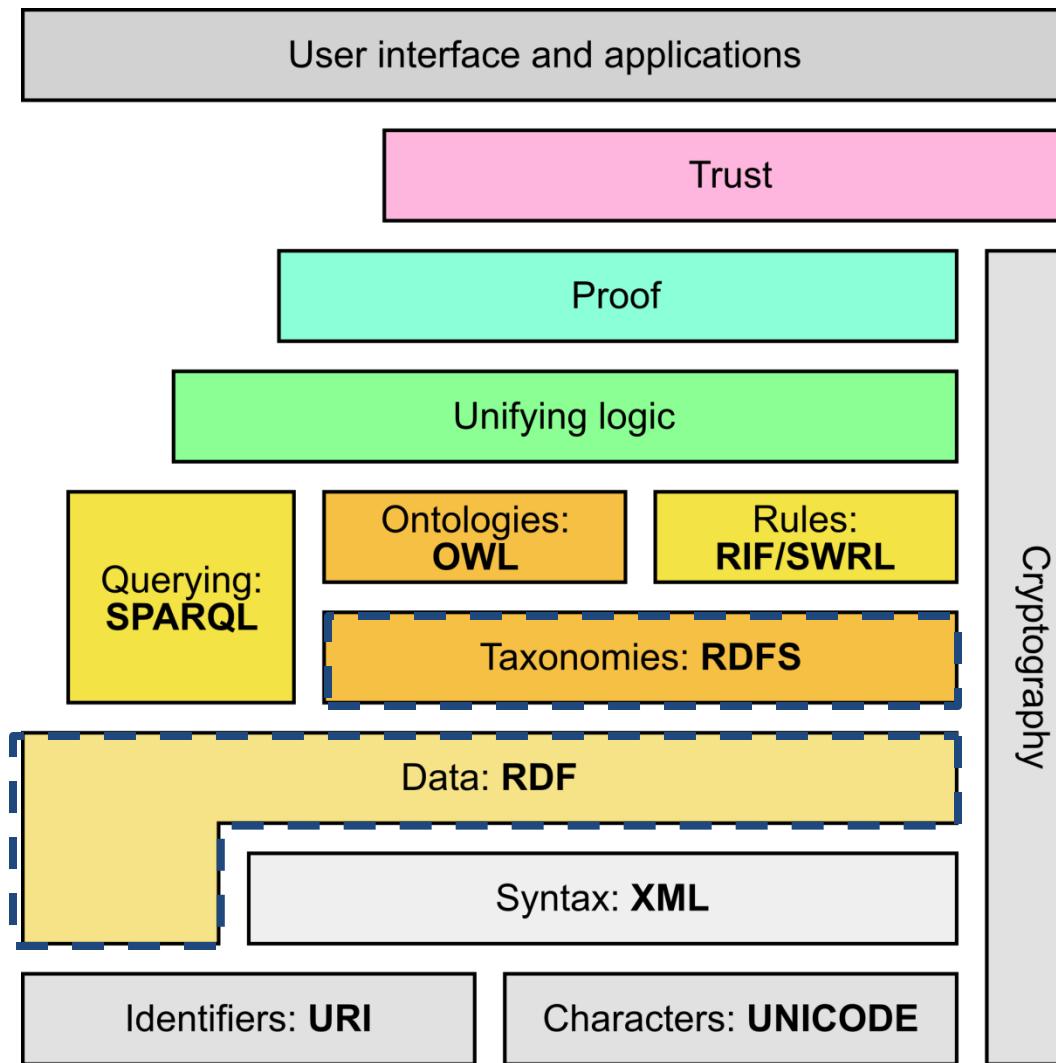
Logic View Properties View

The screenshot shows the Protégé 3.2 beta interface for editing the ontology file pizza.owl. The main window is divided into several panels: 'SUBCLASS EXPLORER' on the left showing the asserted hierarchy of classes like owl:Thing, DomainConcept, Pizza, and various pizza types; 'CLASS EDITOR' in the center displaying annotations for the class 'RealItalianPizza' (with properties rdfs:comment and rdfs:label); 'Asserted Conditions' panel listing constraints such as 'hasCountryOfOrigin' being 'Italy', 'hasBase' being 'ThinAndCrispyBase' (marked as 'NECESSARY'), and 'hasBase' being 'some PizzaBase' (marked as 'INHERITED'); and 'Disjoints' panel at the bottom. The top menu bar includes File, Edit, Project, OWL, Code, Tools, Window, and Help. The bottom navigation bar includes Logic View and Properties View.



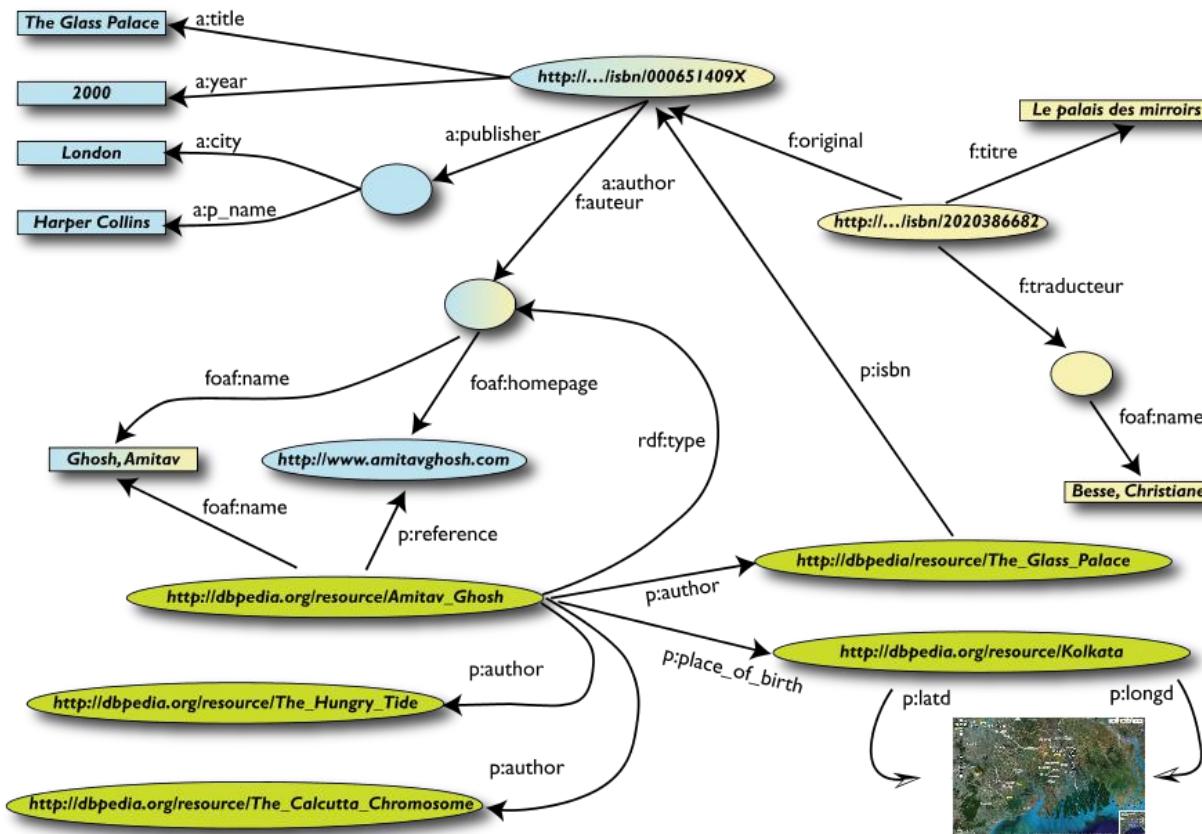
Semantic Web

Semantic Web Stack



RDF: a Direct Connected Graph based Model

- Different interconnected triples lead to a more complex graphic model.
- Basically a RDF document is a direct connect graph.



RDF Basics

- RDF is a language that enable to describe making statements on resources
 - John is father of Bill
- Statement (or triple) as a logical formula $P(x, y)$, where the binary predicate P relates the object x to the object y
- Triple data model:
`<subject, predicate, object>`
 - **Subject**: Resource or blank node
 - **Predicate**: Property
 - **Object**: Resource (or collection of resources), literal or blank node
- Example:
`<ex:john, ex:father-of, ex:bill>`
- RDF offers only binary predicates (properties)

RDF Vocabulary Description Language

- We need a language for defining RDF types:
 - Define classes:
 - “*#Student* is a class”
 - Relationships between classes:
 - “*#Student* is a sub-class of *#Person*”
 - Properties of classes:
 - “*#Person* has a property *hasName*”
- RDF Schema is such a language.

RDF Vocabulary Description Language

- Classes:
`<#Student, rdf:type, #rdfs:Class>`
- Class hierarchies:
`<#Student, rdfs:subClassOf, #Person>`
- Properties:
`<#hasName, rdf:type, rdf:Property>`
- Property hierarchies:
`<#hasMother, rdfs:subPropertyOf, #hasParent>`
- Associating properties with classes (a):
 - “The property `#hasName` only applies to `#Person`”
`<#hasName, rdfs:domain, #Person>`
- Associating properties with classes (b):
 - “The type of the property `#hasName` is `#xsd:string`”
`<#hasName, rdfs:range, xsd:string>`

RDFS Vocabulary

- RDFS extends the RDF vocabulary

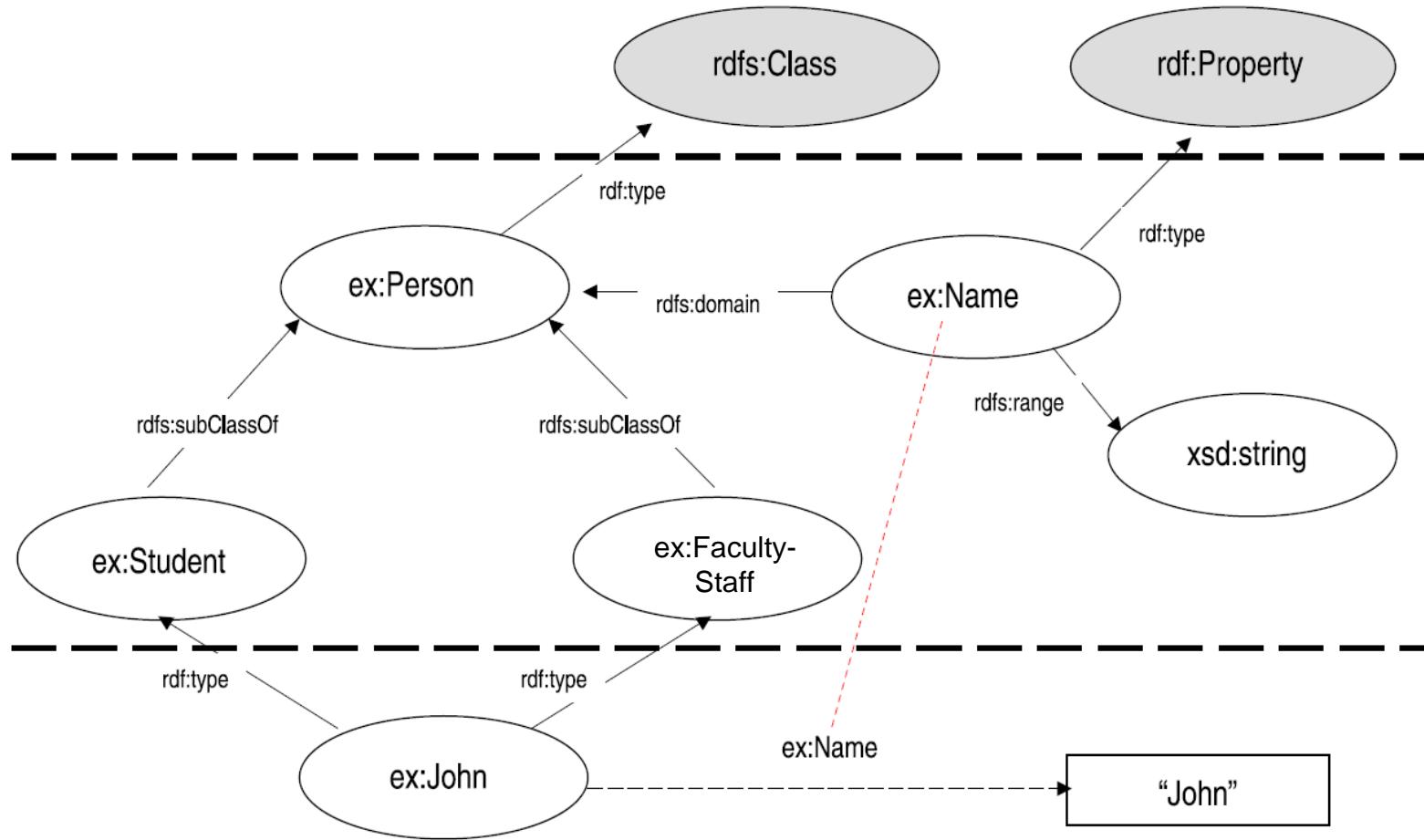
RDFS Classes

- `rdfs:Resource`
- `rdfs:Class`
- `rdfs:Literal`
- `rdfs:Datatype`
- `rdfs:Container`
- `rdfs:ContainerMembershipProperty`

RDFS Properties

- `rdfs:domain`
- `rdfs:range`
- `rdfs:subPropertyOf`
- `rdfs:subClassOf`
- `rdfs:member`
- `rdfs:seeAlso`
- `rdfs:isDefinedBy`
- `rdfs:comment`
- `rdfs:label`

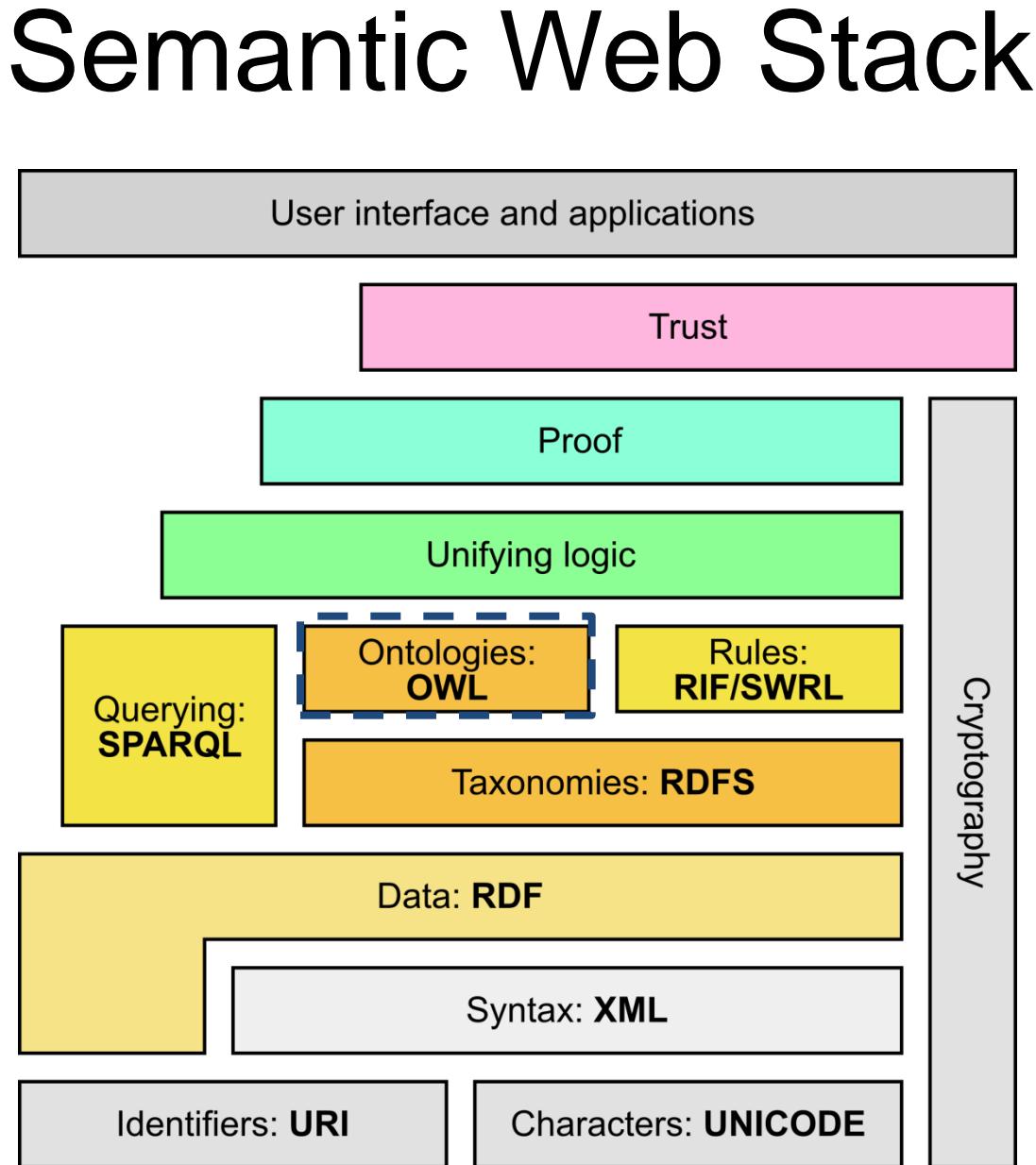
RDFS Example



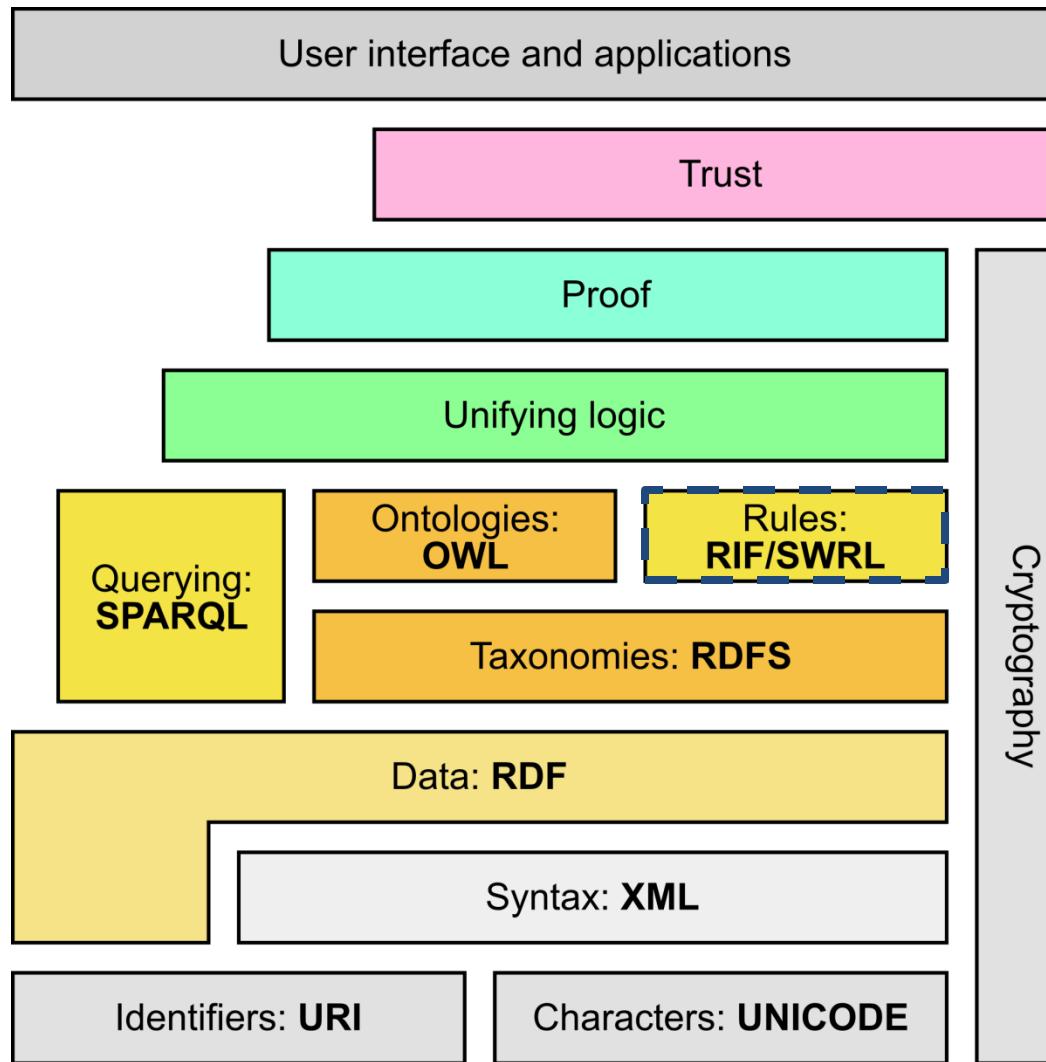
Vocabulary Layer

RDFS Layer

RDF Layer



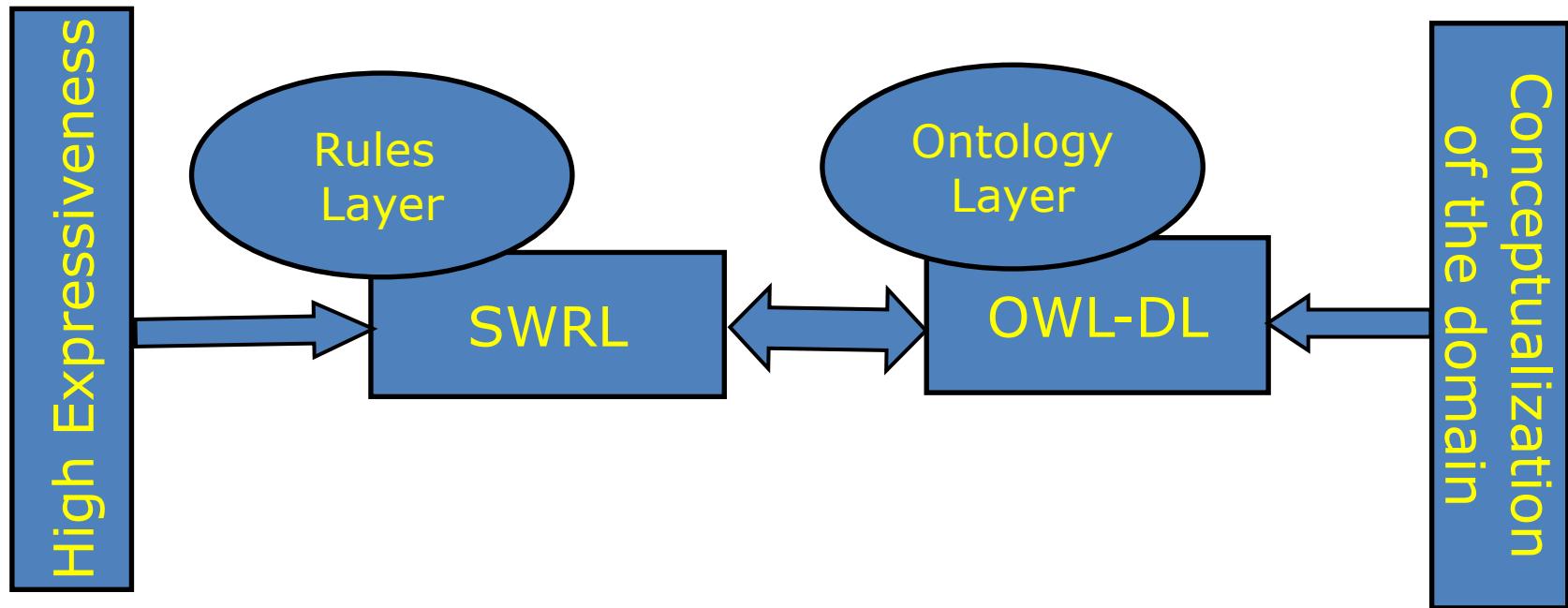
Semantic Web Stack



Adapted from http://en.wikipedia.org/wiki/Semantic_Web_Stack

What is SWRL?

Ontology languages do not offer the expressiveness we want → Rules do it well.



SWRL Rule

An atom is an expression of the form: **P(arg1 arg2,...)**

- **P** is a predicate symbol (classes, properties...)
- Arguments of the expression: **arg1, arg2,...** (individuals, data values or variables)

Example SWRL Rule:

Person(?p) ^ hasSibling(?p,?s) ^ Man(?s) → hasBrother(?p,?s)

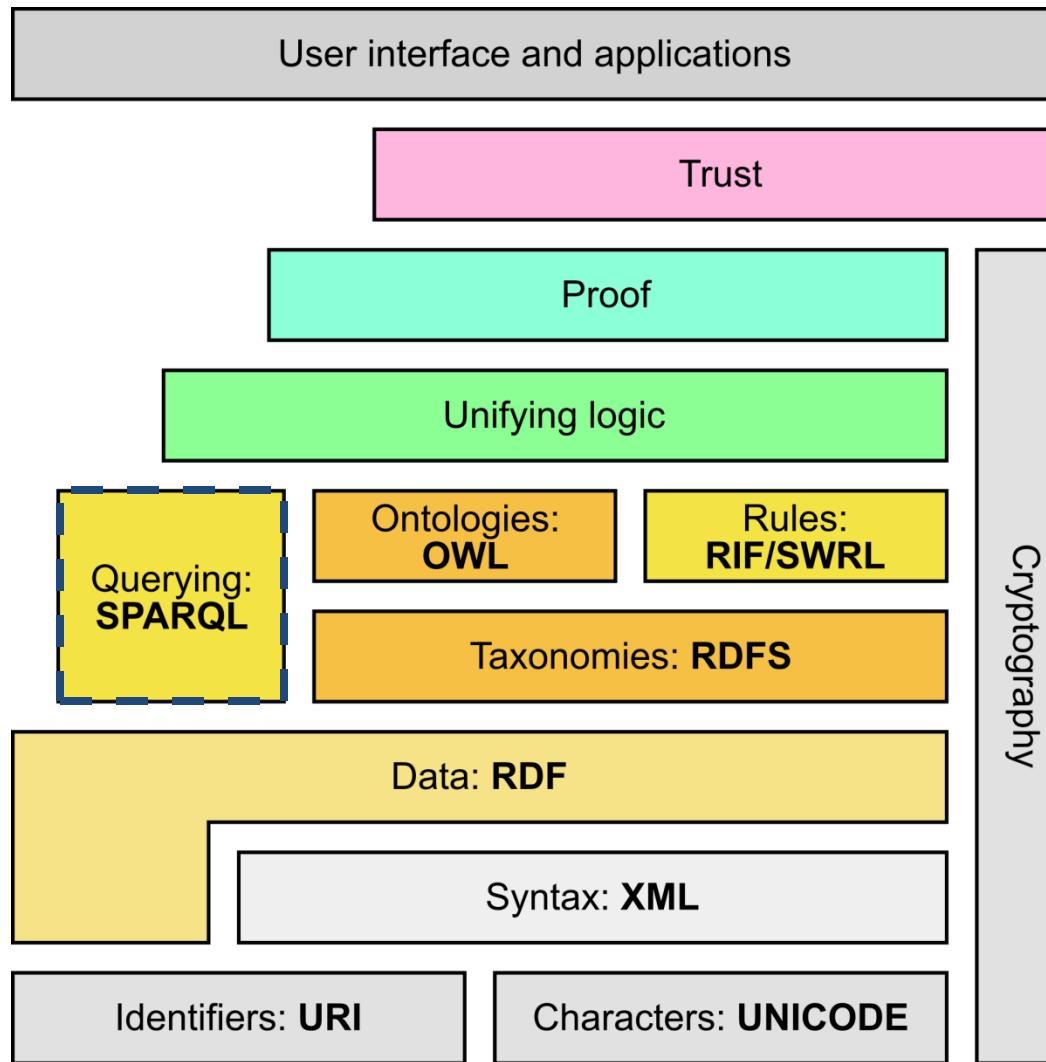


antecedent



consequent

Semantic Web Stack



Adapted from http://en.wikipedia.org/wiki/Semantic_Web_Stack

SPARQL

- Query Language for RDF.

Find all DrugBank drugs along with dosage and disease indication information.

PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>

PREFIX db: <http://bio2rdf.org/drugbank_vocabulary:>

SELECT

?drug_name ?dosage ?indication

WHERE {

?drug a db:Drug .

?drug rdfs:label ?drug_name .

OPTIONAL { ?drug db:dosage ?dosage . }

OPTIONAL { ?drug db:indication ?indication . }

}

SPARQL (Yes/No Query)

Is the Amazon river longer than the Nile River?

```
PREFIX prop: <http://dbpedia.org/property/>
ASK {
    <http://dbpedia.org/resource/Amazon_River> prop:length ?amazon .
    <http://dbpedia.org/resource/Nile> prop:length ?nile .
    FILTER(?amazon > ?nile) .
}
```

http://dbpedia.org/sparql

The screenshot shows a web-based SPARQL query editor. At the top, there is a header bar with navigation icons (back, forward, search) and a URL field containing "de.dbpedia.org/sparql". Below the header is a blue title bar labeled "Virtuoso SPARQL Query Editor". Underneath the title bar, there is a text input field for "Default Data Set Name (Graph IRI)". The main area is titled "Query Text" and contains the following SPARQL query:

```
PREFIX prop: <http://dbpedia.org/property/>
ASK
{
  <http://dbpedia.org/resource/Amazon_River> prop:length ?amazon .
  <http://dbpedia.org/resource/Nile> prop:length ?nile .
  FILTER(?amazon > ?nile)
}
```

APIs

- Helpful for building AI applications.
- Typical Support:
 - A RDF API
 - Serialization: Reading and writing RDF in RDF/XML, N3 and N-Triples
 - An OWL API
 - In-memory and persistent storage
 - SPARQL query engine

Recommended Reading

1

Frames, Concepts, and Conceptual Fields

Lawrence W. Barsalou
University of Chicago

http://barsaloulab.org/Online_Articles/1992-Barsalou-chap-frames.pdf

Recommended Reading

Intelligent Machines

An AI with 30 Years' Worth of Knowledge Finally Goes to Work

An effort to encode the world's knowledge in a huge database has sometimes seemed impractical, but those behind the technology say it is finally ready.

by Will Knight March 14, 2016

Having spent the past 31 years memorizing an astonishing collection of general knowledge, the artificial-intelligence engine created by Doug Lenat is finally ready to go to work.

Lenat's creation is **Cyc**, a knowledge base of semantic information designed to give computers some understanding of how things work in the real world.

<https://www.technologyreview.com/s/600984/an-ai-with-30-years-worth-of-knowledge-finally-goes-to-work/>

Symbolic AI

Andre Freitas



Photo by Vasilyev Alexandr

Today

- Knowledge Bases for supporting AI Systems.
- Knowledge Representation paradigms for KBs.



Powered by Thomson Reuters

[News Feed](#) [Connection](#) [Portfolio](#)

[Settings](#) [All](#) [Pinned](#)

3 number of pinned articles

[BRIEF-Avio receives EUR 40 mln financing from European Investment Bank <SPA2.MI>](#)

Companies: Avio SPA 80%, Avio SPA 80%, Avio SPA 80%

Topics: Business Finance, Contracts / Business Deals Events: ContactDetails

Industry: Spacecraft Manufacturing

Publication date: Oct 6, 2017 10:24:02 AM

Airbus SE

BAE Systems PLC

Boeing Co

[BRIEF-Airtelis orders three H215 Airbus Helicopters<AIR.PA><AIRG.DE>](#)

Companies: Airbus Helicopters SAS 50%, Airbus Helicopters SAS 50%, Airbus Helicopters SAS 50%

Topics: N/A Events: BusinessRelation

Industry: Aerospace & Defense - NEC, Aircraft Parts Manufacturing - NEC

Publication date: Oct 4, 2017 10:46:00 AM

Raytheon Co

BAE Systems PLC

Boeing Co

[BRIEF-British Airline Pilots' Association - pilots union calls for investigation into collapse of Monarch Airlines<MONA.UL>](#)

Companies: Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%

Topics: Other Events: N/A

Industry: Airlines - NEC

Publication date: Oct 9, 2017 1:55:13 PM

Raytheon Co

BAE Systems PLC

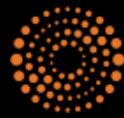
Boeing Co



THOMSON REUTERS

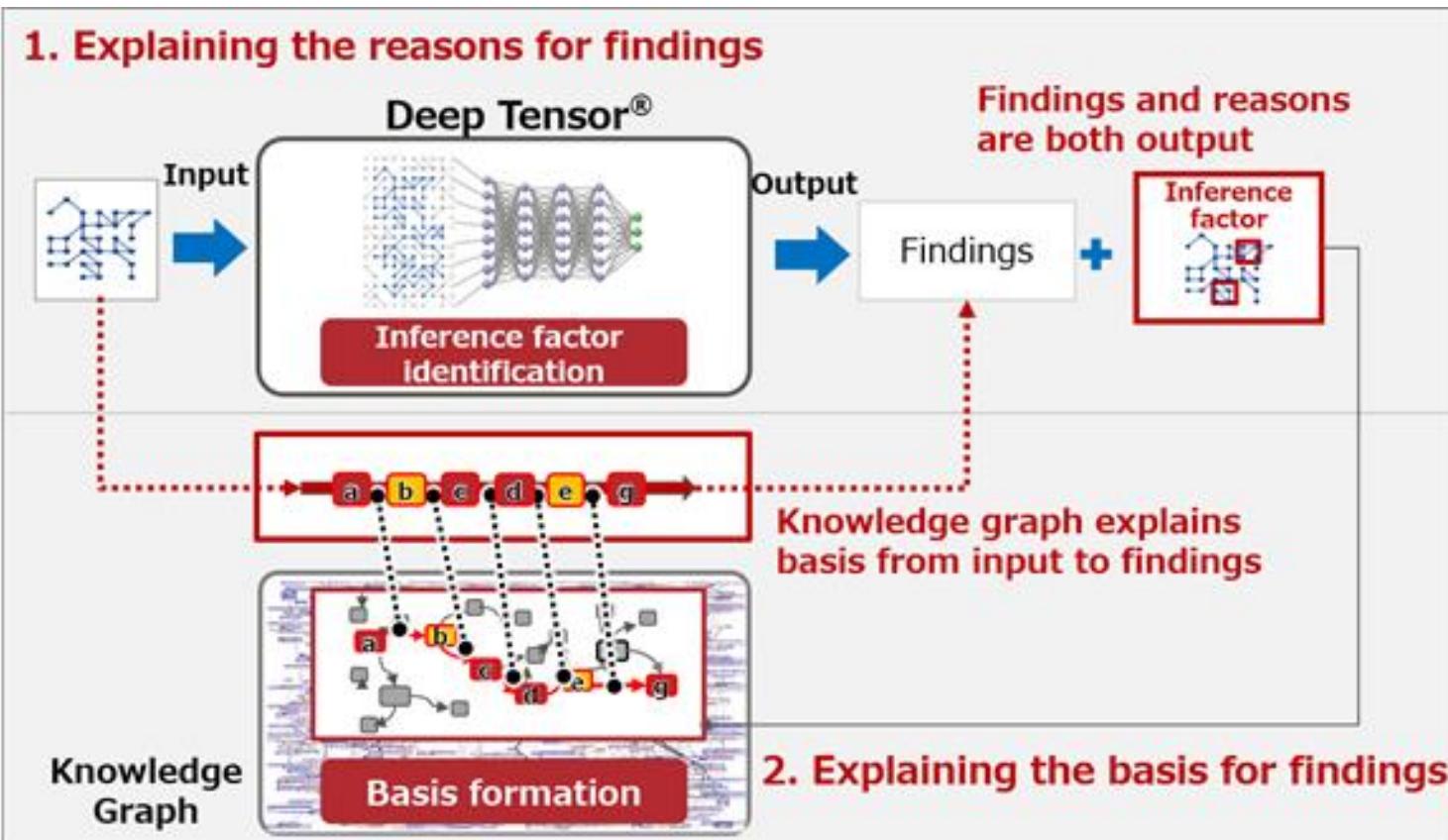
BRIEF-Avio receives EUR 40 mn financing from European Investment Bank <SPA2.MI>

Oct 6 (Reuters) - AVIO SPA <SPA2.MI> * SAYS SIGNED WITH EUROPEAN INVESTMENT BANK CONTRACT FOR EUR 40 MILLION FINANCING Source text for Eikon: [ID:nBIA5D9Ntk] Further company coverage: [SPA2.MI] (Gdynia Newsroom) (gdynia.newsroom@thomsonreuters.com; +48 58 772 0920 ;)



THOMSON REUTERS

Explainable Findings From Tensor Inferences Back to KGs



Machine Knowledge for Answer Engines

Weikum, 2019

Precise and concise answers
for advanced information needs:



properties of entity

- ★ Nobel laureate whose daughter also won a Nobel prize?



sets of entities

- ★ Pop singers who are also poets?

relationships between entities

- ★ Commonalities & relationships among:
Alan Turing, Paul Bocuse, Steve Jobs, Katherine Goble?



Machine Knowledge for Answer Engines

Weikum, 2019

**Precise and concise answers
for advanced information needs:**

real applications

- ★ Proteins that bind to the Zika virus?
- Polymer materials for super-capacitors?
- European politicians mentioned in Panama Leaks?

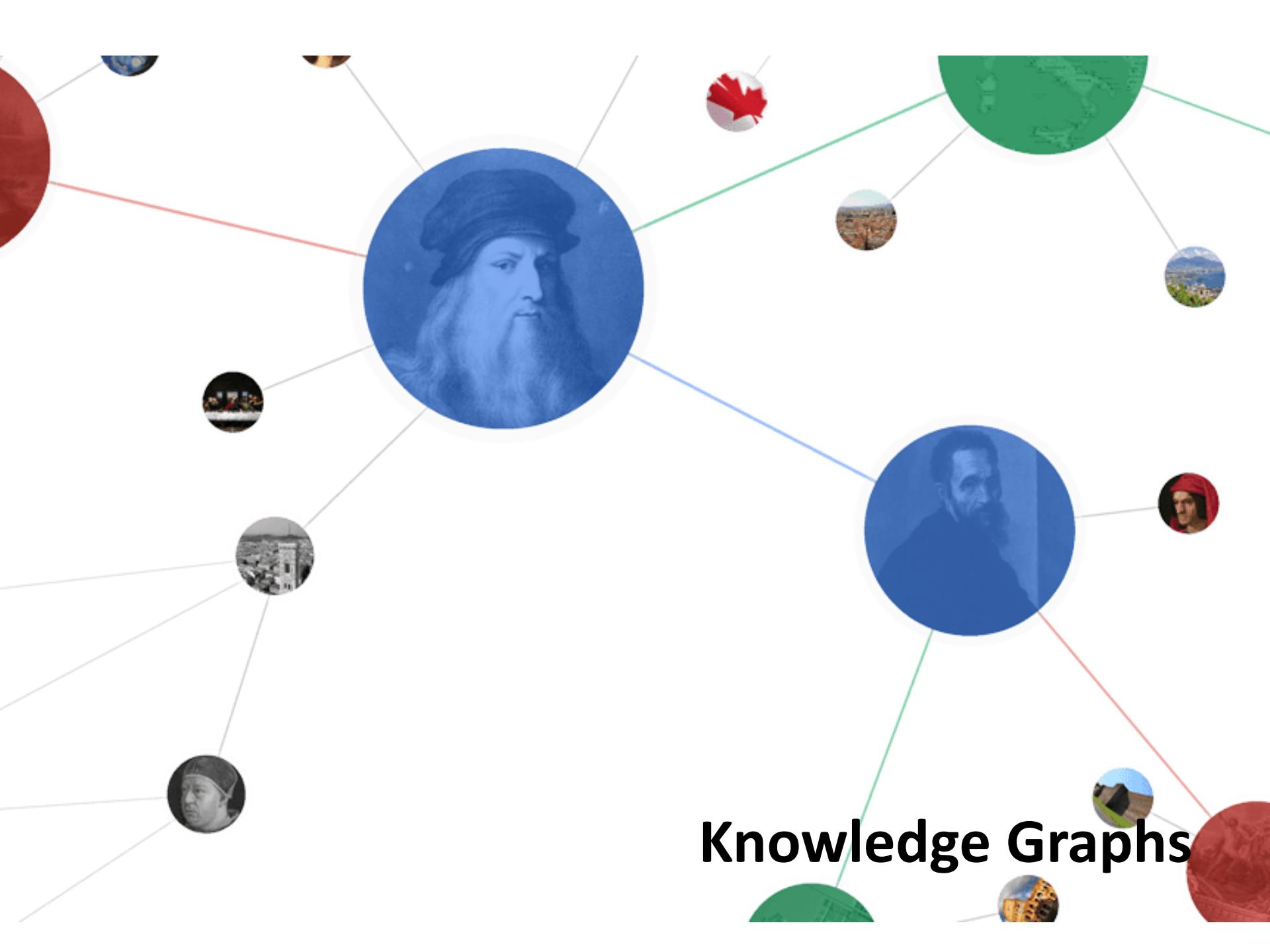
Representation

- **Representation**: organization of a perceptual/symbolic space into an abstraction
 - Attention/selection
 - Intent/goal
- Maximization of Inferential Locality
- Abstraction for Purpose
- Correctness/Completeness for Purpose

Good Knowledge Representation Languages

- Combines the best of natural and formal languages:
 - expressive
 - concise
 - unambiguous
 - independent of context
 - what you say today will still be interpretable tomorrow
 - efficient
 - the knowledge can be represented in a format that is suitable for computers
 - operational
 - there is an inference procedure which can act on it to make new sentences

Knowledge Graphs



Brief History of Knowledge Bases

Weikum, 2019



Vannevar Bush



Cyc



WordNet



Wikipedia



WWW

Cyc



WordNet



freebase™

DBpedia

yAGO
select knowledge

WIKIDATA

BabelNet

IBM Watson

amazon

Alibaba

SIEMENS

Ingenuity for Bloomberg



Terminology

- Ontology
 - provides more complete definitions for concepts
 - Graphical conceptual model
- Thesaurus
 - simple relationships between words
 - synonyms, homonyms, antonyms, etc.
 - often combined with a taxonomy
- Taxonomy
 - hierarchical arrangement of concepts
 - often used as a “backbone” for an ontology
- Lexicon
 - provides natural language descriptions of words and their meanings

Word Senses & Relations

Reminder: lemma and wordform

- A lemma or citation form
 - Same stem, part of speech, rough semantics
- A word form
 - The “inflected” word as it appears in text

Word form	Lemma
banks	bank
sung	sing
duermes	dormir

Lemmas have senses

- One lemma “bank” can have many meanings:

- Sense 1: • ...a **bank** can hold the investments in a custodial account...
 1
- Sense 2: • “...as agriculture burgeons on the east **bank** the river will shrink
 even more”
 2

- Sense (or word sense)
 - A discrete representation
of an aspect of a word’s meaning.
- The lemma **bank** here has two senses

Polysemy

- 1. The **bank** was constructed in 1875 out of local red brick.
- 2. I withdrew the money from the **bank**.
- Are those the same sense?
 - Sense 2: “A financial institution”
 - Sense 1: “The building belonging to a financial institution”
- A polysemous word has **related** meanings
 - Most non-rare words have multiple meanings.

How do we know when a word has more than one sense?

- The “zeugma” test: Two senses of **serve**?
 - Which flights **serve** breakfast?
 - Does Lufthansa **serve** Philadelphia?
 - ?Does Lufthansa serve breakfast and San Jose?
- Since this conjunction sounds weird,
 - we say that these are **two different senses of “serve”**

Synonyms

- Word that have the same meaning in some or all contexts:
 - couch / sofa
 - big / large
 - automobile / car
 - vomit / throw up
 - Water / H₂O
- Two lexemes are synonyms:
 - if they can be substituted for each other in all situations.
 - If so they have the same **propositional meaning.**

Synonymy is a relation between senses rather than words

- Consider the words *big* and *large*.
- Are they synonyms?
 - How **big** is that plane?
 - Would I be flying on a **large** or small plane?
- How about here:
 - Miss Nelson became a kind of **big** sister to Benjamin.
 - ?Miss Nelson became a kind of **large** sister to Benjamin.
- Why?
 - *big* has a sense that means being older, or grown up.
 - *large* lacks this sense.

Antonyms

- Senses that are opposites with respect to one feature of meaning.
- Otherwise, they are very similar!

dark/light short/long fast/slow rise/fall

hot/cold up/down in/out

Hyponymy and Hypernymy

- One sense is a hyponym of another if the first sense is more specific, denoting a subclass of the other:
 - *car* is a hyponym of *vehicle*
 - *mango* is a hyponym of *fruit*
- Conversely hypernym/superordinate (“hyper is super”):
 - *vehicle* is a hypernym of *car*
 - *fruit* is a hypernym of *mango*

Superordinate/hyper	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

Hyponymy more formally

- Extensional:
 - The class denoted by the superordinate extensionally includes the class denoted by the hyponym.
- Hyponymy is usually transitive
 - (A hypo B and B hypo C entails A hypo C).
- Another name: the IS-A hierarchy:
 - A IS-A B (or A ISA B)
 - B **subsumes** A

Hyponyms and Instances

- WordNet has both classes and instances.
- An instance is an individual, a proper noun that is a unique entity:
 - San Francisco is an **instance** of city
 - But city is a class
 - city is a **hyponym** of municipality...location...

WordNet & Other Online Thesauri

Senses of “bass” in Wordnet

Noun

- S: (n) **bass** (the lowest part of the musical range)
- S: (n) **bass**, [bass part](#) (the lowest part in polyphonic music)
- S: (n) **bass**, [basso](#) (an adult male singer with the lowest voice)
- S: (n) [sea bass](#), **bass** (the lean flesh of a saltwater fish of the family Serranidae)
- S: (n) [freshwater bass](#), **bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- S: (n) **bass**, [bass voice](#), [basso](#) (the lowest adult male singing voice)
- S: (n) **bass** (the member with the lowest range of a family of musical instruments)
- S: (n) **bass** (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Adjective

- S: (adj) **bass**, [deep](#) (having or denoting a low vocal or instrumental range) "a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"

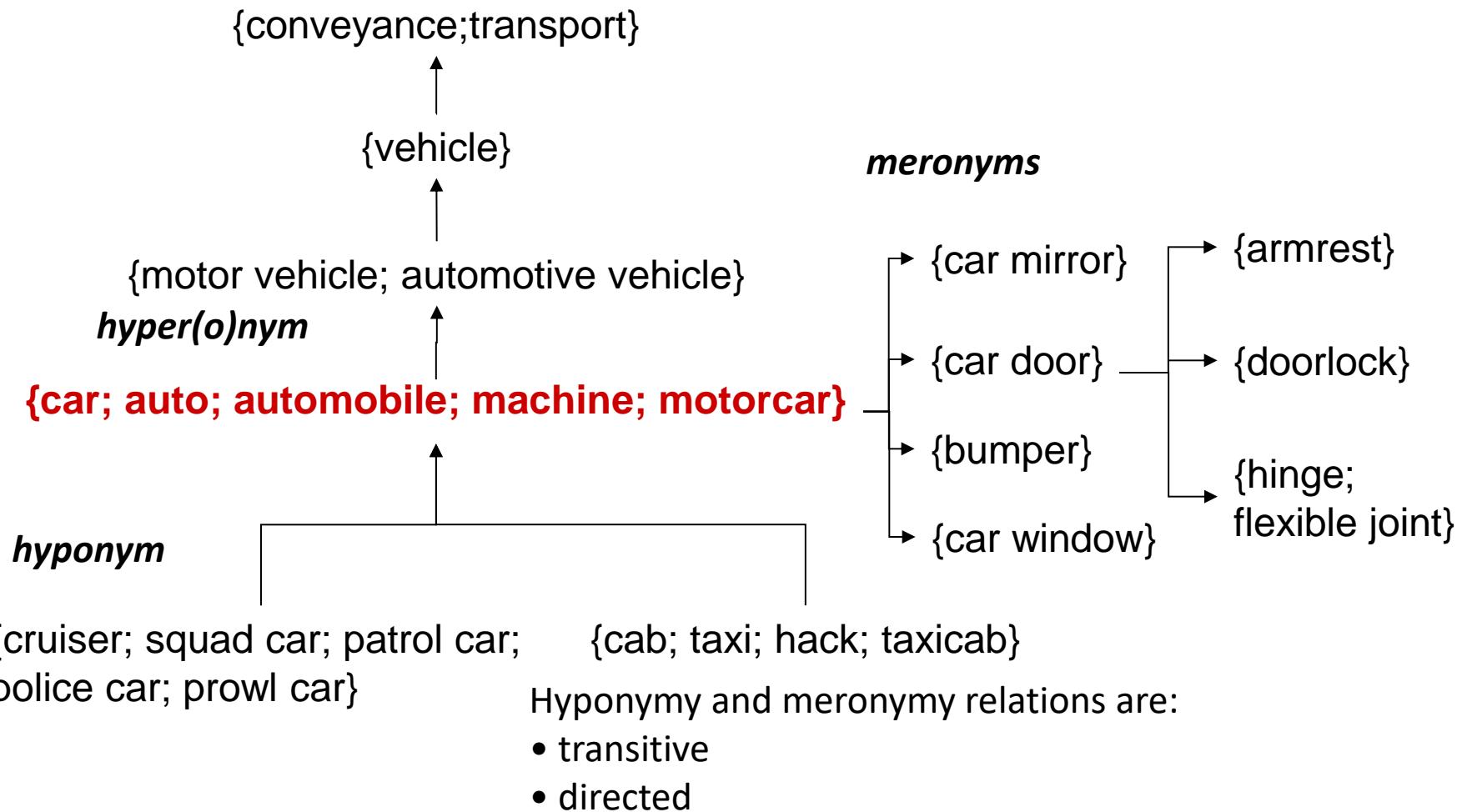
How is “sense” defined in WordNet?

- The synset (synonym set), the set of near-synonyms, instantiates a sense or concept, with a gloss.
- Example: chump as a noun with the gloss:
“a person who is gullible and easy to take advantage of”
- This sense of “chump” is shared by 9 words:
chump¹, fool², gull¹, mark⁹, patsy¹, fall guy¹, sucker¹, soft touch¹, mug²

WordNet Hypernym Hierarchy for “bass”

- S: (n) bass, basso (an adult male singer with the lowest voice)
 - direct hypernym / inherited hypernym / sister term
 - S: (n) singer, vocalist, vocalizer, vocaliser (a person who sings)
 - S: (n) musician, instrumentalist, player (someone who plays a musical instrument (as a profession))
 - S: (n) performer, performing artist (an entertainer who performs a dramatic or musical work for an audience)
 - S: (n) entertainer (a person who tries to please or amuse)
 - S: (n) person, individual, someone, somebody, mortal, soul (a human being) "there was too much for one person to do"
 - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
 - S: (n) living thing, animate thing (a living (or once living) entity)
 - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) "how big is that part compared to the whole?"; "the team is a unit"
 - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) "it was full of rackets, balls and other objects"
 - S: (n) physical entity (an entity that has physical existence)
 - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Wordnet: a network of semantically related words



Wordnet Semantic Relations

WN 1.5 starting point

The ‘synset’ as a weak notion of synonymy:

“two expressions are synonymous in a linguistic context C if the substitution of one for the other in C does not alter the truth value.” (Miller et al. 1993)

Relations between synsets:

HYPONYMY	noun-to-noun	car/ vehicle
	verb-to-verb	walk/ move
MERONYMY	noun-to-noun	head/ nose
ANTONYMY	adjective-to-adjective	good/bad
	verb-to-verb	open/ close
ENTAILMENT	verb-to-verb	buy/ pay
CAUSE	verb-to-verb	kill/ die

Some observations on Wordnet

- Synsets are more compact representations for concepts than word meanings in traditional lexicons.
- Synonyms and hypernyms are substitutional variants:
 - begin – commence
 - I once had a **canary**. The **bird** got sick. The poor **animal** died.
- Hyponymy and meronymy chains are important transitive relations for predicting properties and explaining textual properties:
object -> artifact -> vehicle -> 4-wheeled vehicle -> car
- Strict separation of part of speech (PoS) although concepts are closely related (**bed – sleep**) and are similar (**dead – death**).

PoS (Part-of-Speech)

The Penn Treebank POS tagset.

1. CC	Coordinating conjunction	25. TO	<i>to</i>
2. CD	Cardinal number	26. UH	Interjection
3. DT	Determiner	27. VB	Verb, base form
4. EX	Existential <i>there</i>	28. VBD	Verb, past tense
5. FW	Foreign word	29. VBG	Verb, gerund/present participle
6. IN	Preposition/subordinating conjunction	30. VBN	Verb, past participle
7. JJ	Adjective	31. VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective, comparative	32. VBZ	Verb, 3rd ps. sing. present
9. JJS	Adjective, superlative	33. WDT	<i>wh</i> -determiner
10. LS	List item marker	34. WP	<i>wh</i> -pronoun
11. MD	Modal	35. WP\$	Possessive <i>wh</i> -pronoun
12. NN	Noun, singular or mass	36. WRB	<i>wh</i> -adverb
13. NNS	Noun, plural	37. #	Pound sign
14. NNP	Proper noun, singular	38. \$	Dollar sign
15. NNPS	Proper noun, plural	39. .	Sentence-final punctuation
16. PDT	Predeterminer	40. ,	Comma
17. POS	Possessive ending	41. :	Colon, semi-colon
18. PRP	Personal pronoun	42. (Left bracket character
19. PP\$	Possessive pronoun	43.)	Right bracket character
20. RB	Adverb	44. "	Straight double quote
21. RBR	Adverb, comparative	45. '	Left open single quote
22. RBS	Adverb, superlative	46. "	Left open double quote
23. RP	Particle	47. '	Right close single quote
24. SYM	Symbol (mathematical or scientific)	48. "	Right close double quote

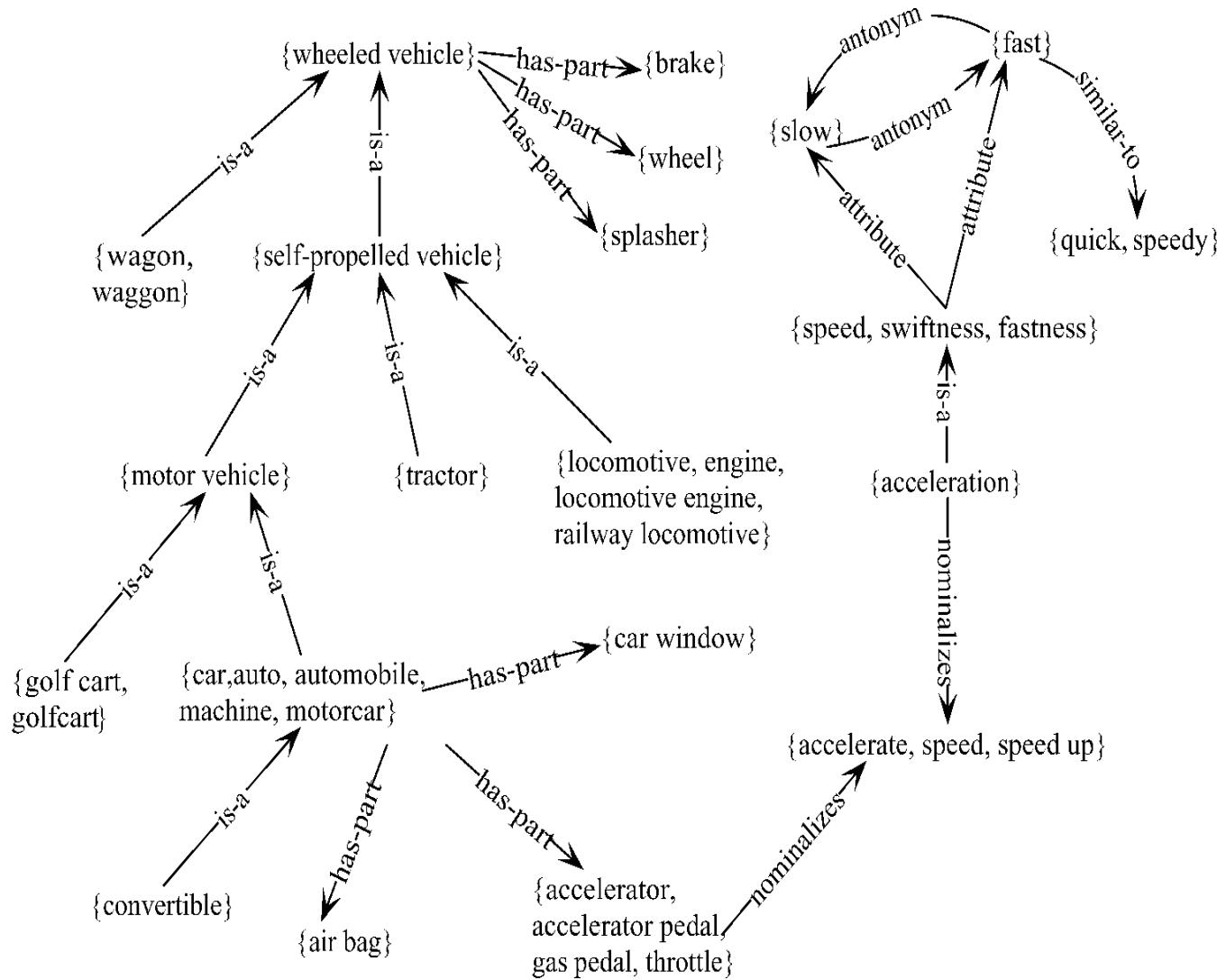
WordNet Noun Relations (Reference)

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	<i>breakfast</i> ¹ → <i>meal</i> ¹
Hyponym	Subordinate	From concepts to subtypes	<i>meal</i> ¹ → <i>lunch</i> ¹
Instance Hypernym	Instance	From instances to their concepts	<i>Austen</i> ¹ → <i>author</i> ¹
Instance Hyponym	Has-Instance	From concepts to concept instances	<i>composer</i> ¹ → <i>Bach</i> ¹
Member Meronym	Has-Member	From groups to their members	<i>faculty</i> ² → <i>professor</i> ¹
Member Holonym	Member-Of	From members to their groups	<i>copilot</i> ¹ → <i>crew</i> ¹
Part Meronym	Has-Part	From wholes to parts	<i>table</i> ² → <i>leg</i> ³
Part Holonym	Part-Of	From parts to wholes	<i>course</i> ⁷ → <i>meal</i> ¹
Substance Meronym		From substances to their subparts	<i>water</i> ¹ → <i>oxygen</i> ¹
Substance Holonym		From parts of substances to wholes	<i>gin</i> ¹ → <i>martini</i> ¹
Antonym		Semantic opposition between lemmas	<i>leader</i> ¹ ⇔ <i>follower</i> ¹
Derivationally		Lemmas w/same morphological root	<i>destruction</i> ¹ ⇔ <i>destroy</i> ¹
Related Form			

WordNet Verb Relations (Reference)

Relation	Definition	Example
Hypernym	From events to superordinate events	<i>fly</i> ⁹ → <i>travel</i> ⁵
Troponym	From events to subordinate event (often via specific manner)	<i>walk</i> ¹ → <i>stroll</i> ¹
Entails	From verbs (events) to the verbs (events) they entail	<i>snore</i> ¹ → <i>sleep</i> ¹
Antonym	Semantic opposition between lemmas	<i>increase</i> ¹ ⇔ <i>decrease</i> ¹
Derivationally	Lemmas with same morphological root	<i>destroy</i> ¹ ⇔ <i>destruction</i> ¹
Related Form		

WordNet: Viewed as a Graph



WordNet 3.0

- Where it is:
 - <http://wordnetweb.princeton.edu/perl/webwn>
- Libraries
 - Python: WordNet from NLTK
 - <http://www.nltk.org/Home>
 - Java:
 - JWNL, extJWNL on sourceforge

Extended WordNet (XWN)

- WordNet with syntactic and semantic annotations over its glosses.
- Contains logical forms and disambiguated glosses.
- XWN 2.0-1 is based on WordNet 2.0.

1. **excellent**, first-class, fantabulous -- (**of the highest quality**; "made an excellent speech"; "the school has excellent teachers"; "a first-class mind")

(TOP (S (NP (JJ **excellent**))
 (VP (VBZ **is**)
 (NP (NP (NN **something**))
 (PP (IN **of**)
 (NP (DT **the**) (JJS **highest**) (NN **quality**))))
 (.)))

excellent:JJ(x1) -> **of**:IN(x1, x2) **highest**:JJ(x1)
quality:NN(x1)

```
<wf pos="IN" >of</wf>
<wf pos="DT" >the</wf>
<wf pos="JJS" lemma="high" quality="silver" wnsn="1"
>highest</wf>
<wf pos="NN" lemma="quality" quality="normal" wnsn="2"
>quality</wf>
```

Accessing WordNet from Prolog

```
? - substance_of(water,X).
```

```
X = [tear|_G407]
```

```
? - has_substance(water,X).
```

```
X = [h2o|_G407]
```

```
?- part_of(leg,X).
```

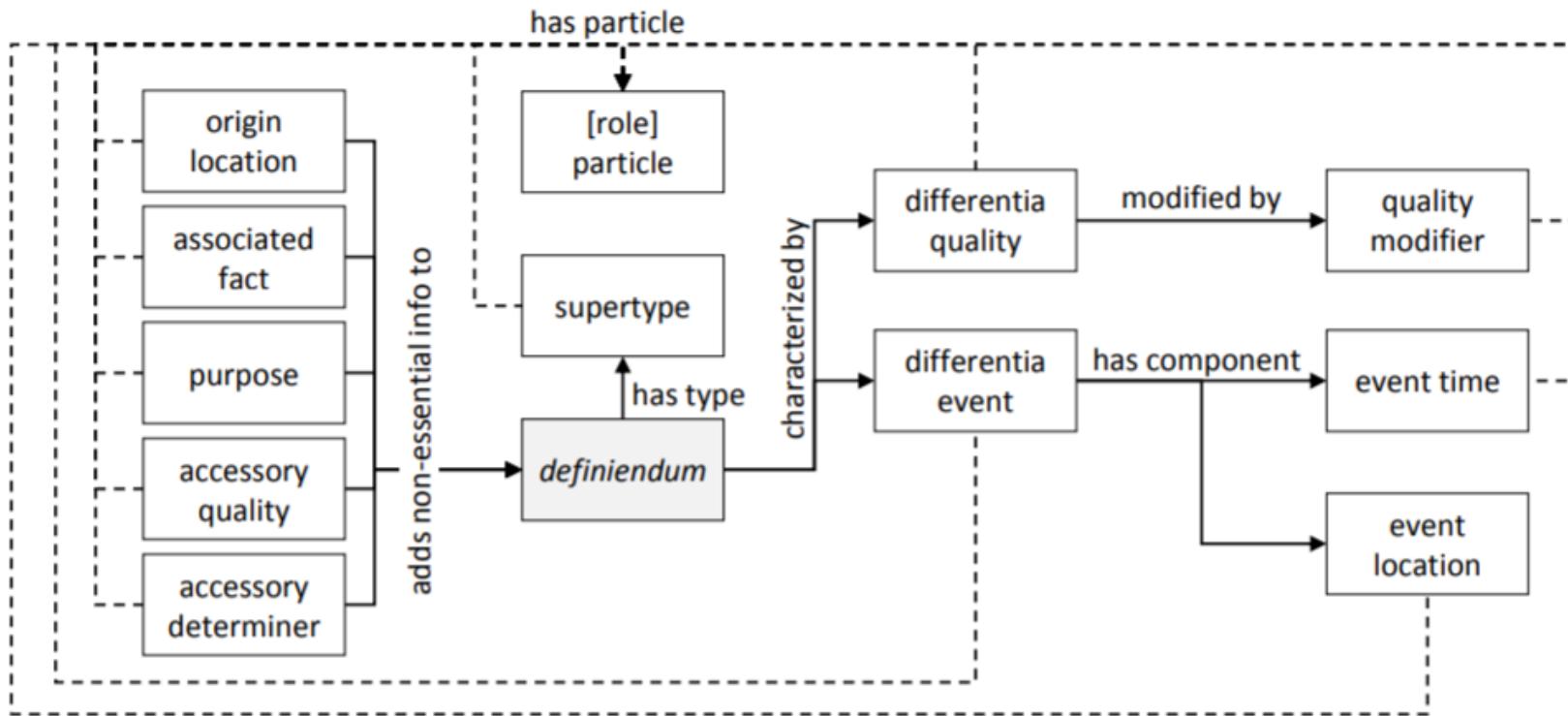
```
X = [table|_G407]
```

```
?- has_part(leg,X).
```

```
X = [knee|_G407]
```

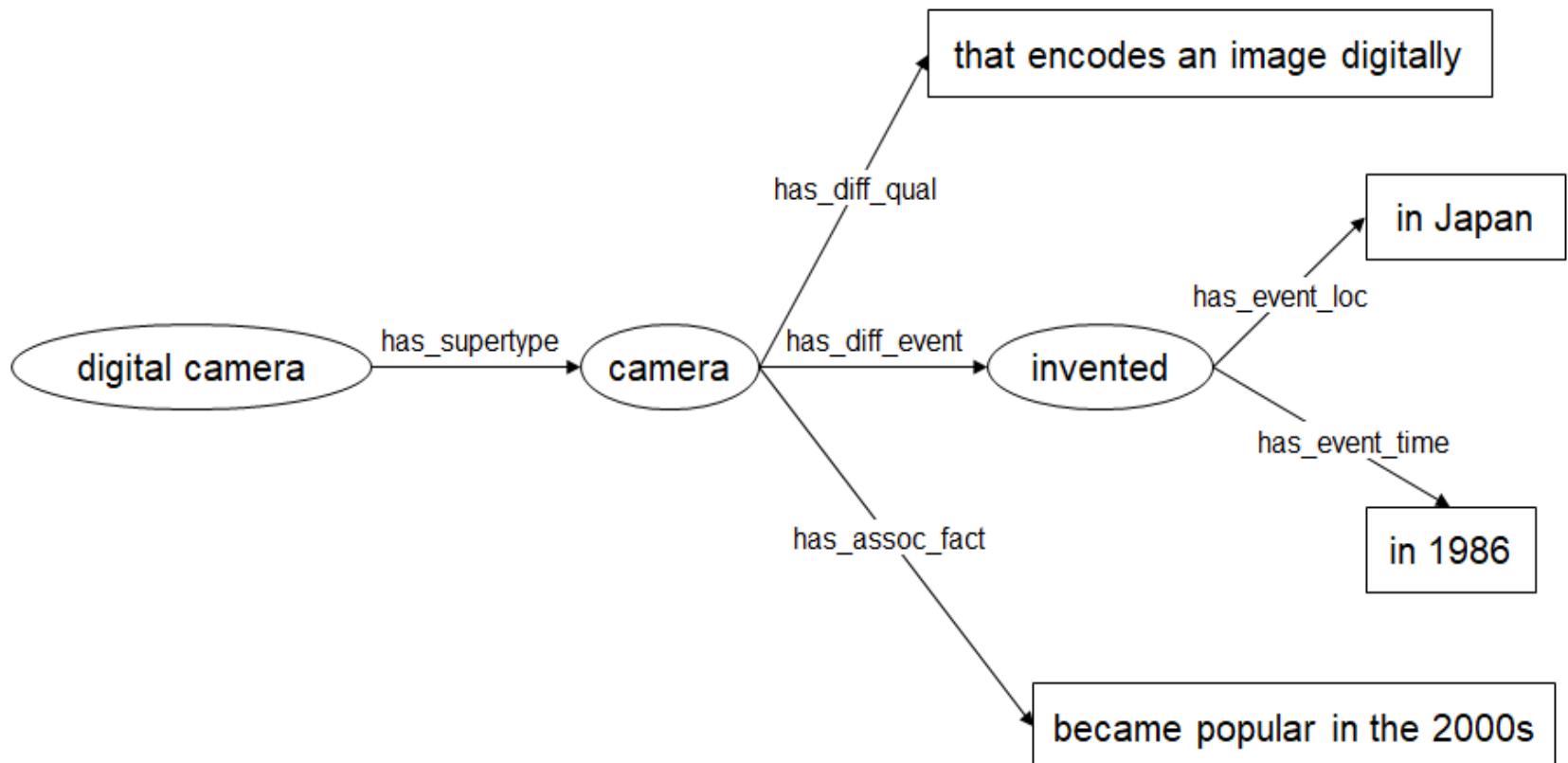
Semantic Roles for Lexical Definitions

Aristotle's classic theory of definition introduced important aspects such as the **genus-differentia definition pattern** and the **essential/non-essential property differentiation**.



Definition Graphs

digital camera: a camera invented in Japan in 1986 that encodes an image digitally and became popular in the 2000s



MeSH: Medical Subject Headings thesaurus from the National Library of Medicine

- **MeSH (Medical Subject Headings)**
 - 177,000 entry terms that correspond to 26,142 biomedical “headings”

Synset

- **Hemoglobins**

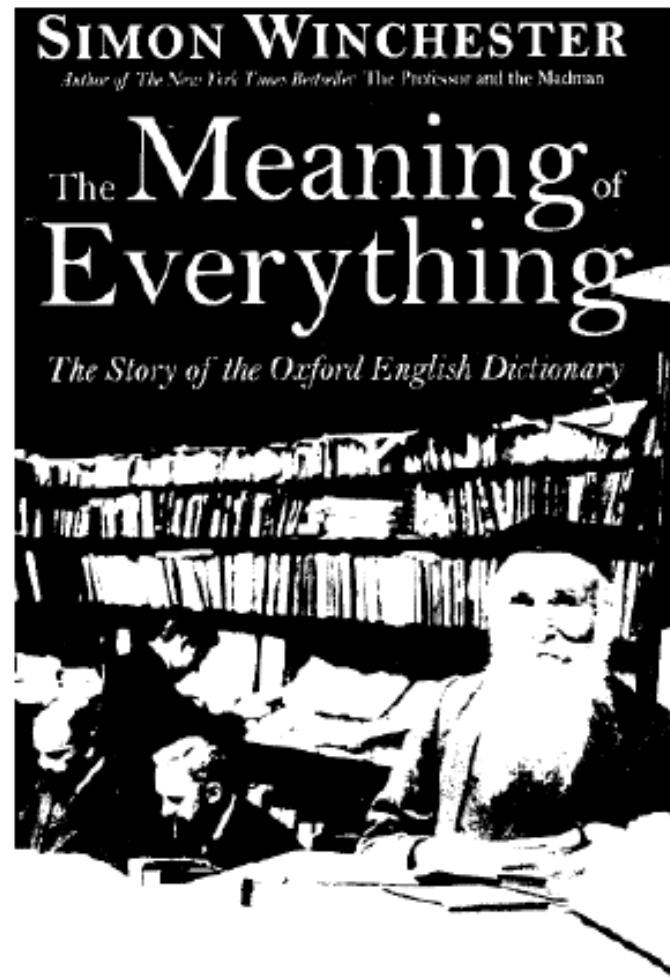
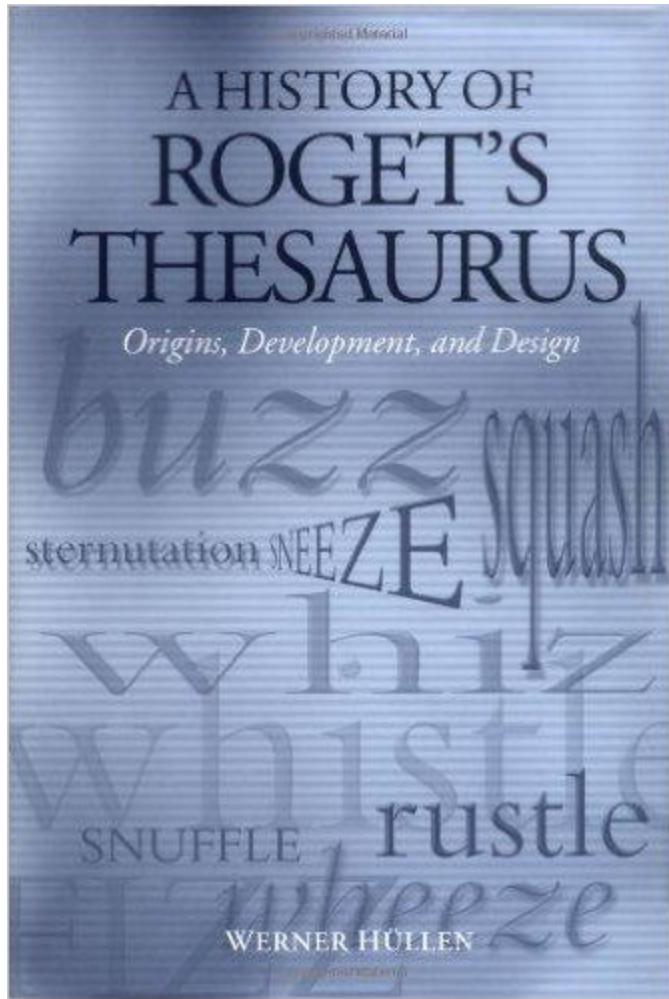
Entry Terms: Eryhem, Ferrous Hemoglobin, Hemoglobin

Definition: The oxygen-carrying proteins of ERYTHROCYTES. They are found in all vertebrates and some invertebrates. The number of globin subunits in the hemoglobin quaternary structure differs between species. Structures range from monomeric to a variety of multimeric arrangements

The MeSH Hierarchy

1. + Anatomy [A]
 2. + Organisms [B]
 3. + Diseases [C]
 4. - Chemicals and Drugs [D]
 - 1 o [Inorganic Chemicals \[D01\]](#) +
 - 2 o [Organic Chemicals \[D02\]](#) +
 - 3 o [Heterocyclic Compounds \[D03\]](#) +
 - 4 o [Polycyclic Compounds \[D04\]](#) +
 - 5 o [Macromolecular Substances \[D05\]](#) +
 - 6 o [Hormones, Hormone Substitutes, and](#)
 - 7 o [Enzymes and Coenzymes \[D08\]](#) +
 - 8 o [Carbohydrates \[D09\]](#) +
 - 9 o [Lipids \[D10\]](#) +
 - 10 o [Amino Acids, Peptides, and Proteins \[D11\]](#)
 - 11 o [Nucleic Acids, Nucleotides, and Nucleosides \[D12\]](#)
 - 12 o [Complex Mixtures \[D20\]](#) +
 - 13 o [Biological Factors \[D23\]](#) +
 - 14 o [Biomedical and Dental Materials \[D25\]](#)
 - 15 o [Pharmaceutical Preparations \[D26\]](#) +
 - 16 o [Chemical Actions and Uses \[D27\]](#) +
 5. + Analytical, Diagnostic and Therapeutic Techniques and Equipment [E]
 6. + Psychiatry and Psychology [F]
 7. + Phenomena and Processes [G]
- [Amino Acids, Peptides, and Proteins \[D12\]](#)
[Proteins \[D12.776\]](#)
[Blood Proteins \[D12.776.124\]](#)
[Acute-Phase Proteins \[D12.776.124.050\]](#) +
[Anion Exchange Protein 1, Erythrocyte \[D12.776.124.078\]](#)
[Ankyrins \[D12.776.124.080\]](#)
[beta 2-Glycoprotein I \[D12.776.124.117\]](#)
[Blood Coagulation Factors \[D12.776.124.125\]](#) +
[Cholesterol Ester Transfer Proteins \[D12.776.124.197\]](#)
[Fibrin \[D12.776.124.270\]](#) +
[Glycophorin \[D12.776.124.300\]](#)
[Hemocyanin \[D12.776.124.337\]](#)
► [Hemoglobins \[D12.776.124.400\]](#)
[Carboxyhemoglobin \[D12.776.124.400.141\]](#)
[Erythrocytins \[D12.776.124.400.220\]](#)

Curating Definitions: A Tour de Force



The theoretical distinction between dictionaries and encyclopaedias has traditionally been an issue of central importance for **lexicologists** (linguists who study word meaning) and **lexicographers** (dictionary writers).

The Dictionary View

- The dictionary view treats knowledge of word meaning as distinct from cultural knowledge, social knowledge and physical knowledge.
- Componential analysis or semantic decomposition approach:
 - word meaning is modelled in terms of semantic features or primitives.

bachelor is represented
as [MALE,ADULT,MARRIED]

Is the pope a *bachelor*?

Prototypes

“best example” of a category: e.g. *blackbird* vs. *penguin* for the category ‘bird’. But notice that the prototype may be abstract.



- not necessarily incompatible with feature theories
- fuzzy boundaries
- family resemblance



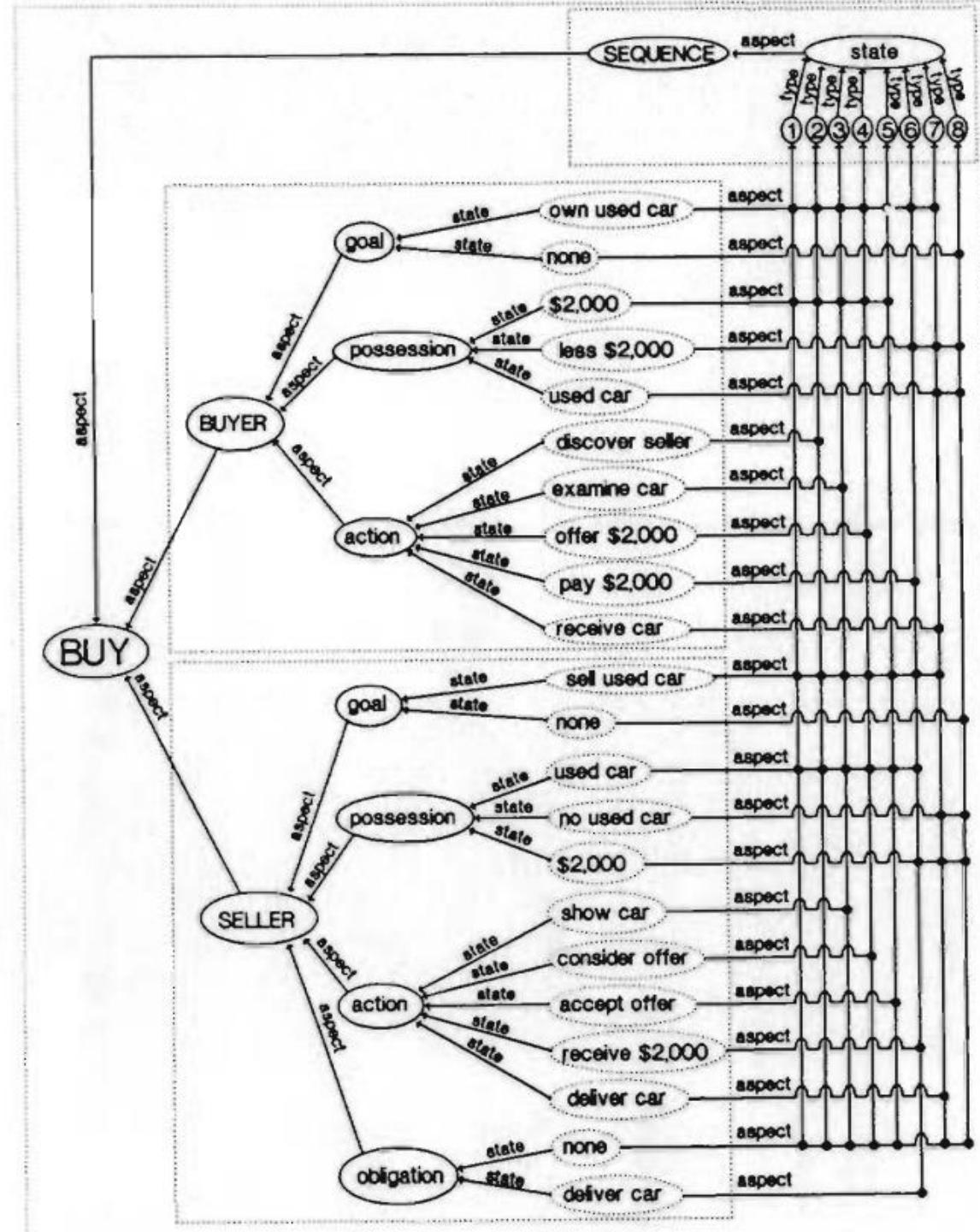
rank	category				
	BIRD	FRUIT	VEHICLE	FURNITURE	WEAPON
top eight					
1	robin	orange	automobile	chair	gun
2	sparrow	apple	station wagon	sofa	pistol
3	bluejay	banana	truck	couch	revolver
4	bluebird	peach	car	table	machine gun
5	canary	pear	bus	easy chair	rifle
6	blackbird	apricot	taxi	dresser	switchblade
7	dove	tangerine	jeep	rocking chair	knife
8	lark	plum	ambulance	coffee table	dagger
...
middle ranks					
26*	hawk	tangelo	subway	lamp	whip
27	raven	papaya	trailer	stool	ice pick
28	goldfinch	honeydew	cart	hassock	slingshot
29	parrot	fig	wheelchair	drawers	fists
30	sandpiper	mango	yacht	piano	axe
...
last five					
51*	ostrich	nut	ski	picture	foot
52	titmouse	gourd	skateboard	closet	car
53	emu	olive	wheelbarrow	vase	glass
54	penguin	pickle	surfboard	fan	screwdriver
55	bat	squash	elevator	telephone	shoes

* Since the total number of listed items varied between 50 and 60, the numbers of middle and bottom ranks are not identical with the original ranks for all categories.

Figure 1.3 A selection of examples from Rosch's goodness-of-example rating tests (Rosch 1975)

Frame Semantics

<https://framenet.icsi.berkeley.edu/fndrupal/frameIndex>



Commonsense Data (ConceptNet)

<http://conceptnet5.media.mit.edu/>

knife

knife – CapableOf → spread butter

knife can spread butter

knife – CapableOf → spread peanut butter

A knife can spread peanut butter

knife – UsedFor → stab

knife is for stabbing

knife – AtLocation → in kitchen

Something you might find in a kitchen is a knife.

knife – UsedFor → cut food

knife may be used to cut food.

cut – RelatedTo → knife

cut is related to knife

knife – AtLocation → kitchen drawer

You are likely to find a knife in a kitchen drawer

knife – MadeOf → steel

a knife can be made from steel.

knife – UsedFor → butter

a knife is used for butter

knife – IsA → tool

A knife is a type of tool

knife – AtLocation → kitchen

*Something you find in the kitchen is knife

knife – UsedFor → cut

When you want to cut, you will use knife.

knife – AtLocation → drawer

*Something you find in a drawer is a knife

knife – IsA → weapon

Kinds of weapons : knife

knife – UsedFor → eat

When you want to eat, you will use a knife.

machete – IsA → knife

a machete is a kind of a knife.

knife – AtLocation → store

*Something you find at a store is knives

blade – PartOf → knife

The blade is part of a knife

knife – CapableOf → butter bread

a knife can butter bread

in kitchen – AtLocation → knife

Something you find a knife is in the kitchen.

$$A \sqsubseteq B$$
$$A \sqsubseteq \neg B$$
$$A \sqcap B$$
$$A \sqcup B$$
$$A \equiv B$$
$$\top \sqsubseteq \forall P.A$$
$$\exists/\forall P.A$$

**Ontologies &
Description Logics**

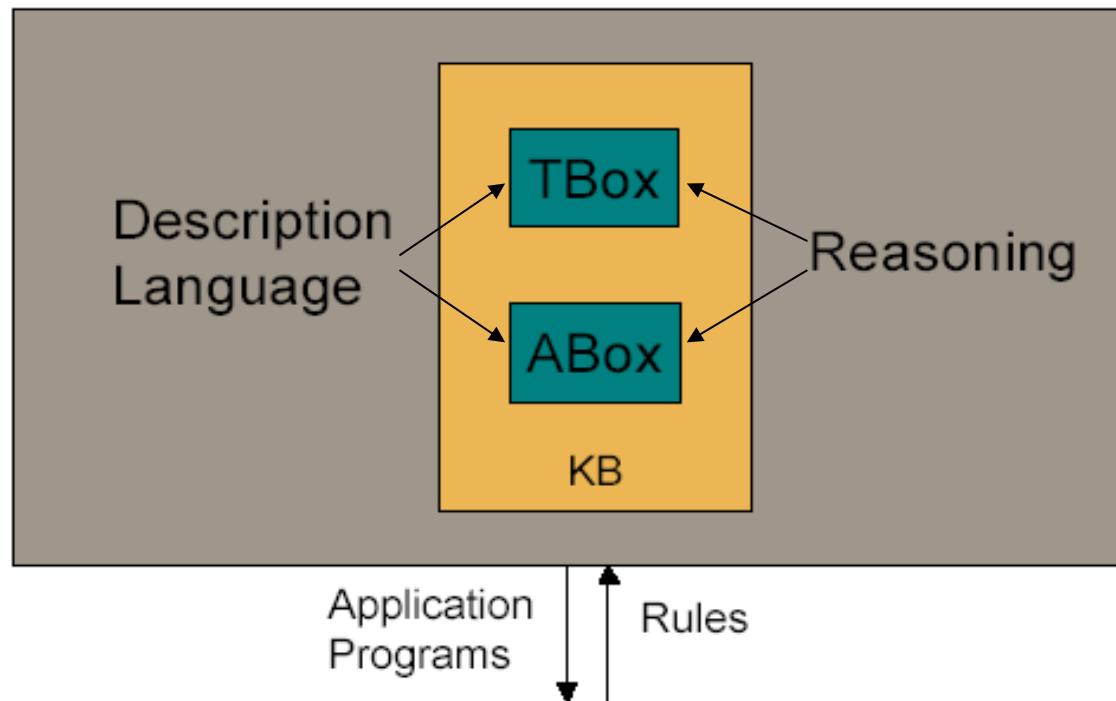
Description Logic

- Description Logics
 - Overcome the ambiguities of early semantic networks and frames
- Well-studied and decidable (most DL languages)
- Tight coupling between theory and practice

TBox and ABox

- TBox: terminology
 - The vocabulary of an application domain:
 - Concepts: sets of individuals
 - Roles: binary relationships between individuals.
 - Examples:
 - Concepts: Person, Female, Mother
 - Role: hasChild, meaning that some person is the child of some other
- ABox: assertions
 - About named individuals in terms of this vocabulary
 - Example
 - Elizabeth and Charles are Persons. We write this as Person(Elizabeth), and Person(Charles).
 - Individuals, like “myCar”, have attributes, like “color”, and those attributes have values, like “red”. When this happens we say that red is the colorOf attribute of myCar.
We write this as colorOf(myCar, red).

Architecture of a DL System

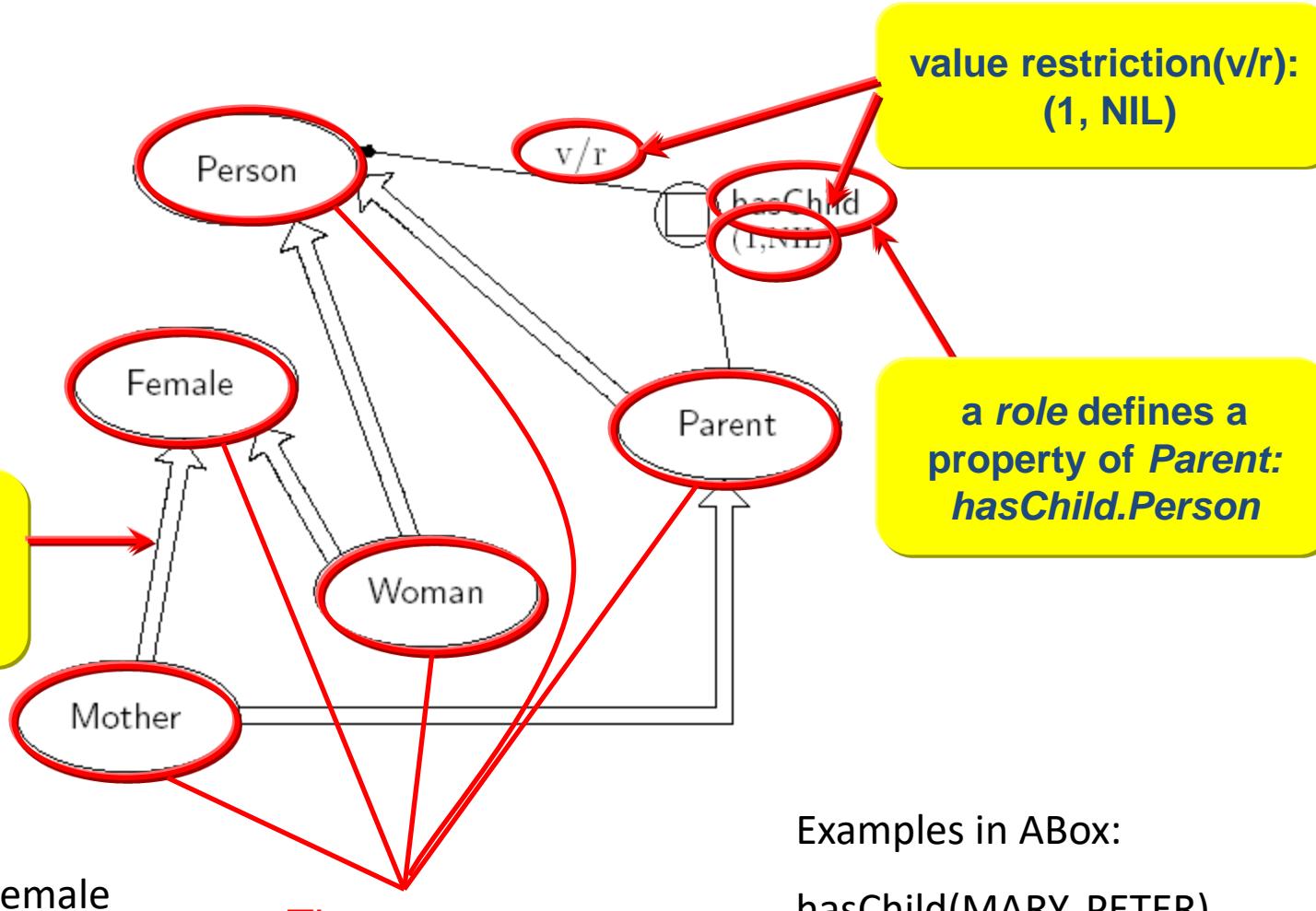


Formulas

- Building blocks that allow complex descriptions of concepts and roles.
 - Example (we'll look at the syntax in more detail soon.)
 - A Woman is a Female Person
 - Woman = Person \sqcup Female
 - A Mother is a Woman and she has a child
 - Mother = Woman \sqcup hasChild.T
- The TBox can be used to assign names to complex descriptions.

*We will use the terms *description* and *concept* interchangeably.*

An Example about Family Relationships



Examples in TBox:

Woman ⊑ Person \sqcup Female

Mother ⊑ Woman \sqcup hasChild.Person

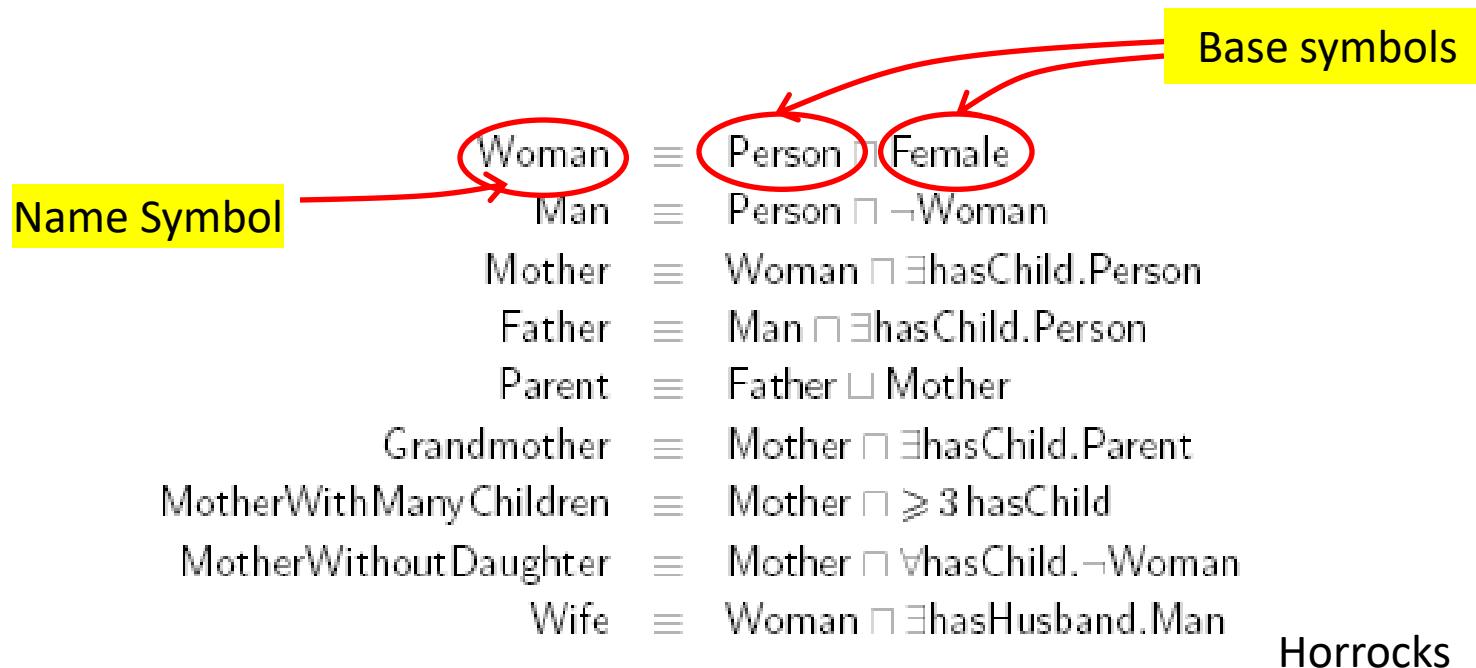
Examples in ABox:

hasChild(MARY, PETER)

Father(PETER)

Name Symbols vs. Base Symbols

- Atomic concepts occurring in a TBox \mathbf{T} can be divided into two sets, name symbols $N_{\mathbf{T}}$ (or defined concepts) and base symbols $B_{\mathbf{T}}$ (or primitive concepts, occur only on the right-hand side)
- A base interpretation for \mathbf{T} only interprets the base symbols.



DL for the Semantic Web

- Web Ontology Language (OWL): W3C Recommendation on 10 Feb 2004
- builds on RDF and RDF Schema and adds more vocabulary for describing properties and classesExtends existing Web standards
- has three increasingly-expressive sublanguages:
 - OWL Lite (based on DL SHIF (D)) ,
 - OWL DL (based on DL SHOIN(D)),
 - and OWL Full (OWL DL + RDF)
- benefits from many years of DL research
 - Well defined **semantics**
 - **Formal properties** well understood (complexity, decidability)
 - Known **reasoning algorithms**
 - Implemented systems (highly optimised)

OWL Class Constructor

Constructor	DL Syntax	Example	Modal Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male	$C_1 \wedge \dots \wedge C_n$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer	$C_1 \vee \dots \vee C_n$
complementOf	$\neg C$	\neg Male	$\neg C$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} \sqcup {mary}	$x_1 \vee \dots \vee x_n$
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor	$[P]C$
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer	$\langle P \rangle C$
maxCardinality	$\leq n P$	≤ 1 hasChild	$[P]_{n+1}$
minCardinality	$\geq n P$	≥ 2 hasChild	$\langle P \rangle_n$

OWL Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} \equiv {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg \{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
inverseOf	$P_1 \equiv P_2^-$	hasChild \equiv hasParent $^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor $^+$ \sqsubseteq ancestor
functionalProperty	$T \sqsubseteq \leqslant 1P$	T $\sqsubseteq \leqslant 1$ hasMother
inverseFunctionalProperty	$T \sqsubseteq \leqslant 1P^-$	T $\sqsubseteq \leqslant 1$ hasSSN $^-$

Ontology Editors



pizza.owl Protégé 3.2 beta (file:\C:\Nick\Applications\Protege_3.2_b235\examples\pizza\pizza.owl.pprj, OWL / RDF...)

File Edit Project OWL Code Tools Window Help

OWLClasses Properties Forms Individuals Metadata

SUBCLASS EXPLORER For Project: pizza.owl

CLASS EDITOR For Class: RealItalianPizza (instance of owl:Class) Inferred View

Annotations

Property	Value	Lang
rdfs:comment	This defined class has conditions that are part of the definition: ie any Pizza that has the country of origin, Italy is a RealItalianPizza. It also has conditions that merely describe the members - that all RealItalianPizzas must only have ThinAndCrispy bases.	en
rdfs:label	PizzaitalianaReal	pt

Asserted Conditions

- Pizza NECESSARY & SUFFICIENT
 - hasCountryOfOrigin **has** Italy ≡
 - hasBase **only** ThinAndCrispyBase NECESSARY
 - hasBase **some** PizzaBase INHERITED [from Pizza]

Disjoints

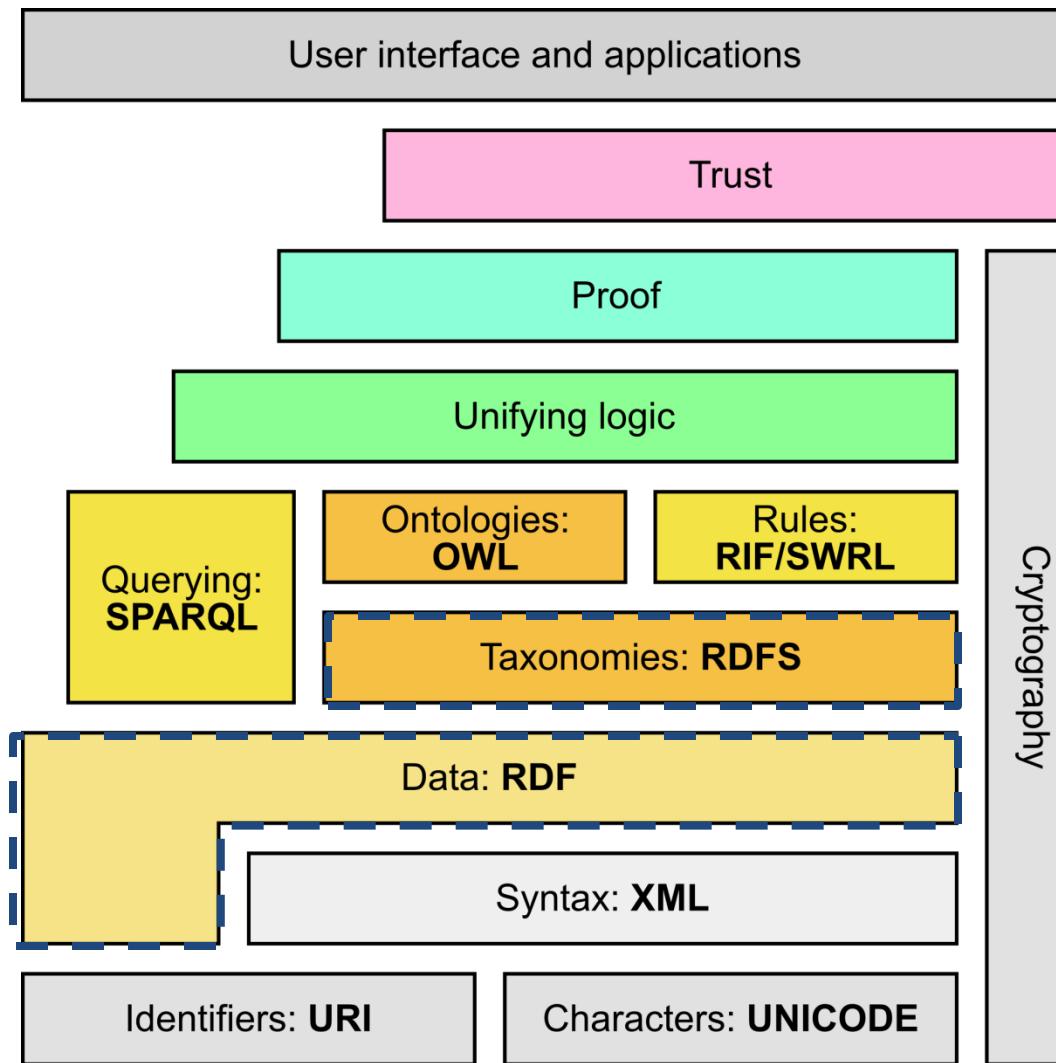
Logic View Properties View

The screenshot shows the Protégé 3.2 beta interface for editing the ontology file pizza.owl. The main window is divided into several panels: 'SUBCLASS EXPLORER' on the left showing the asserted hierarchy of classes like owl:Thing, DomainConcept, Pizza, and various pizza types; 'CLASS EDITOR' in the center displaying annotations for the class 'RealItalianPizza' (with properties rdfs:comment and rdfs:label); 'Asserted Conditions' panel listing constraints such as 'hasCountryOfOrigin has Italy' (necessary & sufficient), 'hasBase only ThinAndCrispyBase' (necessary), and 'hasBase some PizzaBase' (inherited from Pizza); and 'Disjoints' panel at the bottom. The interface uses a color-coded system for classes (yellow), properties (blue), and individuals (green). The title bar indicates the file path and the OWL/RDF format.



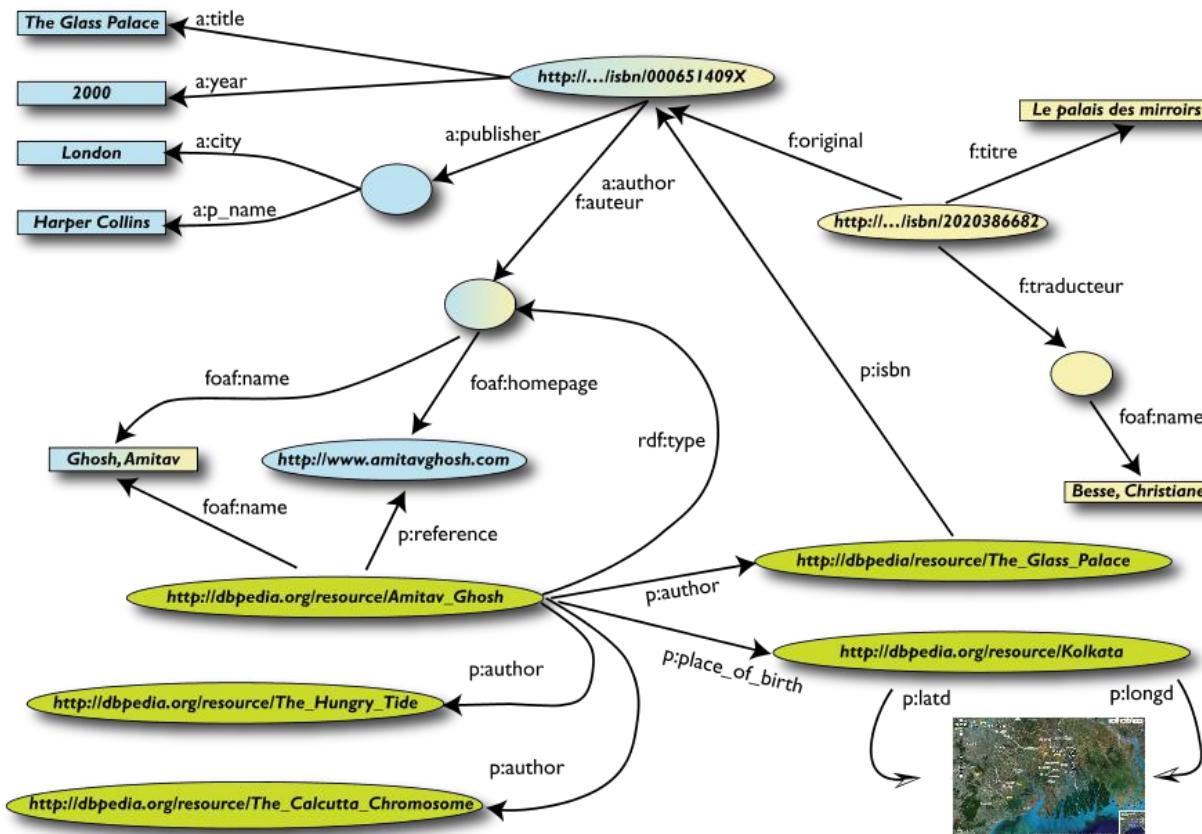
Semantic Web

Semantic Web Stack



RDF: a Direct Connected Graph based Model

- Different interconnected triples lead to a more complex graphic model.
- Basically a RDF document is a direct connect graph.



RDF Basics

- RDF is a language that enable to describe making statements on resources
 - John is father of Bill
- Statement (or triple) as a logical formula $P(x, y)$, where the binary predicate P relates the object x to the object y
- Triple data model:
`<subject, predicate, object>`
 - **Subject**: Resource or blank node
 - **Predicate**: Property
 - **Object**: Resource (or collection of resources), literal or blank node
- Example:
`<ex:john, ex:father-of, ex:bill>`
- RDF offers only binary predicates (properties)

RDF Vocabulary Description Language

- We need a language for defining RDF types:
 - Define classes:
 - “*#Student* is a class”
 - Relationships between classes:
 - “*#Student* is a sub-class of *#Person*”
 - Properties of classes:
 - “*#Person* has a property *hasName*”
- RDF Schema is such a language.

RDF Vocabulary Description Language

- Classes:
`<#Student, rdf:type, #rdfs:Class>`
- Class hierarchies:
`<#Student, rdfs:subClassOf, #Person>`
- Properties:
`<#hasName, rdf:type, rdf:Property>`
- Property hierarchies:
`<#hasMother, rdfs:subPropertyOf, #hasParent>`
- Associating properties with classes (a):
 - “The property `#hasName` only applies to `#Person`”
`<#hasName, rdfs:domain, #Person>`
- Associating properties with classes (b):
 - “The type of the property `#hasName` is `#xsd:string`”
`<#hasName, rdfs:range, xsd:string>`

RDFS Vocabulary

- RDFS extends the RDF vocabulary

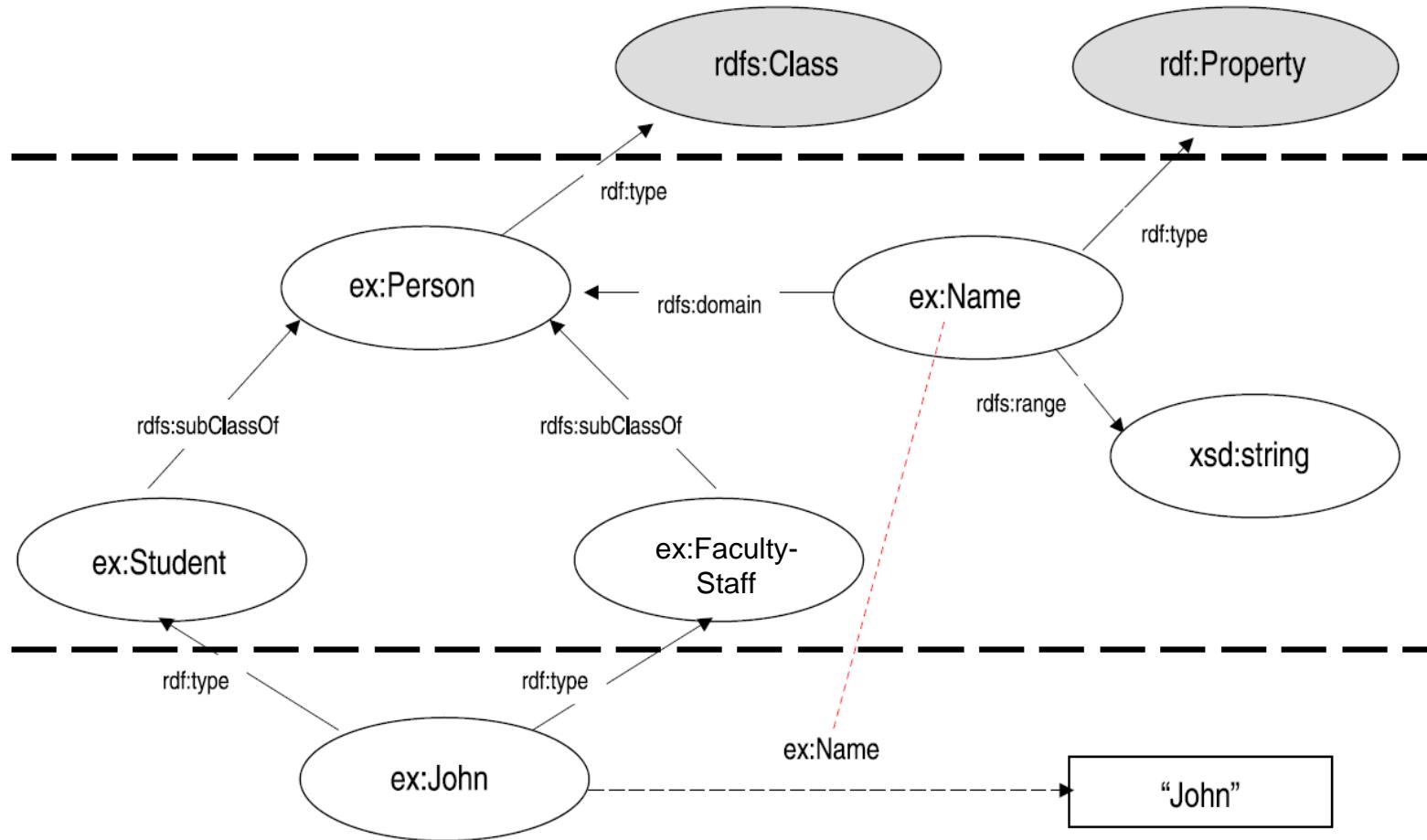
RDFS Classes

- `rdfs:Resource`
- `rdfs:Class`
- `rdfs:Literal`
- `rdfs:Datatype`
- `rdfs:Container`
- `rdfs:ContainerMembershipProperty`

RDFS Properties

- `rdfs:domain`
- `rdfs:range`
- `rdfs:subPropertyOf`
- `rdfs:subClassOf`
- `rdfs:member`
- `rdfs:seeAlso`
- `rdfs:isDefinedBy`
- `rdfs:comment`
- `rdfs:label`

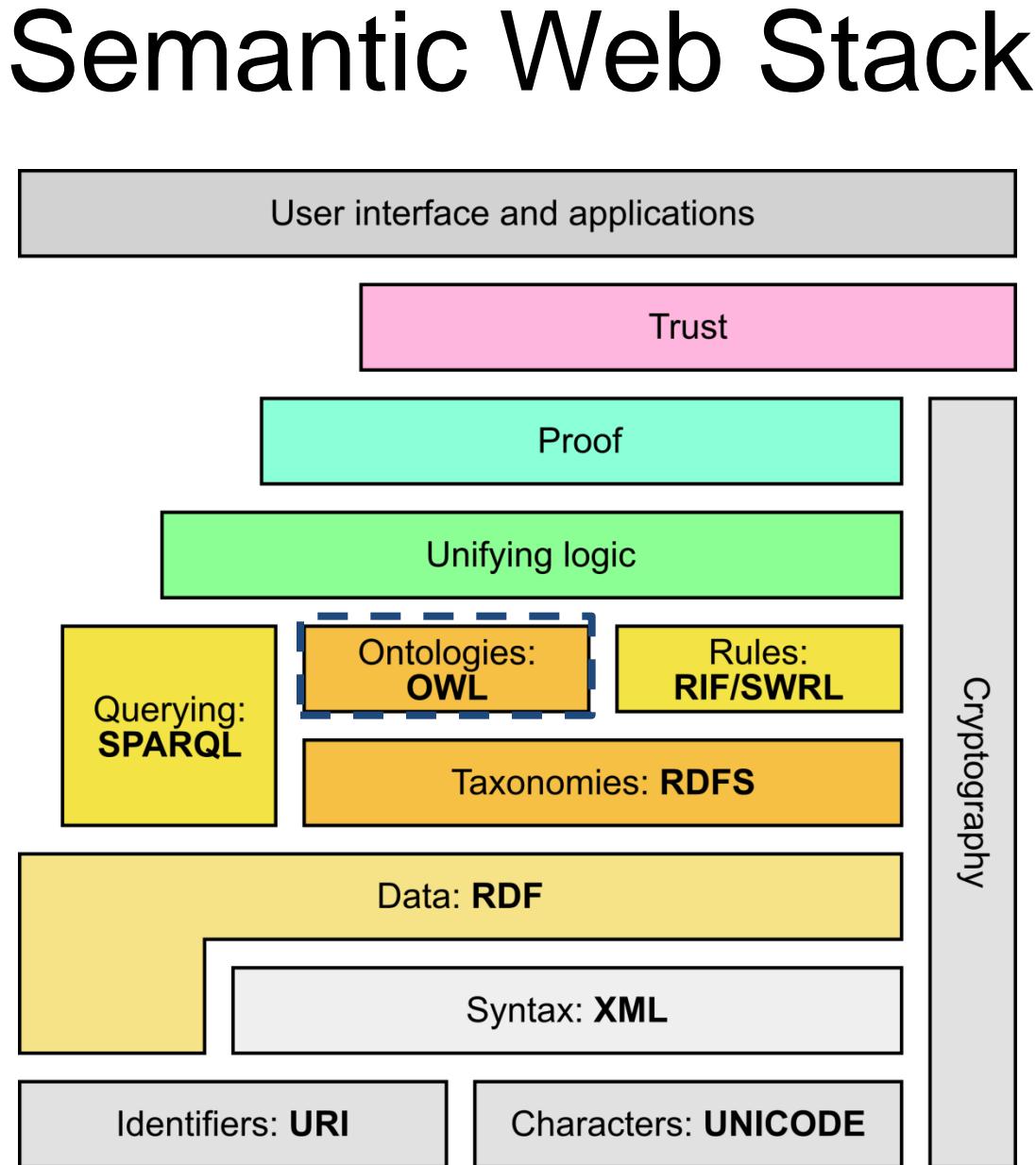
RDFS Example



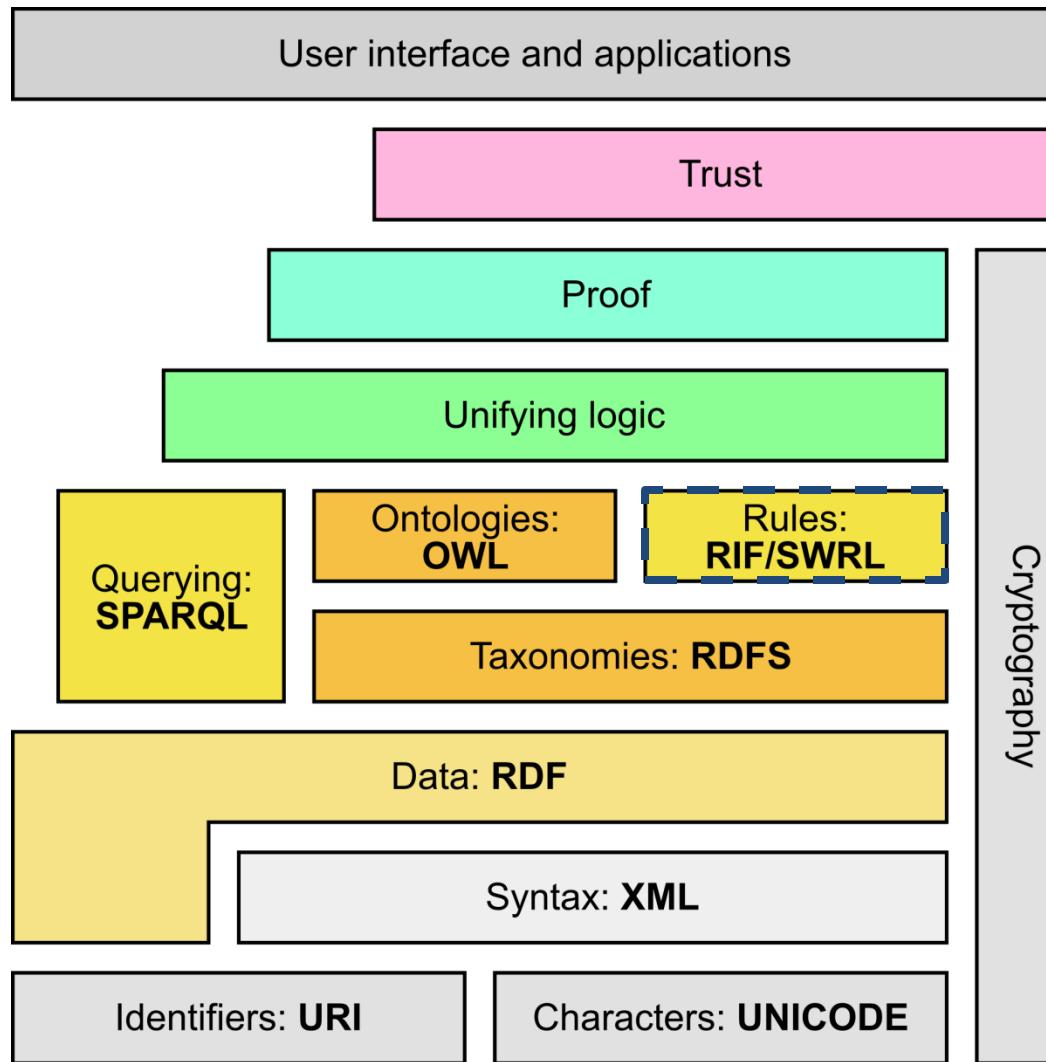
Vocabulary Layer

RDFS Layer

RDF Layer



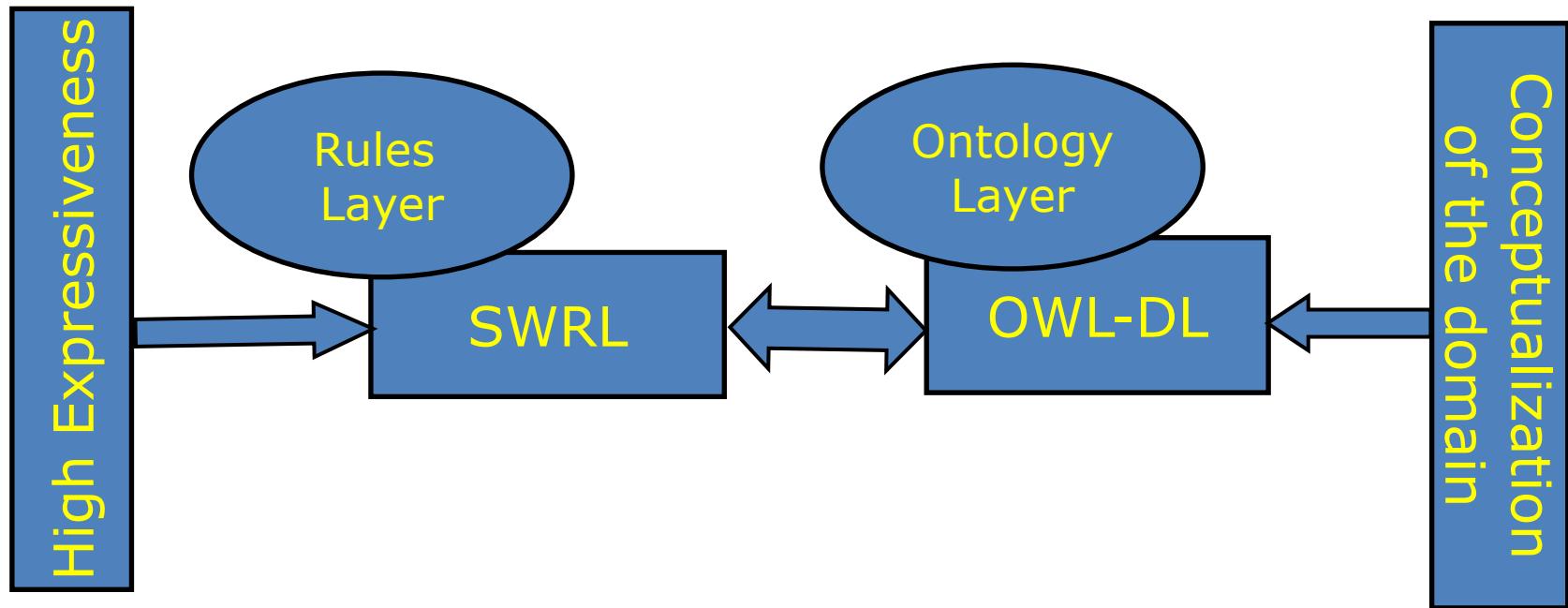
Semantic Web Stack



Adapted from http://en.wikipedia.org/wiki/Semantic_Web_Stack

What is SWRL?

Ontology languages do not offer the expressiveness we want → Rules do it well.



SWRL Rule

An atom is an expression of the form: ***P(arg1 arg2,...)***

- **P** is a predicate symbol (classes, properties...)
- Arguments of the expression: ***arg1, arg2,...*** (individuals, data values or variables)

Example SWRL Rule:

Person(?p) ^ hasSibling(?p,?s) ^ Man(?s) → hasBrother(?p,?s)

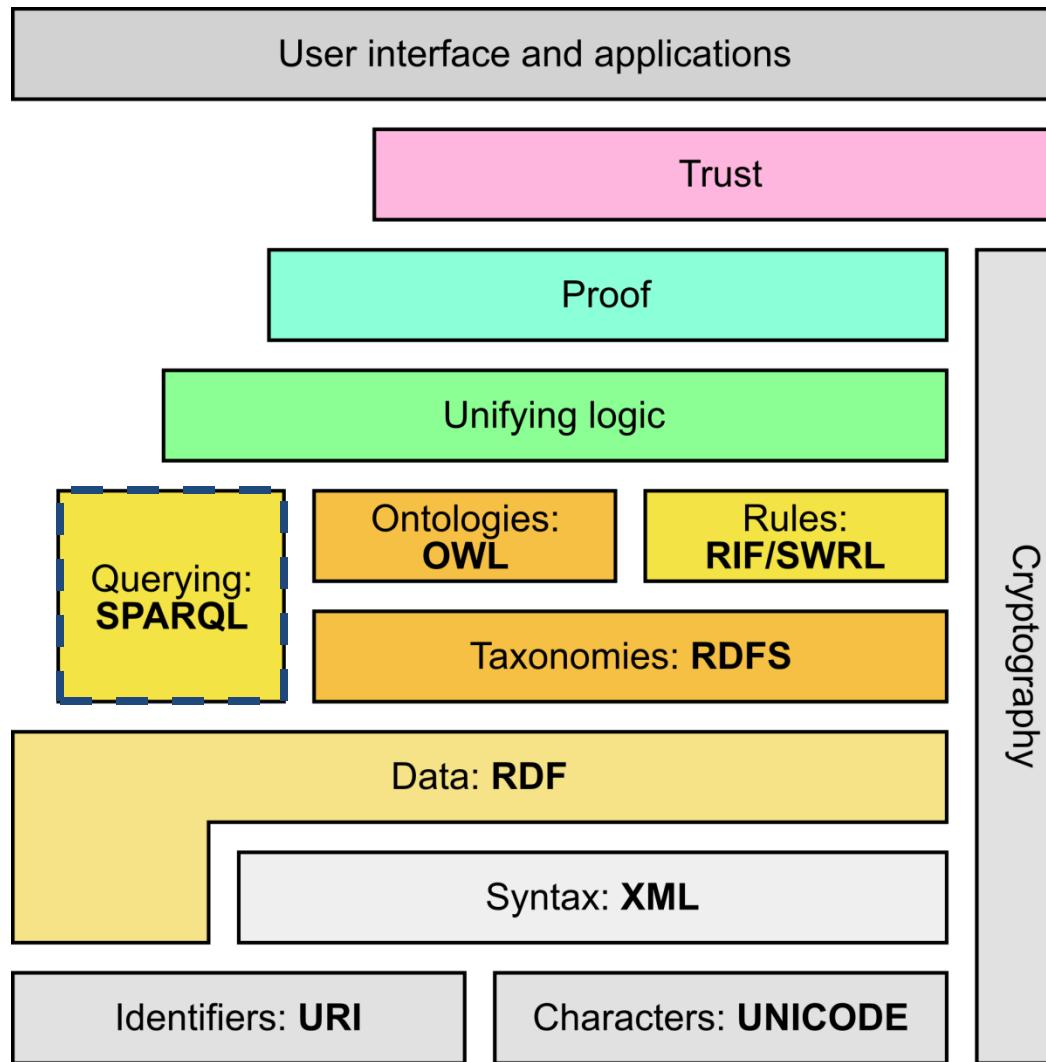


antecedent



consequent

Semantic Web Stack



Adapted from http://en.wikipedia.org/wiki/Semantic_Web_Stack

SPARQL

- Query Language for RDF.

Find all DrugBank drugs along with dosage and disease indication information.

PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>

PREFIX db: <http://bio2rdf.org/drugbank_vocabulary:>

SELECT

?drug_name ?dosage ?indication

WHERE {

?drug a db:Drug .

?drug rdfs:label ?drug_name .

OPTIONAL { ?drug db:dosage ?dosage . }

OPTIONAL { ?drug db:indication ?indication . }

}

SPARQL (Yes/No Query)

Is the Amazon river longer than the Nile River?

```
PREFIX prop: <http://dbpedia.org/property/>
ASK {
    <http://dbpedia.org/resource/Amazon_River> prop:length ?amazon .
    <http://dbpedia.org/resource/Nile> prop:length ?nile .
    FILTER(?amazon > ?nile) .
}
```

http://dbpedia.org/sparql

The screenshot shows a web-based SPARQL query editor. At the top, there is a header bar with navigation icons (back, forward, search) and a URL field containing "de.dbpedia.org/sparql". Below the header is a blue title bar labeled "Virtuoso SPARQL Query Editor". Underneath the title bar, there is a text input field for "Default Data Set Name (Graph IRI)" which is currently empty. The main area is titled "Query Text" and contains the following SPARQL query:

```
PREFIX prop: <http://dbpedia.org/property/>
ASK
{
  <http://dbpedia.org/resource/Amazon_River> prop:length ?amazon .
  <http://dbpedia.org/resource/Nile> prop:length ?nile .
  FILTER(?amazon > ?nile)
}
```

APIs

- Helpful for building AI applications.
- Typical Support:
 - A RDF API
 - Serialization: Reading and writing RDF in RDF/XML, N3 and N-Triples
 - An OWL API
 - In-memory and persistent storage
 - SPARQL query engine

Recommended Reading

1

Frames, Concepts, and Conceptual Fields

Lawrence W. Barsalou
University of Chicago

http://barsaloulab.org/Online_Articles/1992-Barsalou-chap-frames.pdf

Recommended Reading

Intelligent Machines

An AI with 30 Years' Worth of Knowledge Finally Goes to Work

An effort to encode the world's knowledge in a huge database has sometimes seemed impractical, but those behind the technology say it is finally ready.

by Will Knight March 14, 2016

Having spent the past 31 years memorizing an astonishing collection of general knowledge, the artificial-intelligence engine created by Doug Lenat is finally ready to go to work.

Lenat's creation is **Cyc**, a knowledge base of semantic information designed to give computers some understanding of how things work in the real world.

<https://www.technologyreview.com/s/600984/an-ai-with-30-years-worth-of-knowledge-finally-goes-to-work/>

Symbolic AI

Andre Freitas



Photo by Vasilyev Alexandr

Acknowledgements

- Based on the slides of:
 - J. Sowa, Existential & Conceptual Graphs
 - Cristiano Broccias, Cognitive Lexical Semantics
 - Taboada, Introduction to Rhetorical Structure Theory
 - CMSC 473/673 (UMBC), Semantic Roles and Frames

This Lecture

- Representing complex statements.
 - We will focus on events
- Representing discourse elements.

Representing Events

- How do we represent time and temporal relationships between events?
 - It seems only yesterday that Martha Stewart was in prison but now she has a popular TV show. There is no justice.
- Where do we get temporal information?
 - Verb tense
 - Temporal expressions
 - Sequence of presentation

Representing Events

- Temporal, tense logic.
 - I arrived in New York.
 - I am arriving in New York.
 - I will arrive in New York.

$$\exists w \text{ } ISA(w, Arriving) \wedge \text{Arriver}(w, Speaker) \wedge \text{Destination}(w, NewYork)$$

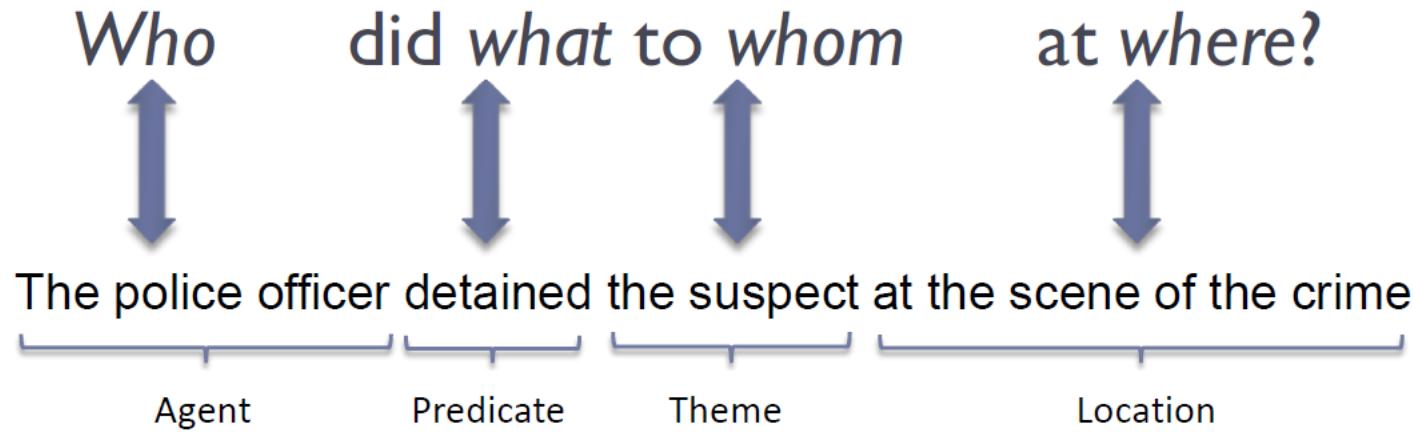
- The temporal information provided by the tense of the verbs can be exploited by predicated additional information about the **event variable w**.

Representing Events

We can add temporal variables representing the interval corresponding to the event, the end point of the event, and temporal predicates relating this end point to the current time as indicated by the tense of the verb.

$$\exists i, e, w, t \ ISA(w, Arriving) \\ \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork) \\ IntervalOf(w, i) \wedge EndPoint(i, e) \wedge Precedes(e, Now)$$
$$\exists i, e, w, t \ ISA(w, Arriving) \\ \wedge Arriver(w, Speaker) \wedge Destination(w, NewYork) \\ IntervalOf(w, i) \wedge MemberOf(i, Now)$$

Semantic Roles: Frame!



Thematic Roles

Sasha broke the window

Pat opened the door

Subjects of break and open:
Breaker and Opener

Specific to each event

Thematic Roles

Sasha broke the window

Breaker and **Opener** have something in common!

Pat opened the door

Volitional actors

Often animate

Direct causal responsibility for their events

Subjects of *break* and *open*:

Breaker and **Opener**

Specific to each event

Thematic roles are a way to capture this semantic commonality between *Breakers* and *Eaters*.

Thematic Roles

Sasha broke the window

Pat opened the door

Subjects of **break** and **open**: **Breaker** and
Opener

Specific to each event

Breaker and **Opener** have something in common!

Volitional actors

Often animate

Direct causal responsibility for their events

Thematic roles are a way to capture this semantic commonality between *Breakers* and *Eaters*.

They are both AGENTS.

The *BrokenThing* and *OpenedThing*, are THEMES.
prototypically inanimate objects affected in some way by the action

Modern formulation from
Fillmore (1966,1968), Gruber (1965)

Fillmore influenced by Lucien Tesnière's (1959) *Éléments de Syntaxe Structurale*,
the book that introduced dependency grammar

Typical Thematic Roles

Thematic Role	Definition	Example
AGENT	The volitional causer of an event	<i>The waiter spilled the soup.</i>
EXPERIENCER	The experiencer of an event	<i>John has a headache.</i>
FORCE	The non-volitional causer of the event	<i>The wind blows debris from the mall into our yards.</i>
THEME	The participant most directly affected by an event	<i>Only after Benjamin Franklin broke the ice...</i>
RESULT	The end product of an event	<i>The city built a <i>regulation-size baseball diamond</i>...</i>
CONTENT	The proposition or content of a propositional event	<i>Mona asked “<i>You met Mary Ann at a supermarket?</i>”</i>
INSTRUMENT	An instrument used in an event	<i>He poached catfish, stunning them with a <i>shocking device</i>...</i>
BENEFICIARY	The beneficiary of an event	<i>Whenever Ann Callahan makes hotel reservations <i>for her boss</i>...</i>
SOURCE	The origin of the object of a transfer event	<i>I flew in <i>from Boston</i>.</i>
GOAL	The destination of an object of a transfer event	<i>I drove to <i>Portland</i>.</i>

Verb Alternations (Diathesis Alternations)

Doris gave the book to Cary.

AGENT THEME GOAL

Doris gave Cary the book.

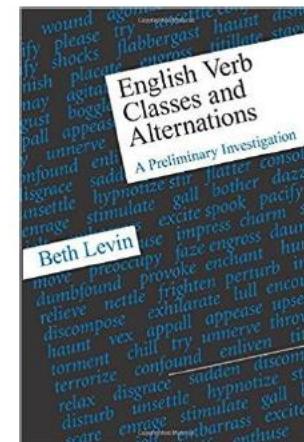
AGENT GOAL THEME

Break: AGENT, INSTRUMENT, or THEME as subject

Give: THEME and GOAL in either order

Levin (1993): 47 semantic classes (“**Levin classes**”) for
3100 English verbs and alternations. In online resource

VerbNet.



Alternative to Thematic Roles

PropBank

1. **Fewer roles:** generalized semantic roles,
defined as prototypes (Dowty 1991)

PROTO-AGENT

PROTO-PATIENT

FrameNet

2. **More roles:** Define roles specific to a group of predicates

PropBank Frame Files

agree.01

Arg0: Agree

Arg1: Proposition

Arg2: Other entity agreeing

Ex1: [Arg0 The group] *agreed* [Arg1 it wouldn't make an offer].

Ex2: [ArgM-TMP Usually] [Arg0 John] *agrees* [Arg2 with Mary]
[Arg1 on everything].

View Commonalities Across Sentences

increase.01 “go up incrementally”

Arg0: causer of increase

Arg1: thing increasing

Arg2: amount increased by, EXT, or MNR

Arg3: start point

Arg4: end point

[Arg0 Big Fruit Co.] increased [Arg1 the price of bananas].

[Arg1 The price of bananas] was increased again [Arg0 by Big Fruit Co.]

[Arg1 The price of bananas] increased [Arg2 5%].

Frege's Begriffsschrift for the Same Sentence

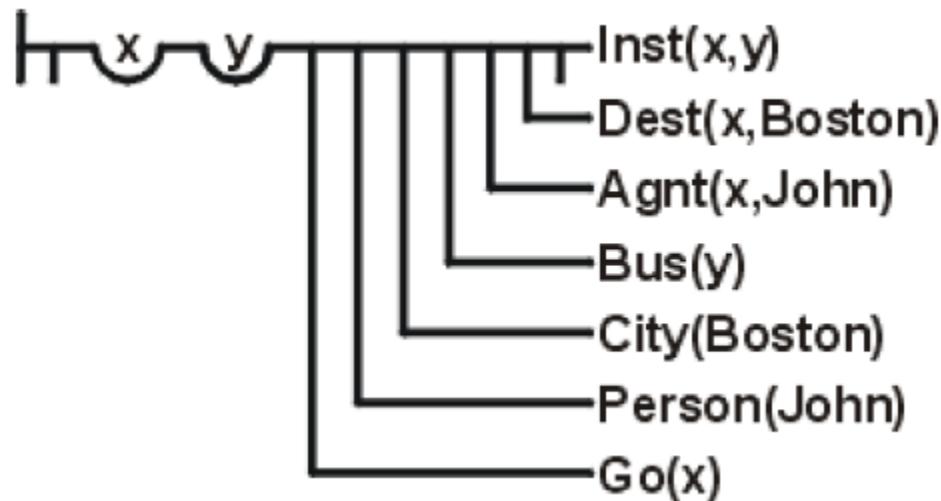
- Peirce's algebraic notation (1885):

$$\Sigma_x \Sigma_y ((\text{Go}(x) \bullet \text{Person(John)} \bullet \text{City(Boston)} \bullet \text{Bus}(y) \bullet \\ \text{Agnt}(x,\text{John}) \bullet \text{Dest}(x,\text{Boston}) \bullet \text{Inst}(x,y)))$$

- With Peano's choice of symbols:

$$(\exists x)(\exists y)(\text{Go}(x) \wedge \text{Person(John)} \wedge \text{City(Boston)} \wedge \text{Bus}(y) \\ \wedge \text{Agnt}(x,\text{John}) \wedge \text{Dest}(x,\text{Boston}) \wedge \text{Inst}(x,y))$$

Frege's Begriffsschrift for the Same Sentence



- Translation to Peirce-Peano notation:

$$\sim(\forall x)(\forall y)(\text{Go}(x) \supset (\text{Person}(\text{John}) \supset (\text{City}(\text{Boston}) \supset (\text{Bus}(y) \supset (\text{Agnt}(x, \text{John}) \supset (\text{Dest}(x, \text{Boston}) \supset \sim\text{Inst}(x, y)))))))$$

Frege's Begriffsschrift for the Same Sentence

- Translation to Peirce-Peano notation:

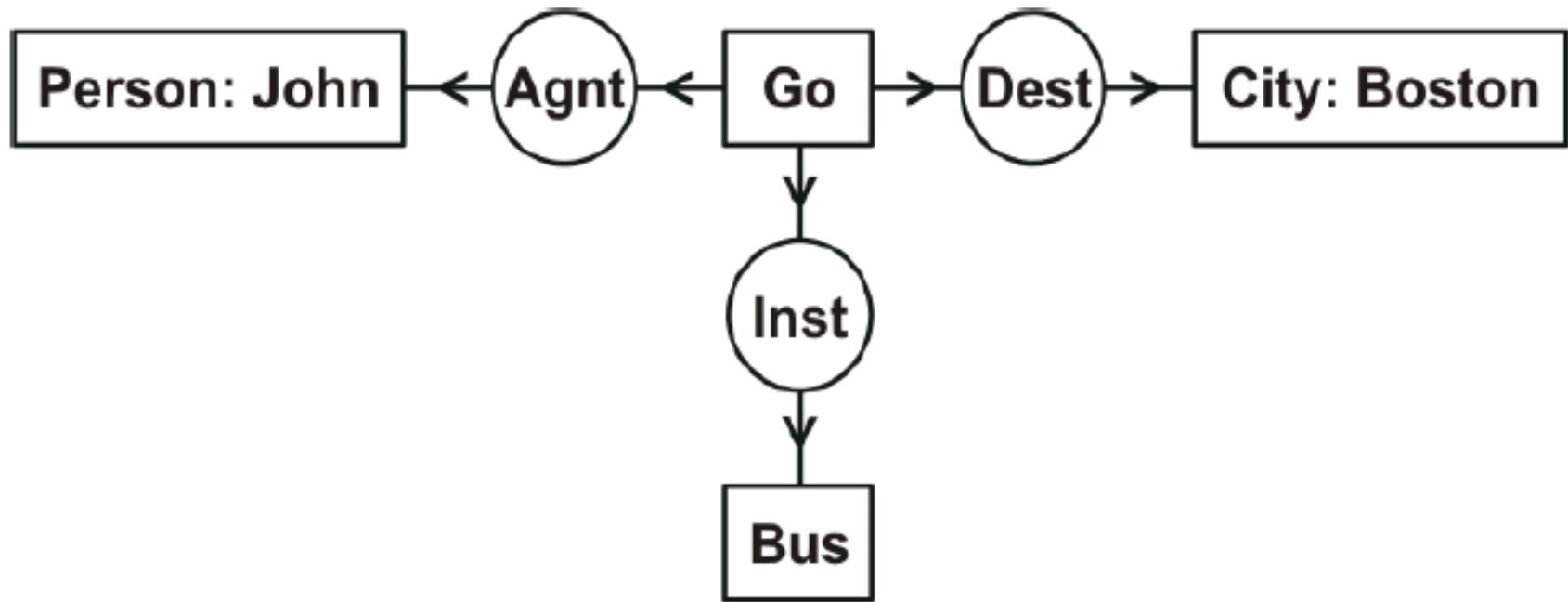
$$\neg(\forall x)(\forall y)(Go(x) \supset (Person(John) \supset (City(Boston) \supset (Bus(y) \supset (Agnt(x, John) \supset (Dest(x, Boston) \supset \neg Inst(x, y)))))))$$

- Equivalent in English:

It is false that for every x and y, if x is an instance of going then if John is a person then if Boston is a city then if y is a bus then if the agent of x is John then if the destination of x is Boston then the instrument of x is not y.

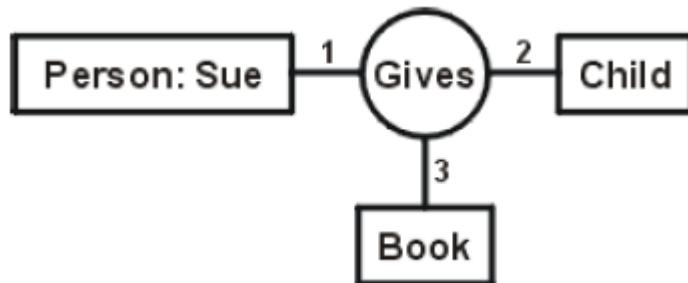
J. Sowa

Sentence: *John is going to Boston by bus*

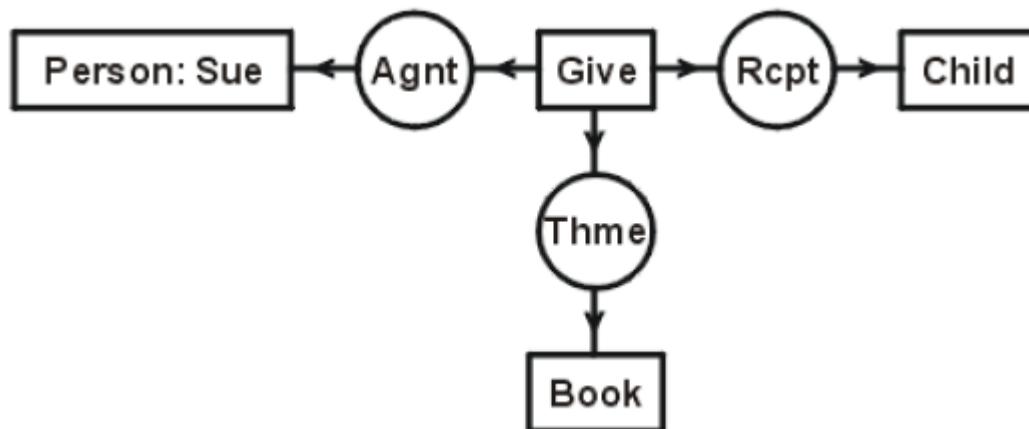


Existential Graphs

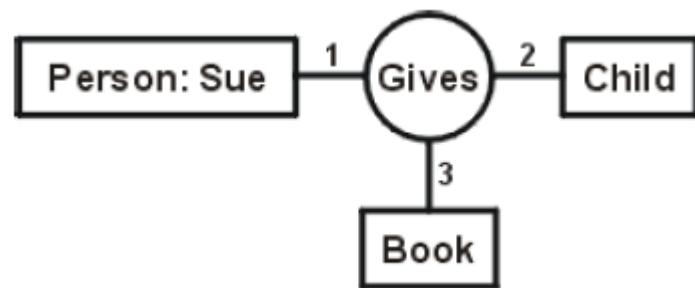
- Existential Graphs vs Conceptual Graphs



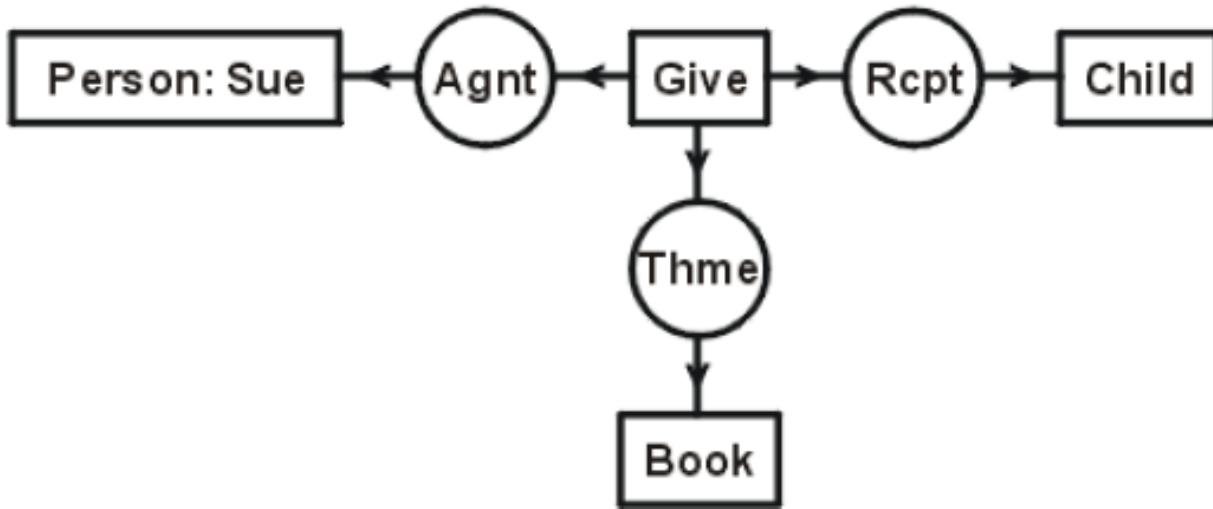
- The concept type **Give** is, in Peirce's terminology, a *hypostatic abstraction* of the relation type **Gives**.
- The idea of representing a verb by an entity that can be related by quantified variables is what Davidson called *event semantics*.



- The equivalent operation can be performed in the algebraic notation, but its effect on the structure is harder to see and to express in a systematic generalization.



$$(\exists x)(\exists y)(\text{Person}(\text{Sue}) \wedge \text{Child}(x) \wedge \text{Book}(y) \wedge \text{Gives}(\text{Sue}, x, y)).$$



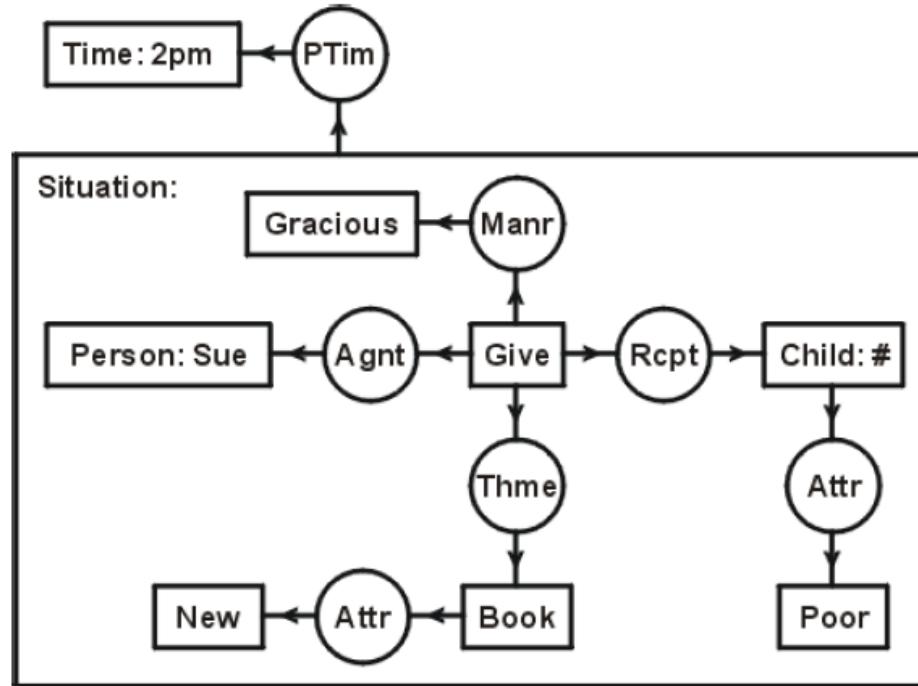
- For the CG above, the triadic connection is represented by five occurrences of the variable z , three of which correspond to the three arcs attached to the concept [Give].
- The conceptual relations (**Rcpt**) for recipient and (**Thme**) for theme are translated to dyadic relations in predicate calculus:

$$(\exists x)(\exists y)(\exists z)(\text{Person}(Sue) \wedge \text{Child}(x) \wedge \text{Book}(y) \wedge \text{Give}(z) \wedge \text{Agnt}(z, \text{Sue}) \wedge \text{Rcpt}(z, x) \wedge \text{Thme}(z, y))$$

EG & CG vs Algebraic Notation

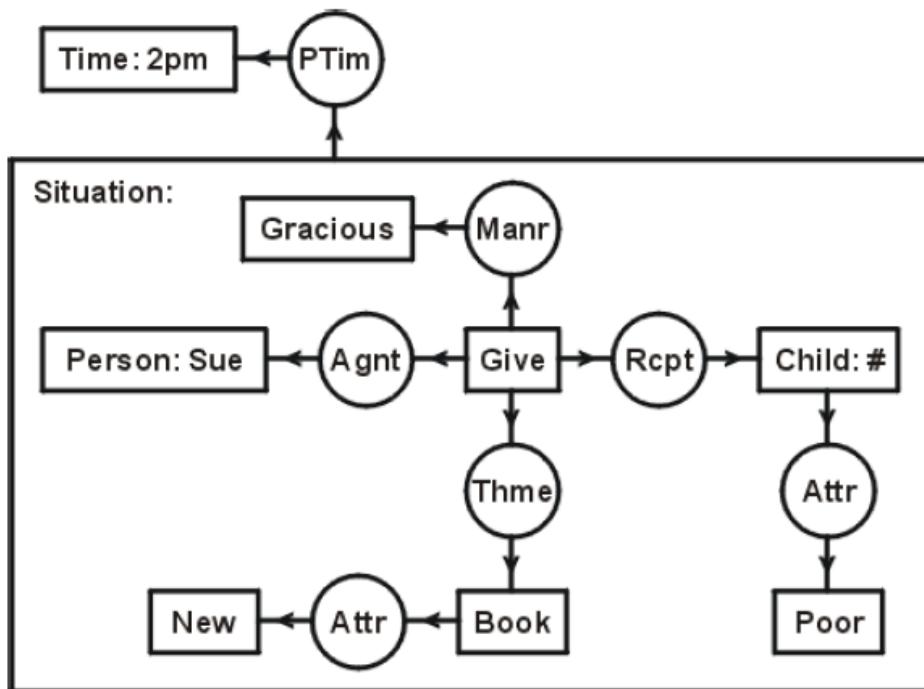
- As this example illustrates, the graph notation directly shows the *topology* of the logic, which is determined by the connectivity of the nodes and the cycles in the graph.
- That same topology is present in the algebraic formulas, but it is obscured by the notation for variables and quantifiers.
- By showing the connections directly, the graph notation in either CG or EG form enables efficient graph operations that are difficult or impossible to apply to the formulas without first converting them to an equivalent graph.

Sentence: *At 2 pm, Sue graciously gave the poor child a new book.*

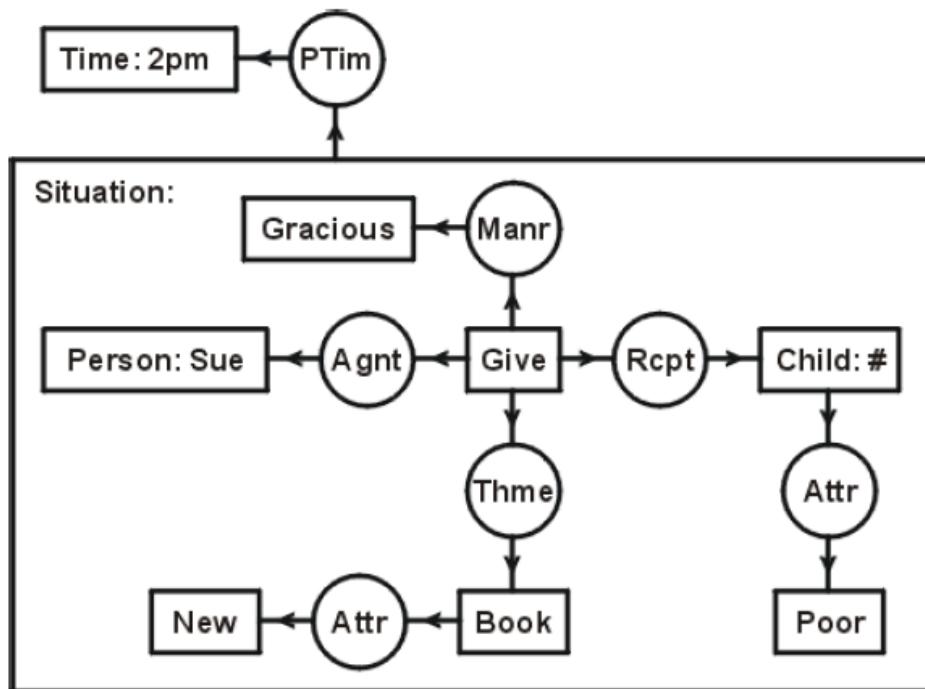


$$\begin{aligned}
 & (\exists s)(\text{Situation}(s) \wedge \text{Time}(2\text{pm}) \wedge \text{PTim}(s, 2\text{pm}) \wedge \text{dscr}(s, \\
 & (\exists y)(\exists z)(\exists u)(\exists v)(\exists w)(\text{Person}(Sue) \wedge \text{Child}(Bob) \wedge \\
 & \text{Book}(y) \wedge \text{Give}(z) \wedge \text{Gracious}(u) \wedge \text{Poor}(v) \wedge \\
 & \text{New}(w) \wedge \text{Manr}(z, u) \wedge \text{Attr}(Bob, v) \wedge \text{Attr}(y, w) \wedge \\
 & \text{Agnt}(z, Sue) \wedge \text{Rcpt}(z, Bob) \wedge \text{Thme}(z, y))).
 \end{aligned}$$

J. Sowa



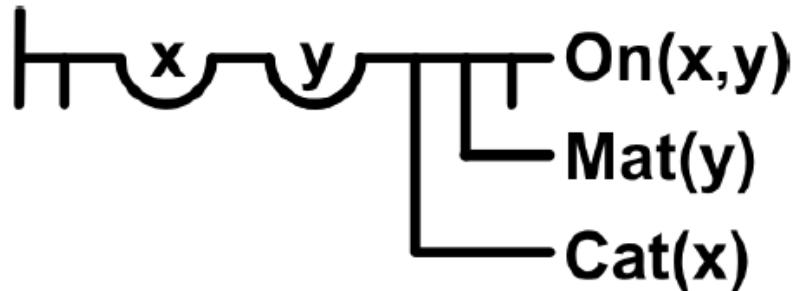
- The concept of type Situation with a nested CG represents a situation described by that CG.
- In the algebraic formula, the relation $dscr(s,p)$ is used to state that a situation s is described by a proposition p .
- The relation **(Ptim)** shows the point in time of that situation.
- The relations **(Manr)** and **(Attr)** represent the manner and attribute relations that are linked to the hypostatic abstractions **[Gracious]** and **[Poor]**, which were derived from an adverb and an adjective in the original sentence.
- Those concepts represent instances of graciousness and poverty, and the graphs allow additional connections to those nodes to represent phrases such as *very graciously* or *poor as a church mouse*.



- The symbol # in the concept **[Child: #]** represents the indexical effect of the phrase *the child*.
- Before the CG can be translated to other versions of logic, the indexical must be resolved to some individual in the context, either in the discourse or in the surrounding environment.
- In the algebraic formula, the symbol # is replaced by the name Bob.
- Hans Kamp developed discourse representation theory as a method of resolving such references. It turns out that the notation Kamp developed has context boxes that are isomorphic to the ovals of Peirce's existential graphs. By following Peirce's structures, the CG boxes turned out to be nested in the same ways as Kamp's.

How to say “A cat is on a mat.”

Gottlob Frege (1879):



Charles Sanders Peirce (1885): $\Sigma_x \Sigma_y \text{Cat}_x \bullet \text{Mat}_y \bullet \text{On}_{x,y}$

Giuseppe Peano (1895): $\exists x \exists y \text{Cat}(x) \wedge \text{Mat}(y) \wedge \text{On}(x,y)$

Charles Sanders Peirce (1897): **Cat — On — Mat**

First-Order Logic

- Shaded ovals are sufficient to express full FOL:

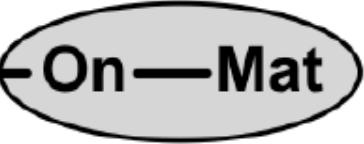
Existence: —

Negation: 

Relations: Cat- -On- -Under- -With- -Mat

A cat is on a mat: **Cat—On—Mat**

Something is under a mat: —Under—Mat

Some cat is not on a mat: **Cat—On—Mat**

Some cat is on something that is not a mat: **Cat—On—Mat**

The Scope of Quantifiers

Cat—Black

Some cat is black.

Cat—Black

Some cat is not black.

Cat—Black

No cat is black.

Cat—Black

*It is false that
some cat is not black.*

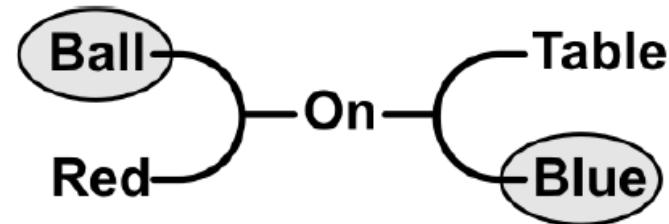
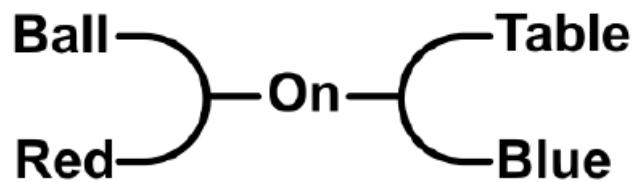
Cat—Black

*If there is a cat,
then it is black.*

Cat—Black

Every cat is black.

Translating EGs to and from English



Left graph:

A red ball is on a blue table.

Some ball that is red is on some table that is blue.

Right graph:

Something red that is not a ball is on a table that is not blue.

A red non-ball is on a non-blue table.

On some non-blue table, there is something red that is not a ball.

Existential Graph Interchange Format

A subset of the Conceptual Graph Interchange Format (CGIF):

Existence: — $[^x]$

Negation:  $\sim[]$

Relations: (Cat ?x) (On ?x ?y) (Under ?x ?y) (Mat ?y)

A cat is on a mat: $[^x] [^y] (\text{Cat} ?x) (\text{On} ?x ?y) (\text{Mat} ?y)$

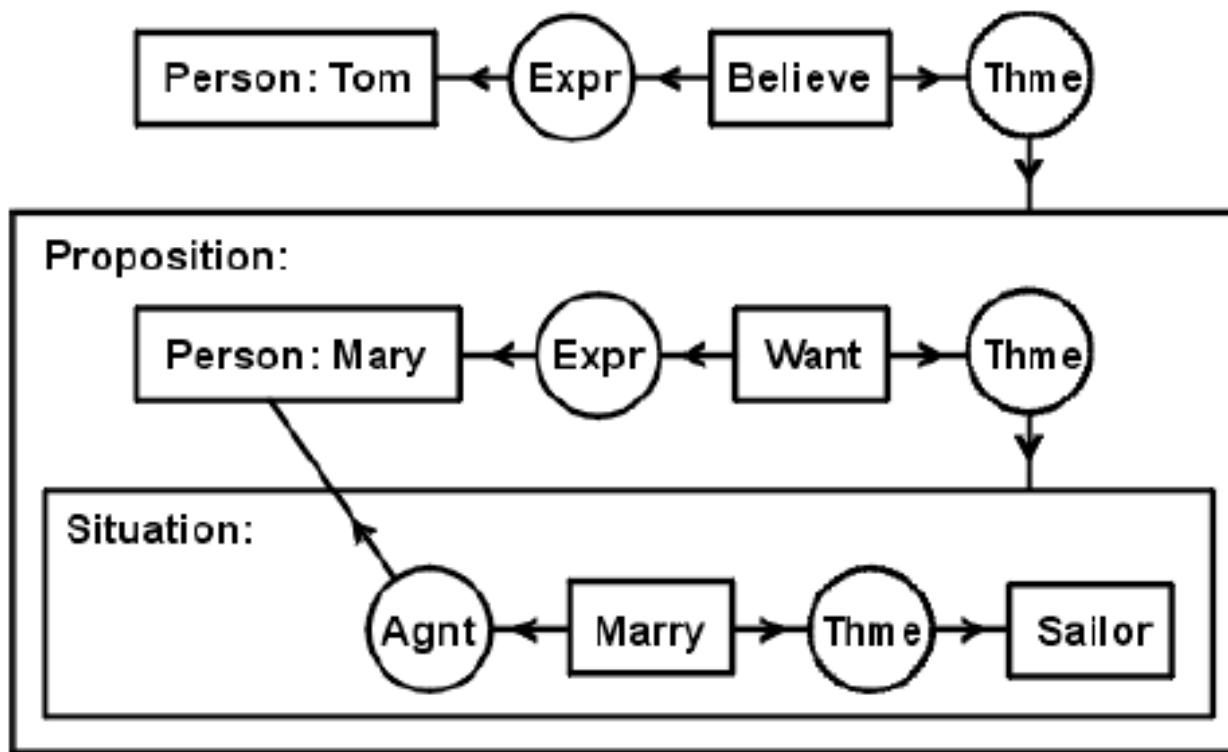
Something is under a mat: $[^x] [^y] (\text{Under} ?x ?y) (\text{Mat} ?y)$

Some cat is not on a mat: $[^x] (\text{Cat} ?x) \sim[[^y] (\text{On} ?x ?y) (\text{Mat} ?y)]$

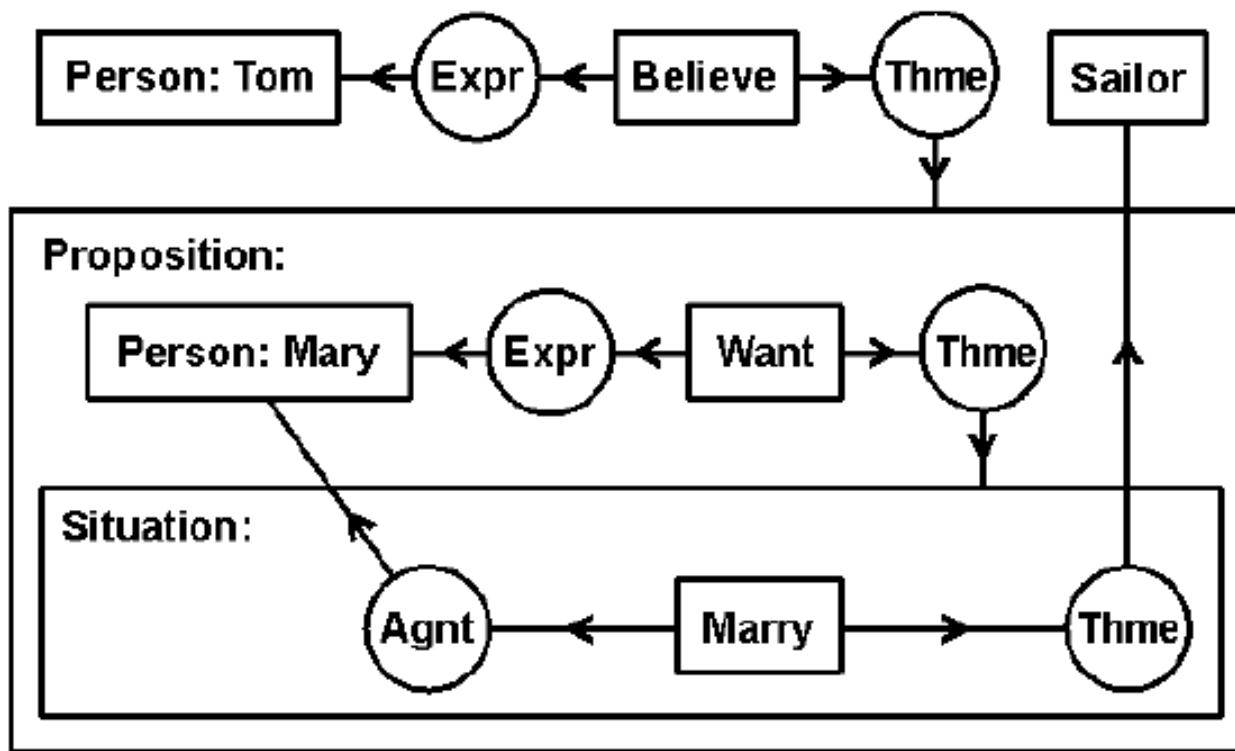
Some cat is on something that is not a mat:

$[^x] [^y] (\text{Cat} ?x) (\text{On} ?x ?y) \sim[(\text{Mat} ?y)]$

Example: *Tom believes Mary wants to marry a sailor.*



Example: *There is a sailor that Tom believes Mary wants to marry.*



Interpreting Discourse

Interpreting Discourse

- Discourse is a sequence of sentences.
- When we look at discourse, interesting challenges arise:
 - Interpreting co-references/anaphoras (pronominal resolution).
 - Representing discourse relations between propositions.

Discourse Representation Theory (DRS)

A woman walks.

$$\exists x(woman(x) \wedge walk(x))$$

She smokes.

$$smoke(x)$$

$$\exists x(woman(x) \wedge walk(x) \wedge smoke(x))$$

Need to Expand the scope of the existential quantifier.

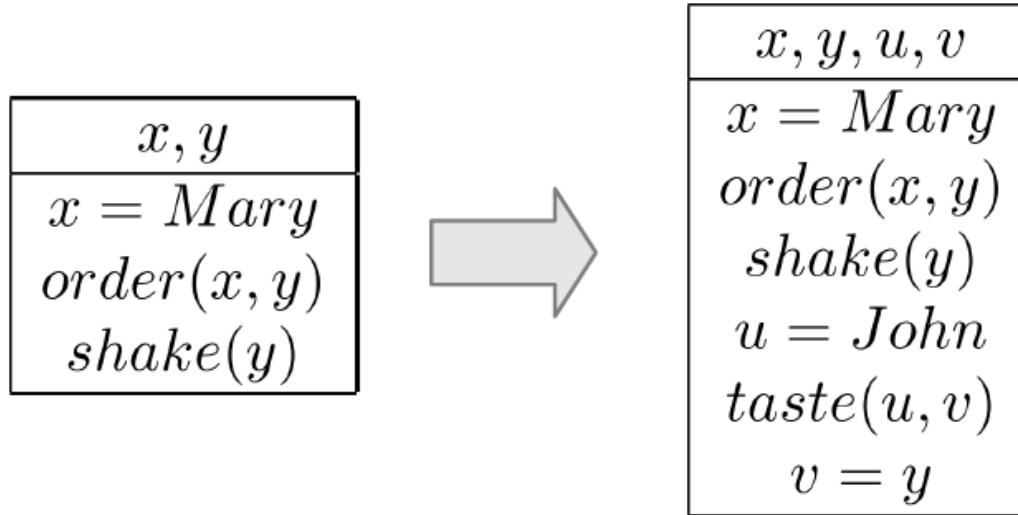
A woman walks. She smokes.

- A woman walks. She smokes.

x, y	discourse referent x, y, in the top part of the box.
$woman(x)$	
$walk(x)$	
$y = x$	conditions upon these discourse referents in the lower part of the box.
$smoke(y)$	

Discourse Structure and Accessibility

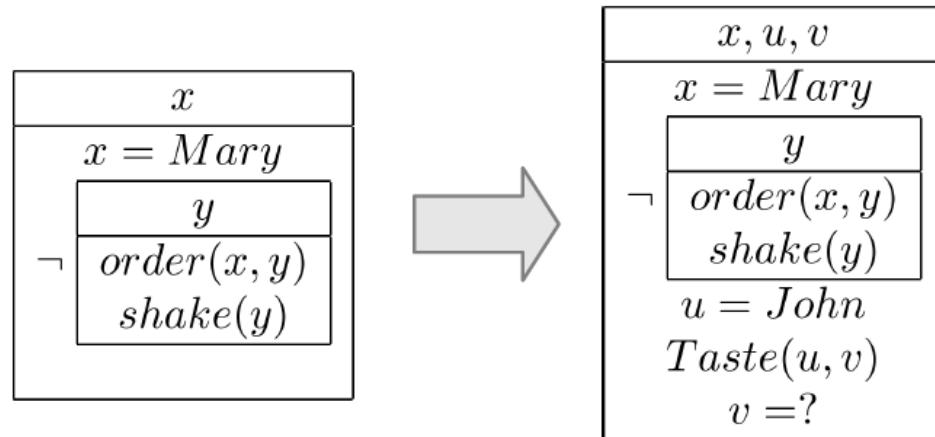
- Mary ordered a milk shake, John tasted **it**.



- The discourse referent **y is accessible** for discourse referent **v**.
- An anaphoric link between **it** and **milk shake** is allowed.

Discourse Structure and Accessibility

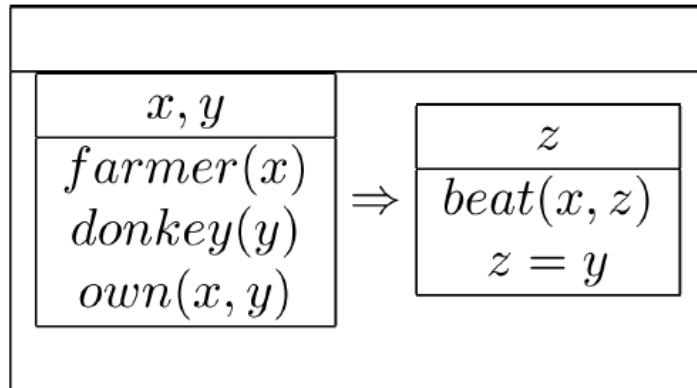
- Mary **did not** ordered a milk shake. John tasted **it**.



- When we introduced **negation**, an anaphoric link is blocked.
- Hence, **y is not accessible for v**

Discourse Structure and Accessibility

- Every farmer that owns a **donkey** beats it.



- Following the definition of accessibility, the discourse referent **y** introduced by a donkey is available as antecedent.
- A link is established by the DRS-condition $z = y$.

Rhetorical Structure Theory (RST)

Principles

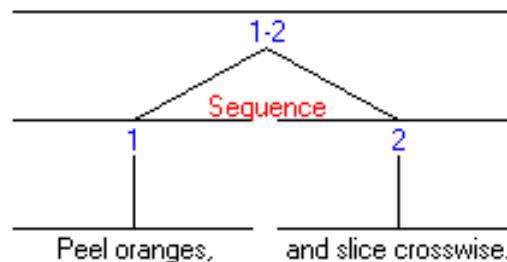
- Coherent texts consist of minimal units, which are linked to each other, recursively, through rhetorical relations
 - Rhetorical relations also known, in other theories, as coherence or discourse relations
- Coherent texts do not show gaps or non-sequiturs
 - Therefore, there must be some relation holding among the different parts of the text

Components

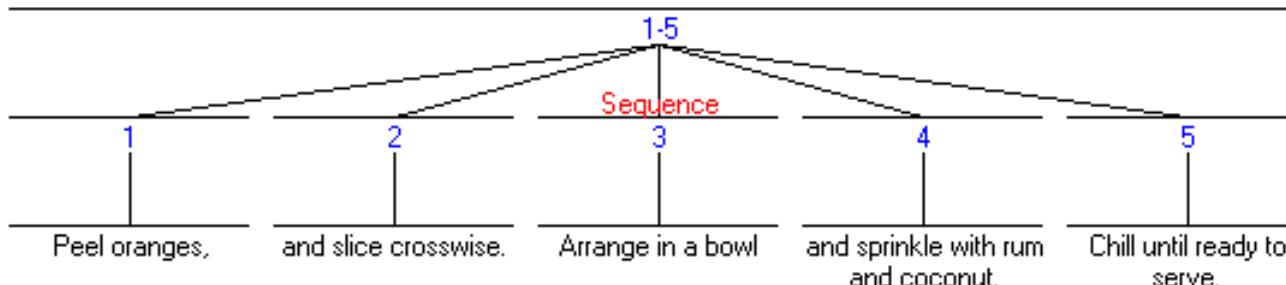
- Units of discourse
 - Texts can be segmented into minimal units, or spans.
- Nuclearity
 - Some spans are more central to the text's purpose (nuclei), whereas others are secondary (satellites).
 - Based on hypotactic and paratactic relations in language.
- Relations among spans
 - Spans are joined into discourse relations.
- Hierarchy/recursion
 - Spans that are in a discourse relation may enter into new relations.

Paratactic (coordinate)

- At the sub-sentential level (traditional coordinated clauses)
 - Peel oranges, and slice crosswise.

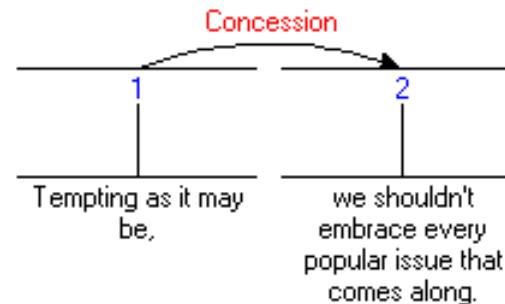


- But also across sentences
 - 1. Peel oranges, 2. and slice crosswise. 3. Arrange in a bowl 4. and sprinkle with rum and coconut. 5. Chill until ready to serve.

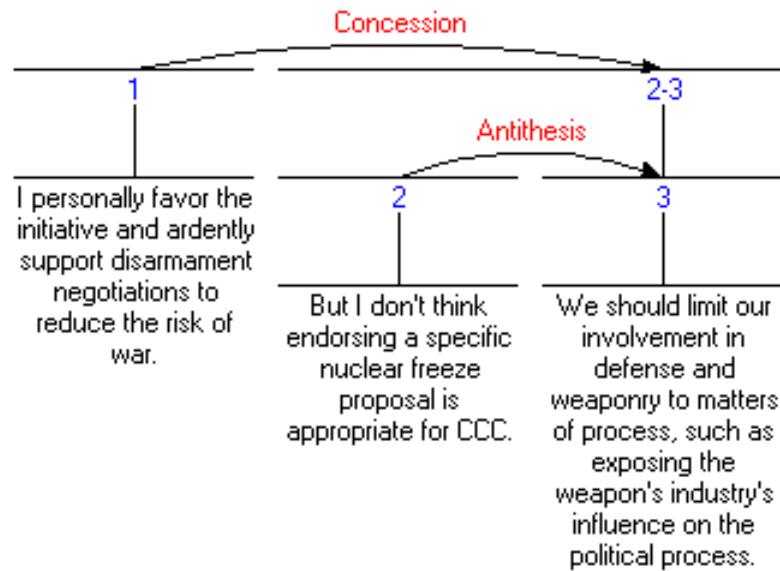


Hypotactic (subordinate)

- Sub-sentential Concession relation



- Concession across sentences
 - Nucleus (spans 2-3) made up of two spans in an Antithesis relation



Relation Types

- Relations are of different types:
 - Subject matter: they relate the content of the text spans
 - Cause, Purpose, Condition, Summary
 - Presentational: more rhetorical in nature. They are meant to achieve some effect on the reader
 - Motivation, Antithesis, Background, Evidence

Other possible classifications

- Relations that hold outside the text
 - Condition, Cause, Result

vs. those that are only internal to the text

 - Summary, Elaboration
- Relations frequently marked by a discourse marker
 - Concession (*although, however*); Condition (*if, in case*)

vs. relations that are rarely, or never, marked

 - Background, Restatement, Interpretation
- Preferred order of spans: nucleus before satellite
 - Elaboration – usually first the nucleus (material being elaborated on) and then satellite (extra information)

vs. satellite-nucleus

 - Concession – usually the satellite (the *although*-type clause or span) before the nucleus

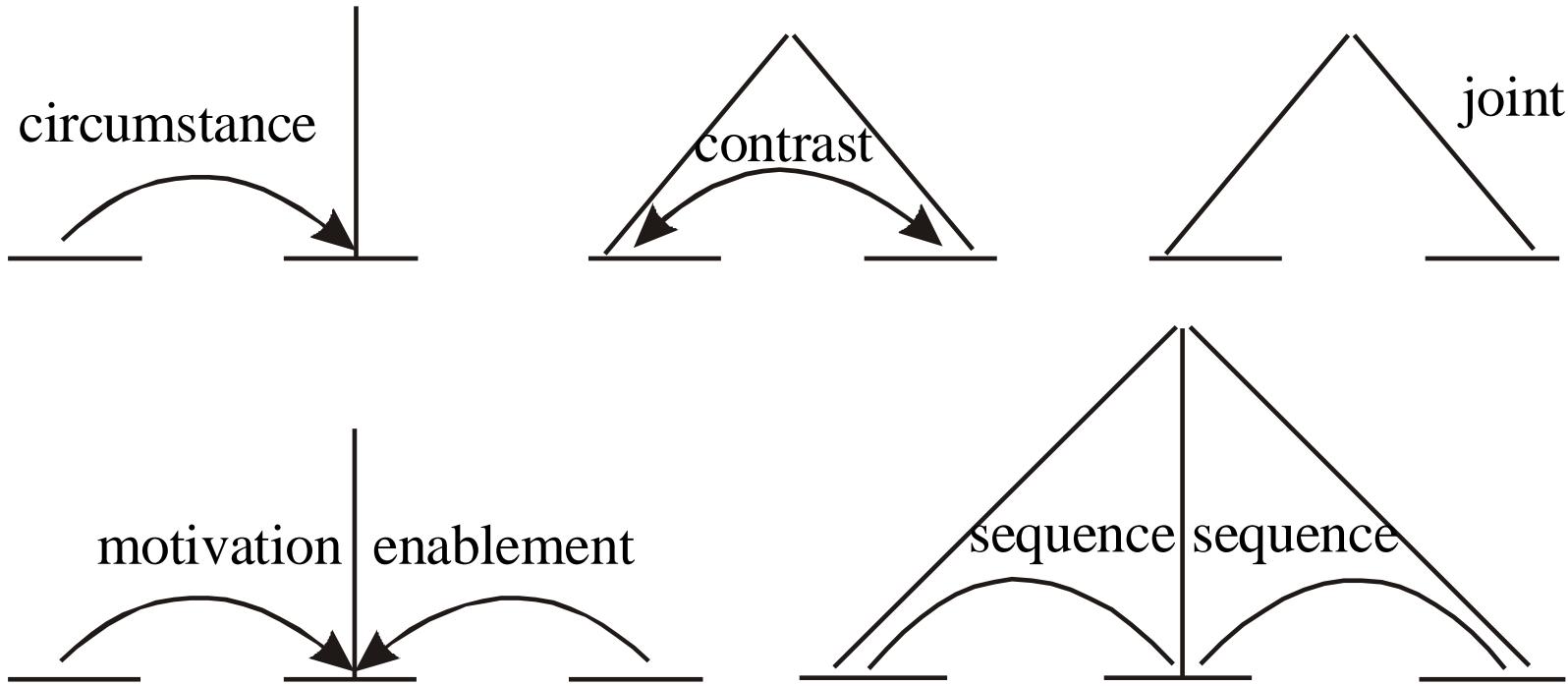
Relation names (in M&T 1988)

Circumstance	Antithesis and Concession
Solutionhood	Antithesis
Elaboration	Concession
Background	Condition and Otherwise
Enablement and Motivation	Condition
Enablement	Otherwise
Motivation	
Evidence and Justify	Interpretation and Evaluation
Evidence	Interpretation
Justify	Evaluation
Relations of Cause	Restatement and Summary
Volitional Cause	Restatement
Non-Volitional Cause	Summary
Volitional Result	
Non-Volitional Result	Other Relations
Purpose	Sequence
	Contrast

Other classifications are possible, and longer and shorter lists have been proposed

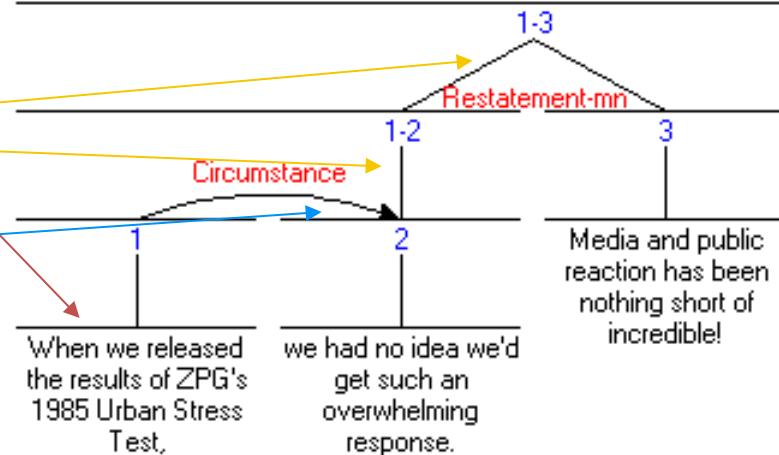
Schemas

- They specify how spans of text can co-occur, determining possible RST text structures



Graphical representation

- A **horizontal line** covers a span of text (possibly made up of further spans)
- A **vertical line** signals the nucleus or nuclei
- A **curve** represents a relation, and the direction of the arrow, the direction of satellite towards nucleus



How to do an RST analysis

1. Divide the text into units
 - Unit size may vary, depending on the goals of the analysis
 - Typically, units are clauses (but not complement clauses)
2. Examine each unit, and its neighbours. Is there a clear relation holding between them?
3. If yes, then mark that relation (e.g., Condition)
4. If not, the unit might be at the boundary of a higher-level relation. Look at relations holding between larger units (spans)
5. Continue until all the units in the text are accounted for
6. Remember, marking a relation involves satisfying all 4 fields (especially the Effect). The Effect is the plausible intention that the text creator had.

Putting All Together

Software: Extracting Knowledge Graphs from Text



<https://github.com/Lambda-3/Graphene>

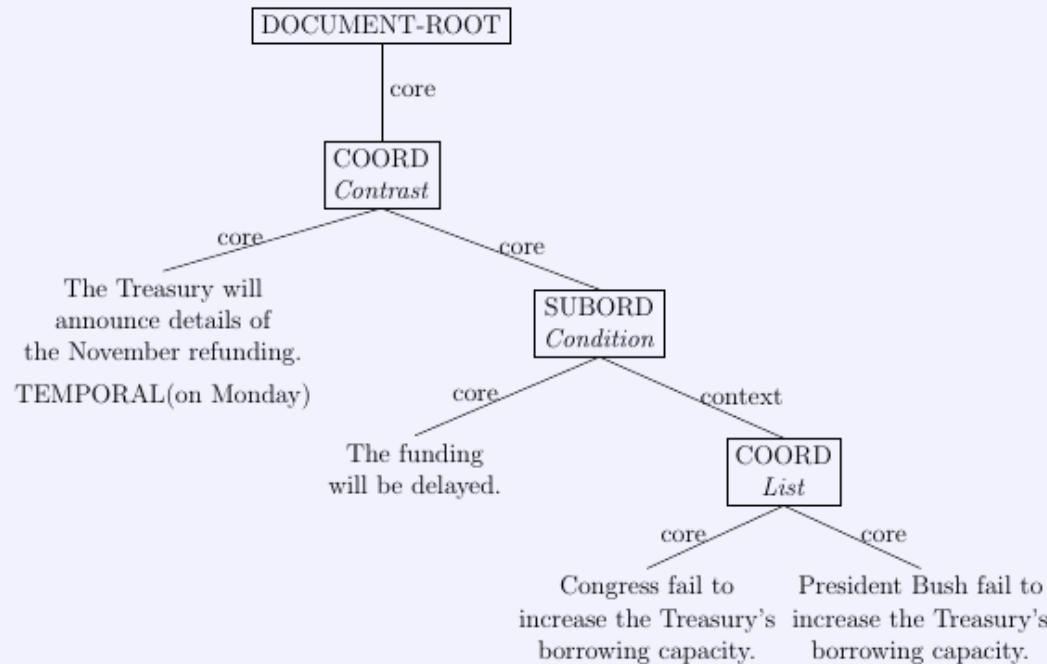
Niklaus et al., A Sentence Simplification System for Improving Relation Extraction, COLING (2017)

Coreference- Resolution

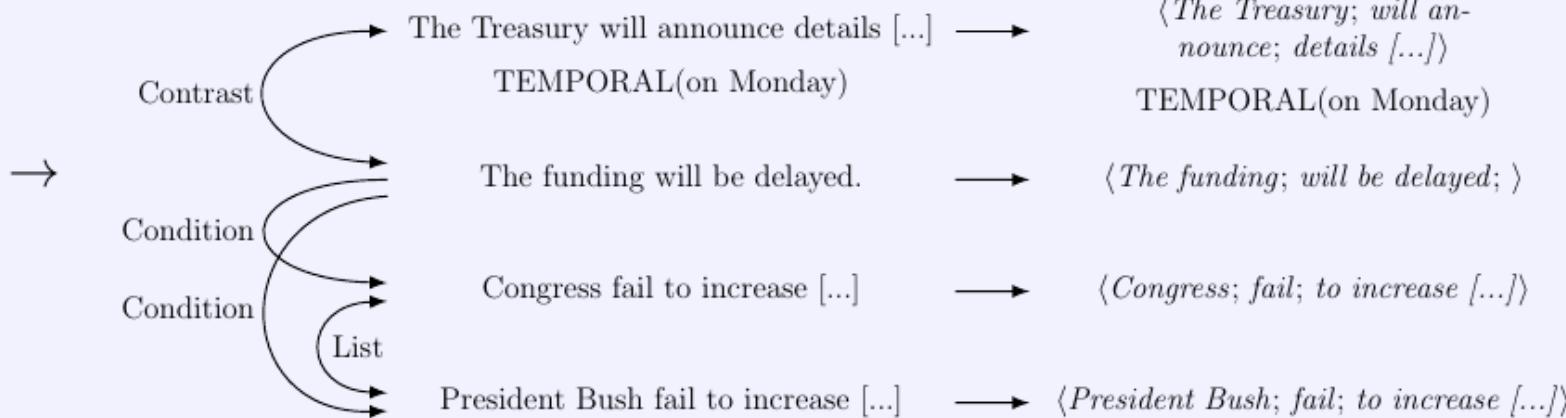
[...] Although the Treasury will announce details of the November refunding on Monday, the funding will be delayed if Congress and President Bush fail to increase the Treasury's borrowing capacity. [...]



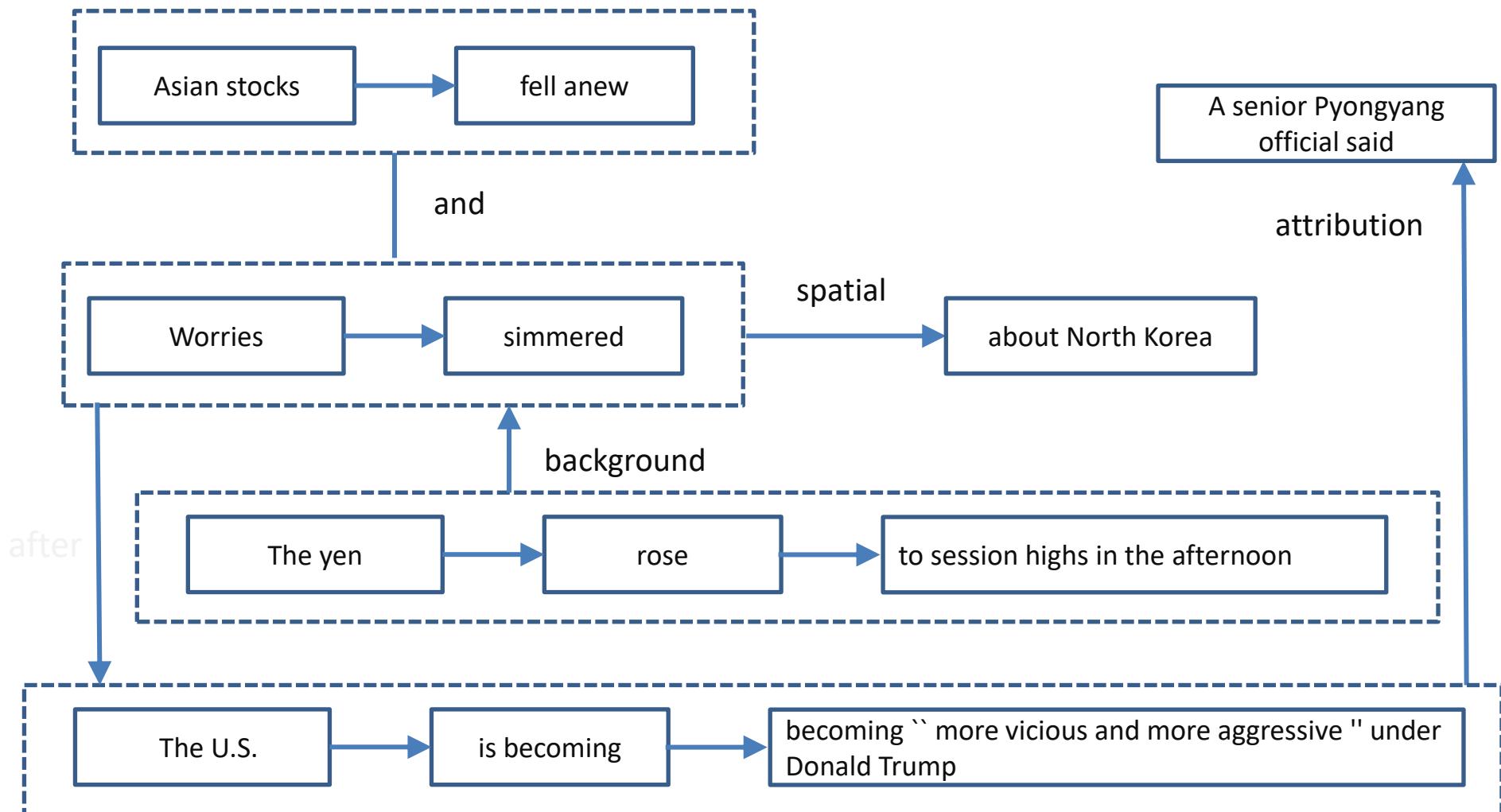
Transformation Stage



Relation Extraction



Asian stocks fell anew and the yen rose to session highs in the afternoon as worries about North Korea simmered, after a senior Pyongyang official said the U.S. is becoming ``more vicious and more aggressive'' under President Donald Trump .



The RDF-NL Format

Although the Treasury will announce details of the November refunding on Monday ,
the funding will be delayed if Congress and President Bush fail to increase the Treasury 's
borrowing capacity .

bacf06771e0f4fc5a8e68c30fc77c9c4 the Treasury will announce details of the November refunding
S:TEMPORAL on Monday .

L:CONTRAST 948eeebd73564adab7dee5c6f177b3b9

948eeebd73564adab7dee5c6f177b3b9 the funding will be delayed

L:CONDITION 006a71e51295440fab7a8e8c697d2ba6

L:CONDITION e4d86228cff443b7a8e9f6d8a5c5987b

L:CONTRAST bacf06771e0f4fc5a8e68c30fc77c9c4

006a71e51295440fab7a8e8c697d2ba6 Congress fail to increase the Treasury 's borrowing capacity

e4d86228cff443b7a8e9f6d8a5c5987b president Bush fail to increase the Treasury 's borrowing capacity

The RDF-NL Format

Although the Treasury will announce details of the November refunding on Monday ,
the funding will be delayed if Congress and President Bush fail to increase the Treasury 's
borrowing capacity .

bacf06771e0f4fc5a8e68c30fc77c9c4 the Treasury will announce details of the November refunding
S:TEMPORAL on Monday .
L:CONTRAST 948eeebd73564adab7dee5c6f177b3b9

948eeebd73564adab7dee5c6f177b3b9 the funding will be delayed
L:CONDITION 006a71e51295440fab7a8e8c697d2ba6
L:CONDITION e4d86228cff443b7a8e9f6d8a5c5987b
L:CONTRAST bacf06771e0f4fc5a8e68c30fc77c9c4

006a71e51295440fab7a8e8c697d2ba6 Congress fail to increase the Treasury 's borrowing capacity

e4d86228cff443b7a8e9f6d8a5c5987b president Bush fail to increase the Treasury 's borrowing capacity

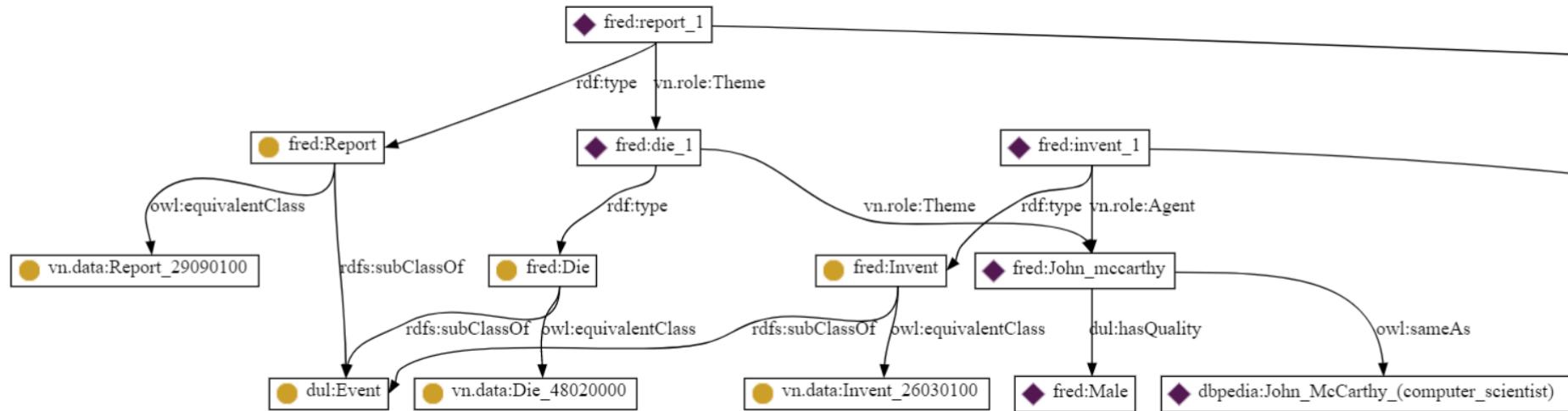
Semafor



<http://www.cs.cmu.edu/~ark/SEMAFOR/>

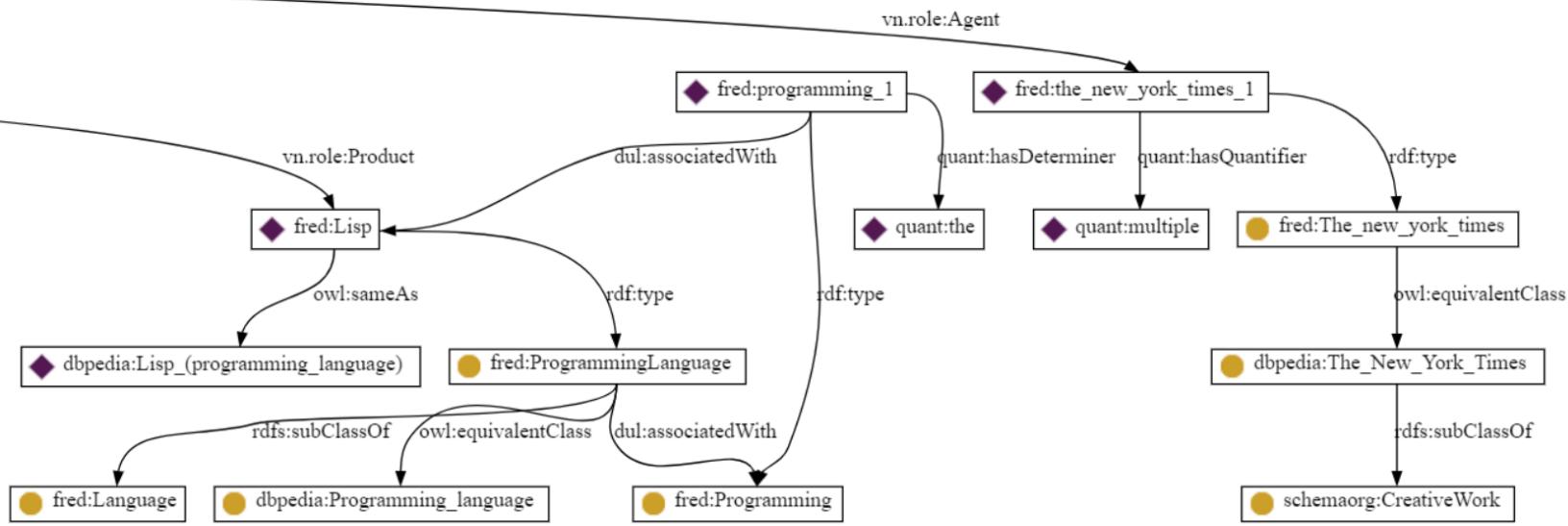
Frame-based Extraction

The New York Times reported that John McCarthy died. He invented the programming language LISP.



Frame-based Extraction

The New York Times reported that John McCarthy died. He invented the programming language LISP.



Software: FRED

FRED

Machine Reading for the Semantic Web

<http://wit.istc.cnr.it/stlab-tools/fred/>

Gangemi et al., Semantic Web Machine Reading with FRED, Semantic Web Journal, 2017

Recommended Reading

Bundy, Alan, & Fiona McNeill (2006)
“Representation as a Fluent: An AI Challenge for
the Next Half Century,” AAAI Fellows
Symposium, Boston, MA.

Symbolic AI

Andre Freitas



Photo by Vasilyev Alexandr

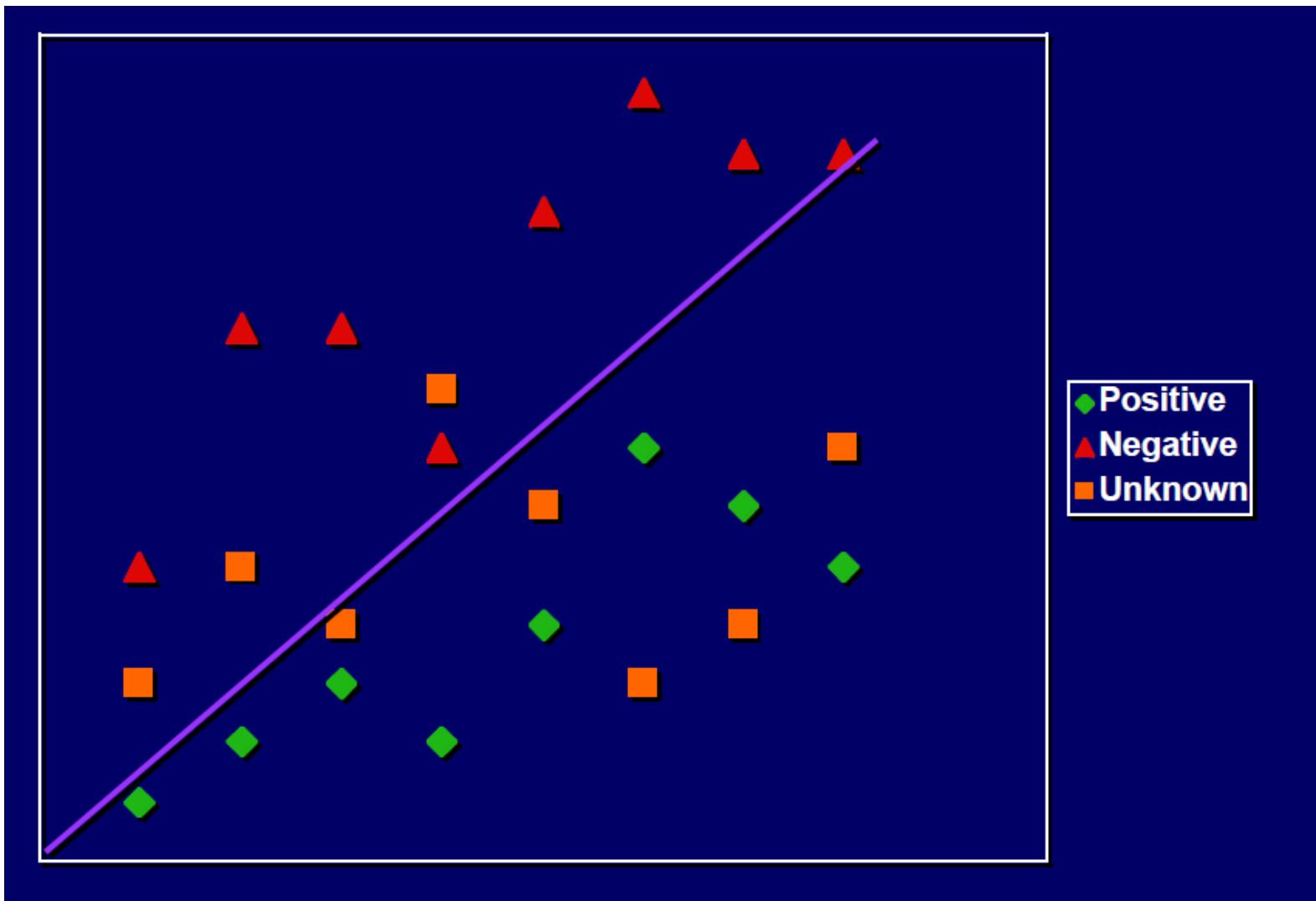
Acknowledgements

- These slides were based on the slides of:
 - Peter A. Flach, Rule induction tutorial, IDA Spring School 2001.
 - Anoop & Hector, Inductive Logic Programming (for Dummies).
 - Gabor Melli, Scribe Notes on FOIL and Inverted Deduction.
 - CS 5751 Machine Learning, Chapter 10 Learning Sets of Rules.

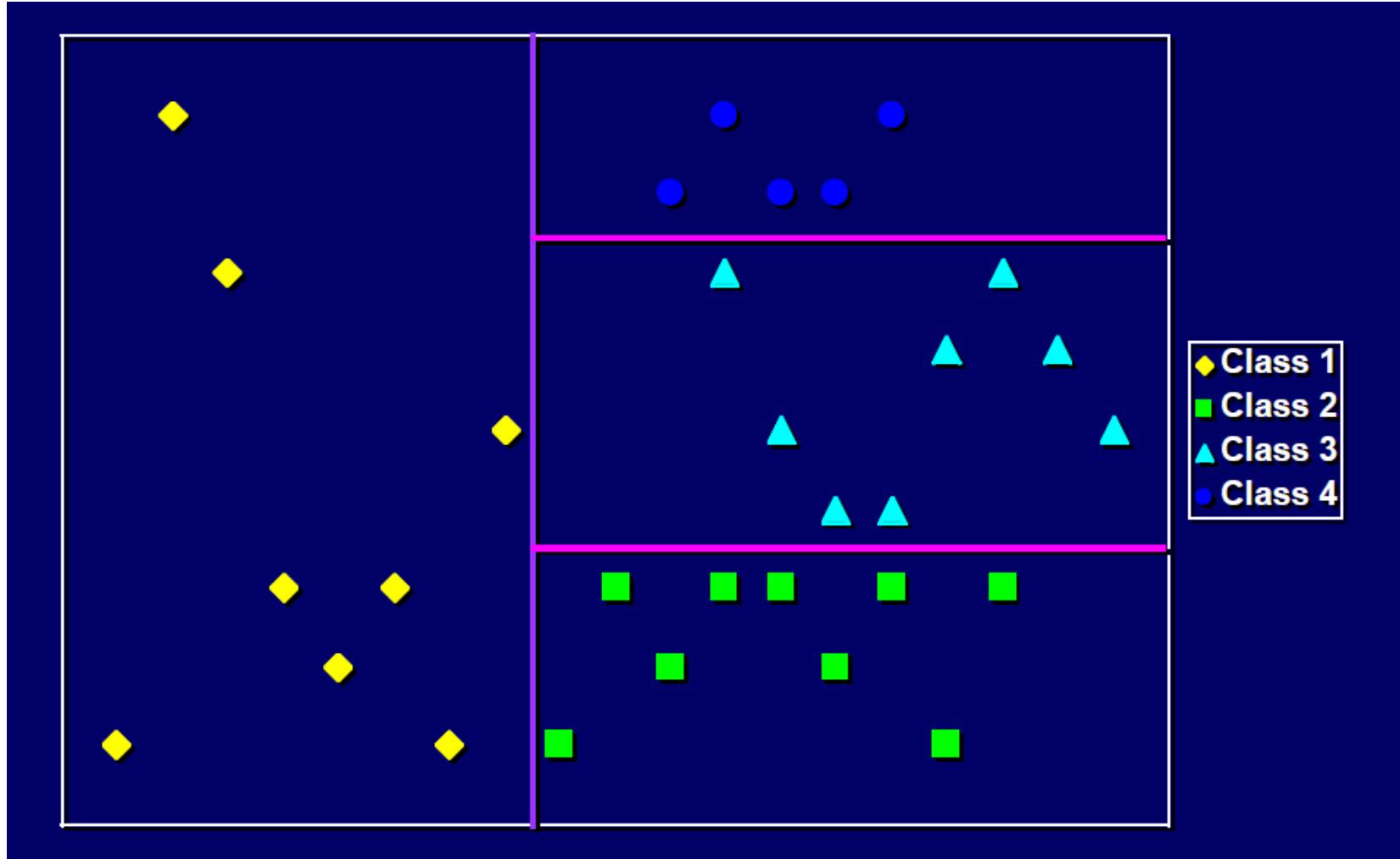
This Lecture

- Introduction to Inductive Logic Programming
- FOIL

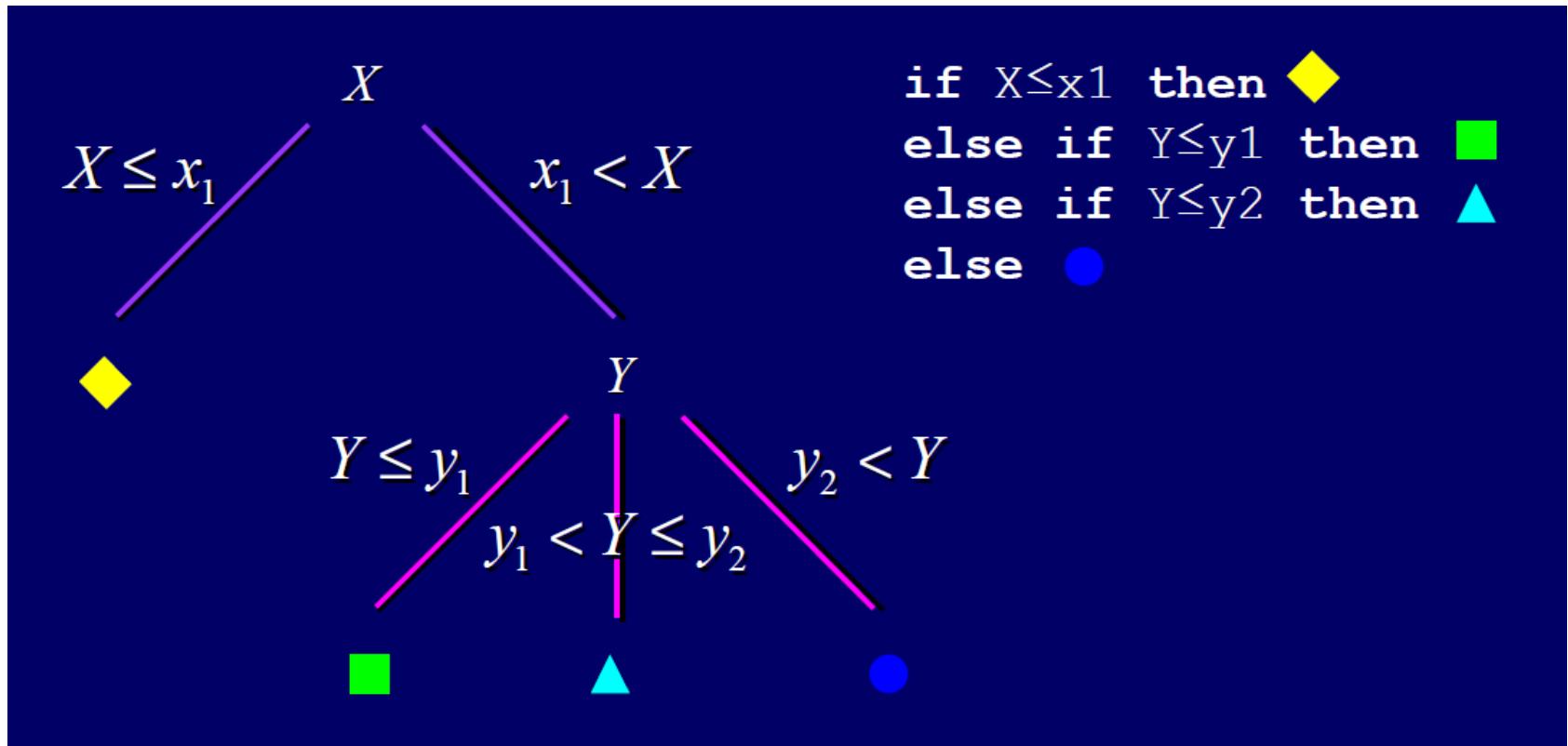
Linear Classifier



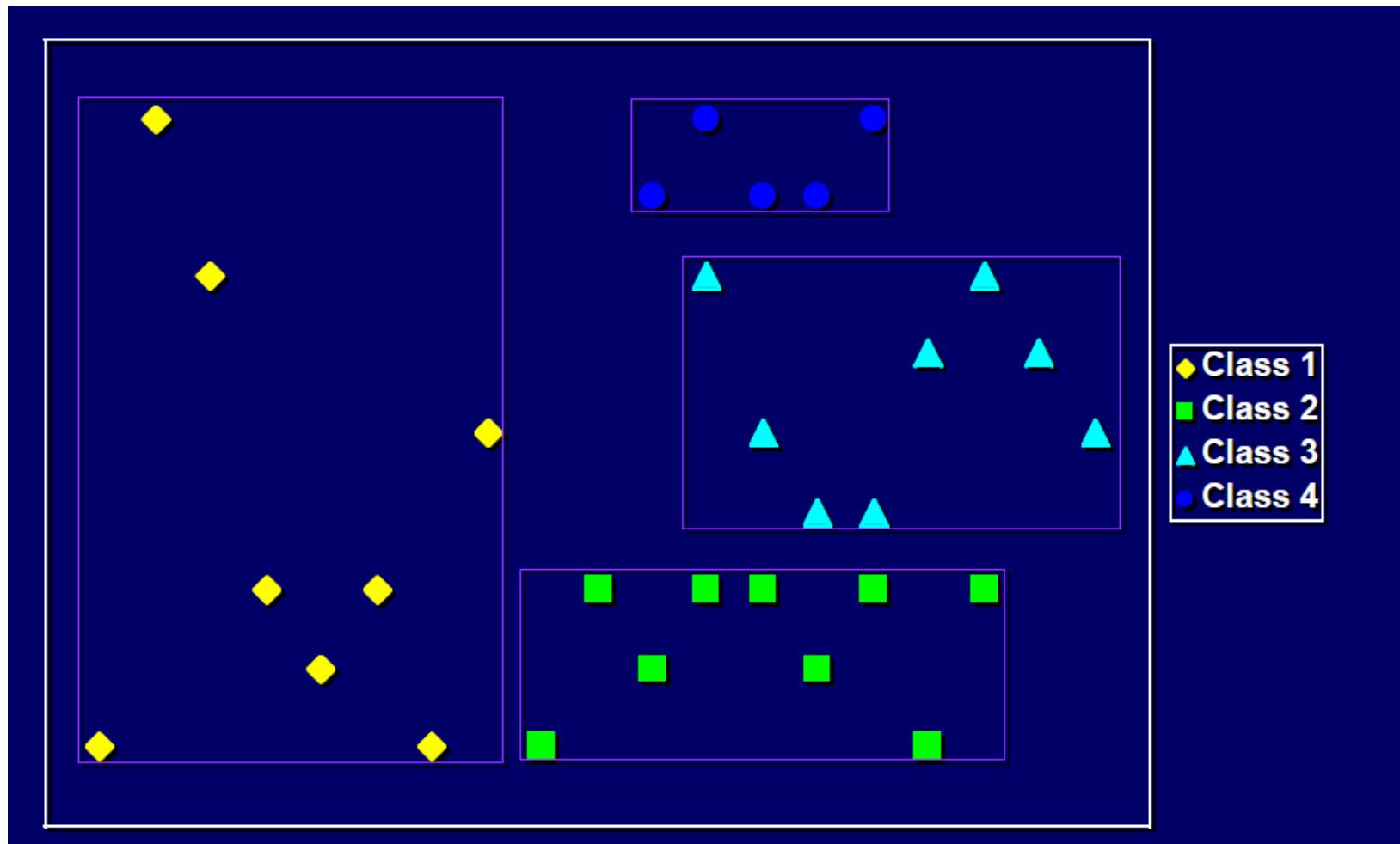
Decision Trees



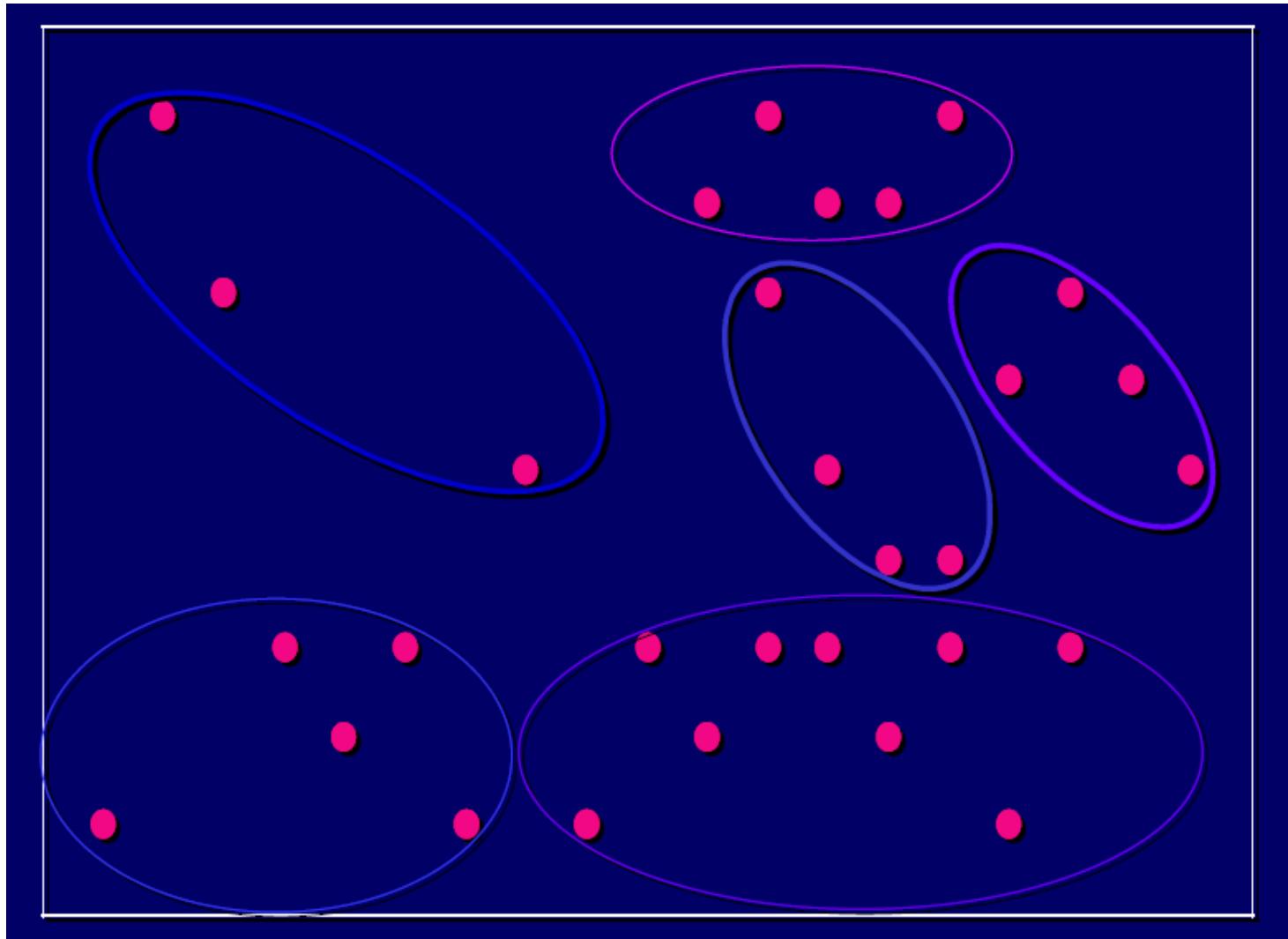
Decision Trees



Rules



Clustering



ILP: Objective

Given a dataset:

- Positive examples ($E+$) and optionally negative examples ($E-$).
- Additional knowledge about the problem/application domain (Background Knowledge B).
- Set of constraints to make the learning process more efficient (C).

Goal of an ILP system is to find a set of hypothesis that:

- Explains (covers) the positive examples – Completeness.
- Are consistent with the negative examples – Consistency.

Generalisation & Specialisation

- **Generalising** a concept involves enlarging its extension in order to cover a given instance or subsume another concept.
- **Specialising** a concept involves restricting its extension in order to avoid covering a given instance or subsuming another concept.

First-order Representations

- **Propositional** representations:
 - datacase is ***fixed-size vector of values***
 - features are those given in the dataset
- **First-order** representations:
 - datacase is ***flexible-size, structured object***
 - sequence, set, graph
 - hierarchical: e.g. set of sequences
 - features need to be **selected** from potentially infinite set

Deductive Vs Inductive Reasoning

$$T \cup B \rightarrow E \text{ (deduce)}$$

```
parent(X,Y) :- mother(X,Y).  
parent(X,Y) :- father(X,Y).
```

```
mother(mary,vinni).  
mother(mary, andre).  
father(carrey, vinni).  
father(carry, andre).
```

```
parent(mary,vinni).  
parent(mary, andre).  
parent(carrey, vinni).  
parent(carrey, andre).
```

$$E \cup B \rightarrow T \text{ (induce)}$$

```
parent(mary,vinni).  
parent(mary, andre).  
parent(carrey, vinni).  
parent(carrey, andre).
```

```
mother(mary,vinni).  
mother(mary, andre).  
father(carrey, vinni).  
father(carry, andre).
```

```
parent(X,Y) :- mother(X,Y).  
parent(X,Y) :- father(X,Y).
```

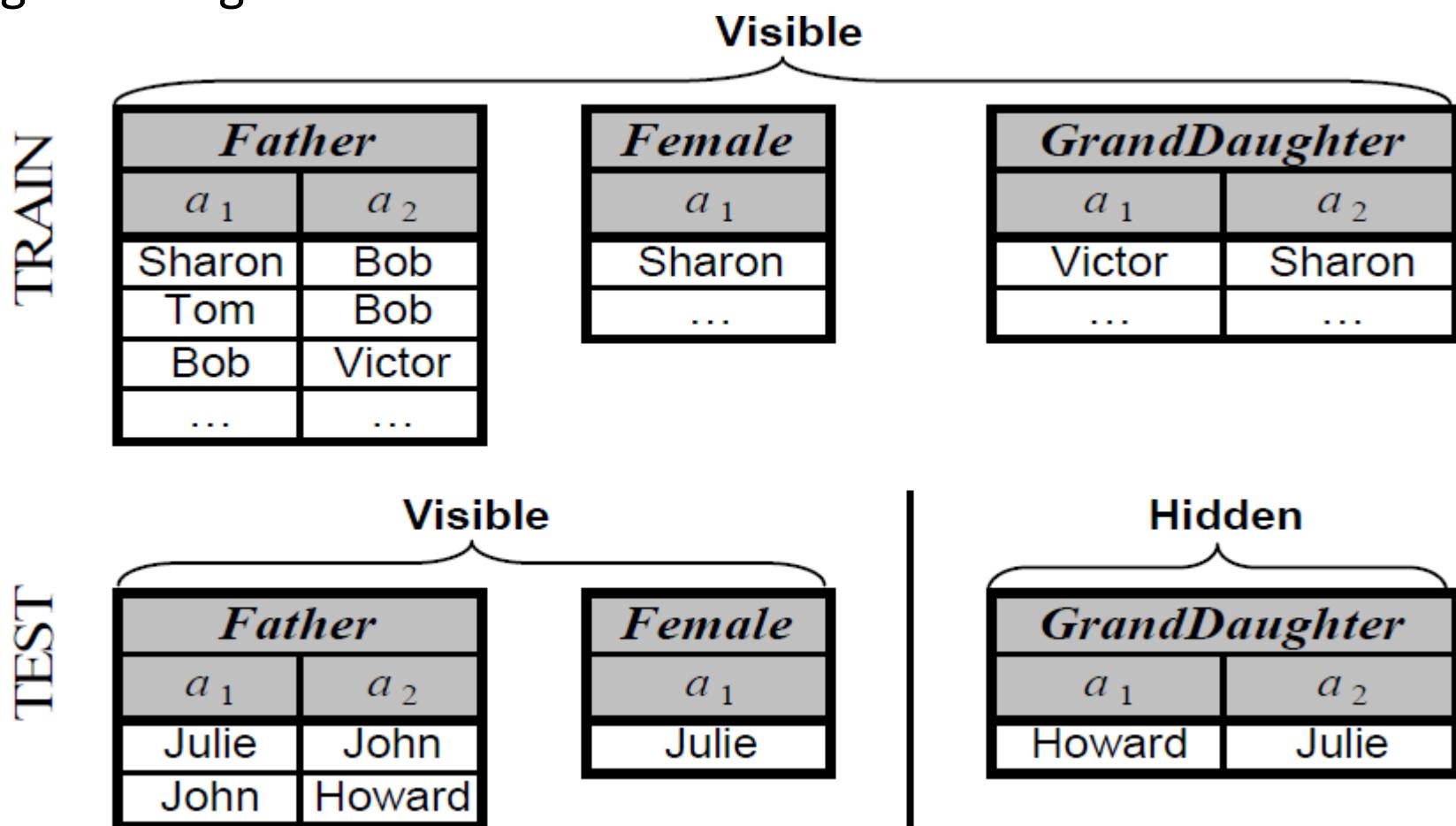
Relational Pattern

IF Customer(C1, Age1, Income1, TotSpent1, BigSpender1)
AND MarriedTo(C1, C2)
AND Customer(C2, Age2, Income2, TotSpent2, BigSpender2)
AND Income2 \geq 10000
THEN BigSpender1 = Yes

big_spender(C1, Age1, Income1, TotSpent1) \leftarrow
married_to(C1, C2) \wedge
customer(C2, Age2, Income2, TotSpent2, BigSpender2) \wedge
Income2 \geq 10000

Example ILP Problem

Discover the rule that describes whether a person has a granddaughter



Propositional Learner with simple data transformation

- One of the first challenges that a propositional learner would encounter with this dataset is that the dataset is not structured as a set of fixed length-vectors of attribute-value pairs. This situation is typically resolved by JOINing the relations.

Predictors			Target
<i>Father</i>	<i>Child</i>	<i>Child is Fem.</i>	<i>Has Gdauh</i>
Bob	Sharon	TRUE	FALSE
Victor	Bob	FALSE	TRUE
...

Propositional Learner with simple data transformation

- A propositional learner would not locate a predictive model for this dataset.
- It would not be able to state that *Sharon* is *Victor's* granddaughter.
- At best it may discover that a child's gender has some influence on the likelihood that that child is a parent, or even a parent to a female child.

Predictors			Target
<i>Father</i>	<i>Child</i>	<i>Child is Fem.</i>	<i>Has Gdauh</i>
Bob	Sharon	TRUE	FALSE
Victor	Bob	FALSE	TRUE
...

Propositional Learner with complex data transformation

- The algorithm cannot make the connection in one observation (*Bob* as a father) and another (*Bob* as child).
- A common way to enable a propositional learner to produce a predictive model on this data is to transform the data so that the required relations appear as attributes in the data.

- This transformation is sometimes referred to as 'flattening' the data.

Predictors						Target
<i>Father</i>	<i>Child</i>	<i>Child is Fem.</i>	<i>Child's Child</i>	<i>C's C is Fem.</i>	<i>Has Gdauh</i>	
Bob	Sharon	TRUE	NULL	NULL	FALSE	
Victor	Bob	FALSE	Sharon	TRUE	TRUE	
...

- Now the search for a rule is trivial. A decision tree would locate the pattern:

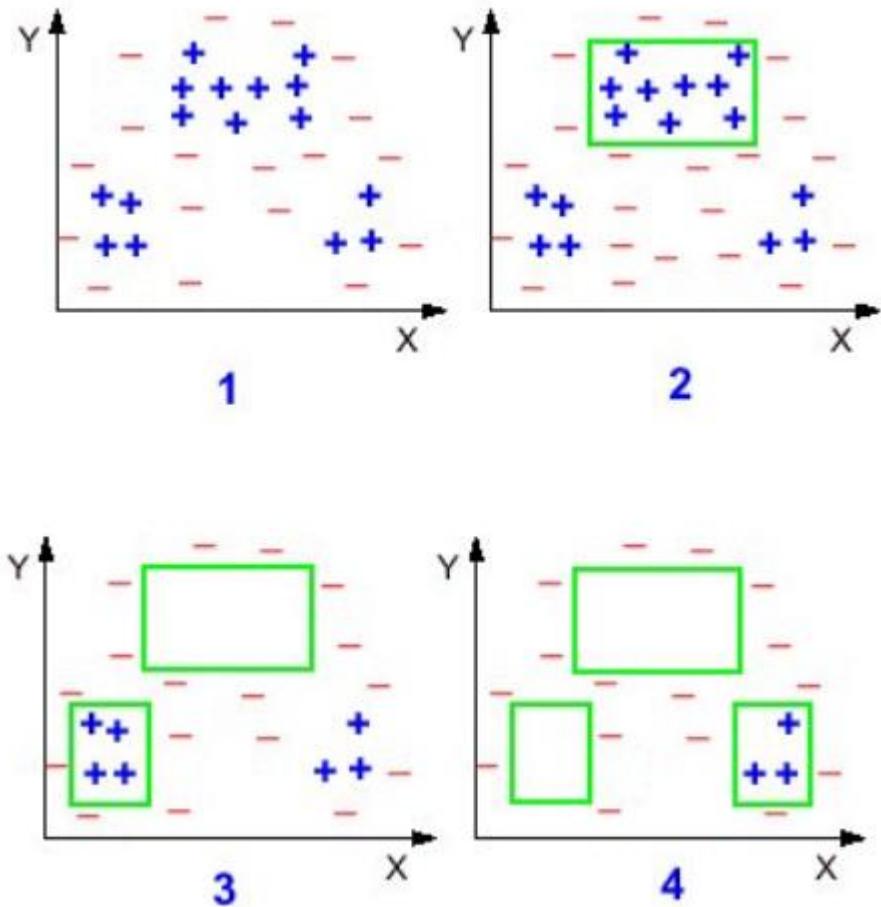
IF Child's Child is Female = TRUE
 THEN HasGrandDaughter = TRUE.
 ELSE HasGrandDaughter = FALSE

Propositional Sequential Covering

- A covering algorithm, in the context of propositional learning systems, is an algorithm that develops a cover for the set of positive examples.
 - that is, a set of hypotheses that account for all the positive examples but none of the negative examples.
- Sequential covering: it learns one rule at a time and repeat this process to gradually cover the full set of positive examples.

Iterate to Learn Multiple Rules

- Select seed from positive examples to build bottom clause.
- Get some rule “If $A \wedge B$ then P”. Now throw away all positive examples that were covered by this rule
- Repeat until there are no more positive examples.



Propositional Sequential Covering

1. Start with an empty **Cover**
2. Use **Learn-One-Rule** to find the best hypothesis.
3. If the Just-Learnt-Rule satisfies the threshold then
 - Put Just-Learnt-Rule to the **Cover**.
 - Remove examples covered by Just-Learnt-Rule.
 - Go to step 2.
4. Sort the **Cover** according to its performance over examples.
5. Return: **Cover**.

Example

Id	Size	Colour	Shape	Weight	Expensive
1	Big	Red	Square	Heavy	Yes
2	Small	Blue	Triangle	Light	Yes
3	Small	Blue	Square	Light	No
4	Big	Green	Triangle	Heavy	No
5	Big	Blue	Square	Light	No
6	Big	Green	Square	Heavy	Yes
7	Small	Red	Triangle	Light	Yes

Expensive = Yes if:

- Colour = Red. (covers example 1,7)
- Or (Colour = Green & Shape = Square). (covers example 6)
- Or (Colour = Blue & Shape = Triangle). (covers example 2)

Complex

- A complex is a conjunction of attribute-value specifications. It forms the **condition** part in a rule, like "if **condition** then predict **class**".

Size=Big	Size=Small	Colour=Red
Colour=Green	Colour=Blue	Shape=Square
Shape=Triangle	Weight=Light	Weight=Heavy

- Specialising a complex is making a conjunction of the complex with one more attribute-value pair. For example:

Colour=Green & Shape=Square

(specialising Colour=Green or Shape=Square)

Colour=Blue & Weight=Heavy

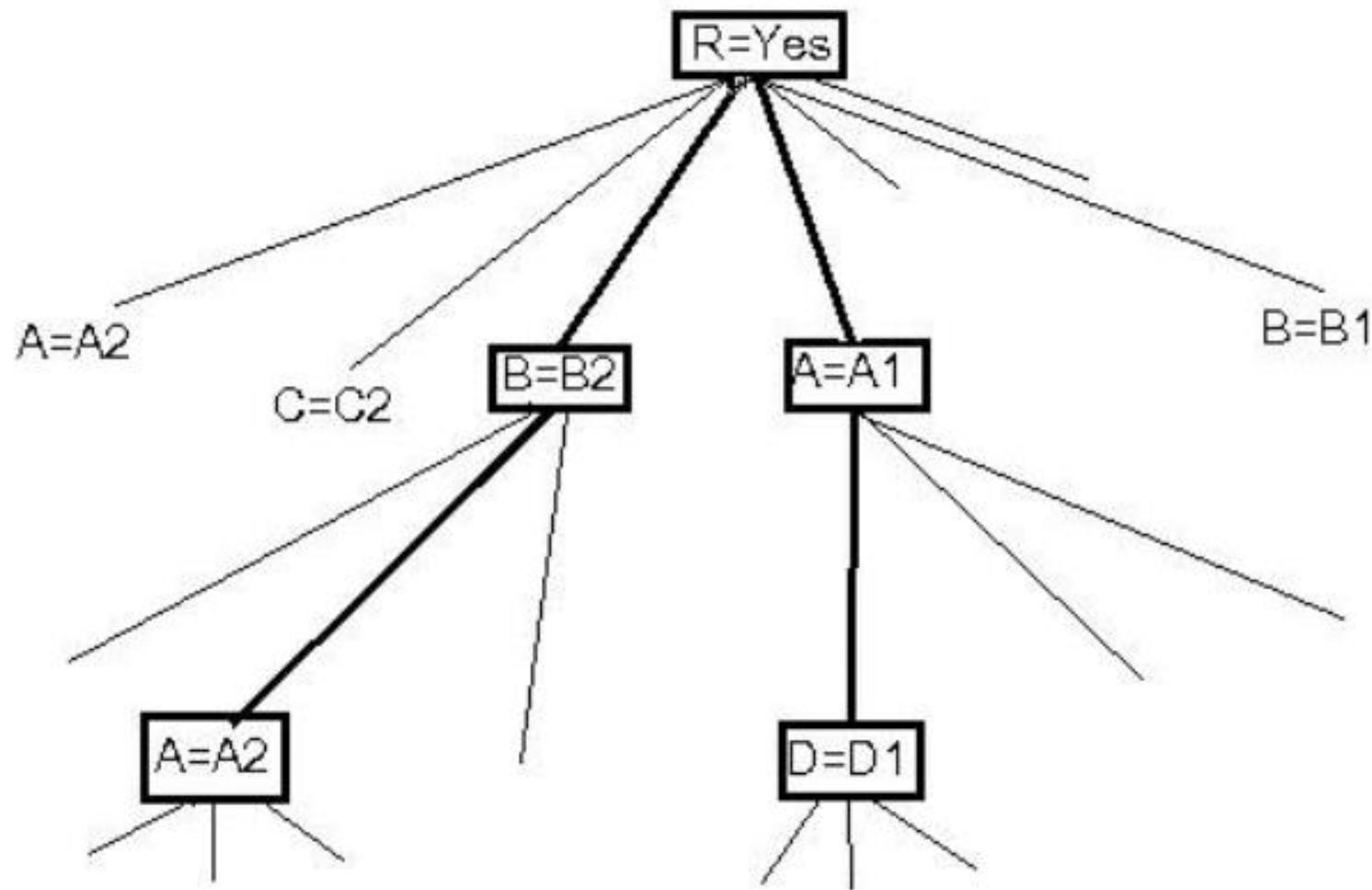
(specialising Colour=Blue or Weight=Heavy)

Learn-One-Rule using Beam Search

1. Initialize a set of most general complexes.
2. Evaluate performances of those complexes over the example set.
 - Count how many positive and negative examples it covers.
 - Evaluate their performances.
3. Sort complexes according to their performances.
4. If the best complex satisfies some **threshold**, form the hypothesis and **return**.
5. Otherwise, pick k best performing complexes for the next generation.
6. Specializing all k complexes in the set to find new set of less general complexes.
7. Go to step 2.

The number k is the beam factor of the search, meaning the maximum number of complexes to be specialized.

Example



General to Specific Beam Search Example

- In the first step, 2 best complexes are found, namely A=A1 and B=B2.
- None of them satisfy the threshold, then the next level complexes are expanded and found 2 best complexes, eg. A=A1 & D=D1 and B=B2 & A=A2.
- The procedure keeps going until we find a complex that satisfies the threshold.

Entropy Evaluation Function

- The evaluation is based on the entropy of the set covered by that complex. Here is an example of a hypothesis covering 8 positive and 2 negative examples.

$$p1 = P(\text{positive}) = 8/(2+8) = 0.8;$$

$$p2 = P(\text{negative}) = 2/(2+8) = 0.2;$$

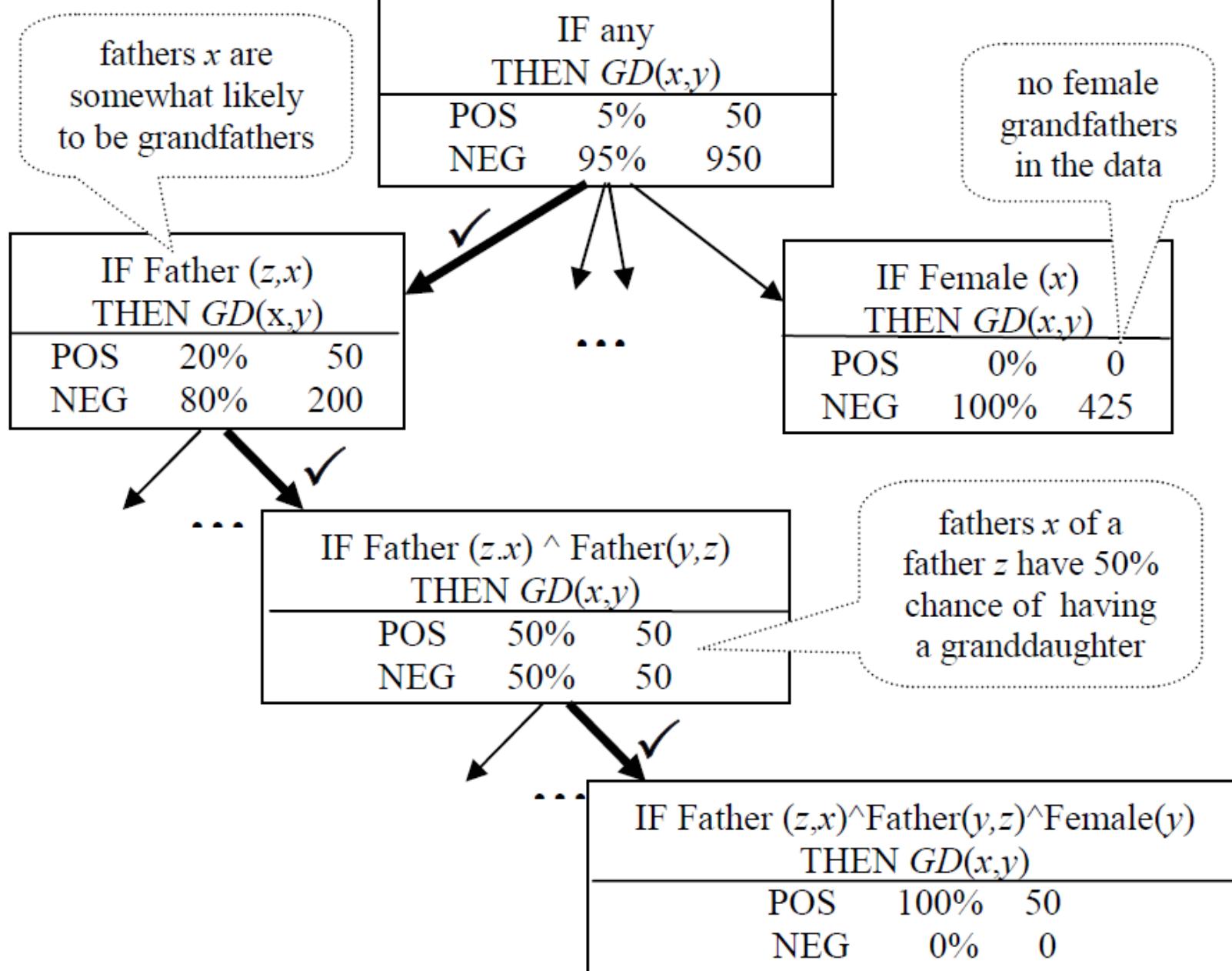
$$\text{Entropy} = - p1 * \log(p1) - p2 * \log(p2) = 0.72.$$

- In this function, the smaller the entropy is, the better the complex.
- In other words, the accuracy function can be defined as (1-Entropy).

The FOIL Algorithm

- The FOIL algorithm is a supervised learning algorithm that produces rules in first-order logic.
- The algorithm is a generalization of the SEQUENTIAL-COVERING and LEARN-ONE-RULE algorithms .
- The main modification is that search can also specialize on predicates with variables.
- The resulting rules differ from Horn clauses in two ways:
 - Negated symbols are allowed within the body.
 - FOIL's rules will not include function symbols.

Back to the Example



FOIL

FOIL(Target_predicate, Predicates, Examples)

Pos \leftarrow positive *Examples*

Neg \leftarrow negative *Examples*

while *Pos* do (*Learn a New Rule*)

NewRule \leftarrow most general rule possible

NegExamplesCovered \leftarrow *Neg*

 while *NegExamplesCovered* do

 Add a new literal to specialize *NewRule*

 1. *Candidate_literals* \leftarrow generate candidates

 2. *Best_literal* $\leftarrow \operatorname{argmax}_{L \in \text{candidate_literal}} \text{FOIL_GAIN}(L, \text{NewRule})$

 3. Add *Best_literal* to *NewRule* preconditions

 4. *NegExamplesCovered* \leftarrow subset of *NegExamplesCovered* that satisfies *NewRule* preconditions

Learned_rules \leftarrow *Learned_rules* + *NewRule*

Pos \leftarrow *Pos* - {members of *Pos* covered by *NewRule*}

Return *Learned_rules*

The FOIL Algorithm

- The *outer loop adds new rules* to the output until no more positive examples are covered.
- The *inner loop searches for the next best rule* by incremental specialization.
- The outer loop corresponds to the SEQUENTIAL-CONVERGING algorithm, the inner to FIND-A-RULE

Specialising Rules in FOIL

Learning rule: $P(x_1, x_2, \dots, x_k) \leftarrow L_1 \dots L_n$

Candidate specializations add new literal of form:

- $Q(v_1, \dots, v_r)$, where at least one of the v_i in the created literal must already exist as a variable in the rule
- $Equal(x_j, x_k)$, where x_j and x_k are variables already present in the rule
- The negation of either of the above forms of literals

Information Gain in FOIL

$$FOIL_GAIN(L, R) \equiv t \left(\log_2 \frac{p_1}{p_1 + n_1} - \log_2 \frac{p_0}{p_0 + n_0} \right)$$

Where

- L is the candidate literal to add to rule R
- p_0 = number of positive bindings of R
- n_0 = number of negative bindings of R
- p_1 = number of positive bindings of $R+L$
- n_1 = number of negative bindings of $R+L$
- t is the number of positive bindings of R also covered by $R+L$

Note

- $-\log_2 \frac{p_0}{p_0 + n_0}$ is optimal number of bits to indicate the class of a positive binding covered by R

Applications

First Order Rule for Classifying Web Pages

From (Slattery, 1997)

```
course(A) ←  
    has-word(A,instructor),  
    NOT has-word(A,good),  
    link-from(A,B)  
    has-word(B,assignment),  
    NOT link-from(B,C)
```

Train: 31/31, Test 31/34

Flach 2001

Early diagnosis of rheumatic diseases

- Sample CN2 rule for an 8-class problem :

**IF Sex = male AND Age > 46 AND
Number_of_painful_joints > 3 AND
Skin_manifestations = psoriasis**

Application

A molecular compound is carcinogenic if:

- (1) it tests positive in the Salmonella assay; or
- (2) it tests positive for sex-linked recessive lethal mutation in Drosophila; or
- (3) it tests negative for chromosome aberration; or
- (4) it has a carbon in a six-membered aromatic ring with a partial charge of -0.13; or
- (5) it has a primary amine group and no secondary or tertiary amines; or
- (6) it has an aromatic (or resonant) hydrogen with partial charge ≥ 0.168 ; or
- (7) it has an hydroxy oxygen with a partial charge ≥ -0.616 and an aromatic (or resonant) hydrogen; or
- (8) it has a bromine; or
- (9) it has a tetrahedral carbon with a partial charge ≤ -0.144 and tests positive on Progol's mutagenicity rules.

Final Considerations

Why ILP is not just Decision Trees.

- Language is First-Order Logic
 - Natural representation for multi-relational settings
 - Thus, a natural representation for *full* databases
- Not restricted to the classification task.
- So then, what is ILP?

Efficiency Issues

- Representational Aspects
- Search
- Evaluation
- Sharing computations
- Memory-wise scalability

Representational Aspects

- Example:
 - Student(string sname, string major, string minor)
 - Course(string cname, string prof, string cred)
 - Enrolled(string sname, string cname)
- In a natural join of these tables there is a one-to-one correspondence between join result and the Enrolled table.
- Data mining tasks on the Enrolled table are really propositional.

Representational Aspects

- Three settings for data mining:
 - Find patterns within individuals represented as tuples (single table, propositional)
 - eg. Which minor is chosen with what major
 - Find patterns within individuals represented as sets of tuples (each individual ‘induces’ a sub-database)
 - Multiple tables, restricted to some individual
 - eg. Student X taking course A, usually takes course B
 - Find patterns within the whole database
 - Multiple tables

Evaluation

- Evaluating a clause: get some measure of coverage
 - Match each example to the clause:
 - Run multiple logical queries.
 - Query optimization methods from DB community
 - Rel. Algebra operator reordering
 - BUT: queries for DB are set oriented (bottom-up), queries in PROLOG find a single solution (top-down).

Sharing Computations

- Materialization of features
- Propositionalization
- Pre-compute some statistics
 - Joint distribution over attributes of a table
 - Query selectivity
- Store proofs, reuse when evaluating new clauses

Summary

- Rules: easy to understand
 - Sequential covering algorithm
 - generate one rule at a time
 - general to specific - add antecedents
 - specific to general - delete antecedents
- First order logic and covering
 - how to connect variables
 - FOIL

Recommended Reading

QuickFOIL: Scalable Inductive Logic Programming

Qiang Zeng
University of
Wisconsin–Madison
qzeng@cs.wisc.edu

Jignesh M. Patel
University of
Wisconsin–Madison
jignesh@cs.wisc.edu

David Page
University of
Wisconsin–Madison
page@biostat.wisc.edu

Symbolic AI

Andre Freitas



•Photo by Vasilyev Alexandr

Acknowledgements

- Based on the slides of:
 - General Ideas in Inductive Logic Programming (FOPI-RG).
 - Lecture 6: Inductive Logic Programming Cognitive Systems II - Machine Learning.
 - CS 391L: Machine Learning: Rule Learning, Mooney.

This Lecture

- Getting deeper into ILP.

Recap: ILP

- Goal is to induce a Horn-clause definition for some target predicate P , given definitions of a set of background predicates Q .
- Goal is to find a syntactically simple Horn-clause definition, D , for P given background knowledge B defining the background predicates Q .
 - For every positive example p_i of P
$$D \cup B \models p_i$$
 - For every negative example n_i of P
$$D \cup B \not\models n_i$$
- Background definitions are provided either:
 - Extensionally: List of ground tuples satisfying the predicate.
 - Intensionally: Prolog definitions of the predicate.

Relational Learning and Inductive Logic Programming (ILP)

- **Fixed feature vectors** are a very limited representation of instances.
- Examples or target concept may require a relational representation that includes multiple entities with relationships between them (e.g. a graph with labeled edges and nodes).
- **First-order predicate logic** is a more powerful representation for handling such relational descriptions.
- **Horn clauses** (i.e. if-then rules in predicate logic, Prolog programs) are a useful restriction on full first-order logic that allows decidable inference.
- Allows learning programs from sample I/O pairs.

Learning Rules

- Rules are fairly easy for people to understand and therefore can help provide insight and comprehensible results for human users.
 - Frequently used in data mining applications where goal is discovering understandable patterns in data.
- Methods for automatically inducing rules from data *have been shown to build more accurate expert systems than human knowledge engineering* for some applications.

Rule Learning vs. Knowledge Engineering

- An influential experiment with an early rule-learning method (AQ) by Michalski (1980) compared results to knowledge engineering (acquiring rules by interviewing experts).
- Knowledge engineered rules:
 - Weights associated with each feature in a rule
 - Method for summing evidence similar to *certainty factors*.
 - No explicit disjunction
- Data for induction:
 - Examples of 15 soybean plant diseases described using 35 nominal and discrete ordered features, 630 total examples.
 - 290 “best” (diverse) training examples selected for training.
Remainder used for testing
 - What is wrong with this methodology?

Experimental Results

- Rule construction time:
 - Human: 45 hours of expert consultation
 - AQ11: 4.5 minutes training on IBM 360/75
 - What doesn't this account for?
- Test Accuracy:

	1 st choice correct	Some choice correct
AQ11	97.6%	100.0%
Manual KE	71.8%	96.9%

Recap: Sequential Covering

- A set of rules is learned one at a time
- each time finding a single rule
- that covers a large number of positive instances
- without covering any negatives,
- removing the positives that it covers,
- and learning additional rules to cover the rest.

Let P be the set of positive examples

Until P is empty do:

Learn a rule R that covers a large number of elements of P but no negatives.

Add R to the list of rules.

Remove positives covered by R from P

- This is an instance of the greedy algorithm for minimum set covering and does not guarantee a minimum number of learned rules.
- Minimum set covering is an NP-hard problem and the greedy algorithm is a standard approximation algorithm.

Strategies for Learning a Single Rule

- Top Down (General to Specific):
 - Start with the most-general (empty) rule.
 - Repeatedly add antecedent constraints on features that eliminate negative examples while maintaining as many positives as possible.
 - Stop when only positives are covered.
- Bottom Up (Specific to General)
 - Start with a most-specific rule (e.g. complete instance description of a random instance).
 - Repeatedly remove antecedent constraints in order to cover more positives.
 - Stop when further generalization results in covering negatives.

Learning a Single Rule in FOIL

- Basic algorithm for instances with discrete-valued features:

Let $A = \{ \}$ (set of rule antecedents)

Let N be the set of negative examples

Let P the current set of uncovered positive examples

Until N is empty do

 For every feature-value pair (literal) $(F_i = V_{ij})$ calculate

$\text{Gain}(F_i = V_{ij}, P, N)$

 Pick literal, L , with highest gain.

 Add L to A .

 Remove from N any examples that do not satisfy L .

 Remove from P any examples that do not satisfy L .

Return the rule: $A_1 \wedge A_2 \wedge \dots \wedge A_n \rightarrow \text{Positive}$

Rule Pruning in FOIL

- Prepruning method based on **minimum description length (MDL)** principle.
- Postpruning to eliminate unnecessary complexity due to limitations of greedy algorithm.

For each rule, R

For each antecedent, A , of rule

If deleting A from R does not cause
negatives to become covered
then delete A

For each rule, R

If deleting R does not uncover any positives (since they
are redundantly covered by other rules)
then delete R

Minimum Description Length

- Devise an encoding that maps a theory (set of clauses) into a bit string.
- Also need an encoding for examples.
- Number of bits required to encode theory should not exceed number of bits to encode +ve examples.

Rule Learning Issues

- Which is better top-down or bottom-up search?
 - **Bottom-up** is more subject to noise, e.g. the random seeds that are chosen may be noisy.
 - **Top-down** is wasteful when there are many features which do not even occur in the positive examples (e.g. text categorization).

Rule Learning Issues

- Which is better rules or trees?
 - **Trees** share structure between disjuncts.
 - **Rules** allow completely independent features in each disjunct.
 - Mapping some rules sets to decision trees results in an exponential increase in size.

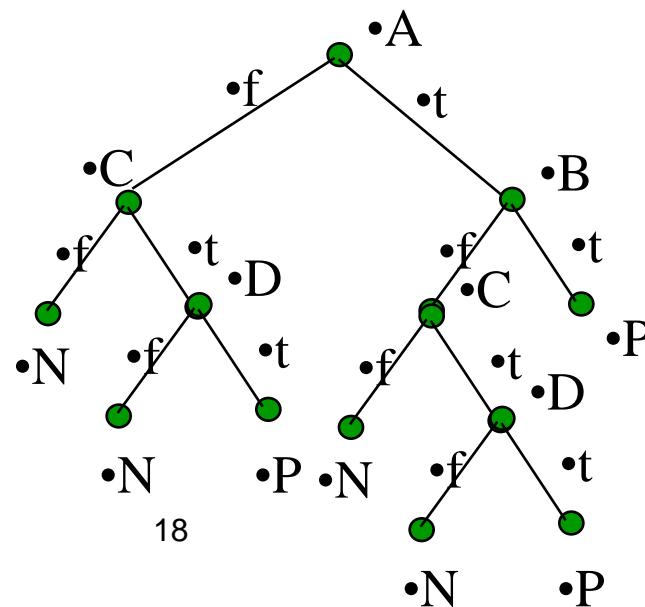
• $A \wedge B \rightarrow P$

• $C \wedge D \rightarrow P$

• What if add rule:

• $E \wedge F \rightarrow P$

• ??



Sequential vs Simultaneous

- **Sequential covering:**
 - learn just one rule at a time, remove the covered examples and
 - repeat the process on the remaining examples
 - many search steps, making independent decisions to select each precondition for each rule
- **Simultaneous covering:**
 - ID3 learns the entire set of disjunct rules simultaneously as part of a single search for a decision tree
 - Fewer search steps, because each choice influences the preconditions of all rules
 - Choice depends of how much data is available
 - Plentiful: sequential covering (more steps supported)
 - Scarce: simultaneous covering (decision sharing effective)

Induction as Inverted Deduction

- **Observation:** induction is just the inverse of deduction.
- In general, machine learning involves building theories that explain the observed data.
- Given some *data* D and some *background knowledge* B , learning can be described as generating a *hypothesis* h that, together with B , explains D .

$$(\forall \langle x_i, f(x_i) \rangle \in D)(B \wedge h \wedge x_i) \vdash f(x_i)$$

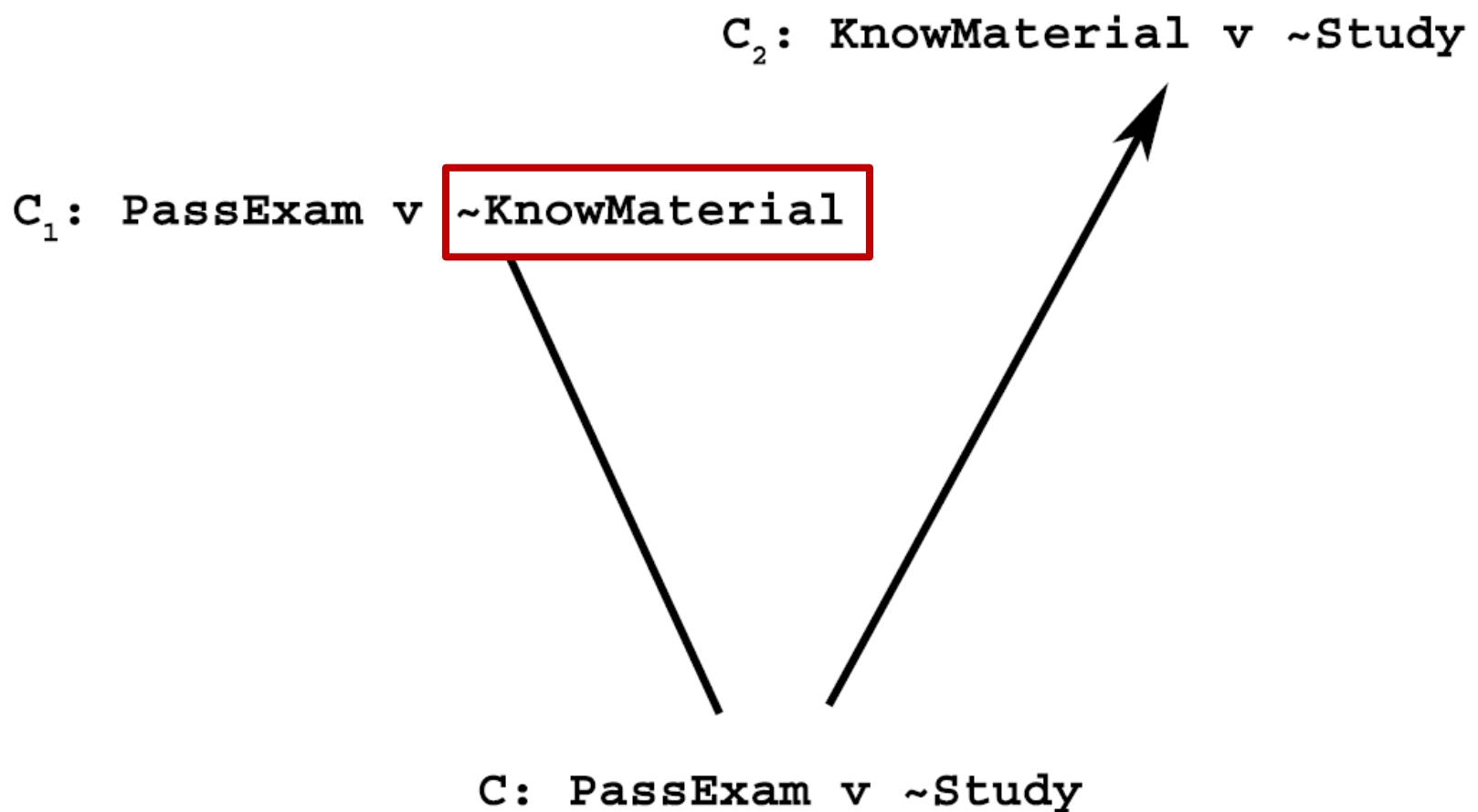
- The above equation casts the learning problem in the framework of deductive inference and formal logic.

Induction as Inverted Deduction

- **Features of inverted deduction:**
 - Subsumes the common definition of learning as finding some general concept.
 - Background knowledge allows a more rich definition of when a hypothesis h is said to “fit” the data.
- **Practical difficulties:**
 - Noisy data makes the logical framework to completely lose the ability to distinguish between truth and falsehood.
 - Search is intractable.
 - Background knowledge often increases the complexity of H .

Inverting Resolution

- Resolution is a general method for automated deduction
- Complete and sound method for deductive inference
- **Inverse Resolution Operator (propositional form):**
 - 1. Given initial clause C_1 and C , find a literal L that occurs in C_1 but not in clause C .

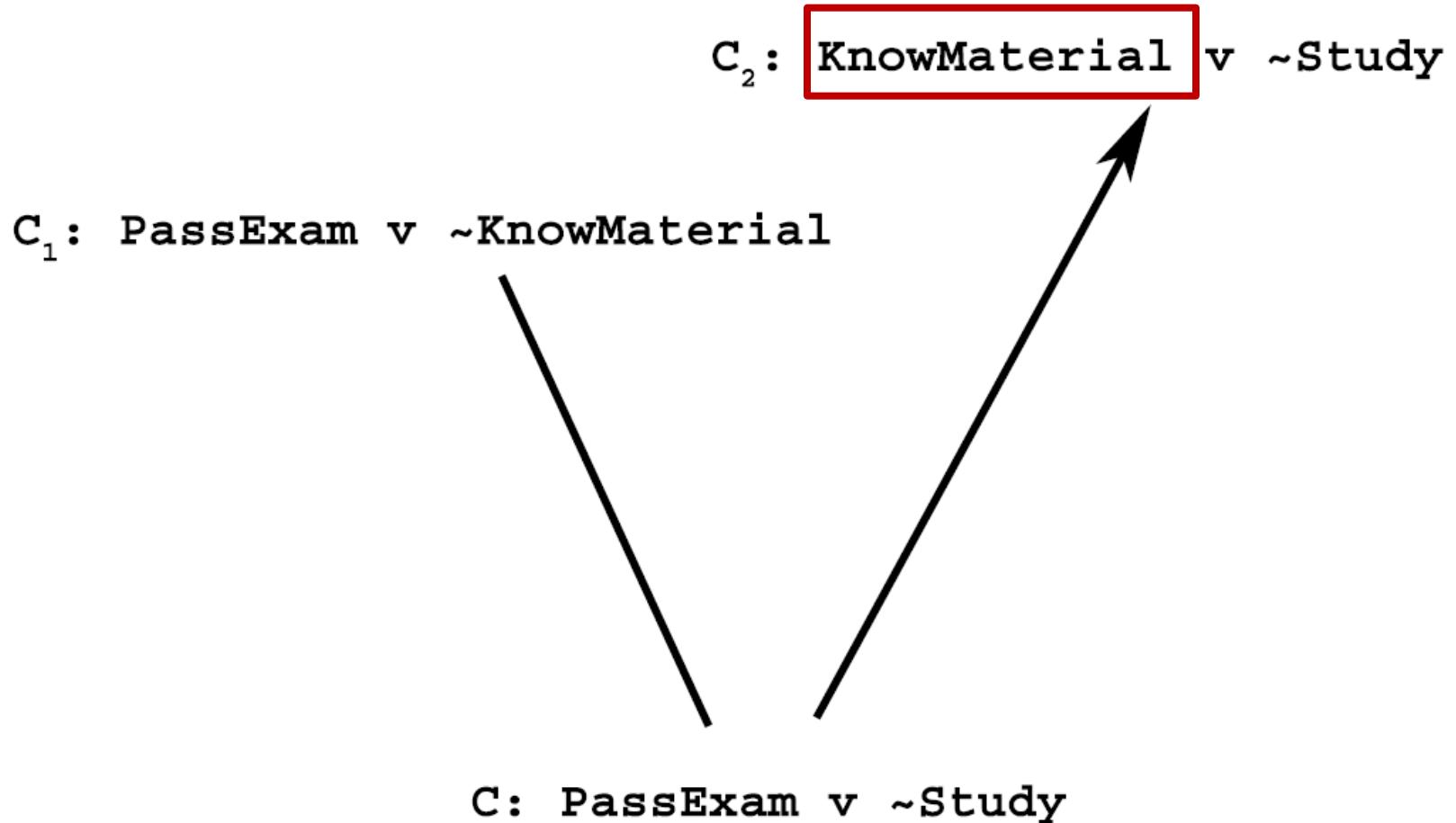


Inverting Resolution

- Resolution is a general method for automated deduction
- Complete and sound method for deductive inference
- **Inverse Resolution Operator (propositional form):**
 - 1. Given initial clause C_1 and C , find a literal L that occurs in C_1 but not in clause C .
 - 2. Form the second clause C_2 by including the following literals

$$C_2 = (C - (C_1 - \{L\})) \cup \{L\}$$

$$C_2 = (C - (C_1 - \{L\})) \cup \{L\}$$



$$\text{GrandChild}(y, x) \vee \neg \text{Father}(x, z) \vee \neg \text{Father}(z, y)$$
$$\text{Father}(\text{Tom}, \text{Bob})$$
$$\{\text{Bob}/y, \text{Tom}/z\}$$
$$\text{GrandChild}(\text{Bob}, x) \vee \neg \text{Father}(x, \text{Tom})$$
$$\text{Father}(\text{Shannon}, \text{Tom})$$
$$\{\text{Shannon}/x\}$$
$$\text{GrandChild}(\text{Bob}, \text{Shannon})$$
$$D = \{\text{GrandChild}(\text{Bob}, \text{Shannon})\}$$
$$B = \{\text{Father}(\text{Shannon}, \text{Tom}), \text{Father}(\text{Tom}, \text{Bob})\}$$

Generalization, θ -Subsumption, Entailment

interesting to consider the relationship between the *more_general_than* relation and inverse entailment

hypothesis can also be expressed as $c(x) \leftarrow h(x)$.

θ – subsumption: Consider two clauses C_j and C_k , both of the form $H \vee L_1 \vee \dots \vee L_n$, where H is a positive literal and the L_i are arbitrary literals. Clause C_j is said to *θ – subsume* clause C_k iff $(\exists \theta)[C_j \theta \subseteq C_k]$.

Entailment: Consider two clauses C_j and C_k . Clause C_j is said to entail clause C_k (written $C_j \vdash C_k$) iff C_j follows deductively from C_k .

ILP Examples

- Learn definitions of family relationships given data for primitive types and relations.

uncle(A,B) :- brother(A,C), parent(C,B).

uncle(A,B) :- husband(A,C), sister(C,D), parent(D,B).

- Learn recursive list programs from I/O pairs.

member(X,[X | Y]).

member(X, [Y | Z]) :- member(X,Z).

append([],L,L).

append([X|L1],L2,[X|L12]):-append(L1,L2,L12).

Ensuring Termination in FOIL

- First empirically determines all binary-predicates in the background that form a well-founded partial ordering by computing their transitive closures.
- Only allows recursive calls in which one of the arguments is reduced according to a known well-founded partial ordering.
 - $\text{path}(X, Y) :- \text{edge}(X, Z), \text{path}(Z, Y).$
 X is reduced to Z by edge so this recursive call is OK
- Due to halting problem, cannot determine if an arbitrary recursive definition is guaranteed to halt.

Inducing Recursive List Programs

- FOIL can learn simple Prolog programs from I/O pairs.
- In Prolog, lists are represented using a logical function
$$:[\text{Head} \mid \text{Tail}].$$
- Since FOIL cannot handle functions, this is re-represented as a predicate:
$$\text{components}(\text{List}, \text{Head}, \text{Tail})$$
- In general, an m -ary function can be replaced by a $(m+1)$ -ary predicate.

Logic Program Induction in FOIL

- FOIL has also learned
 - append given components and null
 - reverse given append, components, and null
 - quicksort given partition, append, components, and null
- Learning recursive programs in FOIL requires a complete set of positive examples for some constrained universe of constants, so that a recursive call can always be evaluated extensionally.
- Negative examples usually computed using a closed-world assumption.
 - Grows combinatorically large for higher arity target predicates.
 - Can randomly sample negatives to make tractable.

FoIL Limitations

- Search space of literals (branching factor) can become intractable.
 - Use aspects of bottom-up search to limit search.
- Requires large extensional background definitions.
 - Use intensional background via Prolog inference.
- Requires complete examples to learn recursive definitions.
 - Use intensional interpretation of learned recursive clauses.

FOIL Limitations (cont.)

- Requires a large set of closed-world negatives.
 - Exploit “output completeness” to provide “implicit” negatives.
- Inability to handle logical functions.
 - Use bottom-up methods that handle functions.
- Background predicates must be sufficient to construct definition, e.g. cannot learn `reverse` unless given `append`.
 - Predicate invention
 - Learn `reverse` by inventing `append`
 - Learn `sort` by inventing `insert`

ILP Settings

Examples:

Positive:	Negative:
bird(penguin)	bird(carp)
bird(eagle)	bird(bat)
bird(crow)	bird(horse)
bird(ostrich)	

Background knowledge:

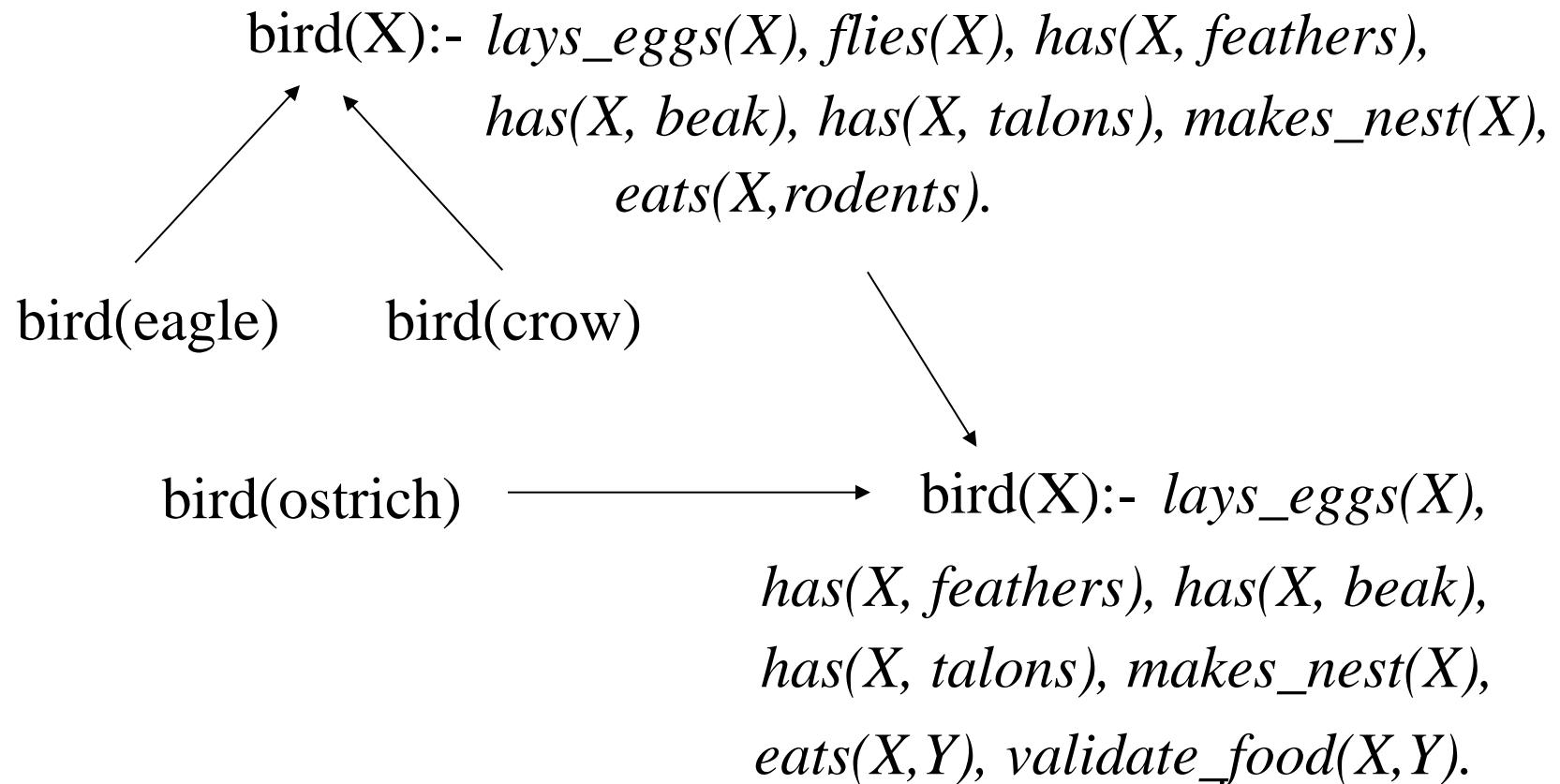
lays_eggs(penguin).	flies(eagle).	swims(carp).	runs(horse).
lays_eggs(crow).	flies(crow).	swims(penguin).	runs(ostrich).
lays_eggs(eagle).	flies(bat).		
lays_eggs(ostrich).	fish(X) :- has_scales(X), swims(X).		
lays_eggs(carp).	mammal(X) :- warm_blooded(X), live_young(X).		

Theory (one or more clauses):

bird(penguin).
bird(X) :- lays_eggs(X), flies(X).
bird(X) :- lays_eggs(X), runs(X).

Bottom-Up Approach

- relative least general generalisation (rlgg)



Used in GOLEM [Muggleton, 90]

Top-down Approach

Some ILP engines use standard top-down search algorithms:
depth-first, breadth-first, A*, etc.

```
bird(X):-  
bird(X):- lays_eggs(X).  
bird(X):- flies(X).  
bird(X):- lays_eggs(X), flies(X).  
...
```

We can improve efficiency by:

- setting a depth-bound (max clauselength).
- paying attention to clause evaluation scores - coverage, MDL.
 - re-ordering candidate clauses based on score
 - pruning candidate clauses below a score threshold
- etc.

Practical Problem Areas

Most commonly encountered:

- Exploring large search spaces
- Positive-only data sets
- Noisy data

Search Space

The hypothesis space is bounded by:

- Maximum clause length
- Size of background knowledge (BK)

Techniques to reduce background knowledge include:

- Excluding redundant predicates
 - Feature subset selection
 - Inverse entailment
- Replacing existing BK with compound predicates (feature construction).

Progol and Aleph's Approach

Uses inverse entailment.

1. Randomly pick a positive example, p .
2. Define the space of possible clauses that could entail that example.
 - Generate the bottom clause, \perp
 - \perp contains all the literals defined in BK that could cover p .
3. Search this space.

Noisy Data

- Techniques to avoid over-fitting.
 - Pre-pruning: limit length of clauses learned
 - Post-pruning: generalise/merge clauses that have a small cover set.
 - Leniency: don't insist on a perfect theory
- Embed the uncertainty into the learning mechanism
 - Stochastic Logic Programs
 - Fuzzy ILP
 - **Diff ILP**

Numerical Reasoning

e.g. `bird(X):- number_of_legs(X,Y), lessthan(Y, 3).`

Many ILP engines don't handle numerical reasoning without help.

- Lazy evaluation [Srinivasan & Camacho, 99]
- Farm it out to another process [Anthony & Frisch, 97]
- (if possible) add predicates to the background knowledge
- First-Order Regression [Karolic & Bratko, 97]

Inventing Predicates

Some ILP engines can invent new predicates and add them to the existing BK.

e.g. Progol uses constraints to call a predicate invention routine.

```
:  
- constraint(invent/2)?
```

```
invent(P,X):- {complicated code that includes asserts}.
```

FOIL only uses extensional BK and so can't use this method.

ILP Systems

- Top-Down:
 - FOIL (Quinlan, 1990)
- Bottom-Up:
 - CIGOL (Muggleton & Buntine, 1988)
 - GOLEM (Muggleton, 1990)
- Hybrid:
 - CHILLIN (Mooney & Zelle, 1994)
 - PROGOL (Muggleton, 1995)
 - ALEPH (Srinivasan, 2000)

Aleph

- file.b: contains the background knowledge (intentional and extensional), the search, language restrictions and types restrictions and the system parameters. (as Prolog clauses).
- file.f: contains the positive examples (only ground facts) to be learned with Aleph;
- file.n: contains the negative examples (only facts without variables) - optional.

Mode Declarations

- Describe the relations (predicates) between the objects and the type of data.
- Declarations inform Aleph if the relation can be used in the head (modeh declarations) or in the body (modeb declarations) of the generated rules.

mode(Recall number, PredicateMode)

- For instance, if we want to declare the predicate `parent_of(P,D)` the recall should be 2, because the daughter D, has a maximum of two parents P.

- Recall number of grandparents(GP, GD) = ?

- The Modes indicates the predicate format, and can be described as:

`predicate(ModeType1, ModeType2, ... , ModeTypen)`

- '+', specifying that when a predicate p appears in a clause, the corresponding argument is an input variable;
- '-', specifying that the corresponding argument is an output variable;
- '#', specifying that the corresponding argument is a constant.

Mode: Example

- Example: for the learning relation uncle_of(U,N) with the background knowledge parent_of(P,D) and sister_of(S1,S2), the mode declarations could be:

```
:‐ modeh(1,uncle_of(+person,+person)).  
:‐ modeb(*,parent_of(-person,+person)).  
:‐ modeb(*,parent_of(+person,-person)).  
:‐ modeb(*,sister_of(+person,-person)).
```

Types

person(john)

person(leihla)

person(richard)

...

Determinations

- Determination statements declare the predicate that can be used to construct a hypothesis

determination(Target Pred/Arity t, Body Pred/Arity b).

determination(aunt_of/2, parent_of/2).

Determinations are only allowed for 1 target predicate on any given run of Aleph: if multiple target determinations occur, the first one is chosen

Positive and Negative Examples

- Positive examples: file with an extension .f
- Negative examples: file with an extension .n

...

```
% Mode declarations
```

```
:‐ modeh(1,aunt_of(+person,+person))?  
:‐ modeb(*,parent_of(-person,+person))?  
:‐ modeb(*,parent_of(+person,-person))?  
:‐ modeb(*,sister_of(+person,-person))?
```

```
% Types
```

```
person(jane).  
person(henry).  
person(sally).  
person(jim).  
person(sam).  
person(sarah).  
person(judy).
```

```
% Background knowledge
```

```
parent_of(Parent,Child) :- father_of(Parent,Child).  
parent_of(Parent,Child) :- mother_of(Parent,Child).
```

```
father_of(sam,henry).
```

```
mother_of(sarah,jim).
```

```
sister_of(jane,sam).  
sister_of(sally,sarah).  
sister_of(judy,sarah).
```

```
% Examples
```

```
aunt_of(jane,henry).          :- aunt_of(henry,sally).  
aunt_of(sally,jim).           :- aunt_of(judy,sarah).  
aunt_of(judy,jim).
```

Output

[Generalising aunt_of(jane,henry).]

[Most specific clause is]

aunt_of(A,B) :- parent_of(C,B), sister_of(A,C).

[Learning aunt_of/2 from positive examples]

[C:-0,12,11,0 aunt_of(A,B).]

[C:6,12,4,0 aunt_of(A,B) :- parent_of(C,B).]

[C:6,12,3,0 aunt_of(A,B) :- parent_of(C,B), sister_of(A,C).]

[C:6,12,3,0 aunt_of(A,B) :- parent_of(C,B), sister_of(A,D).]

[C:4,12,6,0 aunt_of(A,B) :- sister_of(A,C).]

[5 explored search nodes]

f=6,p=12,n=3,h=0

[Result of search is]

aunt_of(A,B) :- parent_of(C,B), sister_of(A,C).

[3 redundant clauses retracted]

aunt_of(A,B) :- parent_of(C,B), sister_of(A,C).

[Total number of clauses = 1]

[Time taken 0.02s]

Symbolic AI

Andre Freitas



Photo by Vasilyev Alexandr

Acknowledgements

- Based on the great slides of:
 - Yoav Artzi, Nicholas FitzGerald and Luke Zettlemoyer, Semantic Parsing with Combinatory Categorial Grammars
 - Combinatory Categorial Grammar: Constraining surface realisation in OpenCCG

This Lecture

- The connection between language, sets and logic
- Semantic Parsing
- Combinatory Categorial Grammars (CCGs)
- How to query KBs using NL

Language to Meaning

at the chair, move forward three steps past the sofa

$$\lambda a. \text{pre}(a, \iota x. \text{chair}(x)) \wedge \text{move}(a) \wedge \text{len}(a, 3) \wedge \\ \text{dir}(a, \text{forward}) \wedge \text{past}(a, \iota y. \text{sofa}(y))$$


Learn

$f : \text{sentence} \rightarrow \text{logical form}$

Lambda Calculus

- Formal system to express computation
- Allows high-order functions

$$\lambda a. move(a) \wedge dir(a, LEFT) \wedge to(a, \iota y. chair(y)) \wedge \\ pass(a, \mathcal{A}y. sofa(y)) \wedge intersect(\mathcal{A}z. intersection(z), y))$$

Lambda Calculus

Base Cases

- Logical constant
- Variable
- Literal
- Lambda term

Lambda Calculus

Logical Constants

- Represent objects in the world

NYC, CA, RAINIER, LEFT, ...

located_in, depart_date, ...

Lambda Calculus

Variables

- Abstract over objects in the world
- Exact value not pre-determined

x, y, z, \dots

Lambda Calculus

Literals

- Represent function application

city(AUSTIN)

located_in(AUSTIN, TEXAS)

Lambda Calculus

Lambda Terms

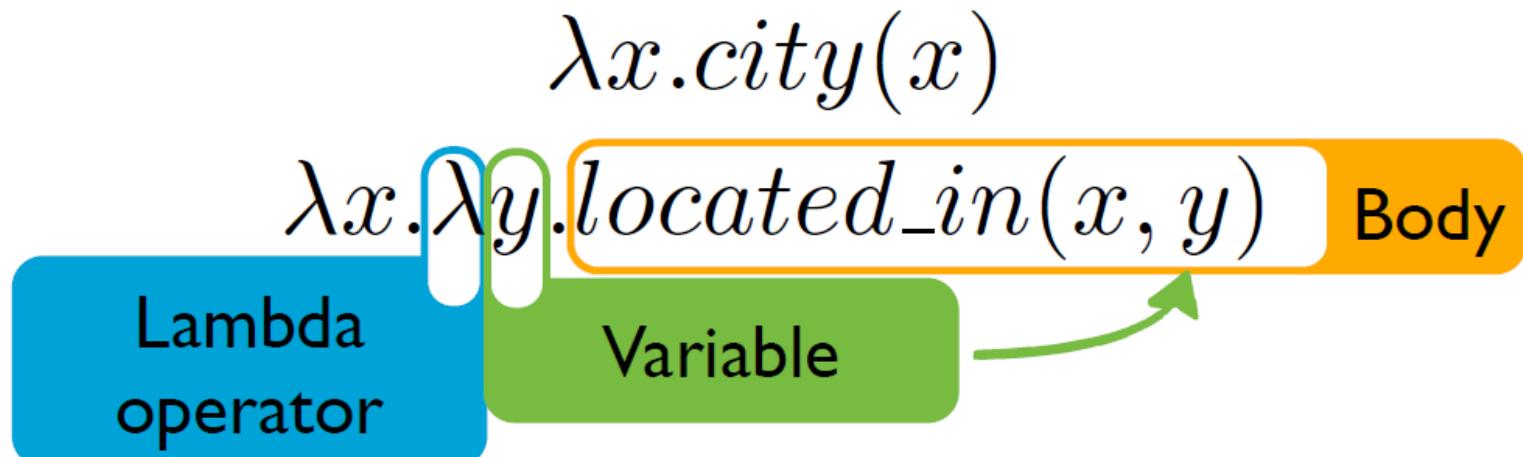
- Bind/scope a variable
- Repeat to bind multiple variables

$$\lambda x.\text{city}(x)$$
$$\lambda x.\lambda y.\text{located_in}(x, y)$$

Lambda Calculus

Lambda Terms

- Bind/scope a variable
- Repeat to bind multiple variables



Capturing Meaning with Lambda Calculus

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV
CA	AZ

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA
Wrangell	AK
Sil	
Ro	



Show me mountains in states
bordering Texas

[Zettlemoyer and Collins 2005]

Capturing Meaning with Lambda Calculus

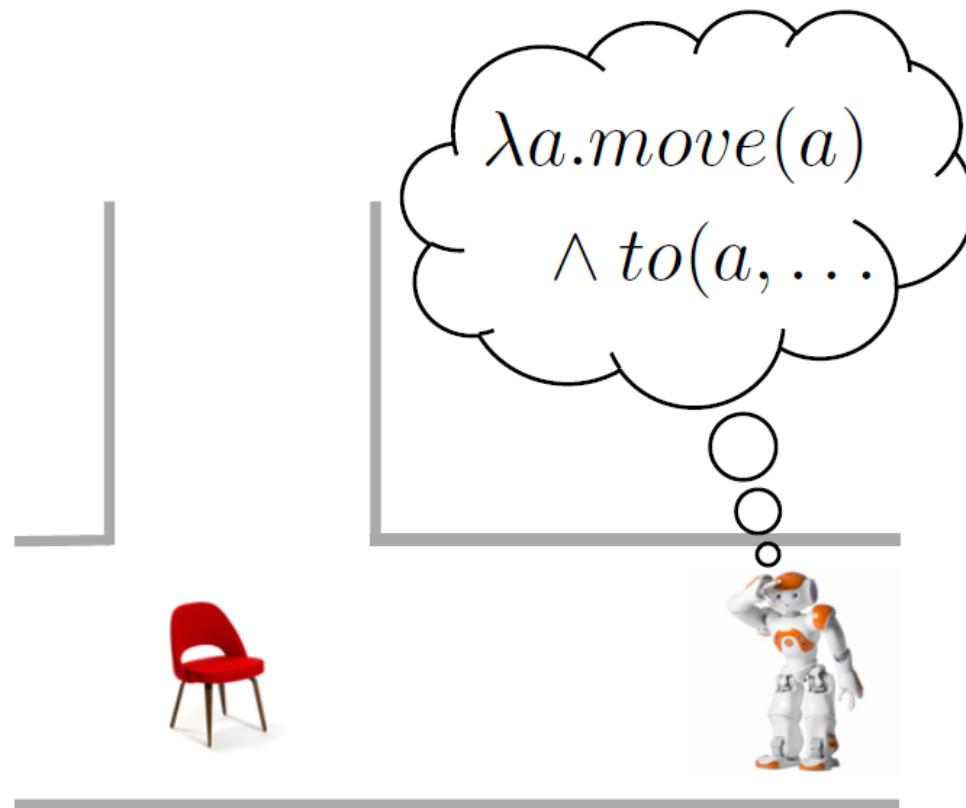
SYSTEM how can I help you ?
USER i ' d like to fly to new york
SYSTEM flying to new york . leaving what city ?
USER from boston on june seven with american airlines
SYSTEM flying to new york . what date would you like to depart boston ?
USER june seventh
SYSTEM do you have a preferred airline ?
USER american airlines
SYSTEM o . k . leaving boston to new york on june seventh flying with
american airlines . where would you like to go to next ?
USER back to boston on june tenth

[CONVERSATION CONTINUES]

[Artzi and Zettlemoyer 2011]

Capturing Meaning with Lambda Calculus

go to the chair
and turn right



[Artzi and Zettlemoyer 2013b]

Capturing Meaning with Lambda Calculus

- Flexible representation.
- Can capture full complexity of natural language.

Constructing Lambda Calculus Expressions

at the chair, move forward three steps past the sofa

Semantic Parsing


$$\lambda a. \text{pre}(a, \iota x. \text{chair}(x)) \wedge \text{move}(a) \wedge \text{len}(a, 3) \wedge \\ \text{dir}(a, \text{forward}) \wedge \text{past}(a, \iota y. \text{sofa}(y))$$

Combinatory Categorial Grammars

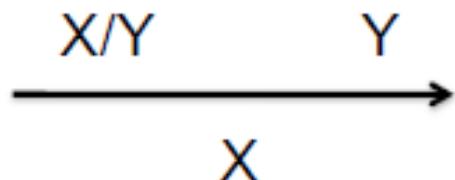
- Categorial formalism.
- Transparent interface between syntax and semantics.
- Designed with computation in mind.

Combinatory Categorial Grammars

$$\begin{array}{ccc} \text{CCG} & \text{is} & \text{fun} \\ \hline NP & \overline{S \setminus NP / ADJ} & \overline{ADJ} \\ CCG & \lambda f. \lambda x. f(x) & \lambda x. fun(x) \\ \hline & \xrightarrow{S \setminus NP} & \\ & \lambda x. fun(x) & \\ \hline & \xleftarrow{S} & \\ & fun(CCG) & \end{array}$$

Formalism

- X/Y : The kind of word or phrase that **combines** with a **following** Y to form an X.

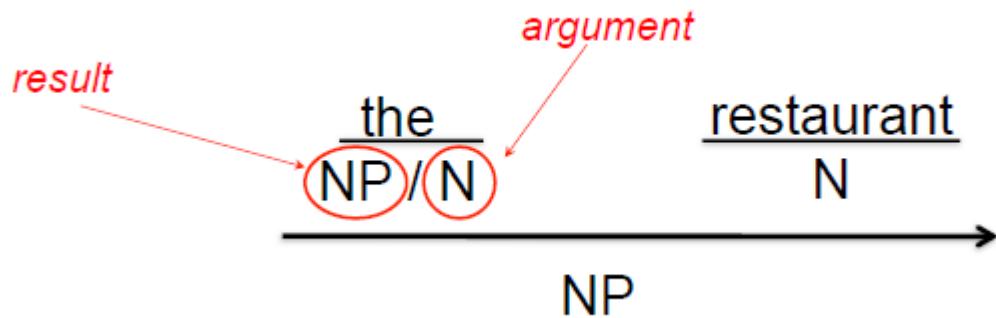


- $X\backslash Y$: kind of word or phrase that combines with a **preceding** Y to form an X.



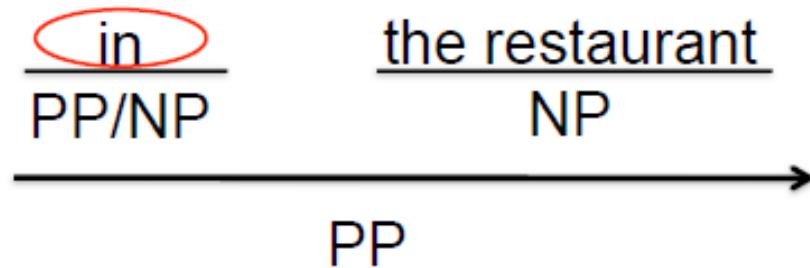
Determiners

- **Determiner:** word that combines with a following N to give an NP, i.e., an NP/N.

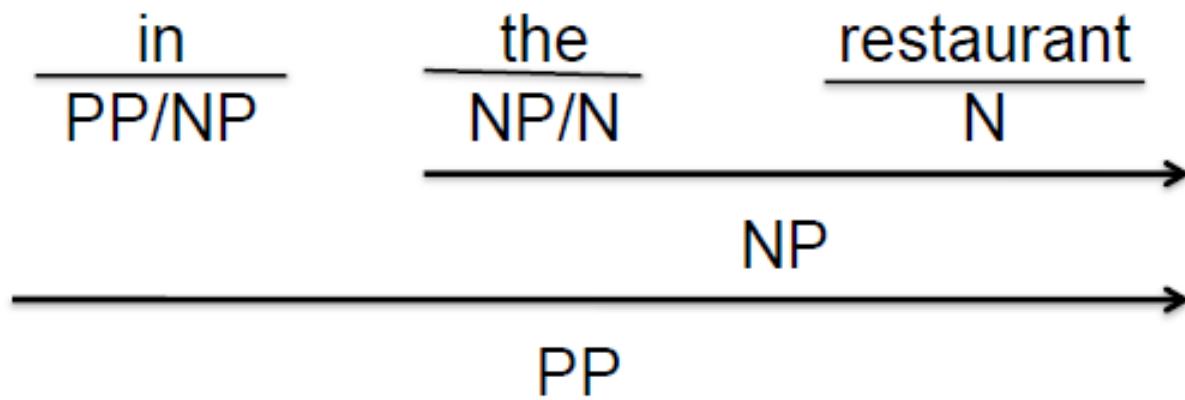


Prepositions

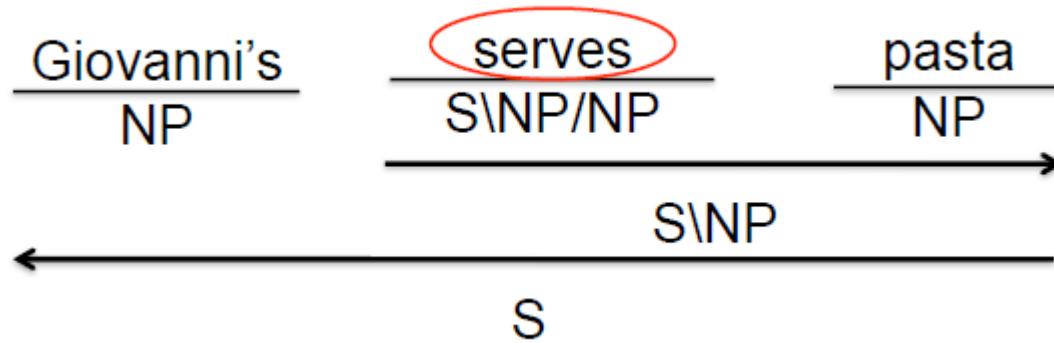
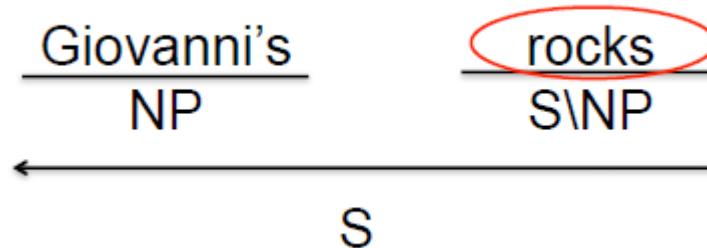
- **Preposition:** word that combines with a following NP to give a PP, i.e., a PP/NP.



Derivation



Verbs



CCG Categories

$$ADJ : \lambda x. fun(x)$$

- Basic building block.
- Capture syntactic and semantic information jointly.

CCG Categories

Syntax

ADJ

$\lambda x. fun(x)$

Semantics

- Basic building block.
- Capture syntactic and semantic information jointly.

CCG Categories

Syntax

$ADJ : \lambda x. fun(x)$

$(S \setminus NP) / ADJ : \lambda f. \lambda x. f(x)$

$NP : CCG$

- Primitive symbols: N, S, NP, ADJ and PP.
- Syntactic combination operator ($/$, \setminus).
- Slashes specify argument order and direction.

CCG Categories

$ADJ : \boxed{\lambda x. fun(x)}$ Semantics

$(S \setminus NP) / ADJ : \lambda f. \lambda x. f(x)$

$NP : CCG$

- λ -calculus expression.
- Syntactic type maps to semantic type.

CCG Lexical Entries

fun $\vdash ADJ : \lambda x. fun(x)$

CCG Lexical Entries

fun

Natural
Language

$ADJ : \lambda x. fun(x)$

CCG Category

CCG Lexicons

fun $\vdash ADJ : \lambda x. fun(x)$

is $\vdash (S \setminus NP) / ADJ : \lambda f. \lambda x. f(x)$

CCG $\vdash NP : CCG$

Parsing with CCGs

CCG

is

fun

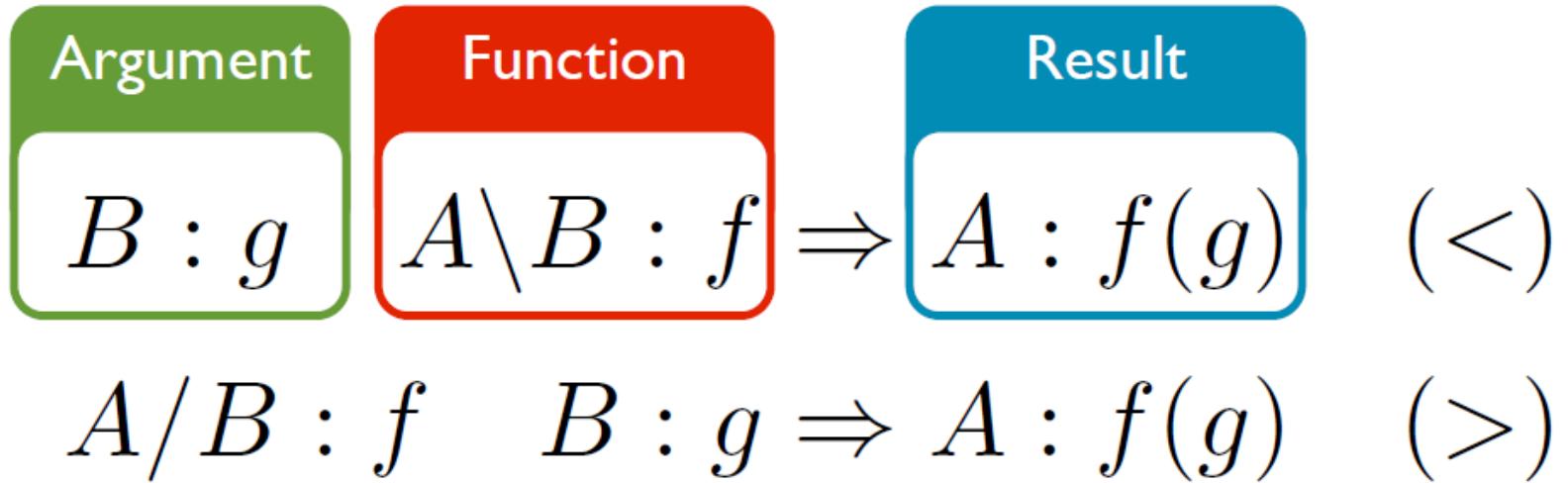
NP
 CCG

$S \setminus NP / ADJ$
 $\lambda f. \lambda x. f(x)$

ADJ
 $\lambda x. fun(x)$

CCG Operations

Application



- Equivalent to function application
- Two directions: forward and backward
 - Determined by slash direction

Parsing with CCGs

$$\frac{\text{CCG}}{\begin{array}{c} NP \\ CCG \end{array}} \quad \frac{\text{is}}{\begin{array}{c} S \setminus NP/ADJ \\ \lambda f. \lambda x. f(x) \end{array}} \quad \frac{\text{fun}}{\begin{array}{c} ADJ \\ \lambda x. fun(x) \end{array}}$$

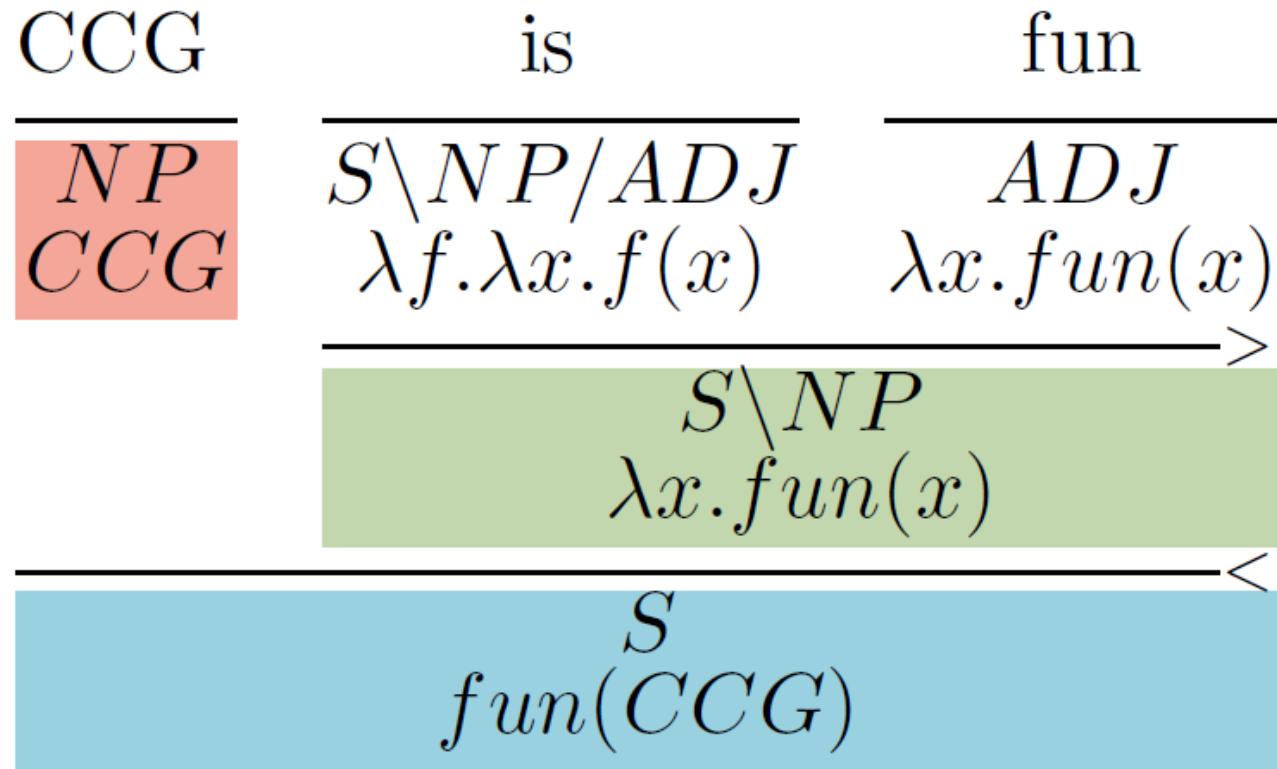
Parsing with CCGs

$$\begin{array}{ccc} \text{CCG} & \text{is} & \text{fun} \\ \hline NP & S \setminus NP / ADJ & ADJ \\ CCG & \lambda f. \lambda x. f(x) & \lambda x. fun(x) \\ \hline & \Rightarrow & \\ & S \setminus NP & \\ & \lambda x. fun(x) & \end{array}$$

Combine categories using operators

$$A/B : f \quad B : g \Rightarrow A : f(g) \quad (>)$$

Parsing with CCGs



Combine categories using operators

$$B : g \quad A \setminus B : f \Rightarrow A : f(g) \quad (<)$$

CCG Operations

Composition

- Equivalent to function composition
- Two directions: forward and backward

$$\begin{array}{c} f \\ A/B : f \end{array} \quad \begin{array}{c} g \\ B/C : g \end{array} \Rightarrow \begin{array}{c} f \circ g \\ A/C : \lambda x. f(g(x)) \end{array} \quad (> B)$$
$$B \setminus C : g \quad A \setminus B : f \Rightarrow A \setminus C : \lambda x. f(g(x)) \quad (< B)$$

Querying Databases

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7
WA	Olympia	4.1
NY	Albany	17.5
IL	Springfield	11.4

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV
CA	AZ

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA
Wrangel	AK
Sill	CA
Bon	AK
Elbe	AK



[Zettlemoyer and Collins 2005]

Querying Databases

State

Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border

State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Mountains

Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Querying Databases

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Noun Phrases

Querying Databases

State	Abbr.	Capital	Pop.
	AL	Montgomery	3.9
	AK	Juneau	0.4
	AZ	Phoenix	2.7

Border	State1	State2
	WA	OR
	WA	ID
	CA	OR
	CA	NV

Mountains	Name	State
	Bianca	CO
	Antero	CO
	Rainier	WA
	Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Verbs

Querying Databases

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV
CA	AZ

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Nouns

Querying Databases

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Prepositions

Querying Databases

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Superlatives

Querying Databases

State

Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border

State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Mountains

Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Determiners

Querying Databases

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

What is the capital of Arizona?

How many states border California?

What is the largest state?

Questions

Referring to DB Entities

Noun phrases

Select single DB entities

Prepositions
Verbs

Relations between entities

Nouns

Typing (i.e., column headers)

Superlatives

Ordering queries

Noun Phrases

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

In this context
Noun phrases name
specific entities

Washington
WA

Florida
The Sunshine State
FL

Noun Phrases

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Noun phrases name specific entities

Washington

*NP
WA*

The Sunshine State

*NP
FL*

Verb Relations

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Verbs express relations between entities

Nevada **borders** California
border(NV, CA)

Verb Relations

State

Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Nevada

NP
 NV

borders

$S \setminus NP/NP$
 $\lambda x. \lambda y. border(y, x)$

California

NP
 CA

$S \setminus NP$
 $\lambda y. border(y, CA)$

S
 $border(NV, CA)$

Nouns

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Nouns are functions
that define entity type

state

$\lambda x.state(x)$

mountain

$\lambda x.mountain(x)$

Nouns

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Nouns are functions
that define entity type

state

$\lambda x.state(x)$

$\{ \text{WA}, \text{AL}, \text{AK}, \dots \}$

$e \rightarrow t$
functions
define sets

mountain

$\lambda x.mountain(x)$

$\{ \text{BIANCA}, \text{ANTERO}, \dots \}$

Nouns

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Nouns are functions
that define entity type

state

N

$\lambda x.state(x)$

mountain

N

$\lambda x.mountain(x)$

Prepositions

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Prepositional phrases are conjunctive modifiers

mountain in Colorado

Prepositions

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Prepositional phrases are conjunctive modifiers

mountain in Colorado

$\lambda x. mountain(x) \wedge$

$in(x, CO)$

{ BIANCA , ANTERO }

Prepositions

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

$$\frac{\begin{array}{c} \text{mountain} \\ \hline N \\ \lambda x.\text{mountain}(x) \end{array} \quad \begin{array}{c} \text{in} \\ \hline PP/NP \\ \lambda y.\lambda x.\text{in}(x, y) \end{array} \quad \begin{array}{c} \text{Colorado} \\ \hline NP \\ CO \end{array}}{\begin{array}{c} PP \\ \lambda x.\text{in}(x, CO) \end{array}} \rightarrow$$

$$\frac{\begin{array}{c} N \setminus N \\ \lambda f.\lambda x.f(x) \wedge \text{in}(x, CO) \end{array}}{\begin{array}{c} N \\ \lambda x.\text{mountain}(x) \wedge \text{in}(x, CO) \end{array}} \leftarrow$$

Function Words

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV

Certain words are used to modify syntactic roles

state **that** borders California

$\lambda x.\text{state}(x) \wedge \text{border}(x, CA)$

$\{ \text{OR}, \text{NV}, \text{AZ} \}$

Function Words

State				
Abbr.	Capital	state	that	borders
AL	Montgomery	$\frac{N}{NV}$	$\frac{PP/(S \setminus NP)}{\lambda f.f}$	$\frac{S \setminus NP/NP}{\lambda x.\lambda y.border(y, x)}$
AK	Juneau			$\frac{NP}{CA}$
AZ	Phoenix			$\frac{S \setminus NP}{\lambda y.border(y, CA)}$
WA	Olympia			$\frac{PP}{\lambda y.border(y, CA)}$
NY	Albany			$\frac{N \setminus N}{\lambda f.\lambda y.f(y) \wedge border(y, CA)}$
IL	Springfield			$\frac{N}{\lambda x.state(x) \wedge (x, CA)}$

Definite Determiners

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Definite determiner
selects the single members
of a set when such exists

$$\iota : (e \rightarrow t) \rightarrow e$$

the mountain in Washington

Definite Determiners

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Definite determiner
selects the single members
of a set when such exists

$$\iota : (e \rightarrow t) \rightarrow e$$

mountain in Washington

$\lambda x.\text{mountain}(x) \wedge \text{in}(x, WA)$

$\{\text{RAINIER}\}$

Definite Determiners

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Definite determiner
selects the single members
of a set when such exists

$$\iota : (e \rightarrow t) \rightarrow e$$

the mountain in Washington

$\iota x.\text{mountain}(x) \wedge \text{in}(x, WA)$



Definite Determiners

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Definite determiner
selects the single members
of a set when such exists

$$\iota : (e \rightarrow t) \rightarrow e$$

the mountain in Colorado

$\iota x.\text{mountain}(x) \wedge \text{in}(x, CO)$

$\left\{ \text{BIANCA}, \text{ANTERO} \right\} \rightarrow \text{X}$

No information to disambiguate

Definite Determiners

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

$$\frac{\text{the}}{NP/N} \frac{}{\lambda f. \iota x. f(x)}$$

mountain in Colorado

.

.

$$\frac{}{\lambda x. \text{mountain}(x) \wedge \text{in}(x, CO)} N$$

$$\frac{}{\iota x. \text{mountain}(x) \wedge \text{in}(x, CO)} NP$$

Indefinite Determiners

State	
Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA

Indefinite determiners are select any entity from a set without a preference

$$\mathcal{A} : (e \rightarrow t) \rightarrow e$$

state with a mountain

$$\lambda x. state(x) \wedge \text{in}(\mathcal{A}y. \text{mountain}(y), x)$$

Superlatives

State	Abbr.	Capital	Pop.
	AL	Montgomery	3.9
	AK	Juneau	0.4
	AZ	Phoenix	2.7
	WA	Olympia	4.1
	NY	Albany	17.5
	IL	Springfield	11.4

Superlatives select optimal entities according to a measure

the largest state

$\text{argmax}(\lambda x.\text{state}(x), \lambda y.\text{pop}(y))$

Min or max ... over this set ... according to this measure

{ WA , AL , AK , ... }

AL	3.9
AK	0.4
Seattle	2.7
San Francisco	4.1
NY	17.5
IL	11.4

Superlatives

State	Abbr.	Capital	Pop.
	AL	Montgomery	3.9
	AK	Juneau	0.4
	AZ	Phoenix	2.7
	WA	Olympia	4.1
	NY	Albany	17.5
	IL	Springfield	11.4

Superlatives select optimal entities according to a measure

the largest state

$\text{argmax}(\lambda x.\text{state}(x), \lambda y.\text{pop}(y))$

Min or max ... over this set ... according to this measure

CA

AL	3.9
AK	0.4
Seattle	2.7
San Francisco	4.1
NY	17.5
IL	11.4

Superlatives

State	
Abbr.	Capit
AL	Montgo
AK	Junea
AZ	Phoen
WA	Olym
NY	Albany
IL	Springf

$$\frac{\text{the largest } NP/N}{\lambda f.\text{argmax}(\lambda x.f(x), \lambda y.\text{pop}(y))} \xrightarrow{\lambda x.\text{state}(x)} NP \text{ argmax}(\lambda x.\text{state}(x), \lambda y.\text{pop}(y))$$

state

N

$\lambda x.\text{state}(x)$

→

Superlatives

State

Abbr.	Capital
AL	Montgomery
AK	Juneau
AZ	Phoenix
WA	Olympia
NY	Albany
IL	Springfield

$$\begin{array}{c}
 \text{the most} \\
 \hline
 NP/N/N \\
 \lambda g. \lambda f. argmax(\lambda x. f(x), \lambda y. g(y)) \\
 \xrightarrow{\quad N \quad} \\
 \lambda x. pop(x) \\
 \xrightarrow{\quad N \quad} \\
 \lambda x. state(x)
 \end{array}$$

$$\begin{array}{c}
 NP/N \\
 \lambda f. argmax(\lambda x. f(x), \lambda y. pop(y)) \\
 \xrightarrow{\quad NP \quad} \\
 argmax(\lambda x. state(x), \lambda y. pop(y))
 \end{array}$$

Representing Questions

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4

Border	
State1	State2
WA	OR
WA	ID
CA	OR

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA

Which mountains are in Arizona?

Represent questions as the queries that generate their answers

Representing Questions

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4

State1	State2
WA	OR
WA	ID
CA	OR

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA

Which mountains are in Arizona?

$$\lambda x.\text{mountain}(x) \wedge \text{in}(x, AZ)$$

Represent questions as the queries that generate their answers

Representing Questions

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4

Border	
State1	State2
WA	OR
WA	ID
CA	OR

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA

How many states border California?

$\text{count}(\lambda x. \text{state}(x) \wedge \text{border}(x, CA))$

Represent questions as the queries that generate their answers

Spatial and Instructional Language

Name objects

Noun phrases

Specific entities

Nouns

Sets of entities

Prepositional phrases

Adjectives

Constrain sets

Instructions to execute

Verbs

Davidsonian events

Imperatives

Sets of events

Neo-Davidsonian Event Semantics

- Vincent shot Marvin in the car accidentally

$$\exists a. shot(a) \wedge agent(a, VINCENT) \wedge \\ patient(a, MARVIN) \wedge in(a, \iota x. car(x)) \wedge \neg intentional(a)$$

Summary

- The connection between language, sets and logic
- Semantic Parsing
- Combinatory Categorial Grammars (CCGs)
- How to query KBs using NL

Recommended Reading

A Very Short Introduction to CCG*

Mark Steedman

Draft, November 1, 1996

<http://cs.brown.edu/courses/csci2952d/readings/lecture5-steedman.pdf>

Recommended Reading

Open-Domain Semantic Parsing with Boxer

Johan Bos

Center for Language and Cognition

University of Groningen

johan.bos@rug.nl

<http://cs.brown.edu/courses/csci2952d/readings/lecture8-bos.pdf>