Symbolic AI

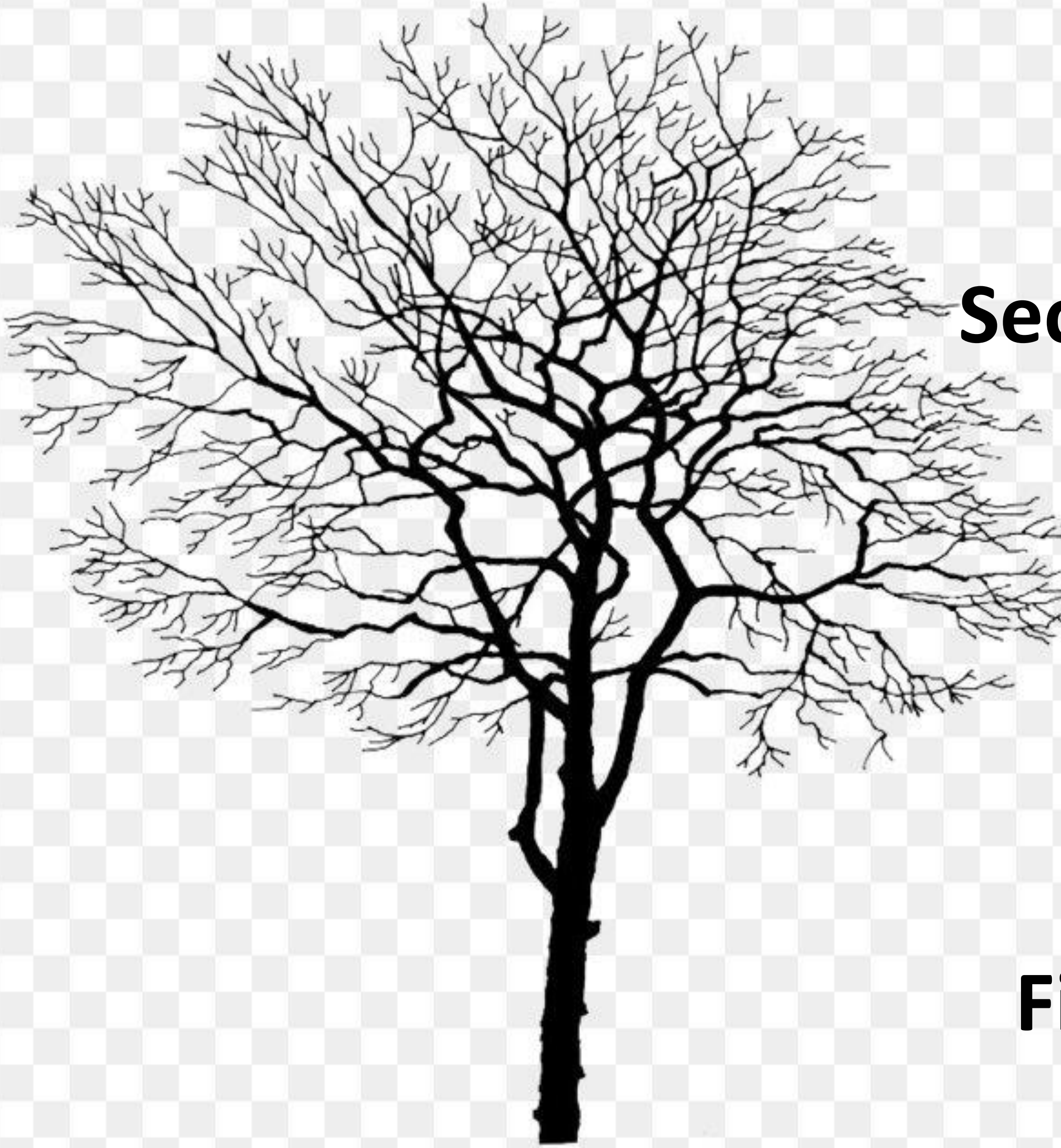
Andre Freitas

Photo by Vasilyev

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.

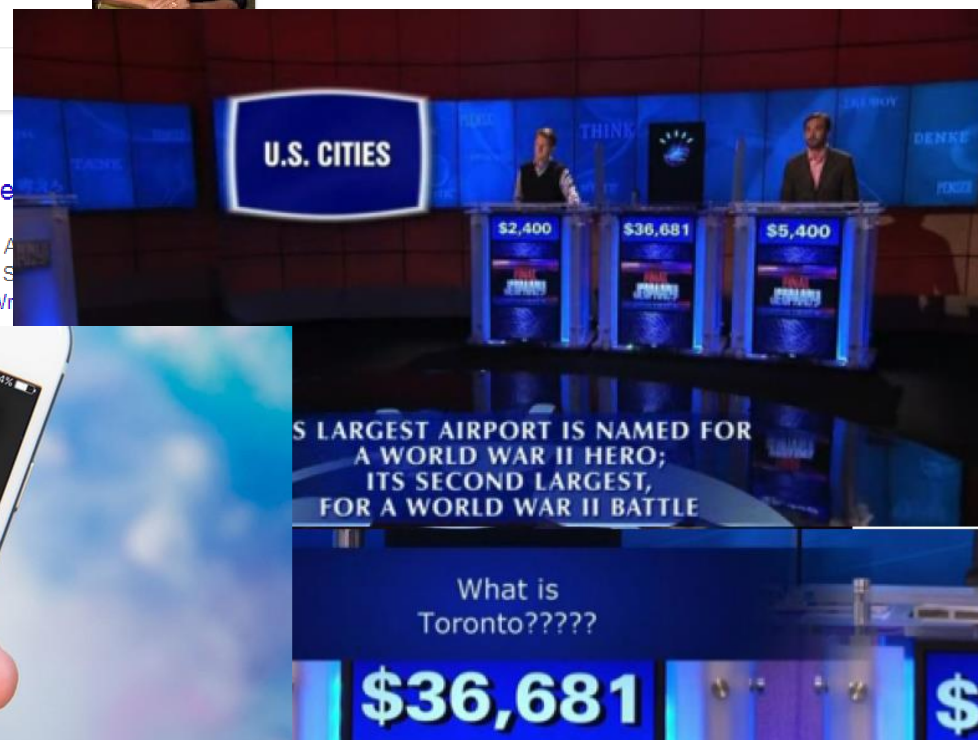
Employability Skills

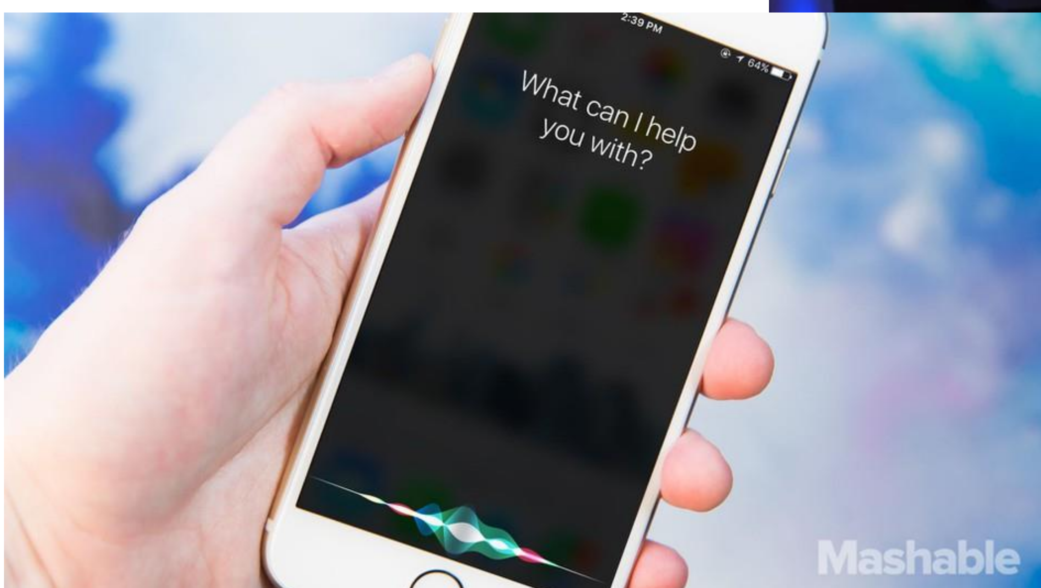
- This unit is heavily complementary to Machine Learning and Text Mining.

- Contemporary AI is evolving rapidly in the direction of hybrid **neuro-symbolic models**.

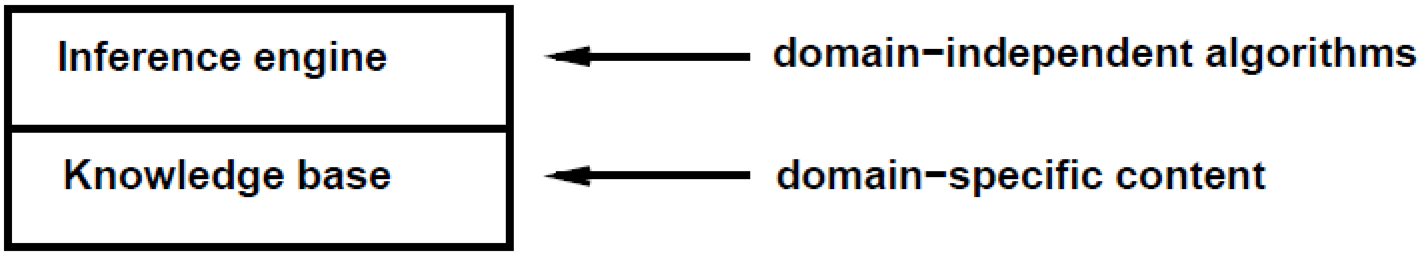
- Larger palette to build AI Systems.


who is the wife of barack obama
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 encyclope
en.wikipedia.org/wiki/Michelle_Obama
Wikipedia
Michelle LaVaughn Robinson Obama (born January 17, 1964), an A
writer, is the wife of the 44th and current President of the United S
Capers Funnye - Sidley Austin - Hyde Park, Chicago - Jeremiah Wr


U.S. CITIES
$2,400
$36,681
$5,400
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


2:39 PM
What can I help
you with?
Mashable

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

Russell & Norvig

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*

Russell & Norvig

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

Russell & Norvig

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

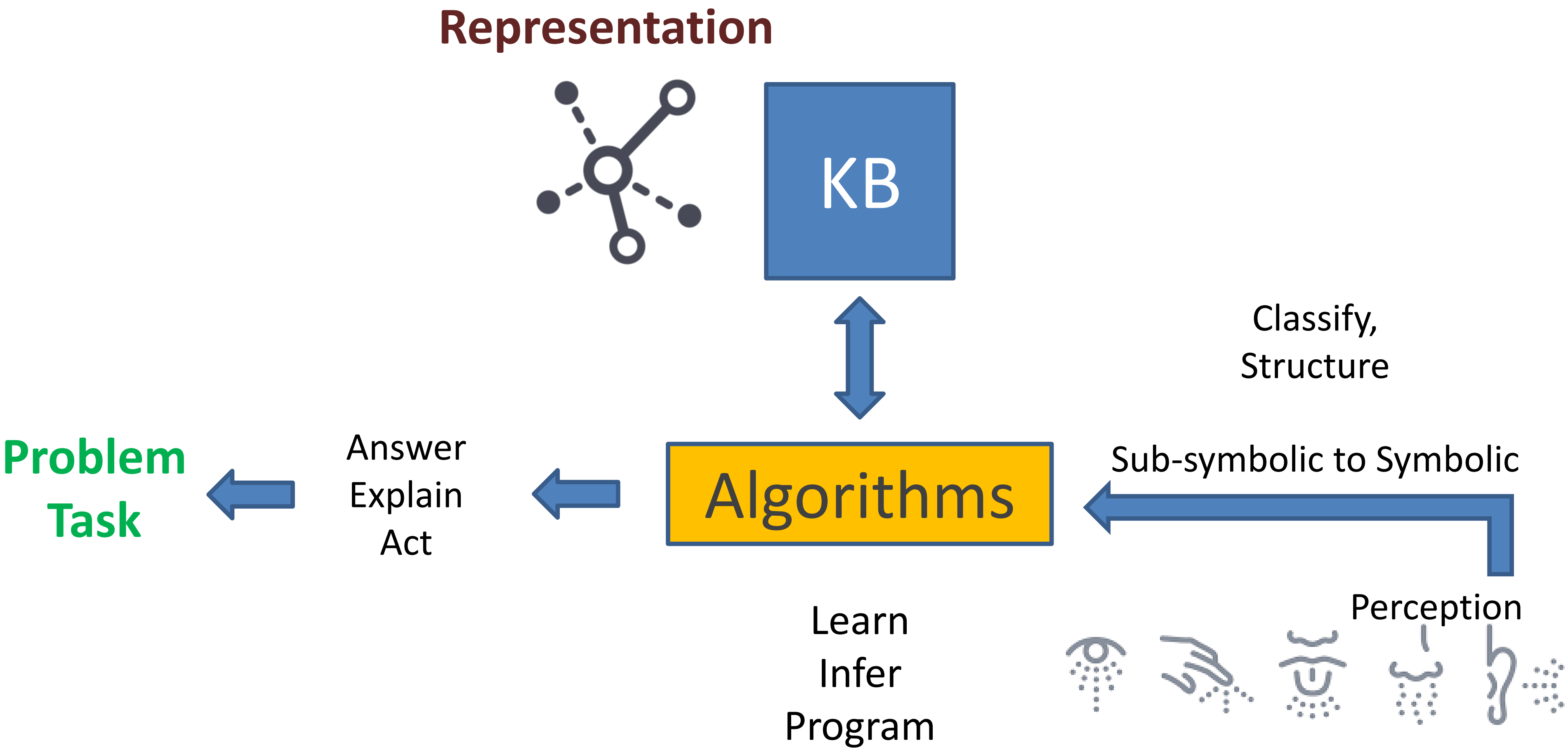
Russell & Norvig

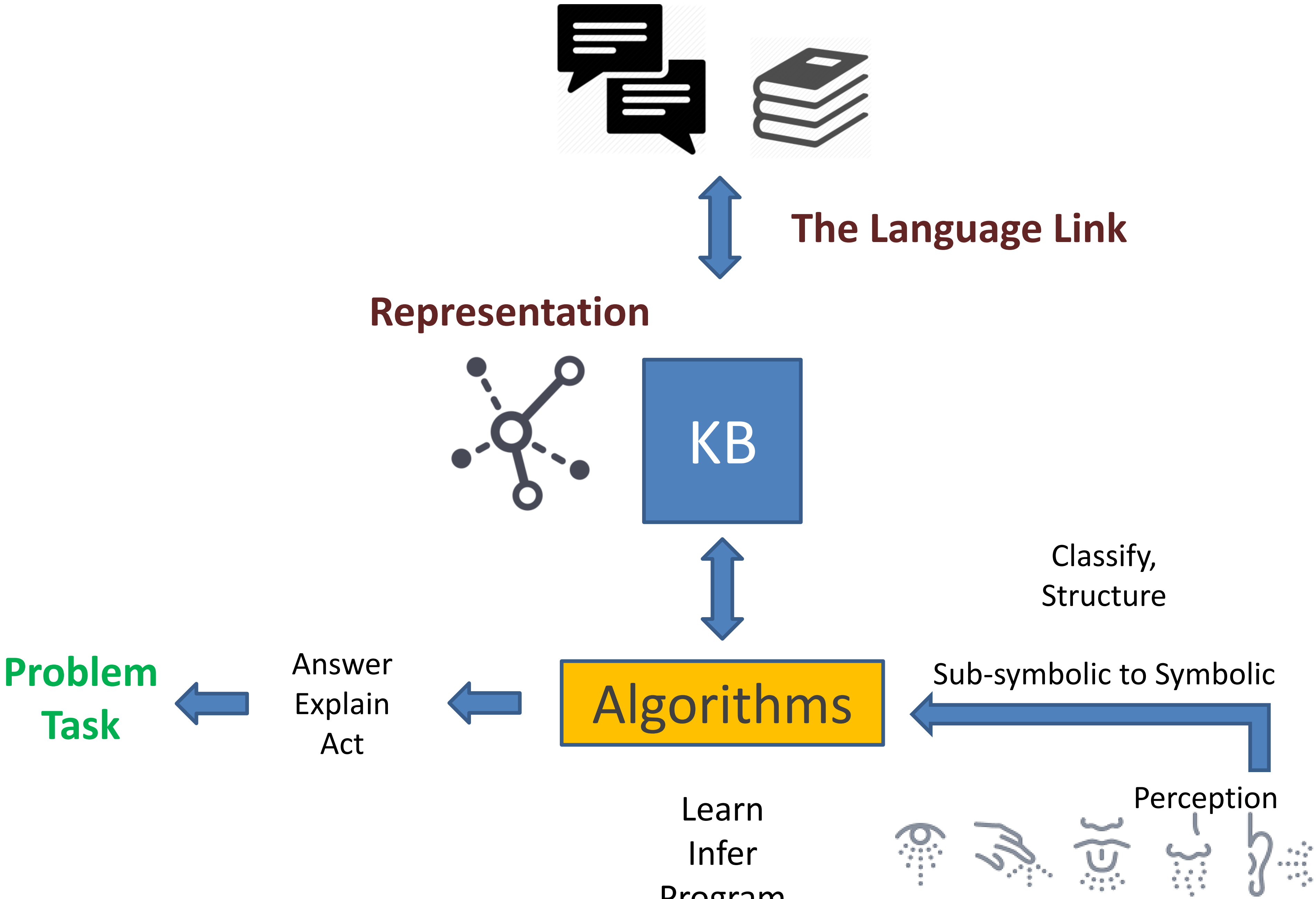
Goals of this second part of the course

- 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
KB
Classify,
Structure
Sub-symbolic to Symbolic
Problem
Task
Answer
Explain
Act
Algorithms
Learn
Infer
Program
Perception

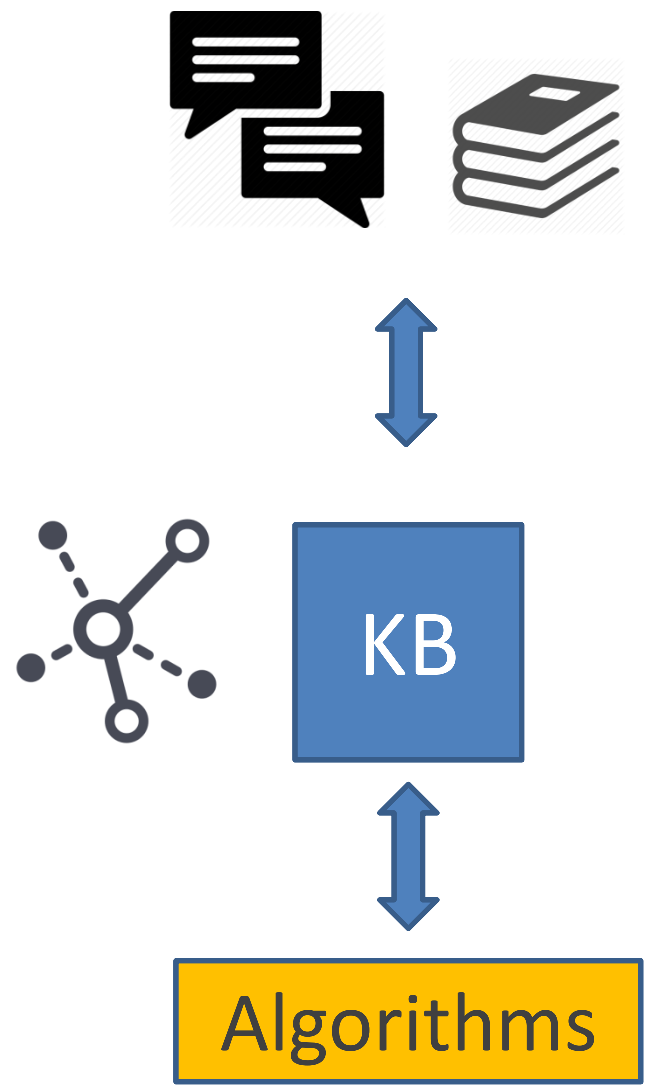

The Language Link
Representation
KB
Classify,
Structure
Problem
Task
Answer
Explain
Act
Algorithms
Sub-symbolic to Symbolic
Perception
Learn
Infer
Program

Syllabus

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

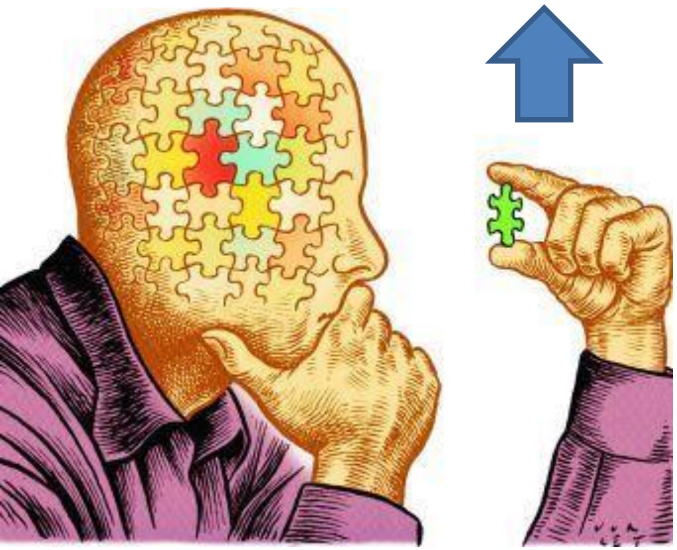
1. Knowledge Representation

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



Knowledge Representation

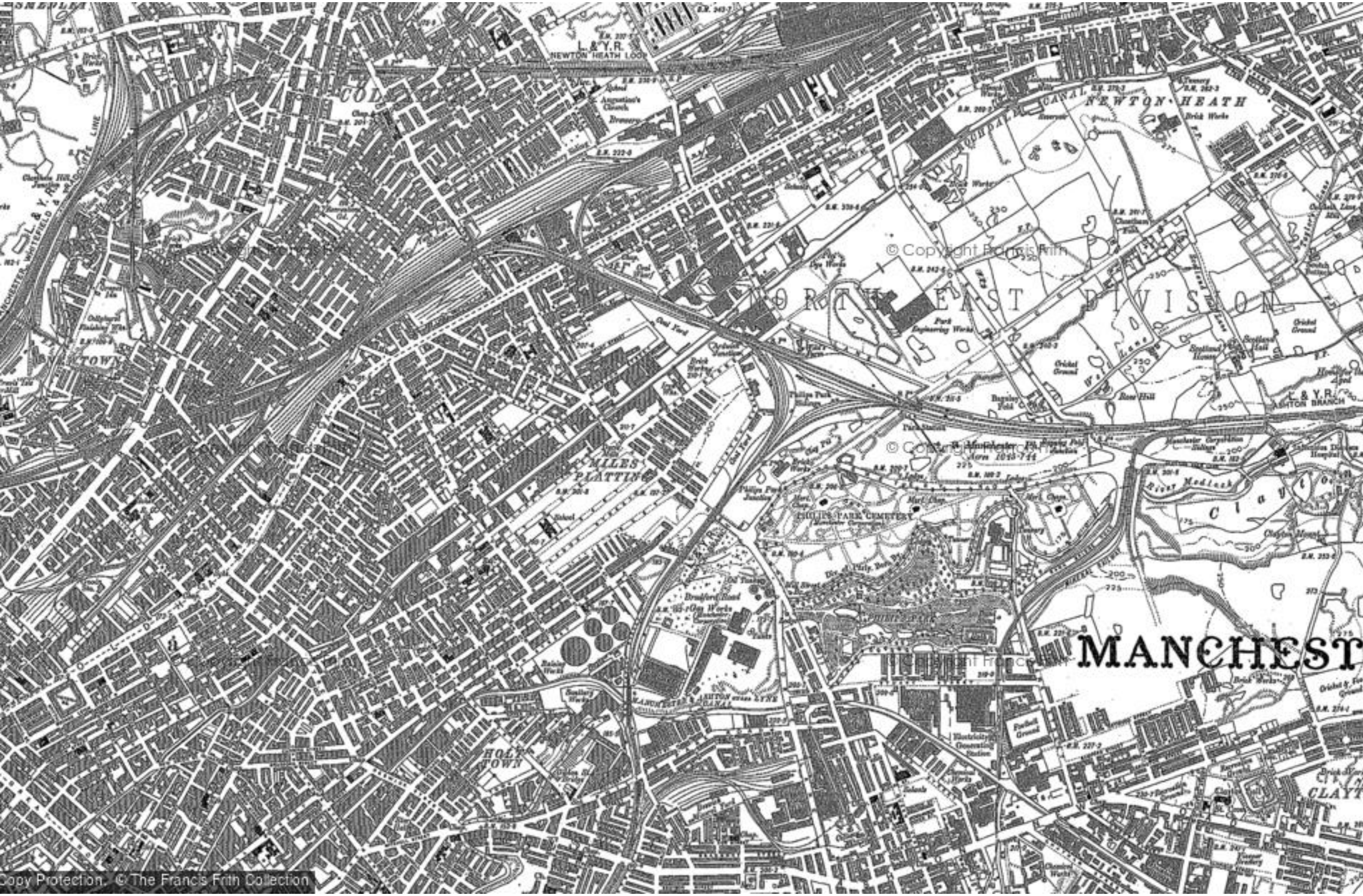
Communication of the Representation

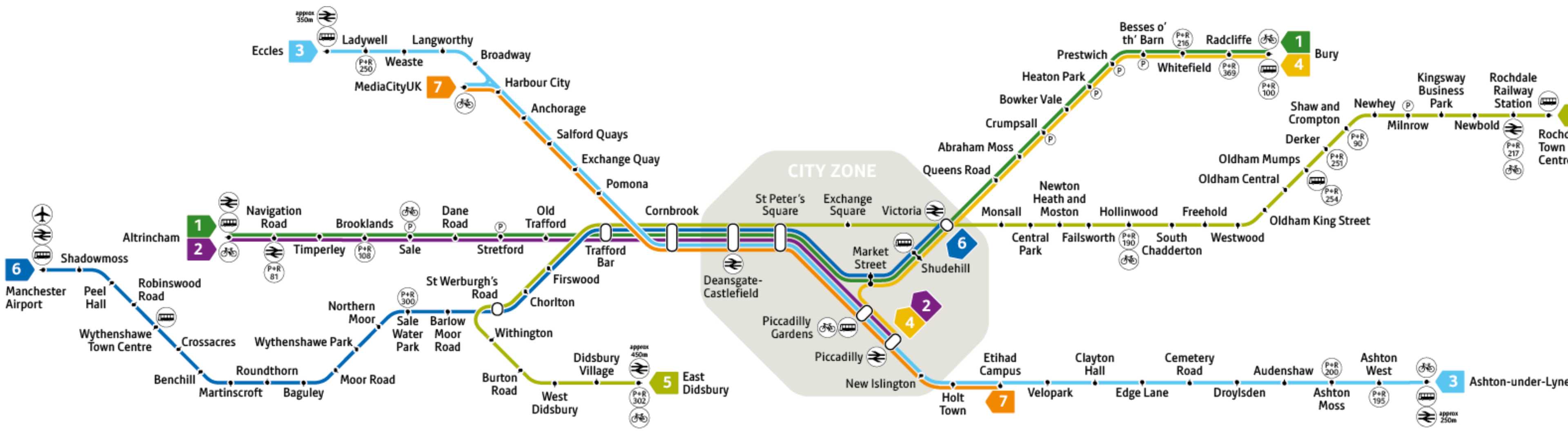


Representation of the Reality

Structure of the Reality


MANCHESTER
City Centre
Manchester Victoria
Victoria
Shudehill
Market Street
Piccadilly Gardens
St Peter's Square
Manchester Piccadilly
G-Mex
Deansgate
Manchester Oxford Road
Salford Central
Cornbrook
Wallness
New Windsor
Ancoats
Brunswick
University
Hulme
Ardwick
River Irwell
River Medlock
Silk Street
Blackfriars Road
Chapel Street
Deansgate
Oldfield Road
Liverpool Road
Lever Street
Oldham Road
Pollard Street
Stretford Road
Whitworth Street
Water Street
St Stephen Street
Elton Street
Dantzic Street
Livesey Street
Butler Street
Old Mill Street
Ducie Street
Store Street
Every Street
Ordsall La
Royce Road
City Road
Brunswick Street
Higher Ardwick
Corn Brook
Bridgewater Way
Metrolink
A664
A6042
A6
A57
A57(M)
A5067
A34
A635
A665


L.&Y.R.
NEWTON HEATH LOCO
NEWTON HEATH
NORTH EAST DIVISION
MILES PLATTING
PHILIPS PARK CEMETERY
L.&Y.R.
ASHTON BRANCH
River Medlock
HOLT TOWN
MANCHESTER ASHTON under LYNE CANAL
MANCHESTER
CLAYTON
Copy Protection. © The Francis Frith Collection



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

* Early services operate on a 20 minute frequency between Manchester Airport and Deansgate-Castlefield. Please check journey planning posters or **tfgm.com** before travelling.

Key

Metrolink stop

Change with other servic

Bus interchange

Rail interchange

Park + Ride with number of spaces

Car park fewer than 50 spac

Cycle Hub membership required

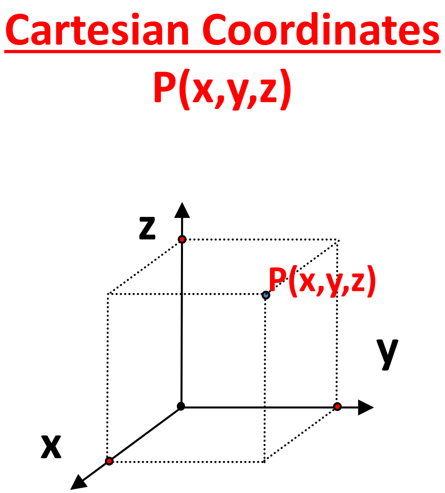

Ceci n'est pas une pipe.
Magritte

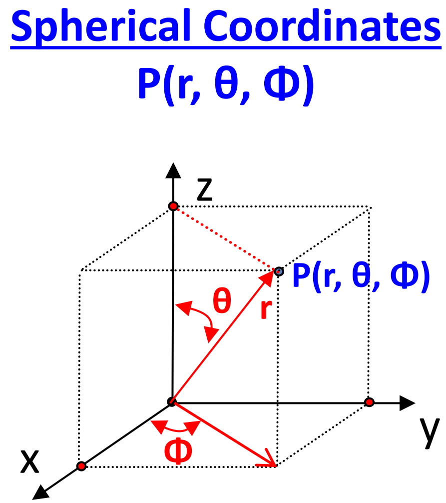
"Human knowledge is a process of approximation. In the focus of experience, there is comparative clarity. But the discrimination of this clarity leads into the penumbral background. There are always questions left over. The problem is to discriminate exactly what we know vaguely."

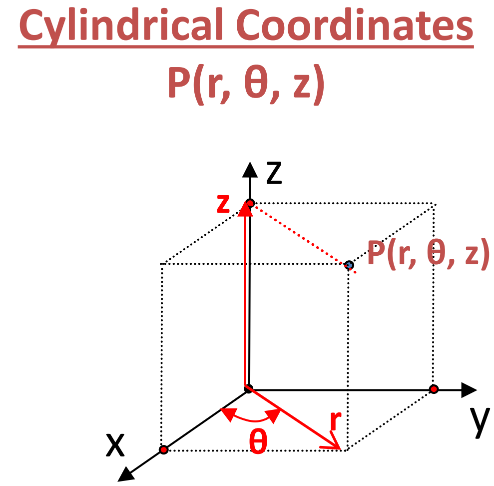
Alfred North Whitehead

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



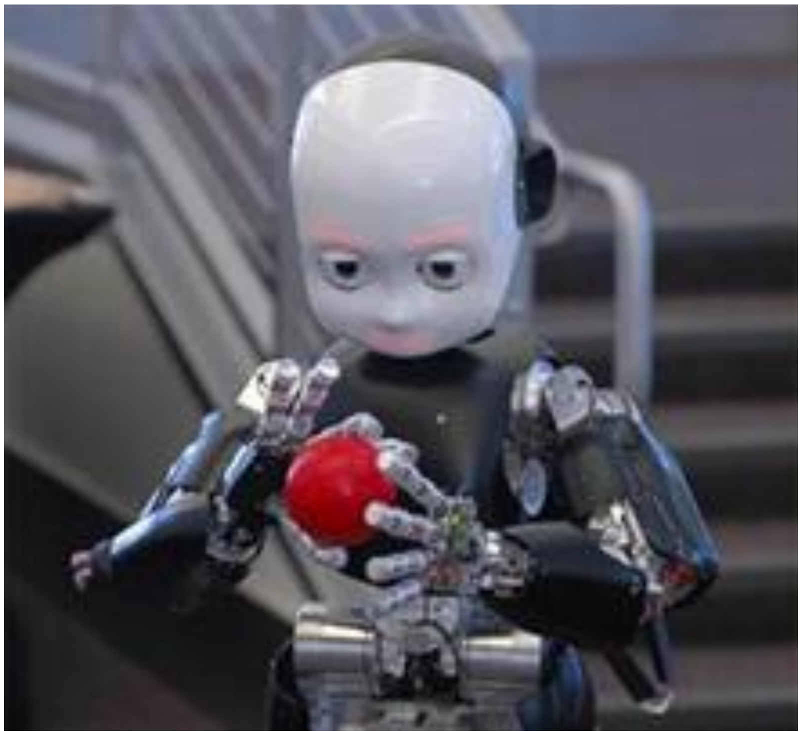




$$ds^2 = \left(1 - \frac{2m}{r}\right) dt^2 - \frac{1}{\left(1 - \frac{2m}{r}\right)} dr^2$$

$$-(r)^2 (d\theta^2 + \sin^2(\theta) d\phi^2)$$

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 <u>their uncanny ability to create maps</u>... 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!

Semantics

=

Formal meaning representation model (lots of data)

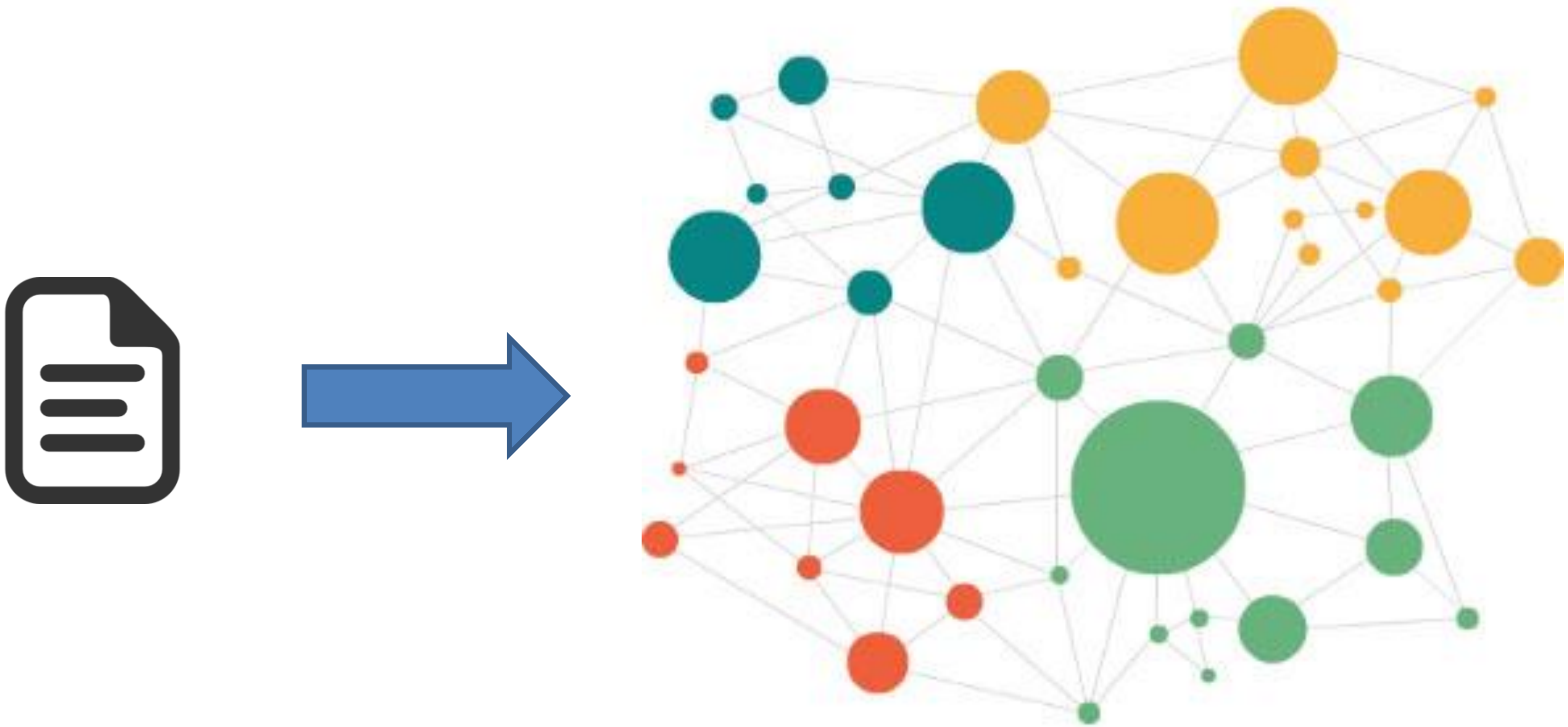
+

inference model

> 2000 years of tradition!

Logics, linguistics, philosophy, cognitive sciences, computer science

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.

...

Named Entity recognition, disambiguation

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**.

...

Relation extraction

Regularities in Natural Language

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

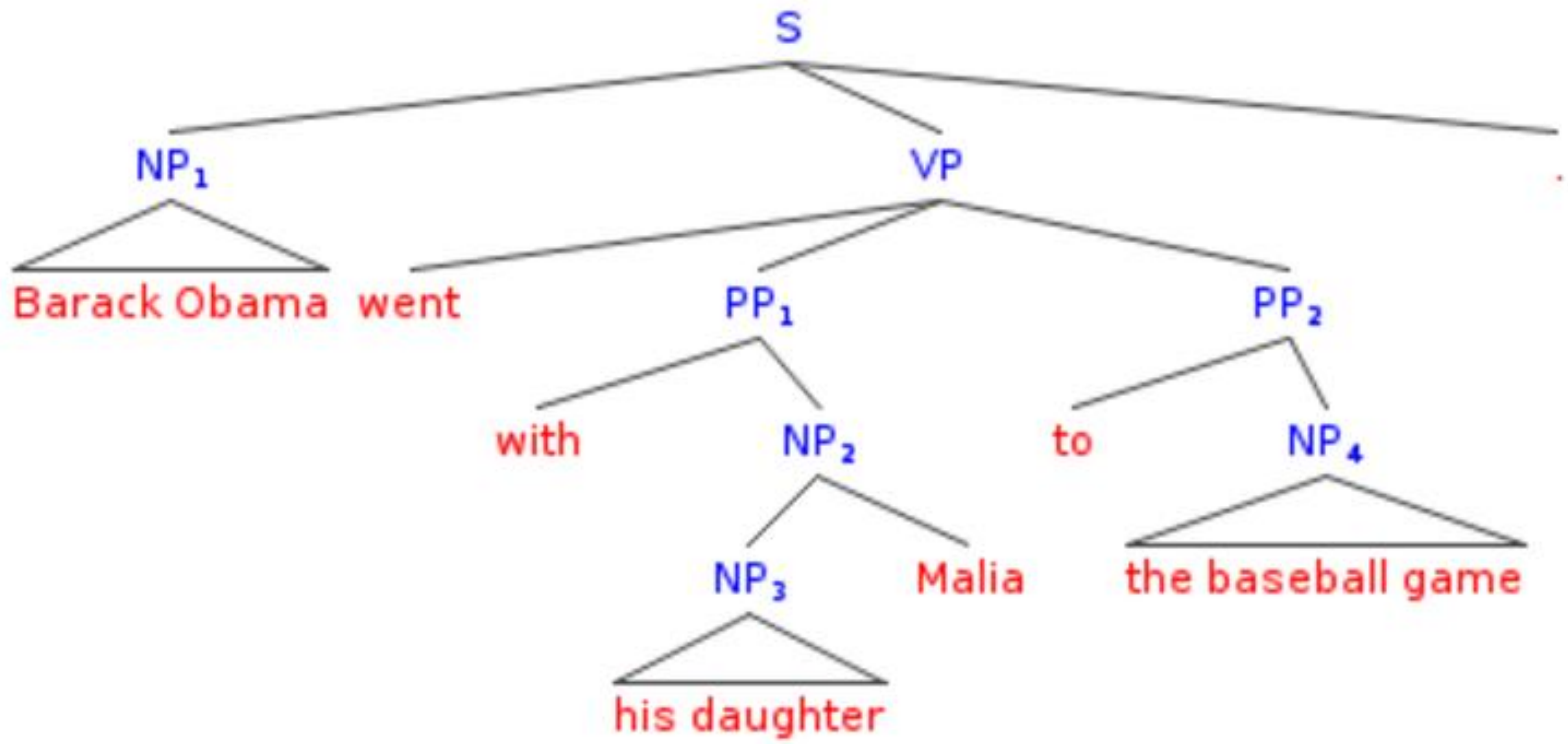
Barack/NNP
Obama/NNP
went/VBD
with/IN
his/PRP$
daughter/NN
Malia/NN
to/TO
the/DT
baseball/NN
game/NN
./.

Regularities in Natural Language

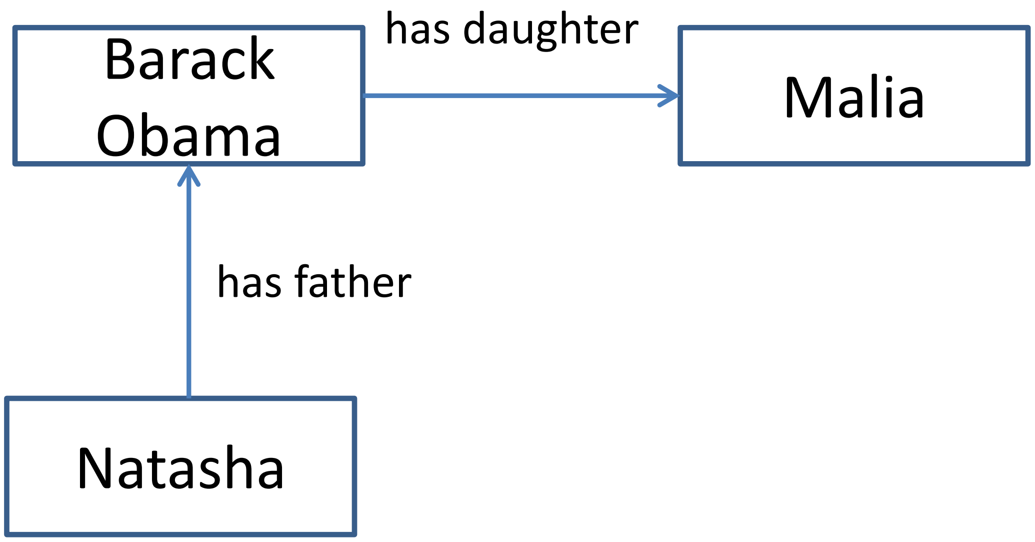
Barack Obama went with his daughter Malia to the baseball game.

Barack/**NNP** → Proper noun
Obama/**NNP**
went/VBD
with/IN
his/PRP$
daughter/NN
Malia/NN
to/TO
the/DT
baseball/NN
game/NN
./.

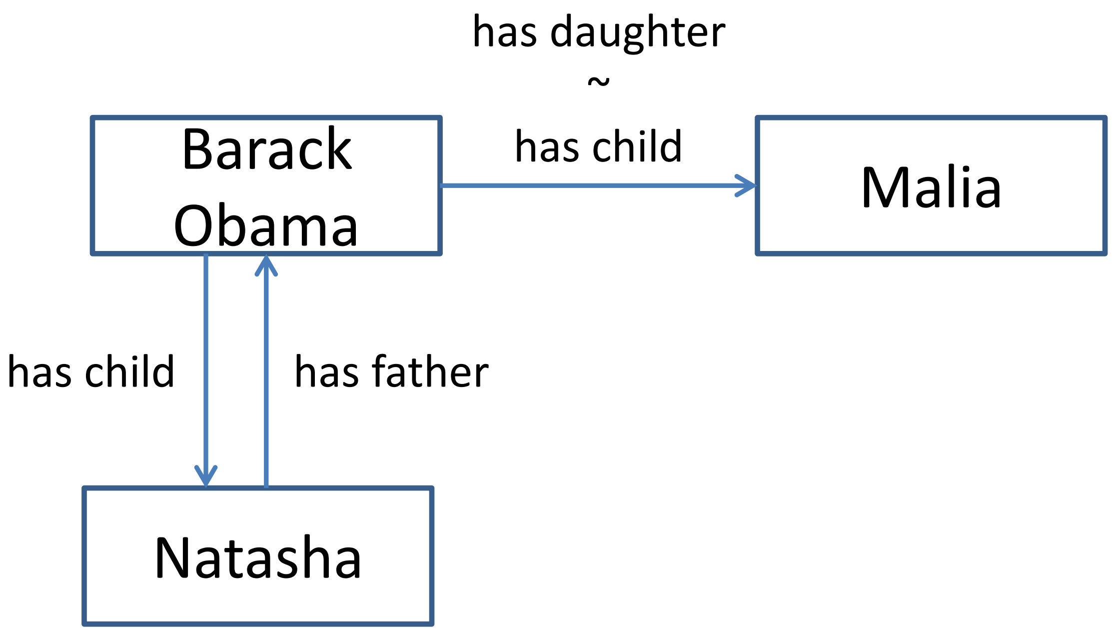
Regularities in Natural Language



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

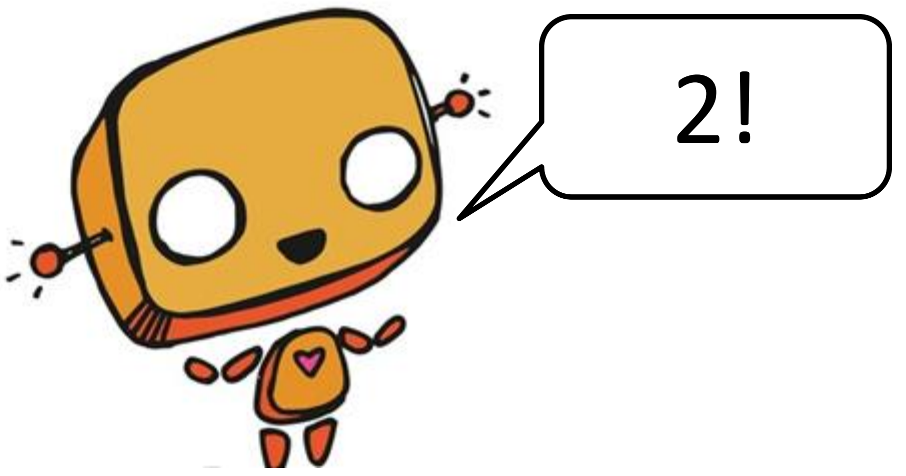
- *How many children does Barack Obama have?*

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

- count(has_child(Barack_Obama, ?x))

It computes!

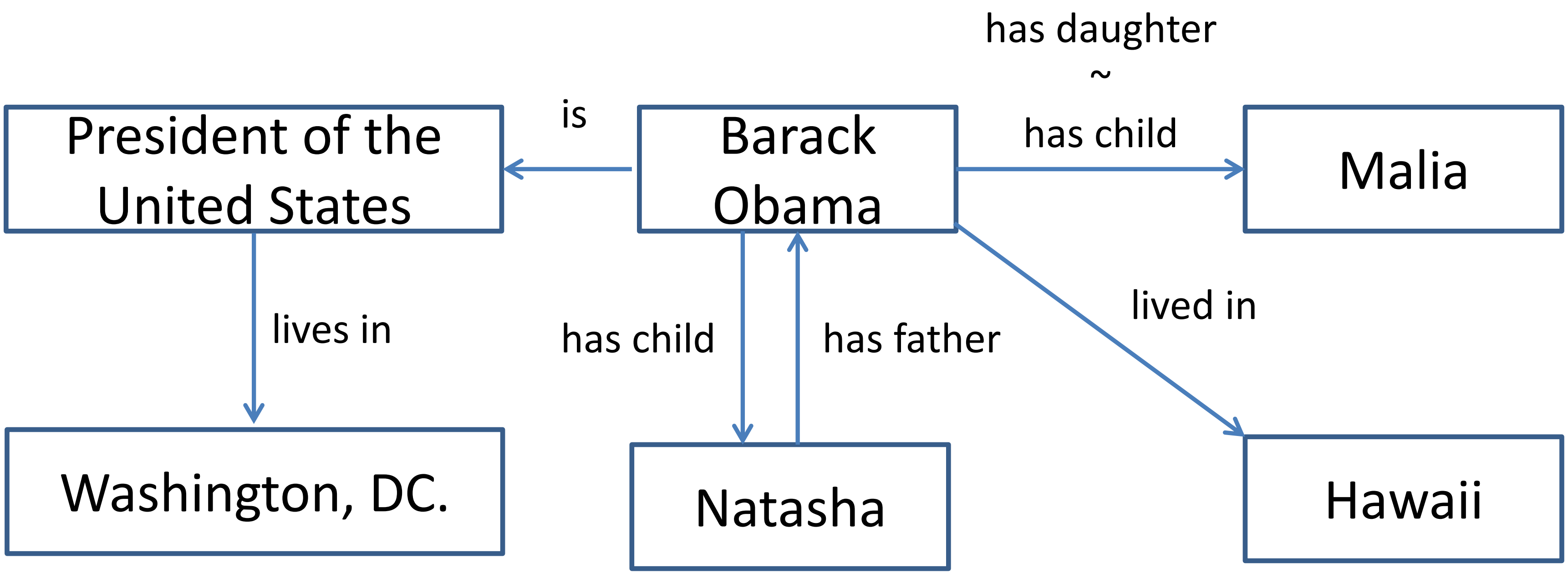
Query: count(has_child(Barack_Obama, ?x))



KB:

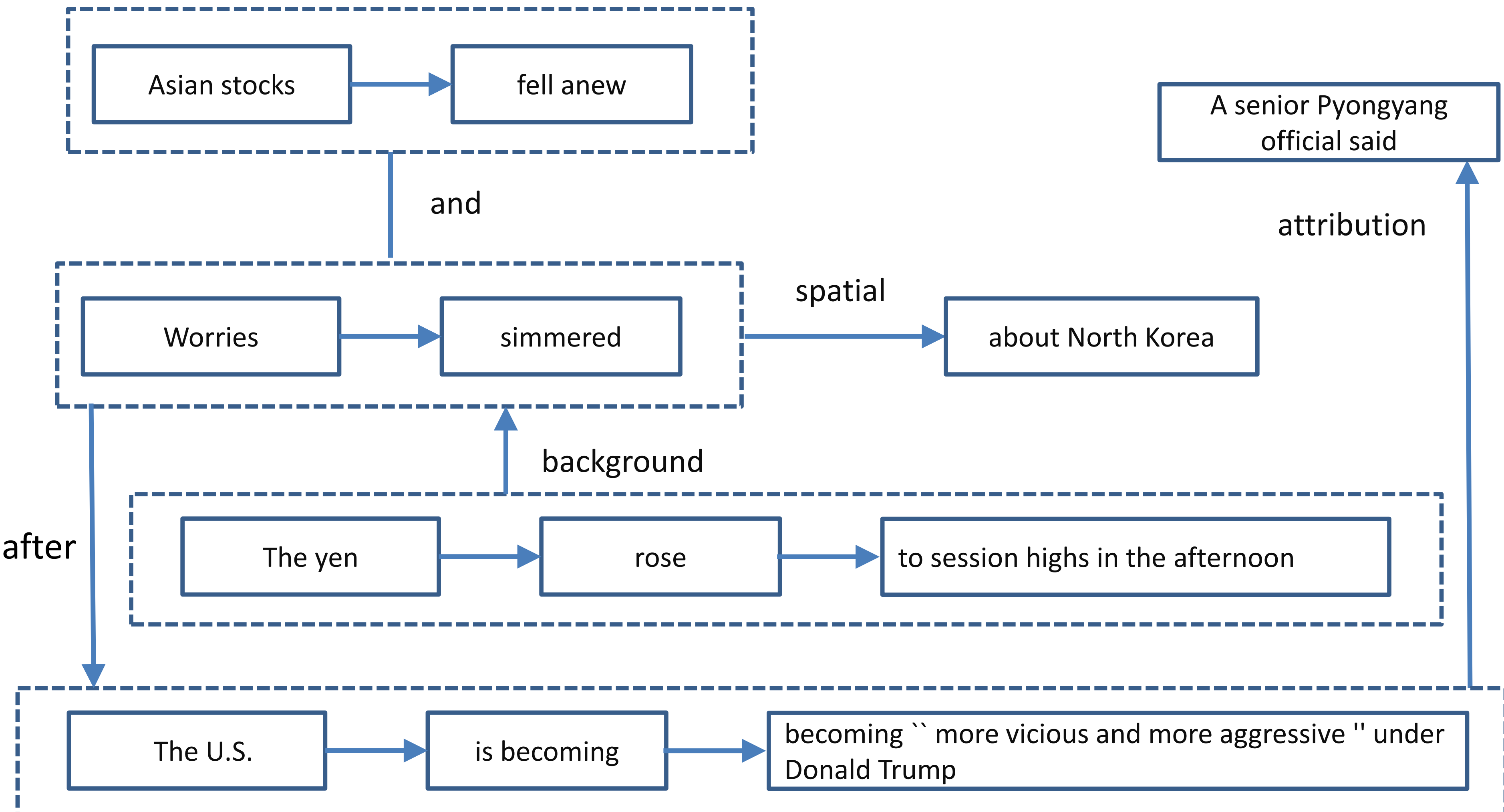
- has_child(Barack_Obama, Malia)
- has_child(Barack_Obama, Natasha)
- …

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

$$
\begin{array}{cccc}
\text{show} \quad \text{me} & \text{flights} & \text{to} & \text{Boston} \\
\hline
S/N & N & PP/NP & NP \\
\lambda f.f & \lambda x.flight(x) & \lambda y.\lambda x.to(x,y) & BOSTON \\
\end{array}
$$

$$
\frac{PP/NP \quad NP}{PP} > \qquad \lambda x.to(x, BOSTON)
$$

$$
N\backslash N \qquad \lambda f.\lambda x.f(x) \wedge to(x, BOSTON)
$$

$$
\frac{N \quad N\backslash N}{N} < \qquad \lambda x.flight(x) \wedge to(x, BOSTON)
$$

$$
\frac{S/N \quad N}{S} > \qquad \lambda x.flight(x) \wedge to(x, BOSTON)
$$

Artzi

Symbol

Tree

Tree is the name of a set

Tree is a noun

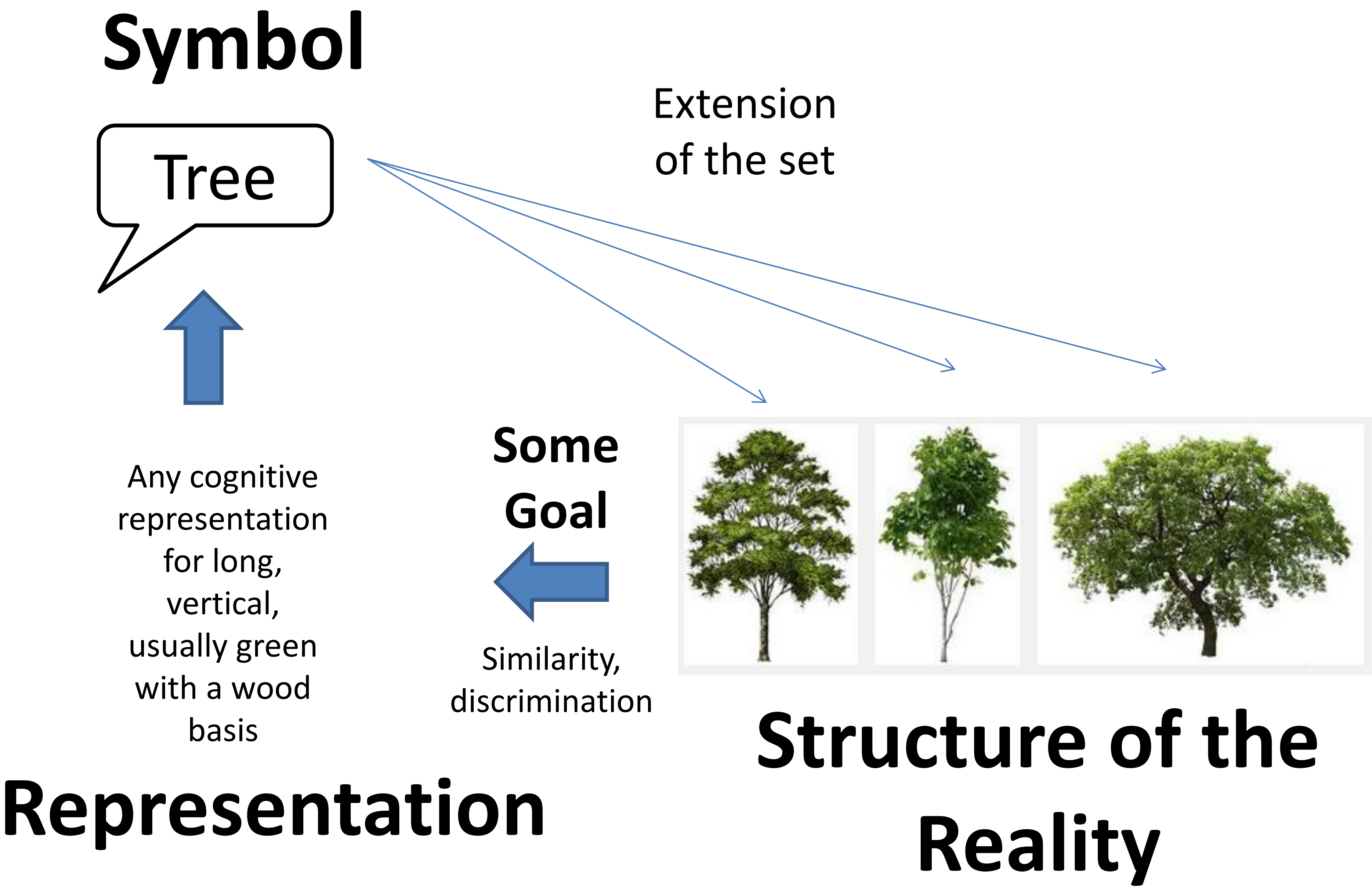
Any cognitive representation for long, vertical, usually green with a wood basis

Some Goal

Similarity, discrimination

Representation

Structure of the Reality


Symbol
Tree
Extension
of the set
Any cognitive representation for long, vertical, usually green with a wood basis
Some Goal
Similarity, discrimination
Representation
Structure of the Reality

Operating on the Representation

Symbol

Tree

The tallest tree

Some Goal

Similarity, discrimination

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

Representation

Structure of the Reality

Operating on the Representation

Definite article: “get me one”

Symbol

Tree

The tallest tree

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

Some Goal

Similarity, discrimination

Representation

Structure of the Reality

Operating on the Representation

Superlative adjective : “top most”

Symbol

Tree

The **tallest** tree

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

Some Goal

Similarity, discrimination

Structure of the Reality

Representation

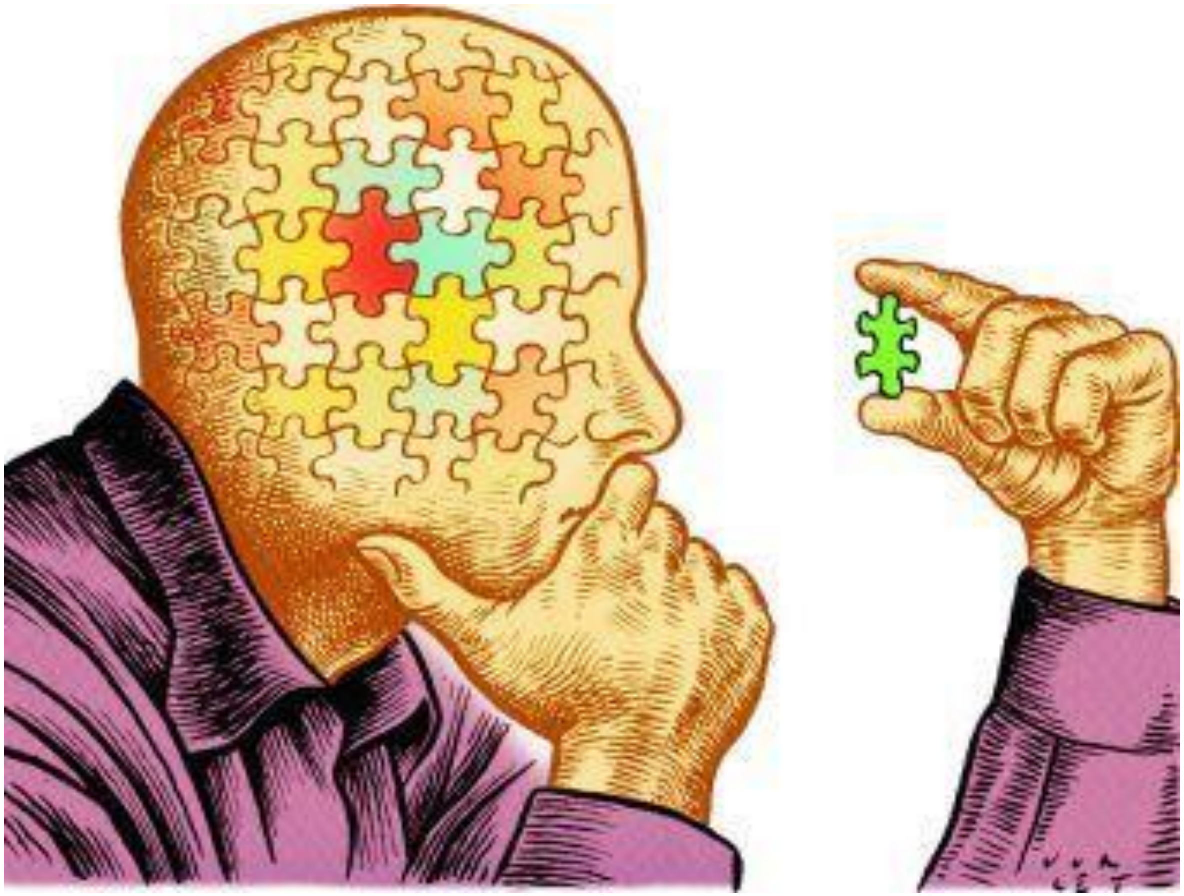
Natural Language ⟺ Representation

- 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

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

Chu

Inductive Logic Programming

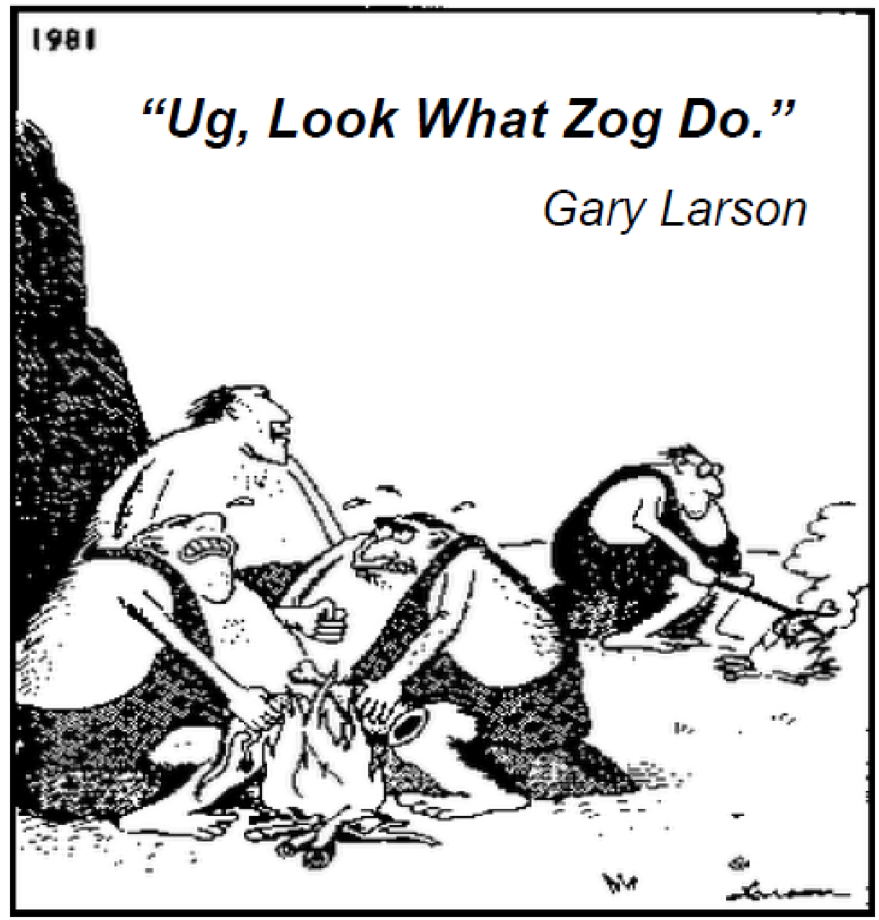
- By creating a new predicate, the definition of Grandparent can be reduced

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

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

Explanation-based Learning

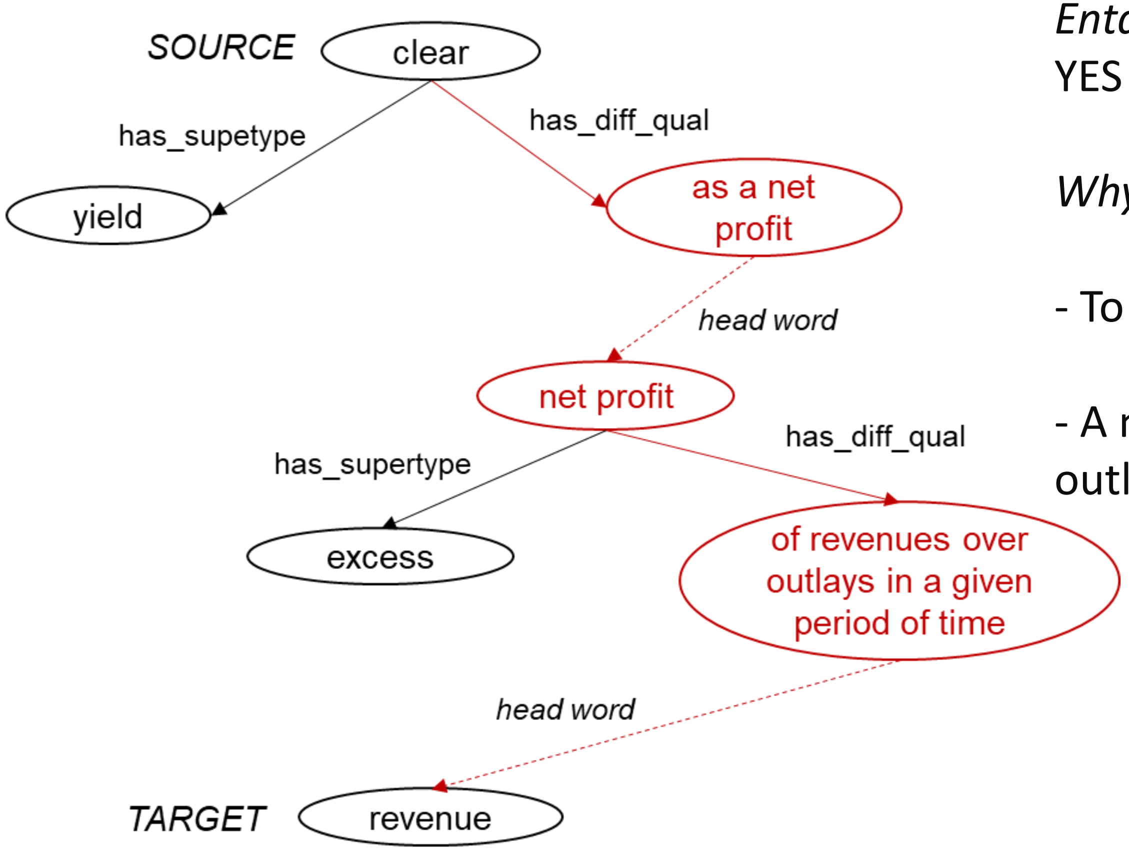
- **Explanation-based Learning (EBL)**
  - Method for extracting rules from individual observations through an explanation.
- **Explanation**
  - Stick holds the food over the fire while keeping hands safe.
- **Generalization**
  - Any long, rigid, sharp object can be used to toast food over the fire.
  - 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

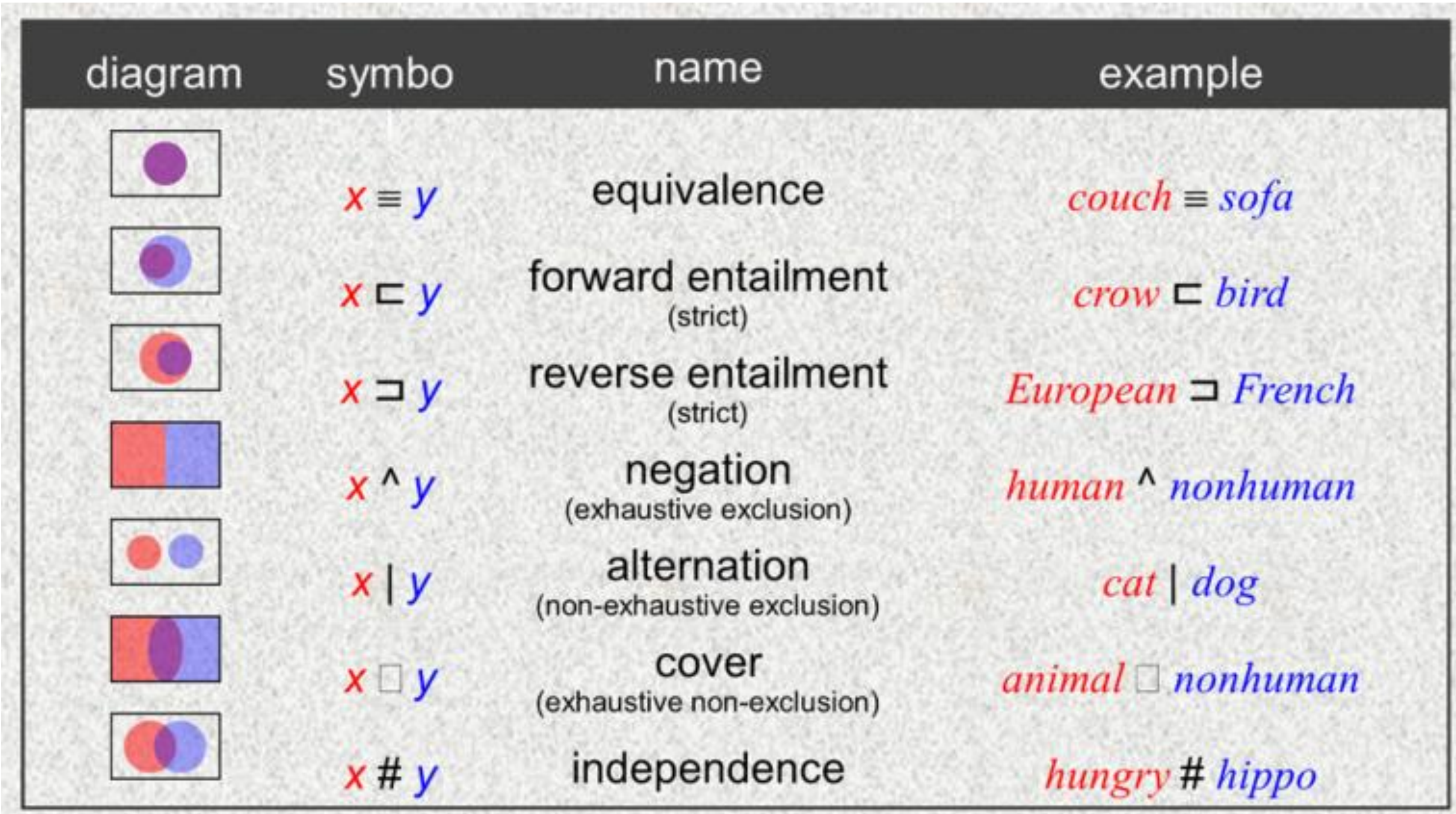
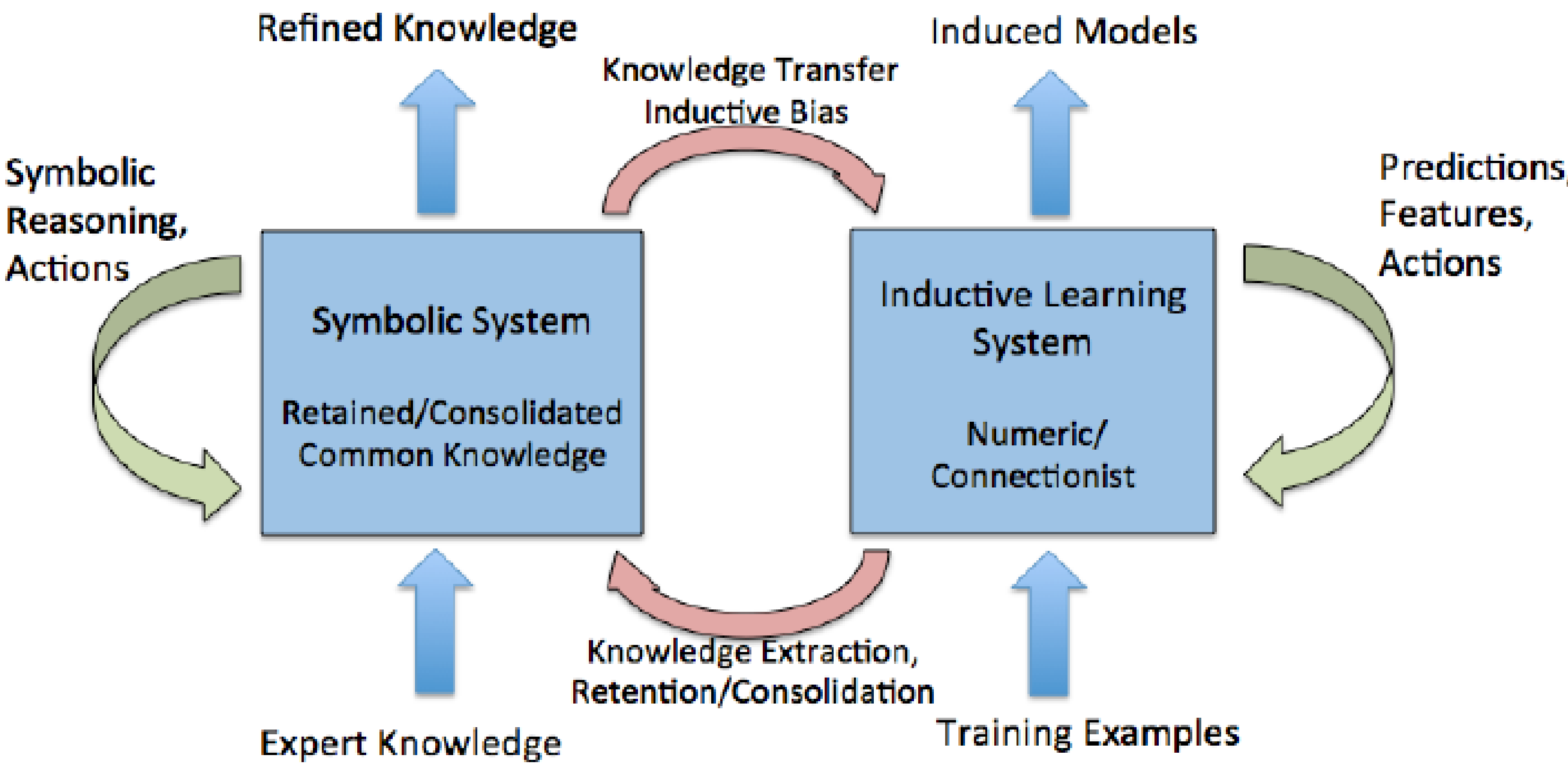


diagram	symbo	name	example
	$x \equiv y$	equivalence	*couch* ≡ *sofa*
	$x \sqsubset y$	forward entailment (strict)	*crow* ⊏ *bird*
	$x \sqsupset y$	reverse entailment (strict)	*European* ⊐ *French*
	$x \wedge y$	negation (exhaustive exclusion)	*human* ^ *nonhuman*
	$x \mid y$	alternation (non-exhaustive exclusion)	*cat* | *dog*
	x □ y	cover (exhaustive non-exclusion)	*animal* □ *nonhuman*
	$x \# y$	independence	*hungry* # *hippo*

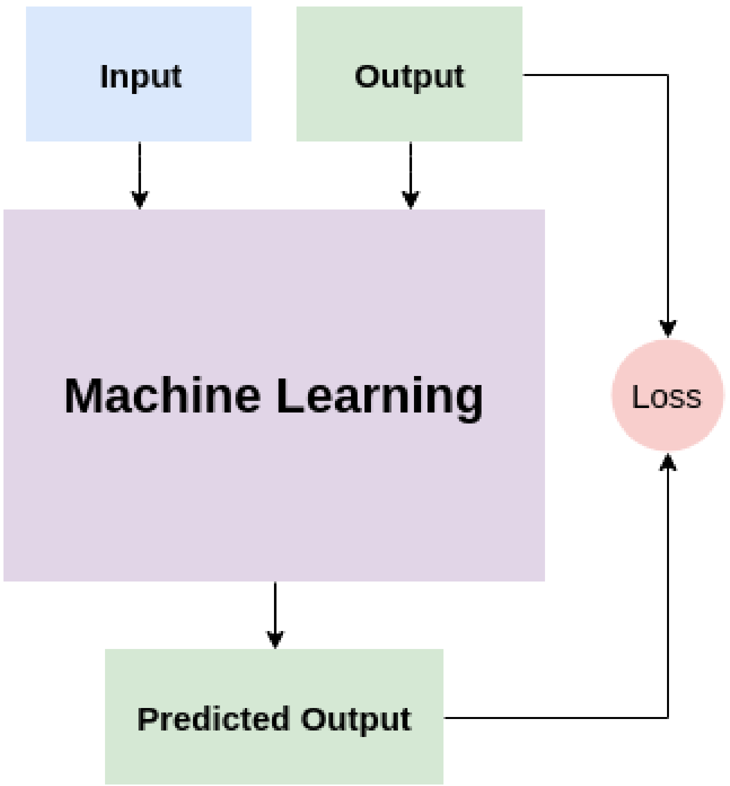
MacCartney

Neuro-Symbolic Models



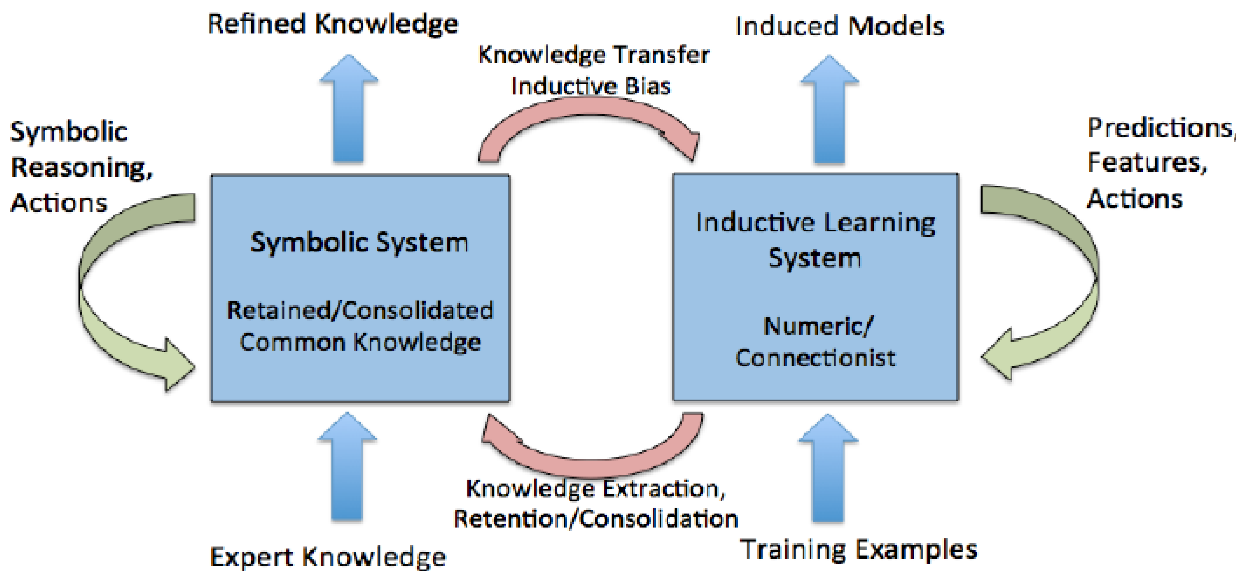
Current Limits of Deep Learning

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



Marcus, Gary. "Deep learning: A critical appraisal." *arXiv preprint arXiv:1801.00631* (2018).

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

Evans, Richard, and Edward Grefenstette. "Learning explanatory rules from noisy data." *Journal of Artificial Intelligence Research*61 (2018): 1-64.

Gated Graph Neural Networks

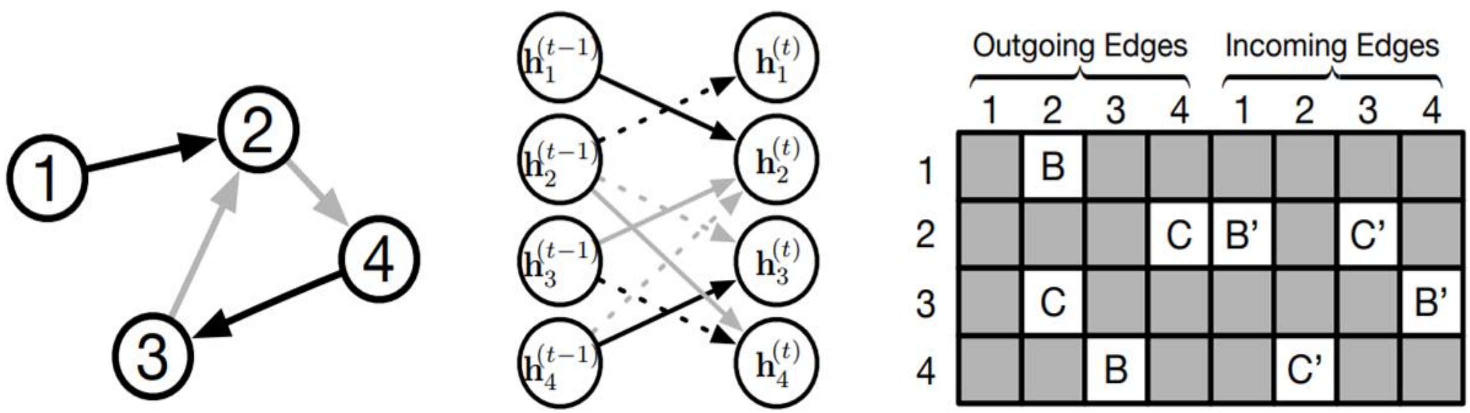
GGNNs are a neural network architecture defined according to a graph structure G = (V, E)

Nodes v ∈ V take unique values from 1, . . . , |V|, and edges are pairs e = (v, v0) ∈ V × 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 ∈ 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.

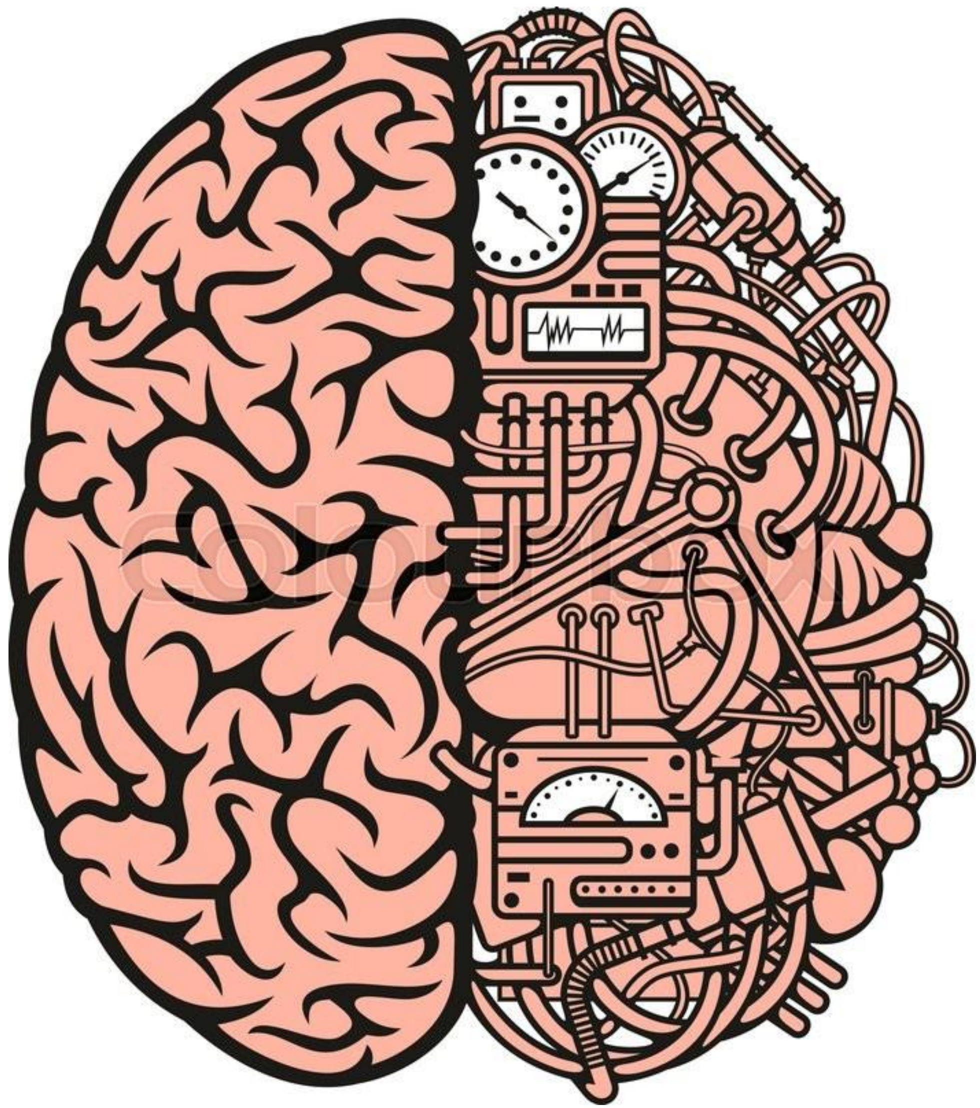
The structure of the graph, encoded in a matrix A serves to retrieve the hidden states of adjacent nodes based on the edge types between them. The hidden states are then updated by a a gated update module.



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

Li, Yujia, et al. "Gated graph sequence neural networks." *arXiv preprint arXiv:1511.05493* (2015).

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.

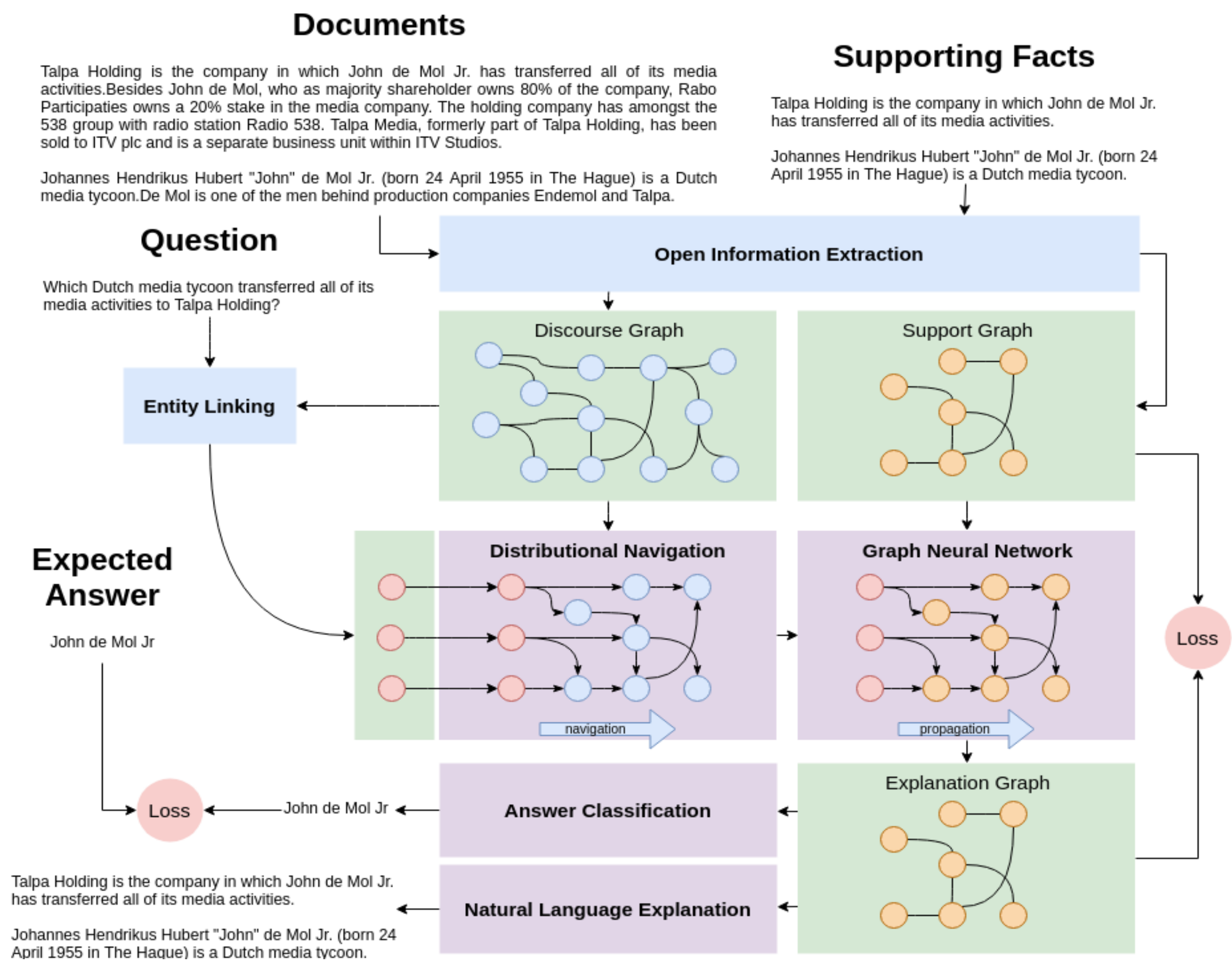
Paragraph B: Toxics Release Inventory

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.

Yang, Zhilin, et al. "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering." *arXiv preprint arXiv:1809.09600*(2018).

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. <u>Which action most likely made the small pieces of rock?</u>

[0]: sand blowing into cracks

[1]: leaves pressing down tightly

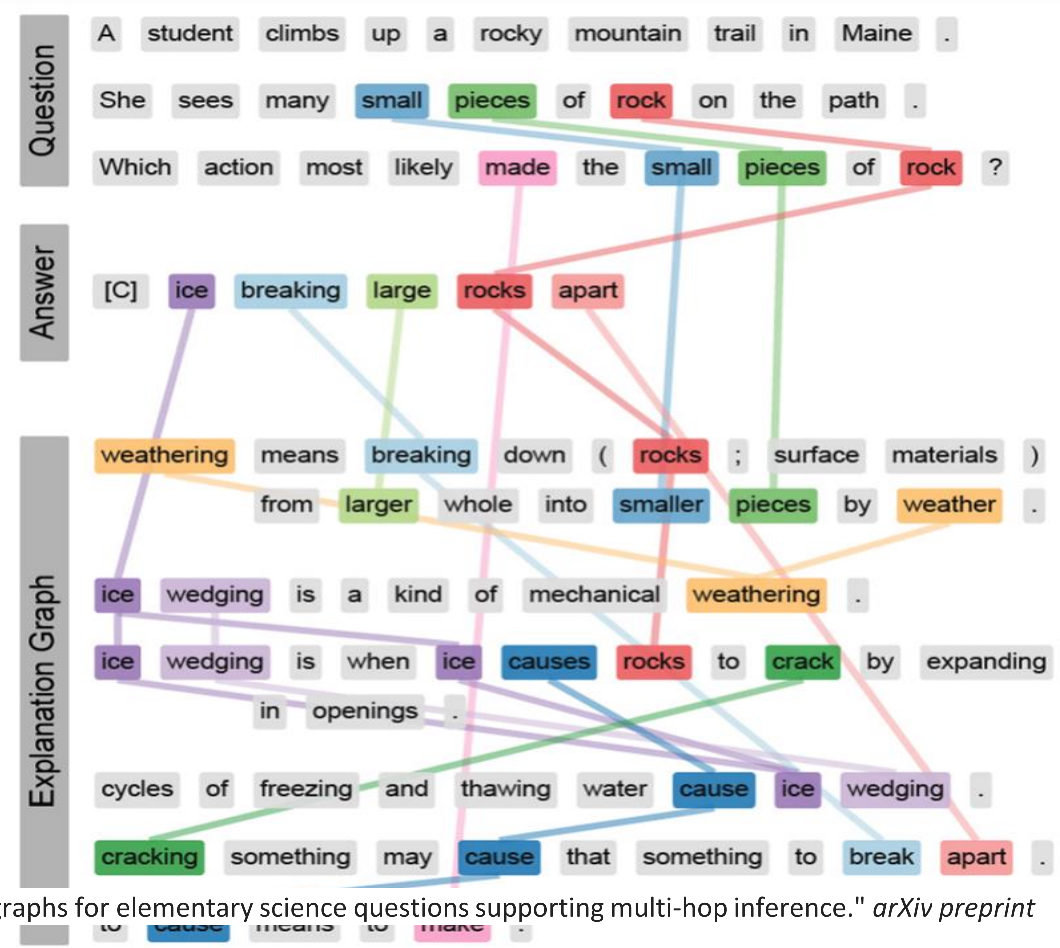
[2]: ice breaking large rocks apart

[3]: shells and bones sticking together

A: ice breaking large rocks apart

Explanation:

- weathering means breaking down (rocks weather
- ice wedging is a kind of mechanical weat
- ice wedging is when ice causes rocks to c
- cycles of freezing and thawing water cau
- cracking something may cause that som
- to cause means to make



Jansen, Peter A., et al. "Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference." *arXiv preprint arXiv:1802.03052* (2018).

LINKQ

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

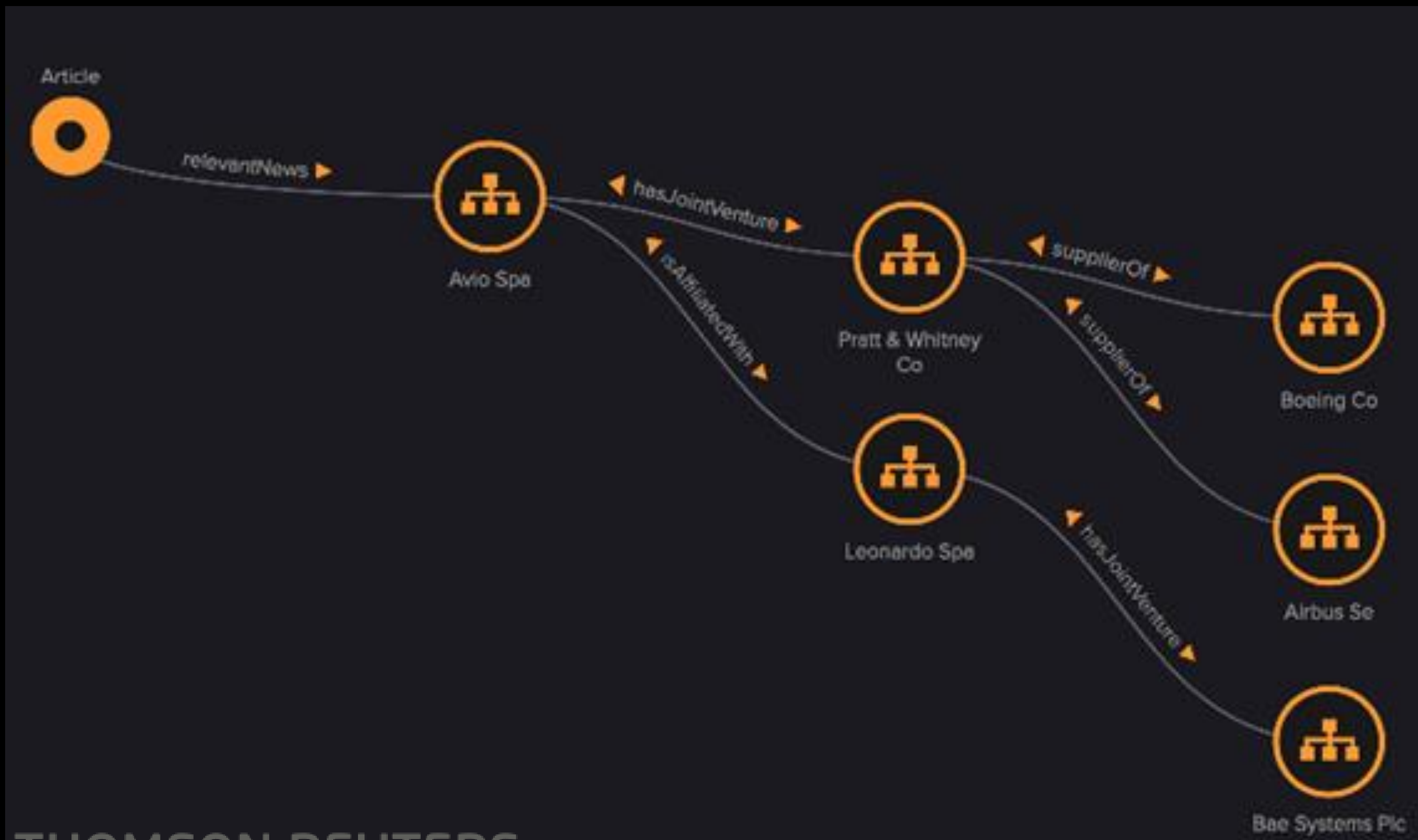
BRIEF-British Airline Pilots' Association - pilots union calls for investigation into collapse of Monarch Airlines<MONA.UL>

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

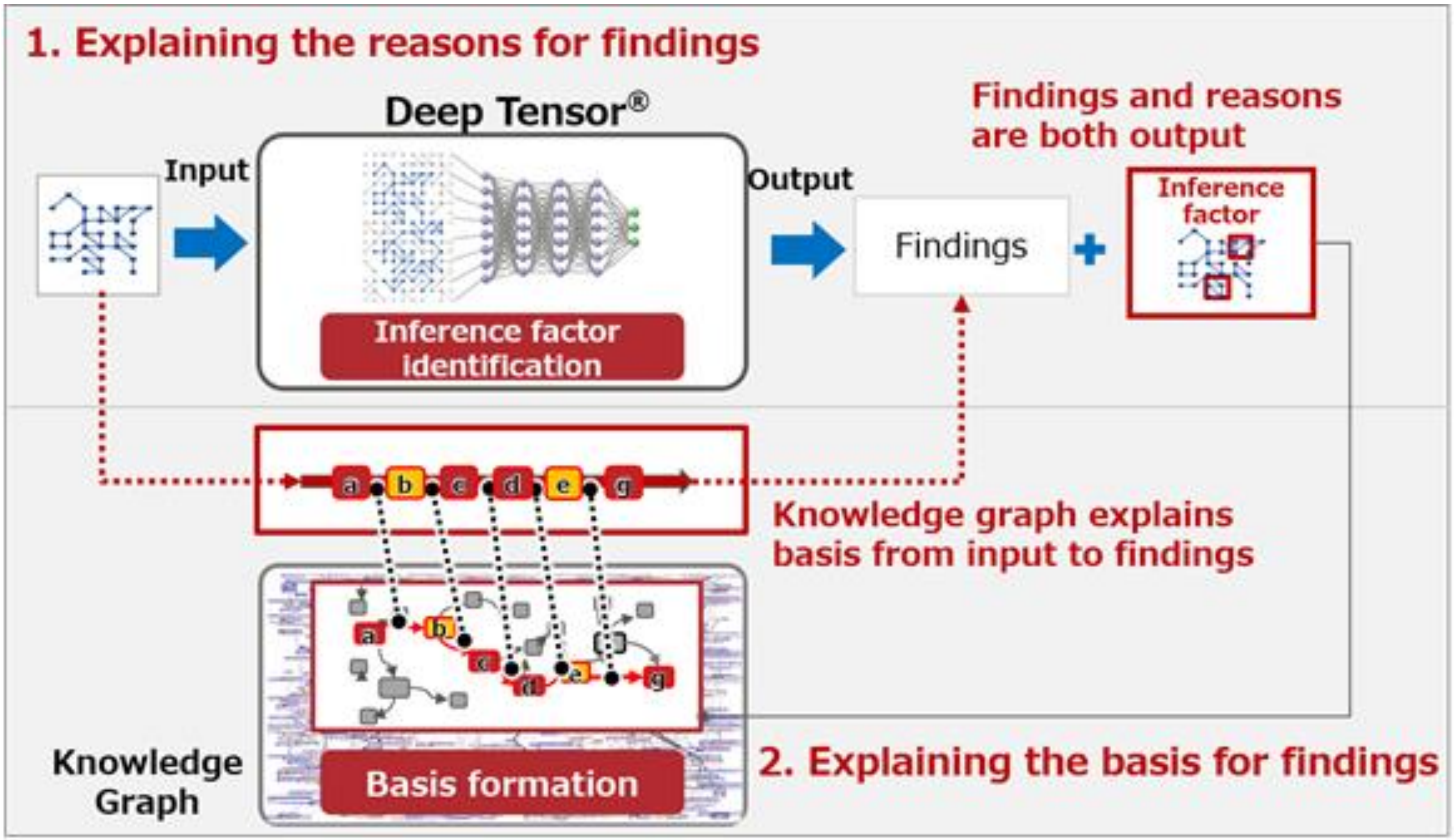
Raytheon Co
BAE Systems PLC
Boeing Co

BRIEF-Avio receives EUR 40 mln 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 ;))



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 Marcus[1]
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.

LINKQ

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

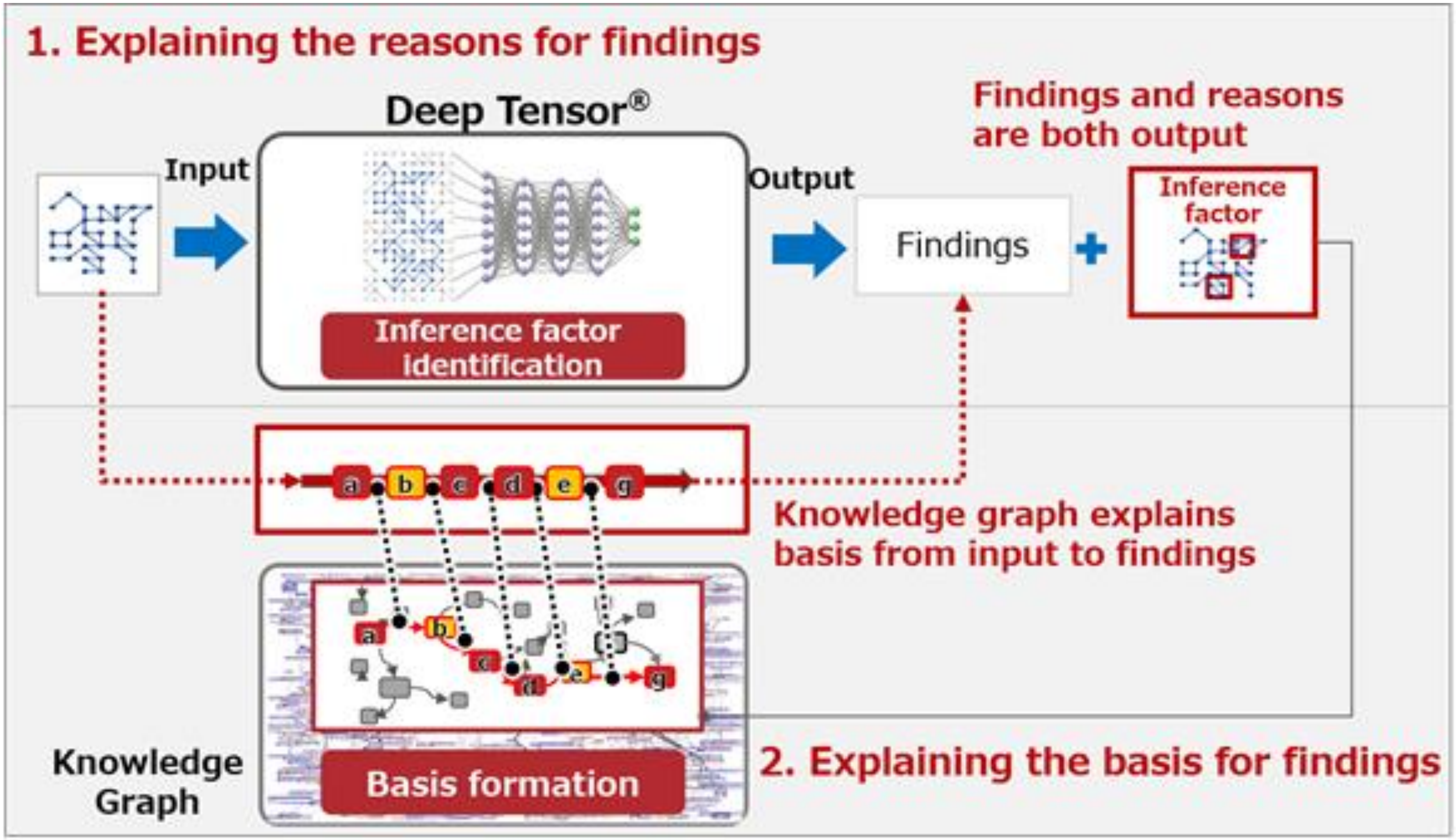
THOMSON REUTERS

BRIEF-Avio receives EUR 40 mln 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 ;))

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-capacitators?
  European politicians mentioned in Panama Leaks?

Representation

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

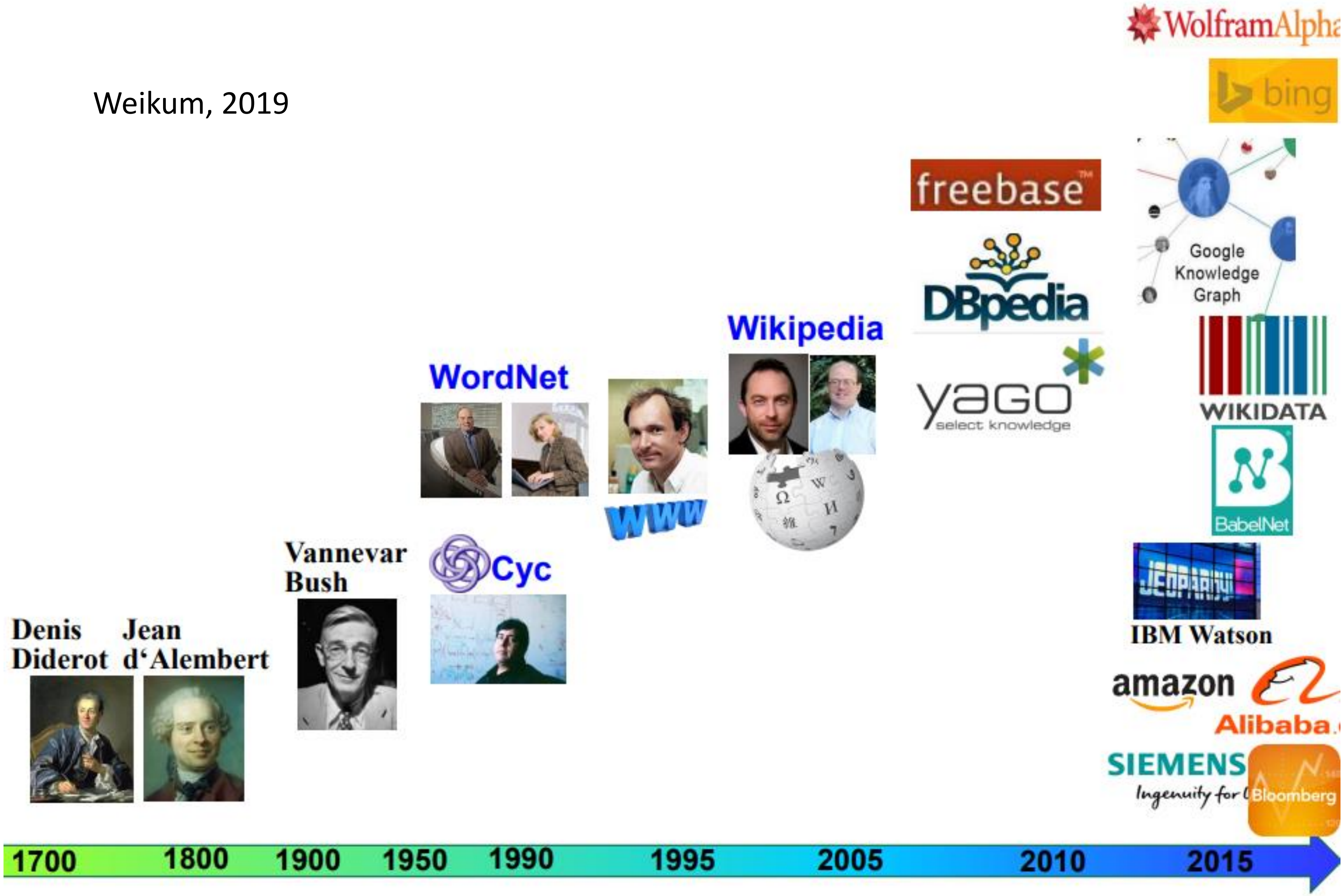
Good Knowledge Representation Languages

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

Knowledge Graphs

Brief History of Knowledge Bases

Weikum, 2019



Terminology

- <u>Ontology</u>
  - provides more complete definitions for concepts
  - Graphical conceptual model
- <u>Thesaurus</u>
  - simple relationships between words
    - synonyms, homonyms, antonyms, etc.
  - often combined with a taxonomy
- <u>Taxonomy</u>
  - hierarchical arrangement of concepts
  - often used as a “backbone” for an ontology
- <u>Lexicon</u>
  - 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 <u>polysemous</u> 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 <u>in some or all contexts</u>:
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H_20
- 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 <u>with respect to one feature of meaning</u>.
- 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 <u>IS-A hierarchy</u>:
  - A IS-A B (or A ISA B)
  - B **subsumes** A

Hyponyms and Instances

- WordNet has both <u>classes</u> and <u>instances</u>.
- An <u>instance</u> 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”

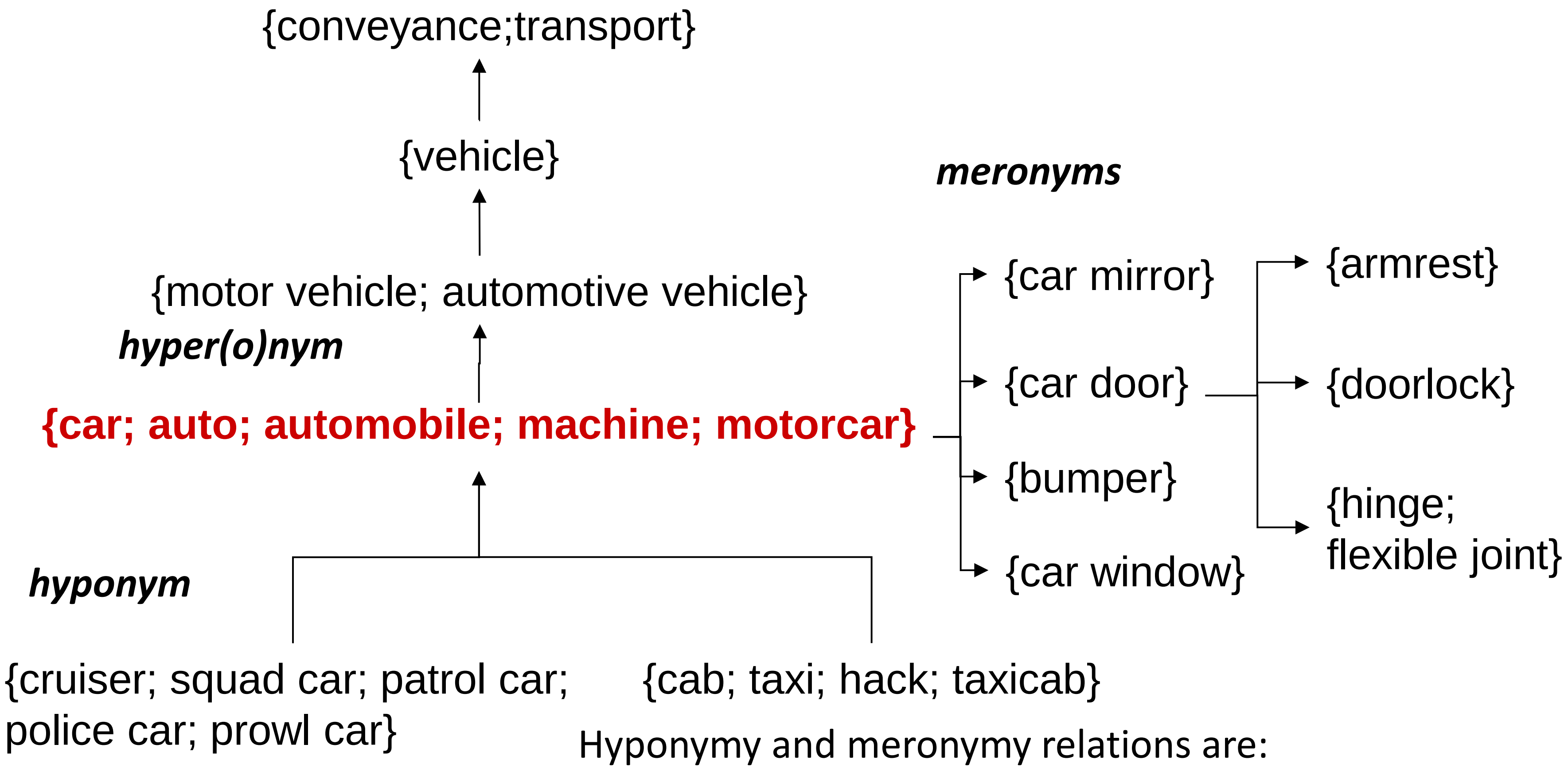
- This sense of “chump” is shared by 9 words:

  chump^1, fool^2, gull^1, mark^9, patsy^1, fall guy^1, sucker^1, soft touch^1, mug^2

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



Hyponymy and meronymy relations are:

- transitive
- directed

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 <u>substitutional variants</u>:
  - begin – commence
  - I once had a canary. The bird got sick. The poor animal died.

- <u>Hyponymy and meronymy chains</u> are important transitive relations for predicting properties and explaining textual properties:

  object -> artifact -> vehicle -> 4-wheeled vehicle -> car

- Strict separation of <u>part of speech (PoS)</u> 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	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	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	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-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

http://wenhoujx.blogspot.com/2013/06/penn-treebank-pos-tags.html

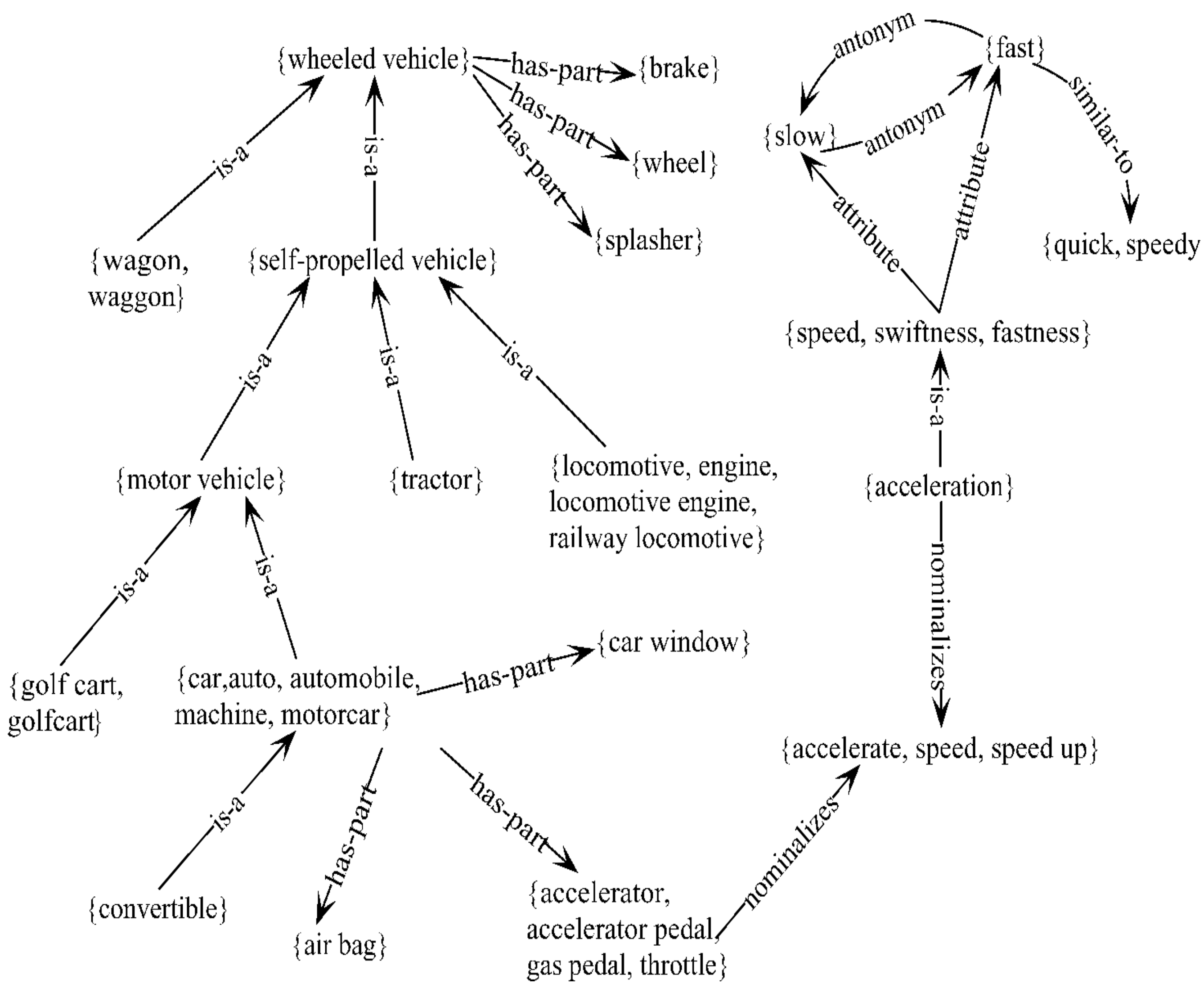
WordNet Noun Relations (Reference)

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

WordNet Verb Relations (Reference)

Relation	**Definition**	**Example**
Hypernym	From events to superordinate events	*fly*^{9} $\rightarrow$ *travel*^{5}
Troponym	From events to subordinate event (often via specific manner)	*walk*^{1} $\rightarrow$ *stroll*^{1}
Entails	From verbs (events) to the verbs (events) they entail	*snore*^{1} $\rightarrow$ *sleep*^{1}
Antonym	Semantic opposition between lemmas	*increase*^{1} $\Longleftrightarrow$ *decrease*^{1}
Derivationally Related Form	Lemmas with same morphological root	*destroy*^{1} $\Longleftrightarrow$ *destruction*^{1}

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>
```

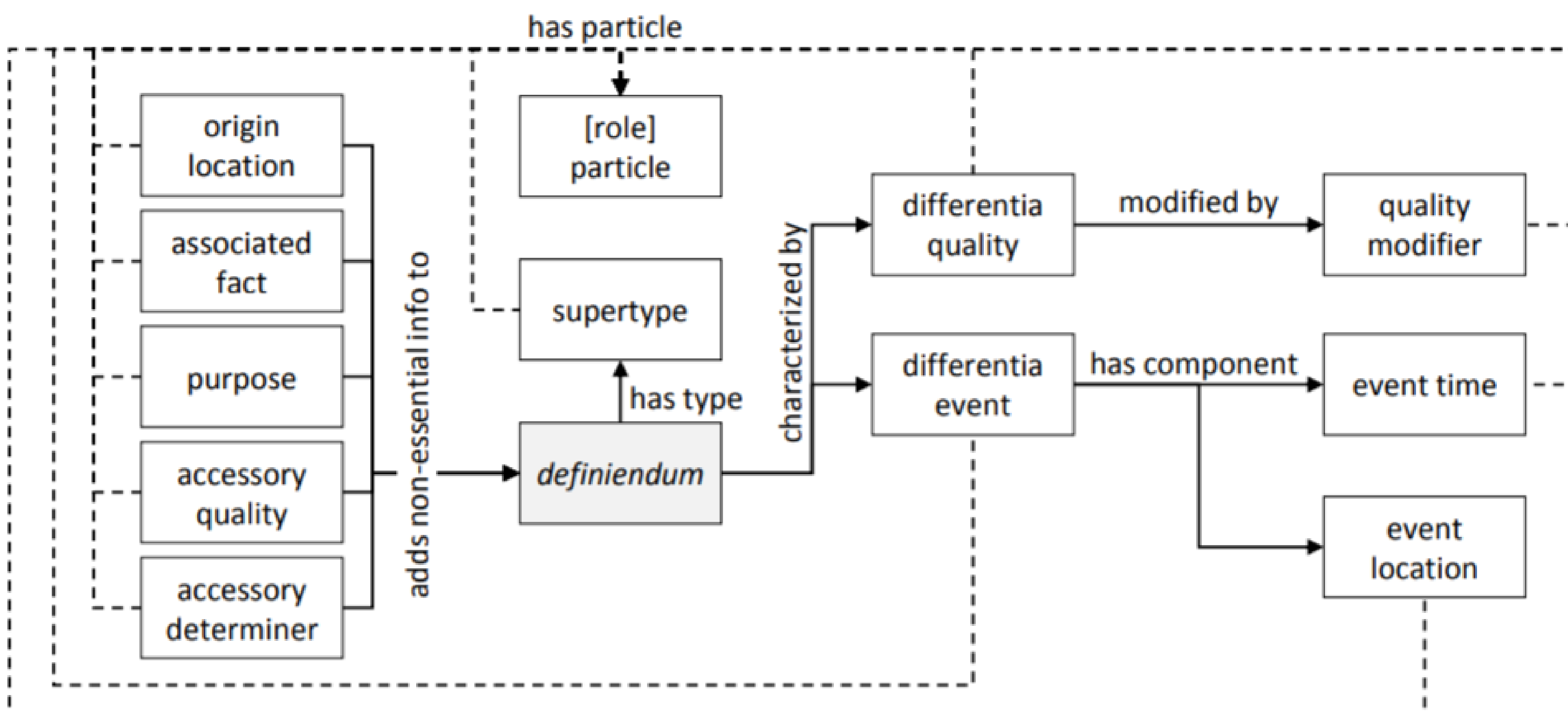
Accessing WordNet from Prolog

```
? - 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]
```

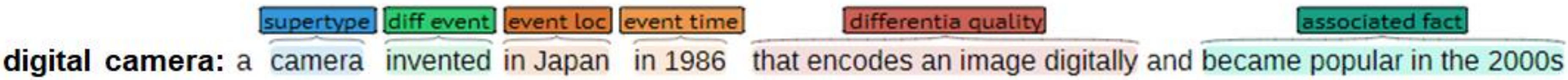
https://github.com/ekaf/wordnet-prolog

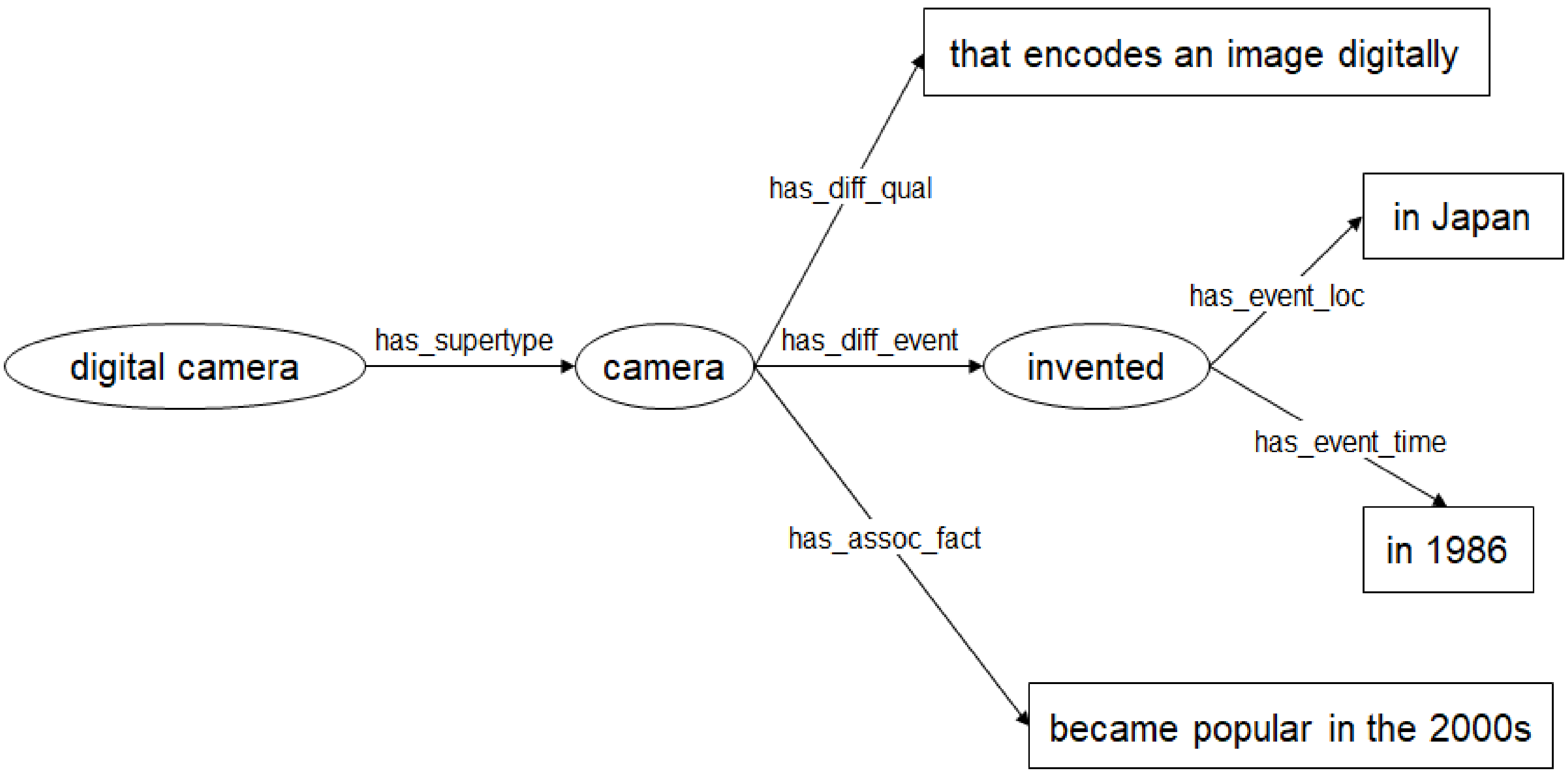
Semantic Roles for Lexical Definitions

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





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]
   - Inorganic Chemicals [D01] +
   - Organic Chemicals [D02] +
   - Heterocyclic Compounds [D03] +
   - Polycyclic Compounds [D04] +
   - Macromolecular Substances [D05] +
   - Hormones, Hormone Substitutes, and
   - Enzymes and Coenzymes [D08] +
   - Carbohydrates [D09] +
   - Lipids [D10] +
   - Amino Acids, Peptides, and Proteins [
   - Nucleic Acids, Nucleotides, and Nucle
   - Complex Mixtures [D20] +
   - Biological Factors [D23] +
   - Biomedical and Dental Materials [D25
   - Pharmaceutical Preparations [D26] +
   - 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]

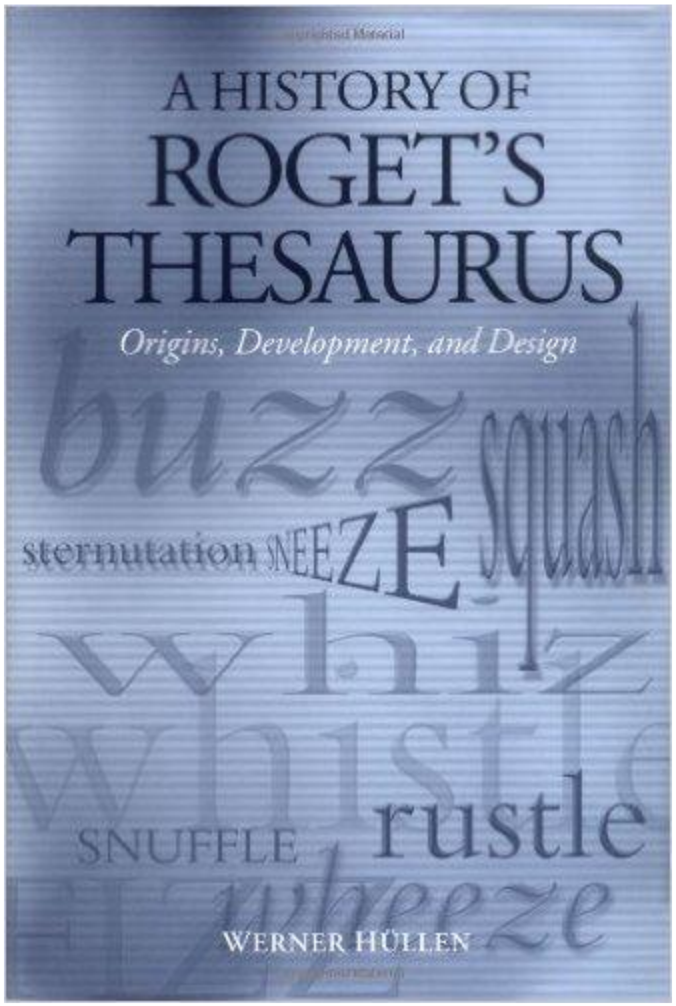
Hemocyanin [D12.776.124.337]

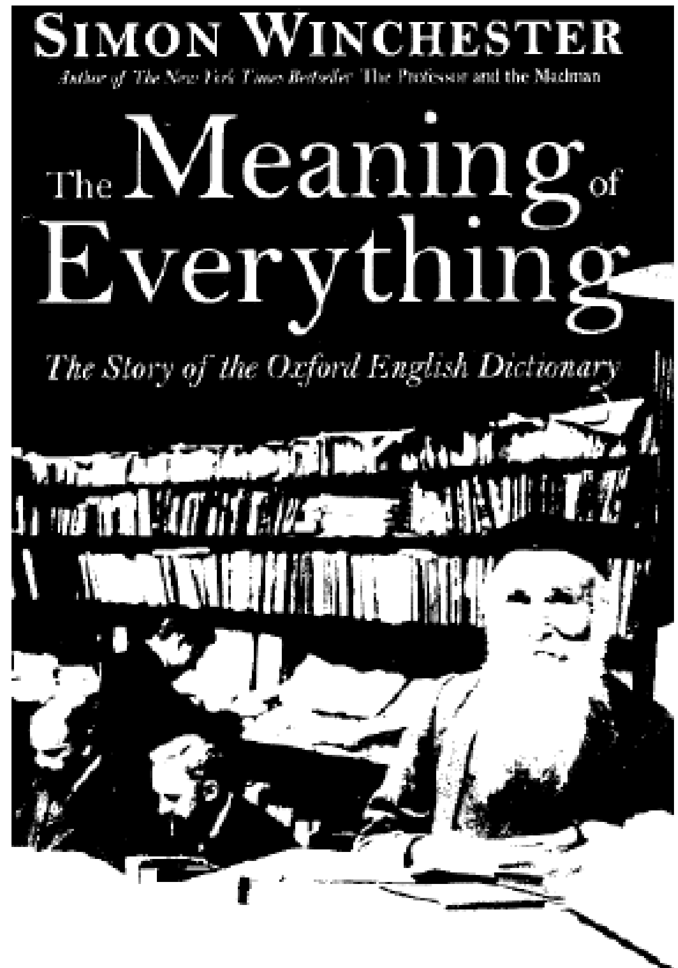
Hemoglobins [D12.776.124.400]

Carboxyhemoglobin [D12.776.124.400.141]

Erythrocruorins [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

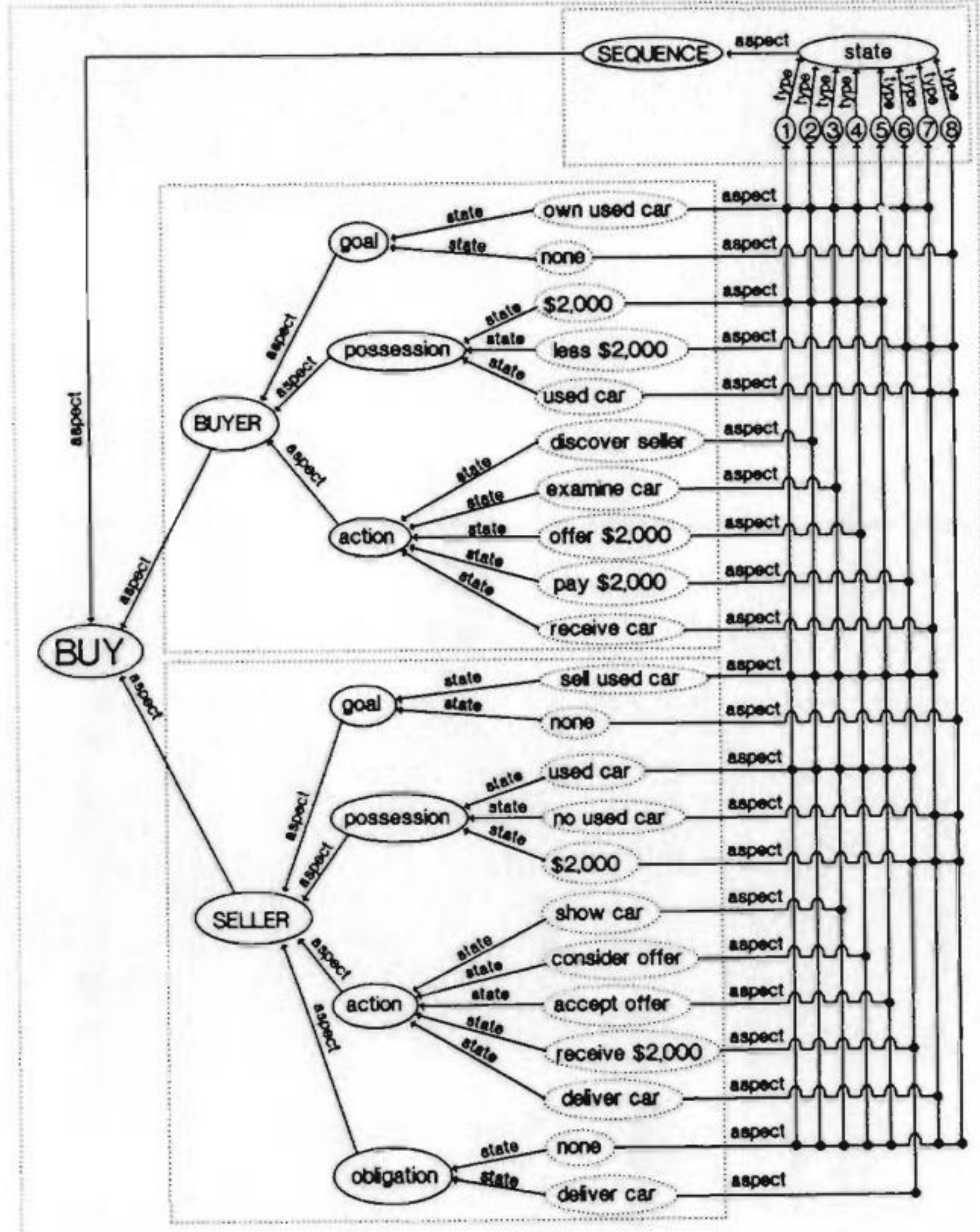
	category				
rank	BIRD	FRUIT	VEHICLE	FURNITURE	WEAPON
top eight					
1	robin	orange	automobile	chair	gun
2	sparrow	apple	station wagon	sofa	pistol
3	bluejay	banana	truck	couch	revolver
4	bluebird	peach	car	table	machine gun
5	canary	pear	bus	easy chair	rifle
6	blackbird	apricot	taxi	dresser	switchblade
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

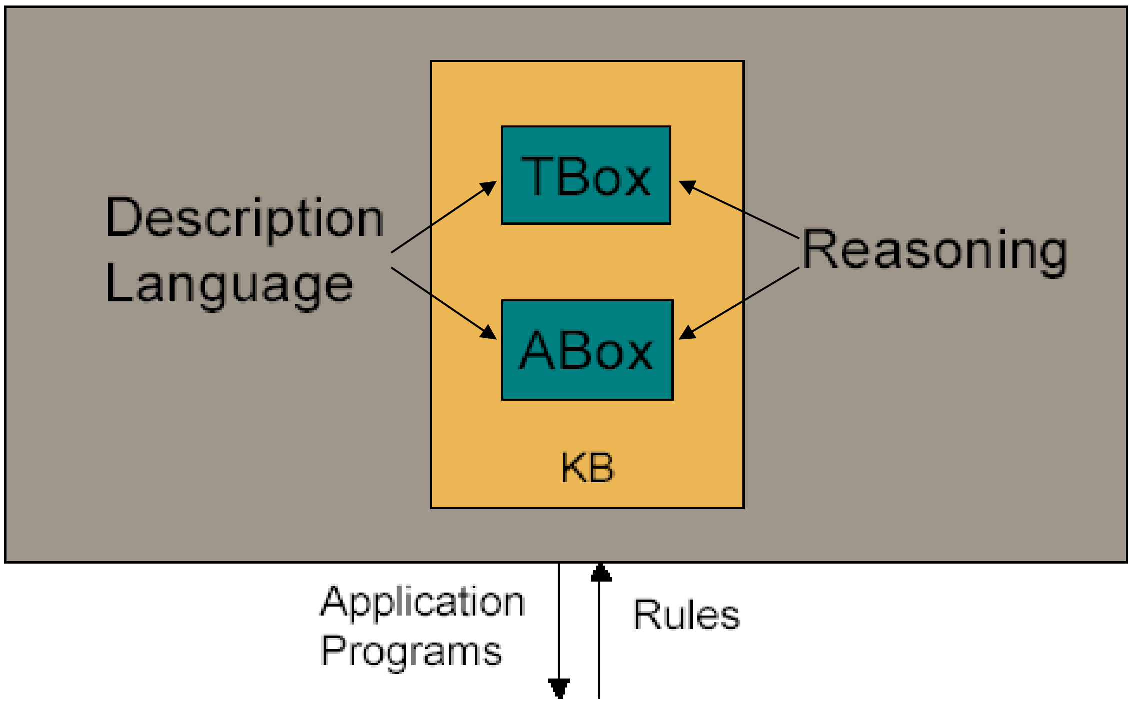
Horrocks

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).

Horrocks

Architecture of a DL System



Horrocks

Formulas

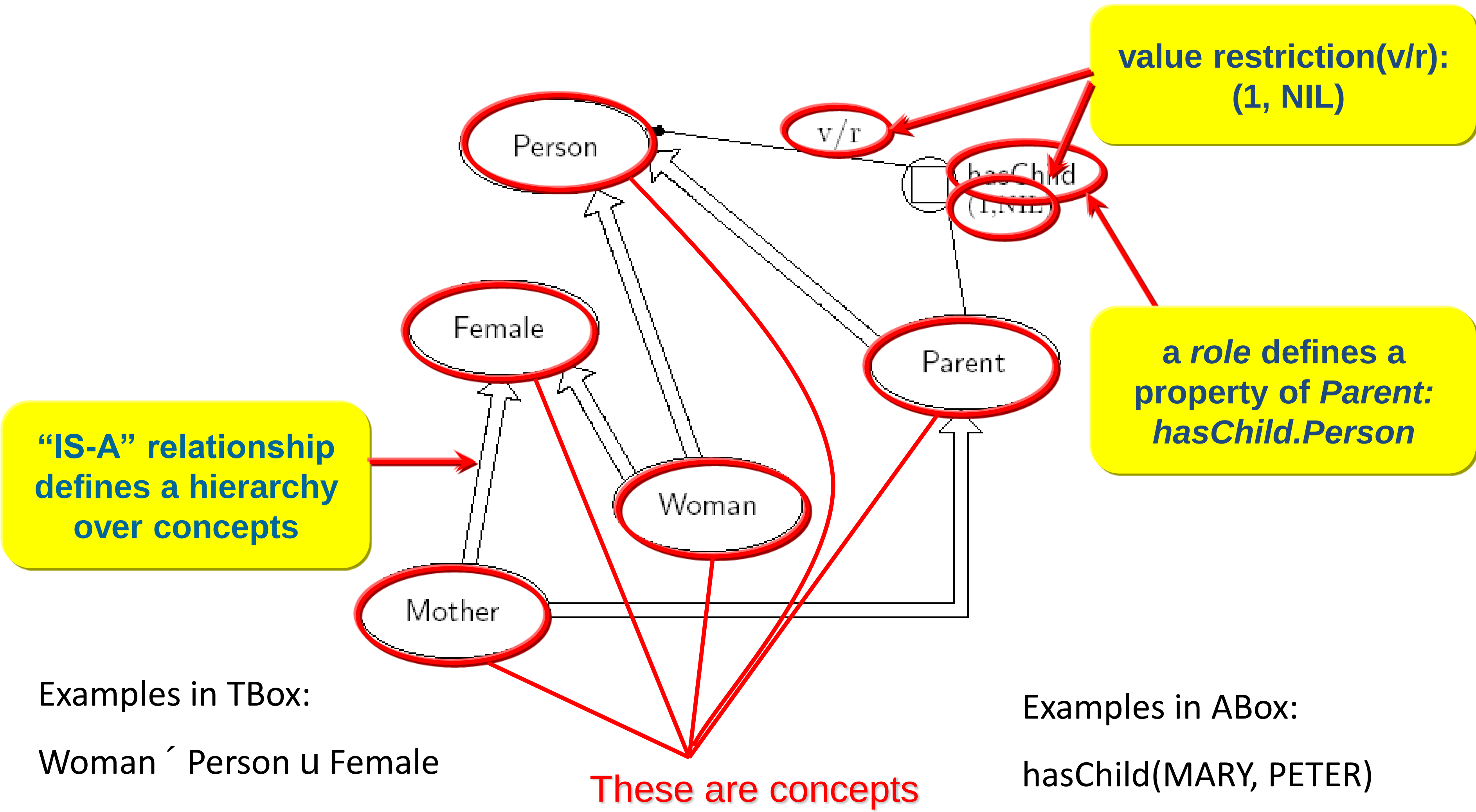
- Building blocks that allow complex descriptions of concepts and roles.
  - Example (we'll look at the syntax in more detail soon.)
    - A Woman is a Female Person
      - Woman = Person u Female
    - A Mother is a Woman and she has a child
      - Mother = Woman u hasChild.T

- The TBox can be used to assign names to complex descriptions.

We will use the terms description and concept interchangeably.

Horrocks

An Example about Family Relationships



Examples in TBox:

Woman ´ Person u Female

Mother ´ Woman u hasChild.Person

Examples in ABox:

hasChild(MARY, PETER)

Father(PETER)

Name Symbols vs. Base Symbols

- Atomic concepts occurring in a TBox **T** can be divided into two sets, name symbols N_T (or defined concepts) and base symbols B_T (or primitive concepts, occur only on the right-hand side)

- A base interpretation for **T** only interprets the base symbols.

Base symbols

Name Symbol

$$
\begin{array}{rcl}
\mathsf{Woman} & \equiv & \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} & \equiv & \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} & \equiv & \mathsf{Woman} \sqcap \exists \mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Father} & \equiv & \mathsf{Man} \sqcap \exists \mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Parent} & \equiv & \mathsf{Father} \sqcup \mathsf{Mother} \\
\mathsf{Grandmother} & \equiv & \mathsf{Mother} \sqcap \exists \mathsf{hasChild}.\mathsf{Parent} \\
\mathsf{MotherWithManyChildren} & \equiv & \mathsf{Mother} \sqcap \geqslant 3\, \mathsf{hasChild} \\
\mathsf{MotherWithoutDaughter} & \equiv & \mathsf{Mother} \sqcap \forall \mathsf{hasChild}.\neg\mathsf{Woman} \\
\mathsf{Wife} & \equiv & \mathsf{Woman} \sqcap \exists \mathsf{hasHusband}.\mathsf{Man}
\end{array}
$$

Horrocks

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)

Horrocks

OWL Class Constructor

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

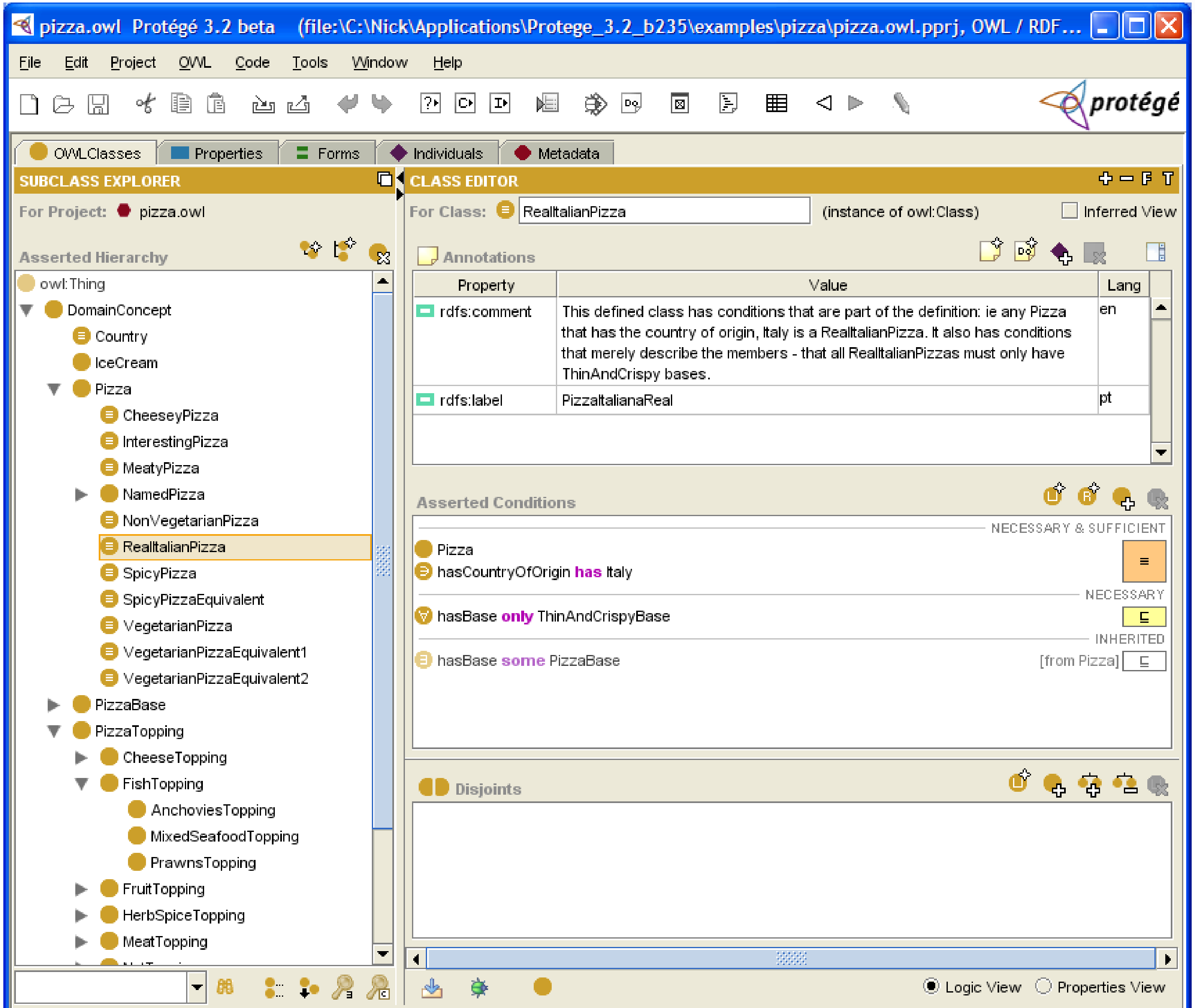
Horrocks

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	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant$1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant$1hasSSN$^-$

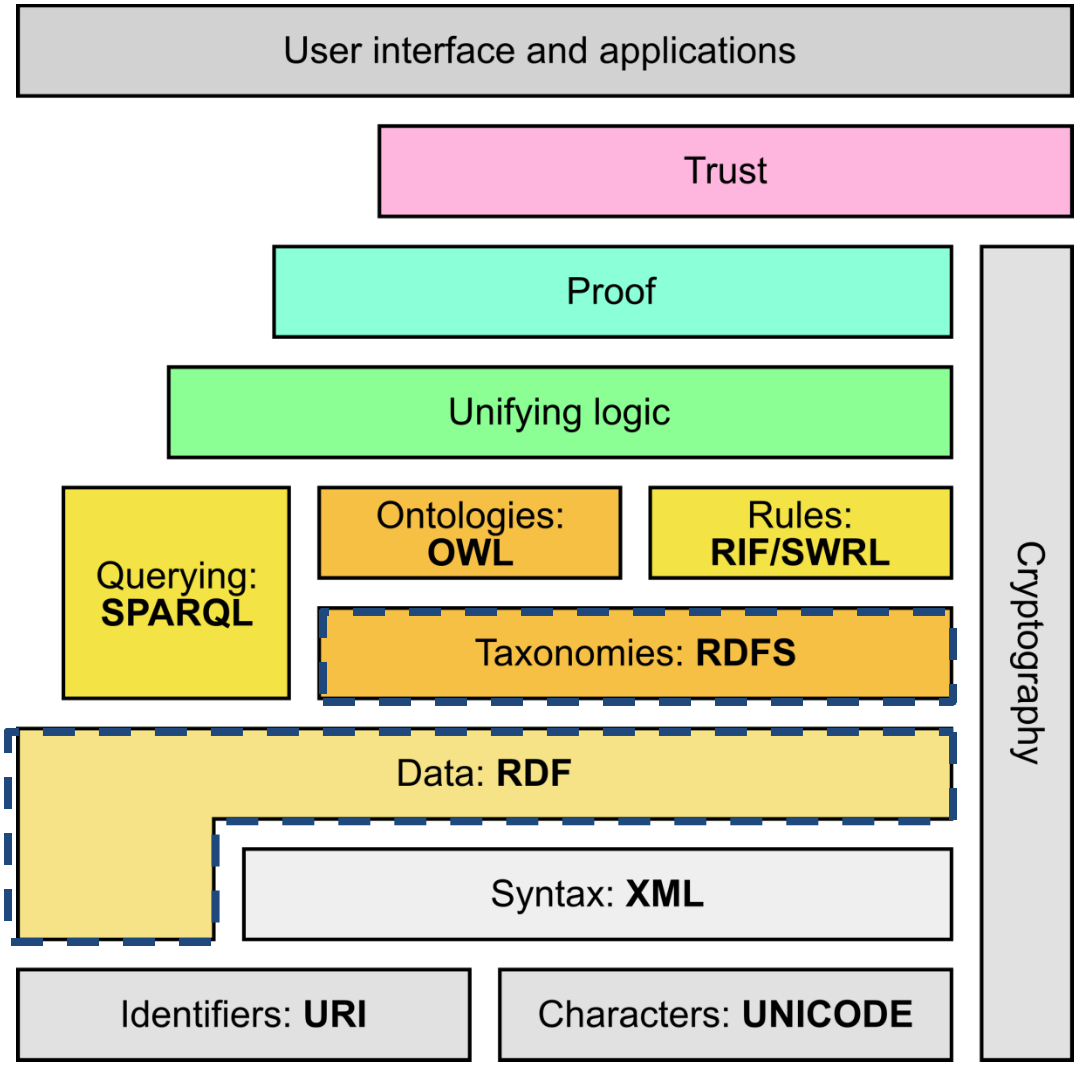
Horrocks

Ontology Editors



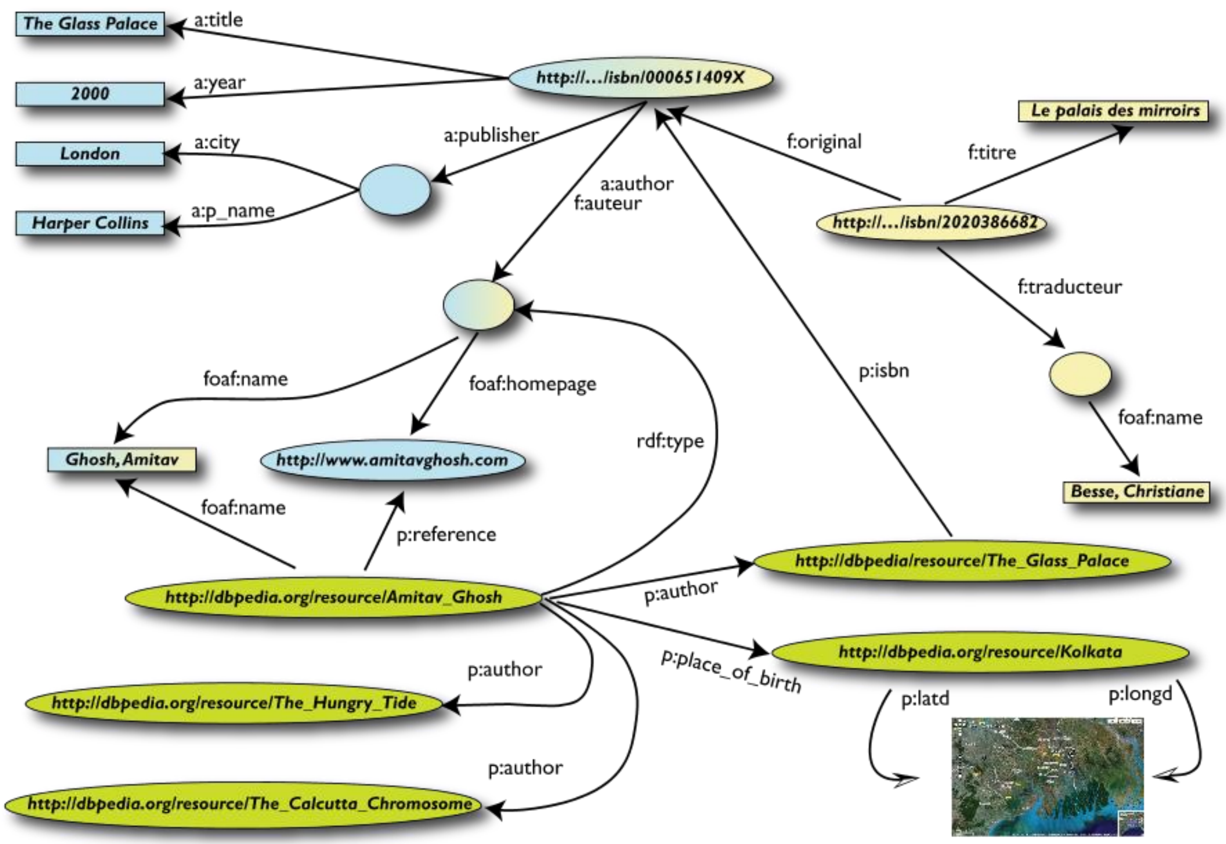
Semantic Web

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

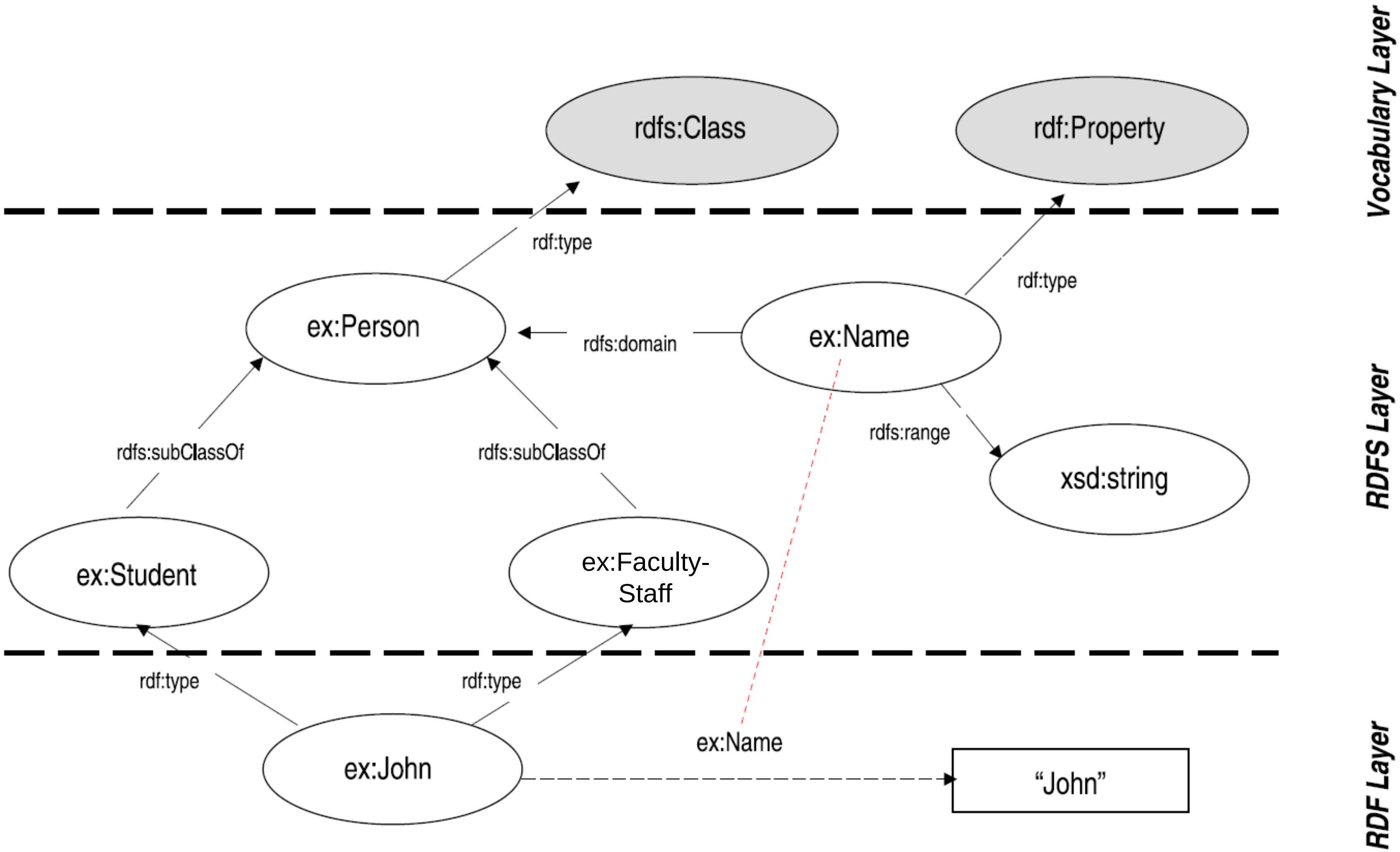
RDFS Classes

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

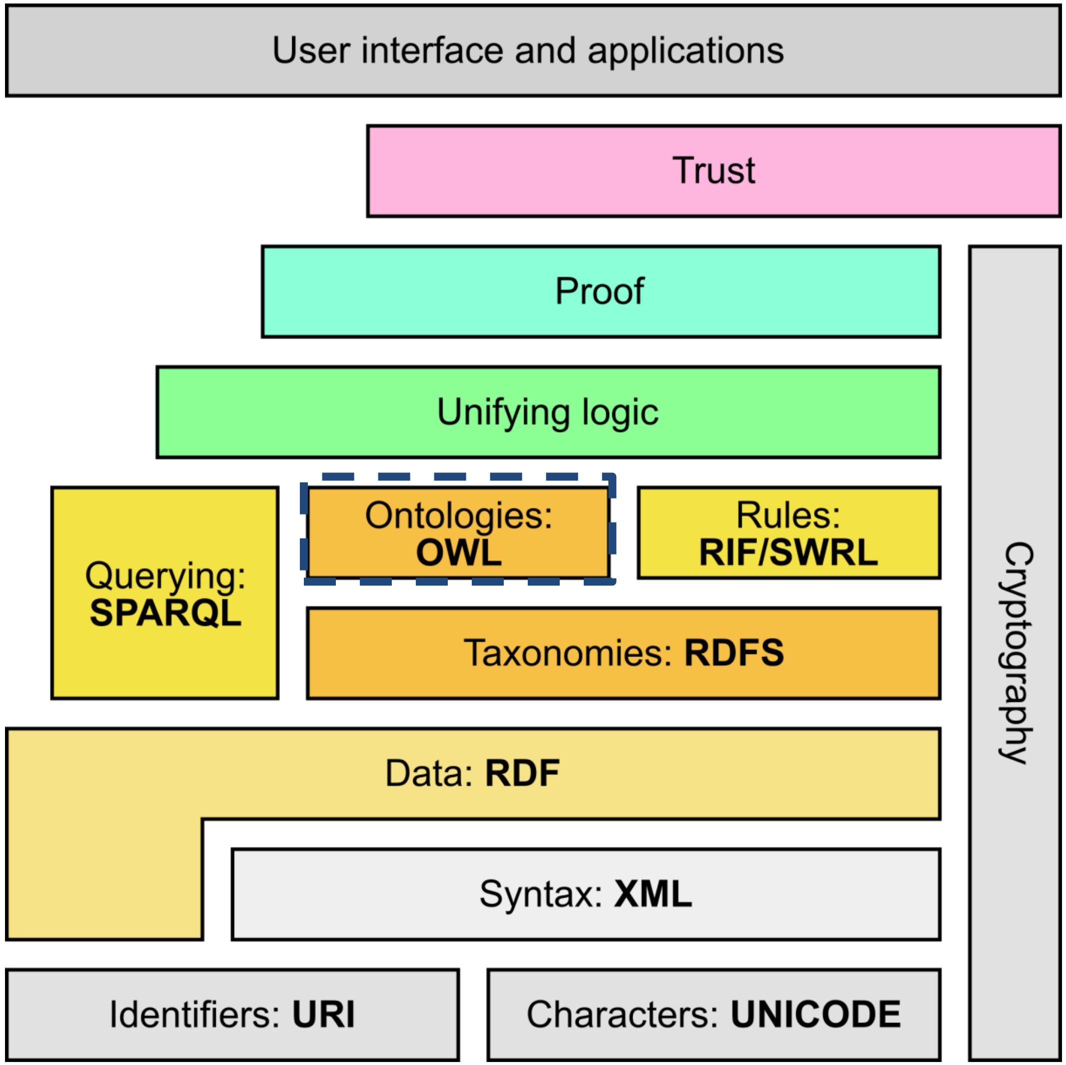
RDFS Properties

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

RDFS Example

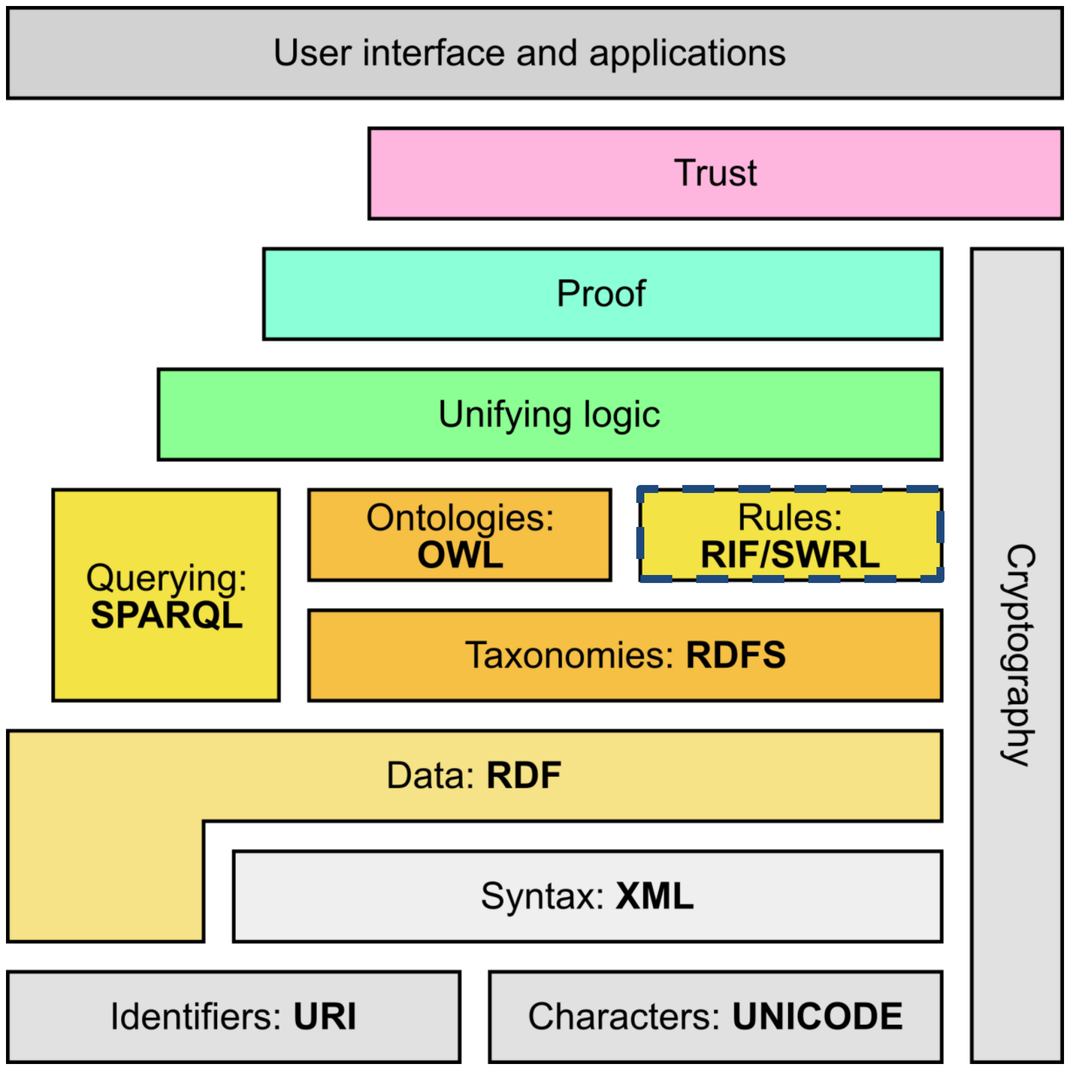


Semantic Web Stack



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

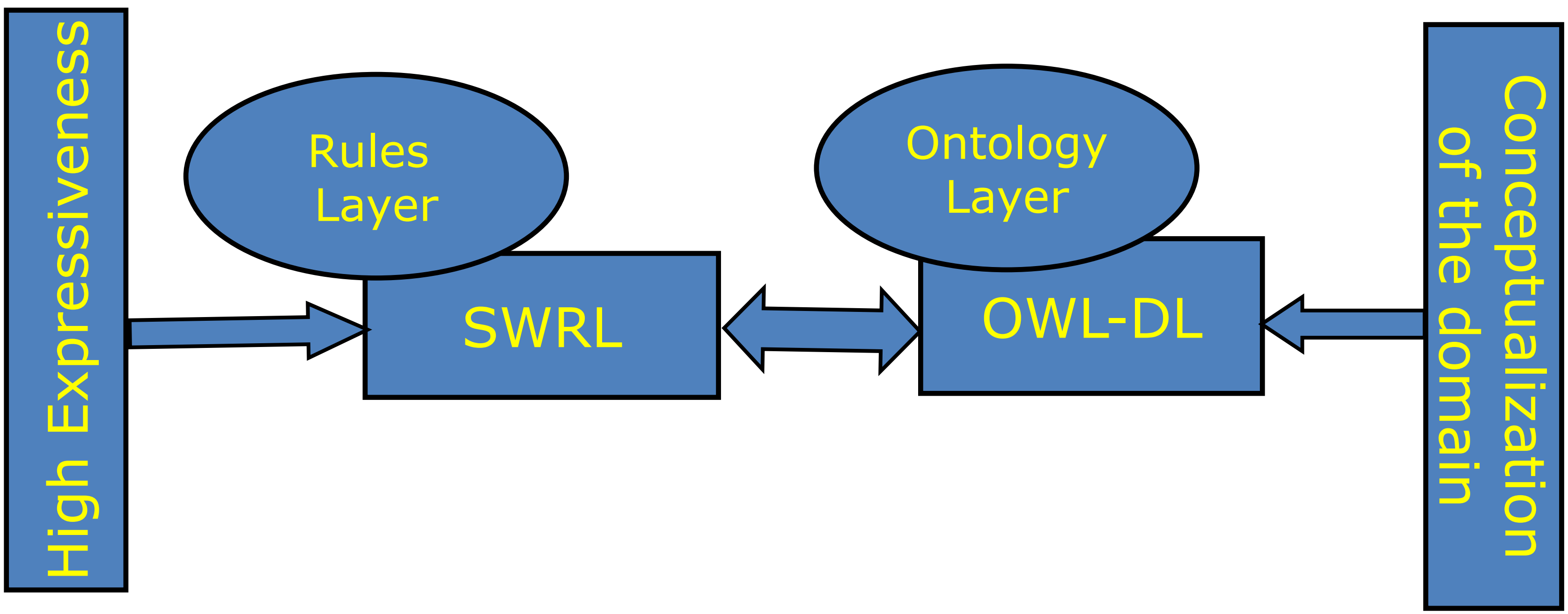
Semantic Web Stack



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

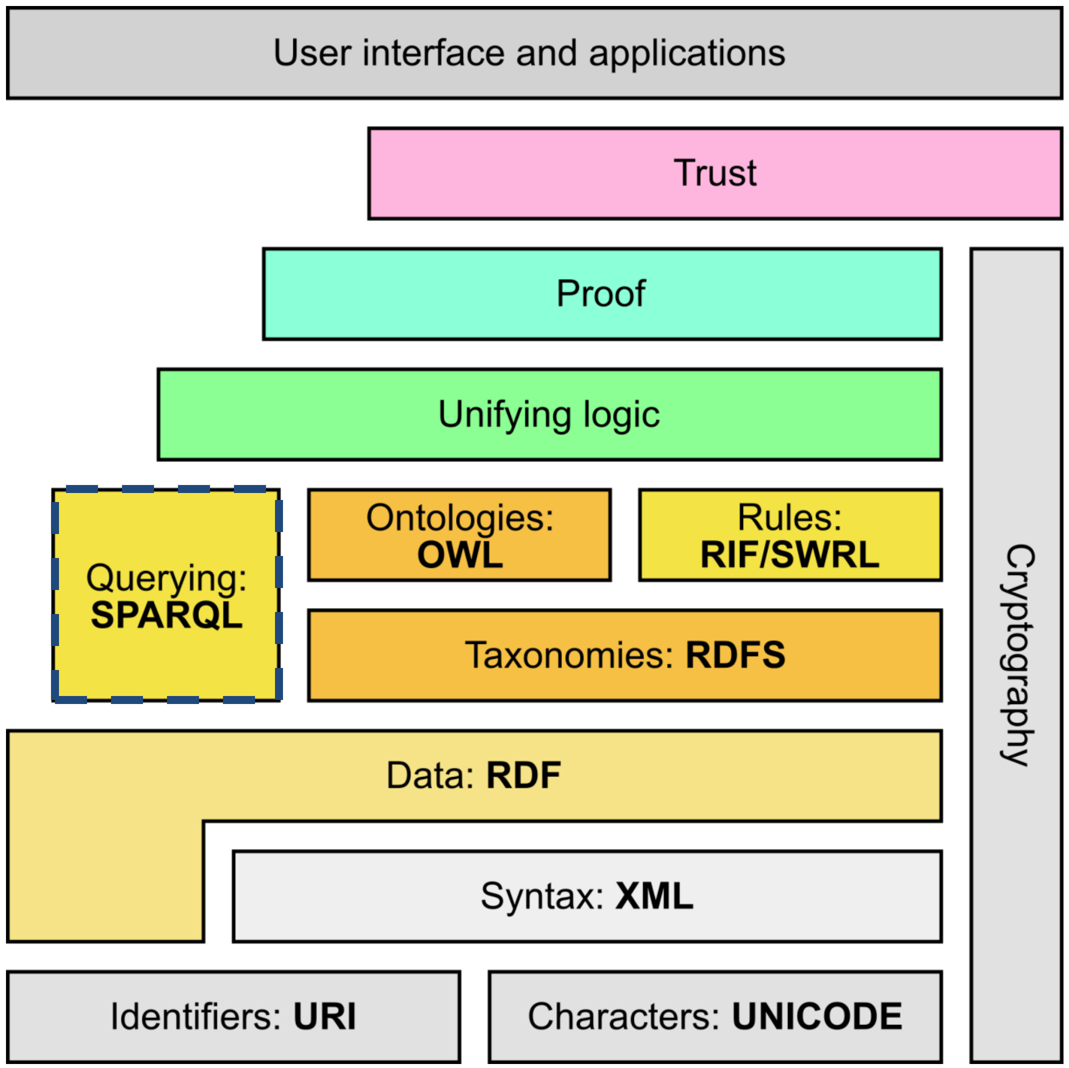
An atom is an expression of the form: ***P(arg1 arg2,...)***

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

Example SWRL Rule:

$$\underbrace{\textbf{Person(?p) ^ hasSibling(?p,?s) ^ Man(?s)}}_{\textbf{antecedent}} \rightarrow \underbrace{\textbf{hasBrother(?p,?s)}}_{\textbf{consequent}}$$

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.

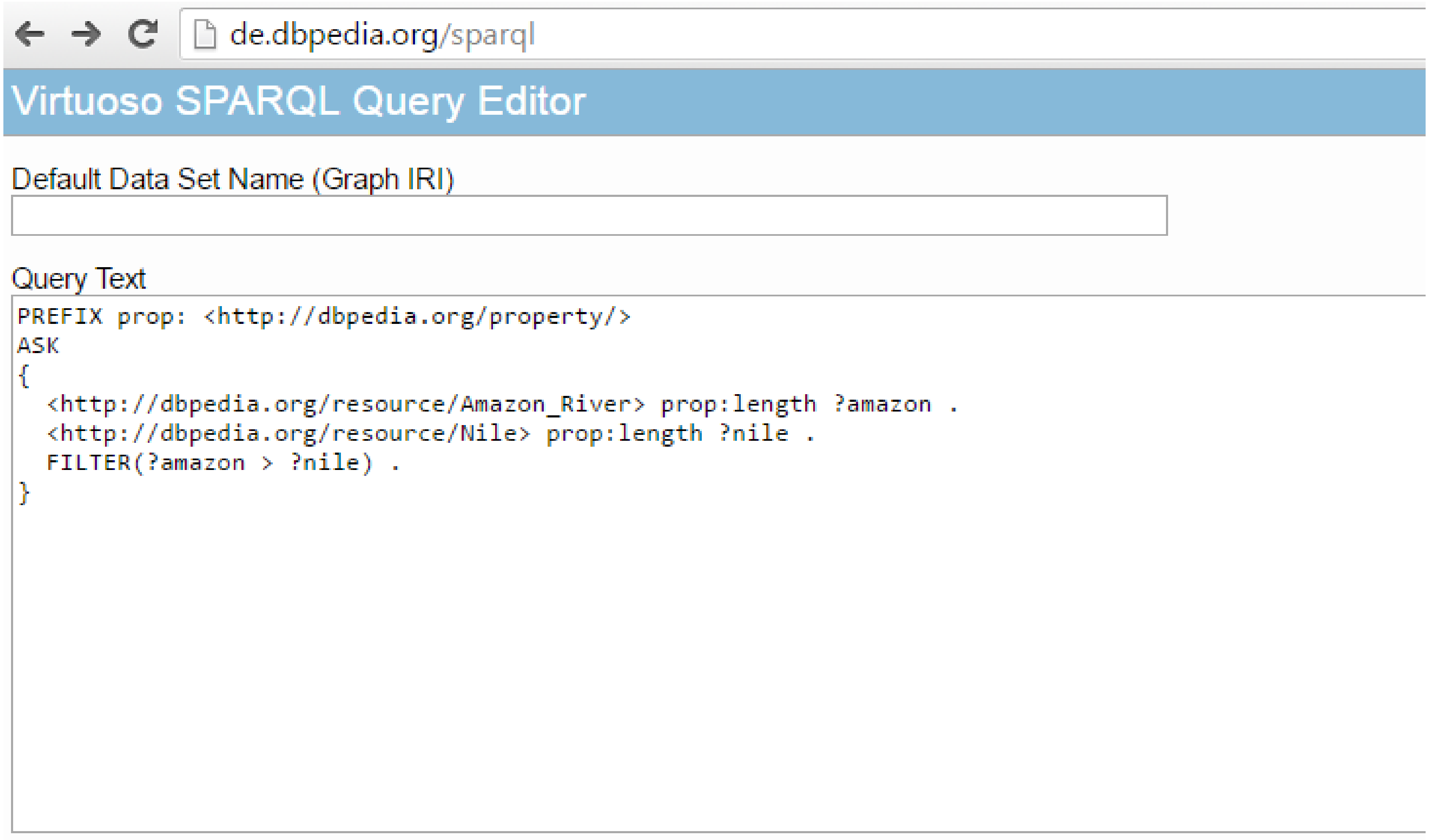
```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX db: <http://bio2rdf.org/drugbank_vocabulary:>

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



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.

LINKQ

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>

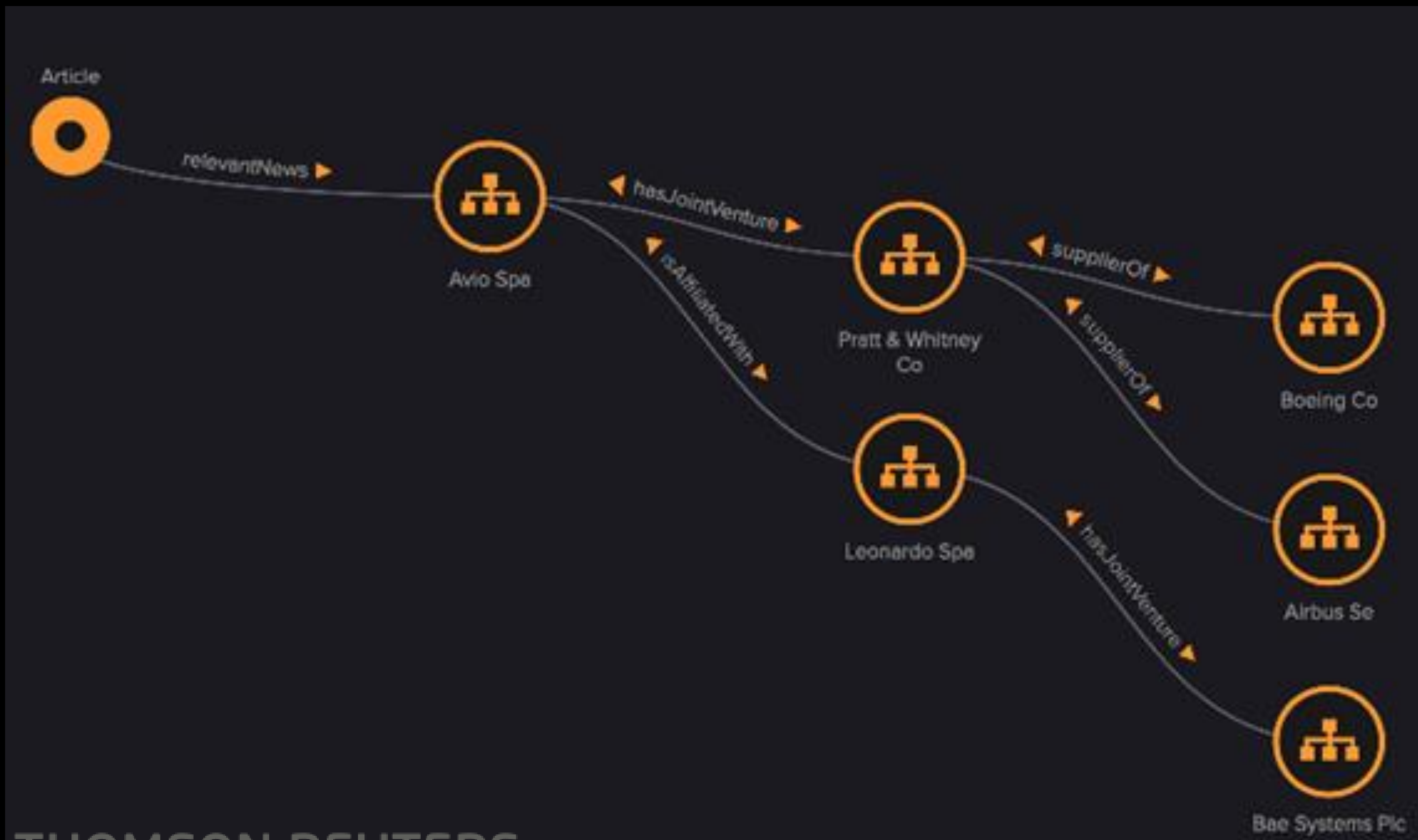
Companies: Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%, Monarch Airlines Ltd 80%
Topics: Other Events: N/A
Industry: Airlines - NEC
Publication date: Oct 9, 2017 1:55:13 PM

Raytheon Co | BAE Systems PLC | Boeing Co

THOMSON REUTERS

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

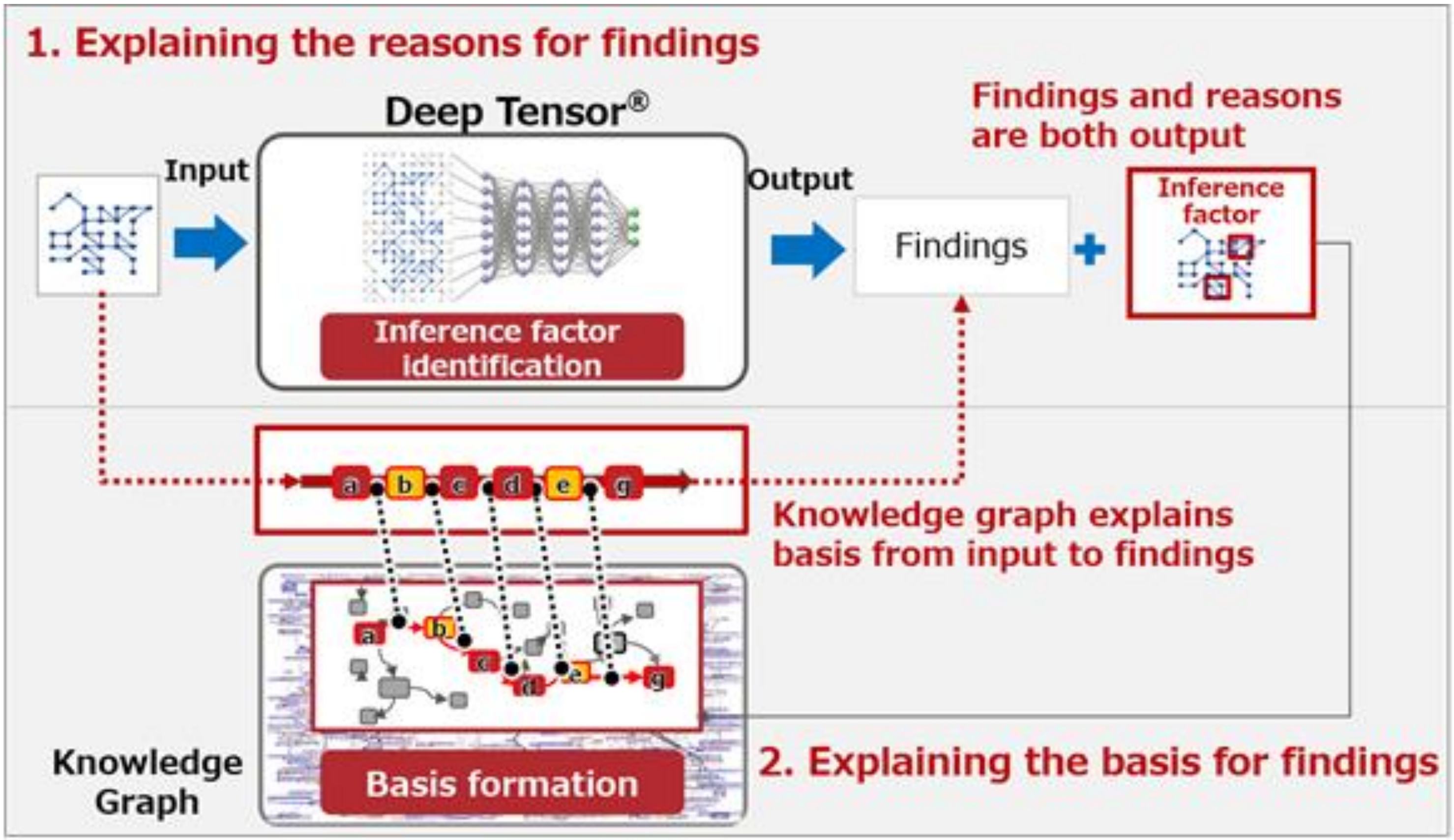
Oct 6 (Reuters) - AVIO SPA <SPA2.MI> * SAYS SIGNED WITH EUROPEAN INVESTMENT BANK CONTRACT FOR EUR 40 MILLION FINANCING Source text for Eikon: [ID:nBIA5D9Ntk] Further company coverage: [SPA2.MI] (Gdynia Newsroom) ((gdynia.newsroom@thomsonreuters.com; +48 58 772 0920 ;))



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

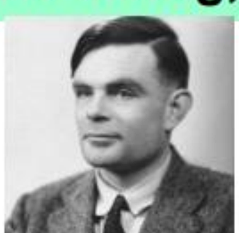
★ **Nobel laureate whose daughter also won a Nobel prize?**

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-capacitators?
  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

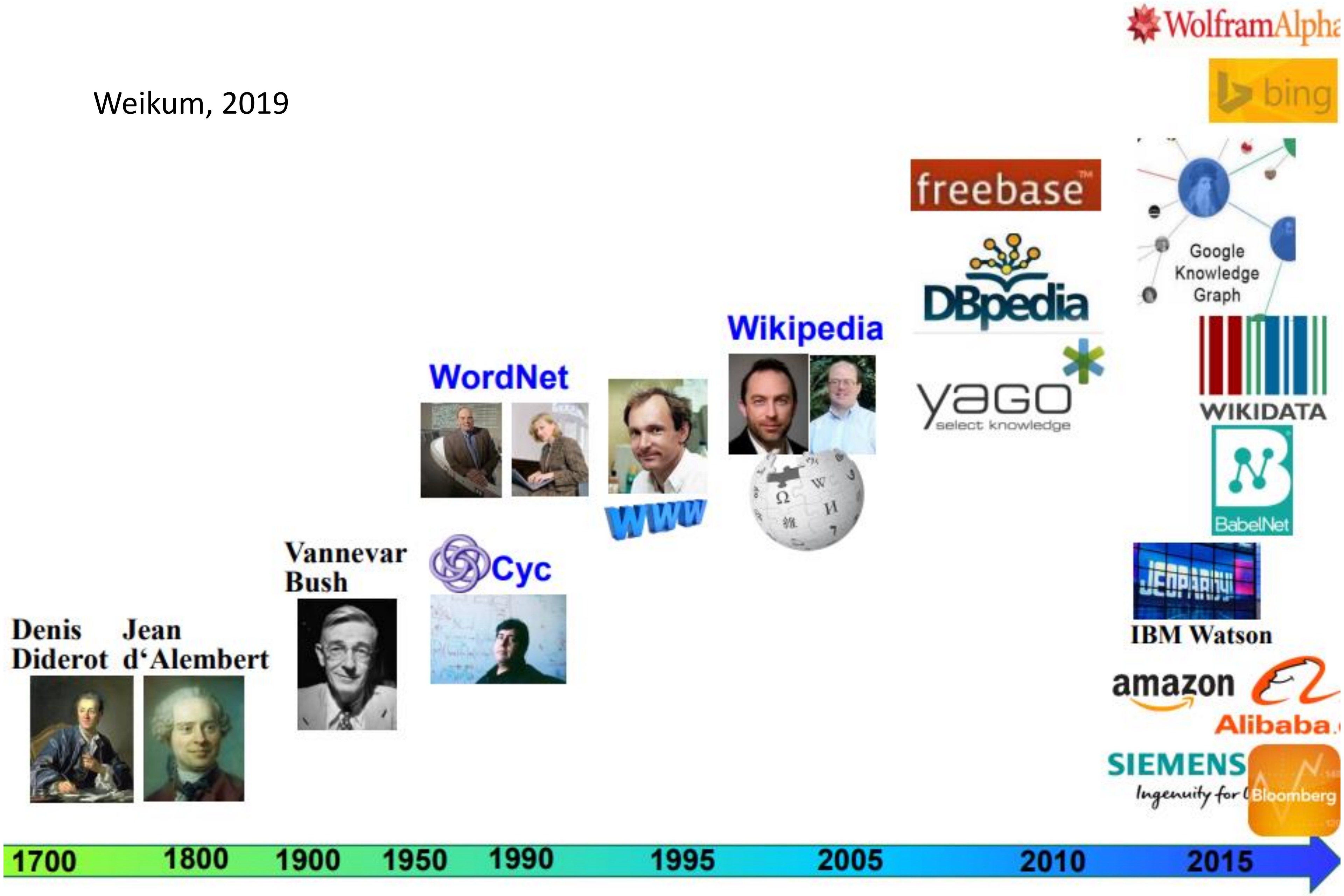
Good Knowledge Representation Languages

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

Knowledge Graphs

Brief History of Knowledge Bases

Weikum, 2019



Terminology

- <u>Ontology</u>
  - provides more complete definitions for concepts
  - Graphical conceptual model
- <u>Thesaurus</u>
  - simple relationships between words
    - synonyms, homonyms, antonyms, etc.
  - often combined with a taxonomy
- <u>Taxonomy</u>
  - hierarchical arrangement of concepts
  - often used as a "backbone" for an ontology
- <u>Lexicon</u>
  - 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 <u>polysemous</u> 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 <u>in some or all contexts</u>:
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H_20
- 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 <u>classes</u> and <u>instances</u>.
- An <u>instance</u> 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”

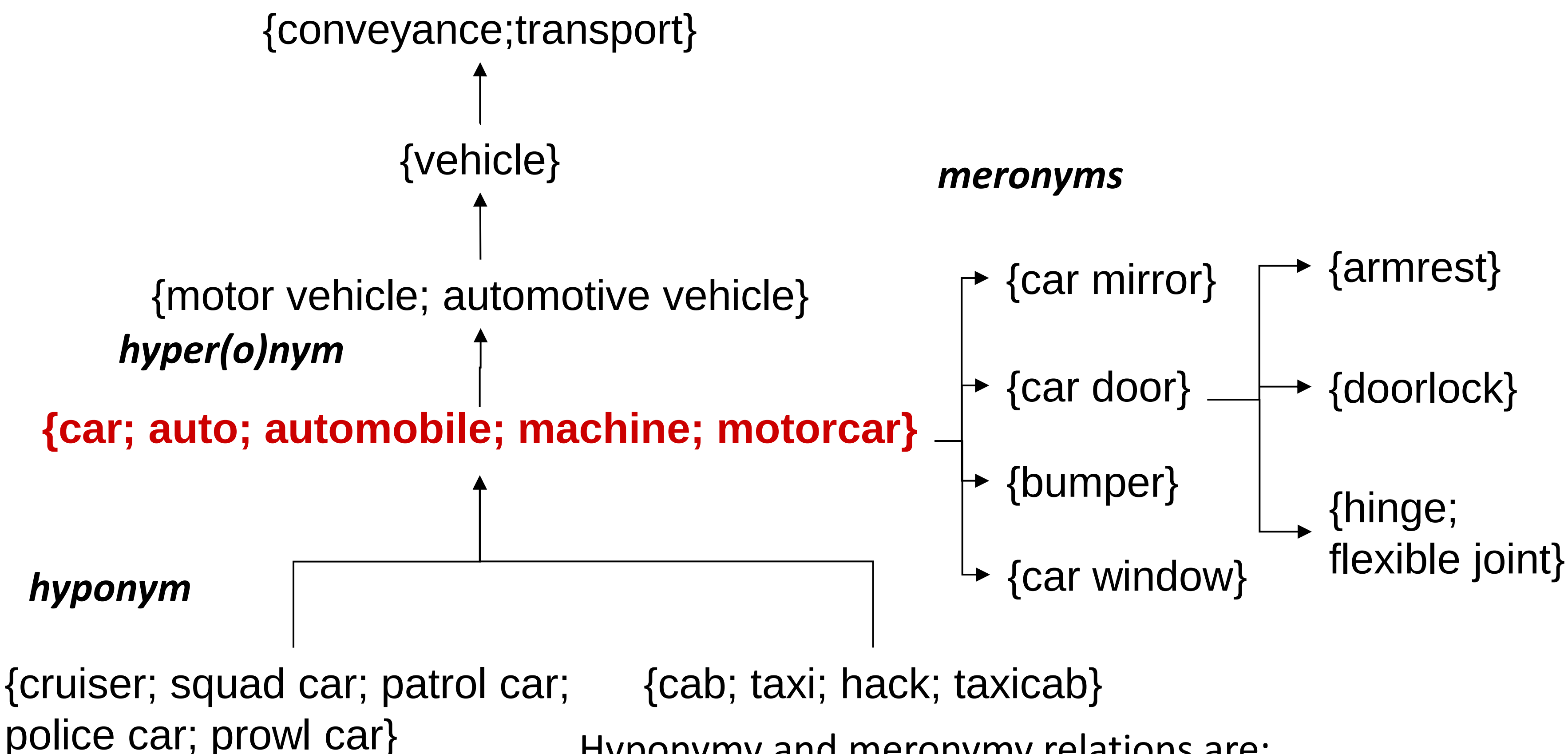
- This sense of “chump” is shared by 9 words:

  chump^{1}, fool^{2}, gull^{1}, mark^{9}, patsy^{1}, fall guy^{1}, sucker^{1}, soft touch^{1}, mug^{2}

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



Hyponymy and meronymy relations are:

- transitive
- directed

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	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	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	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-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

http://wenhoujx.blogspot.com/2013/06/penn-treebank-pos-tags.html

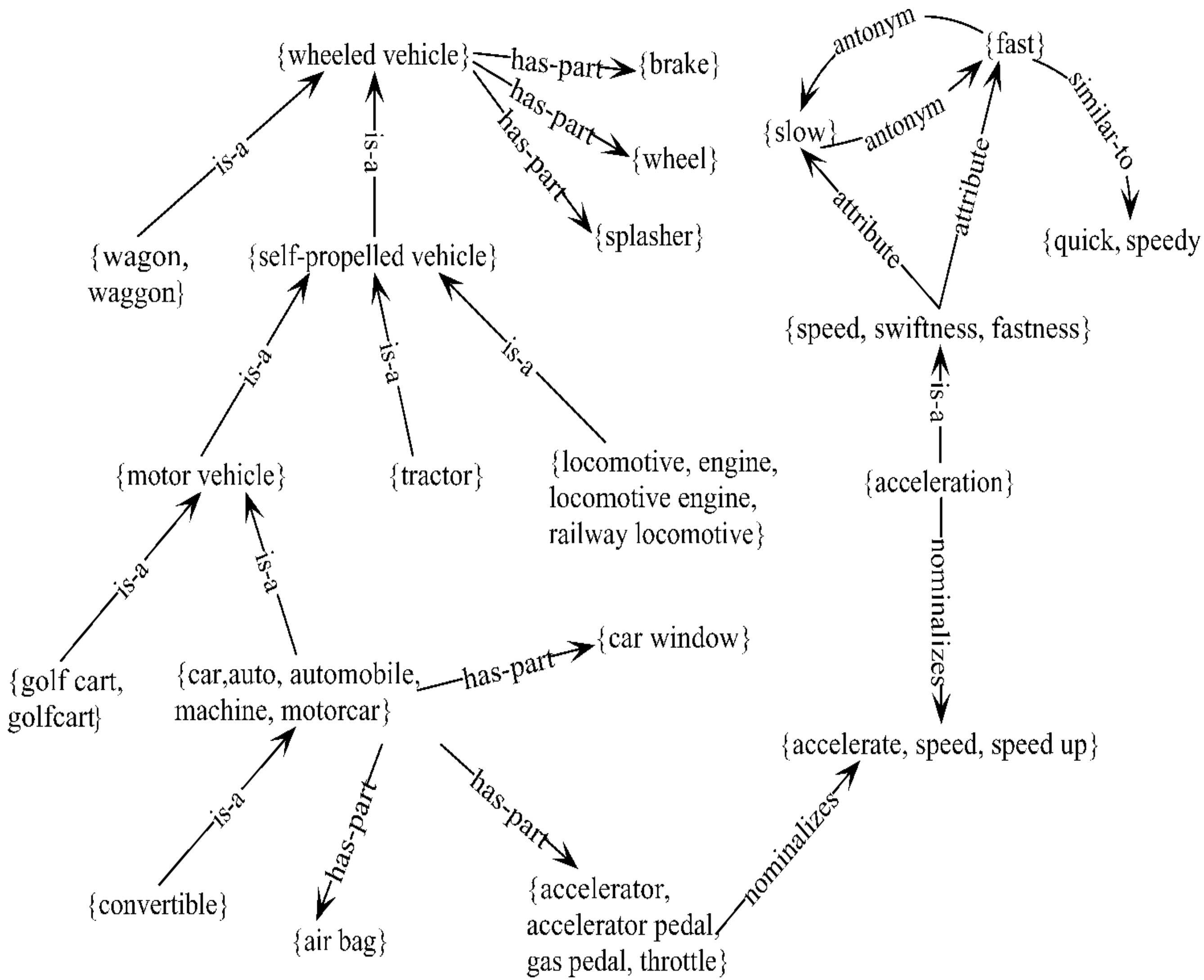
WordNet Noun Relations (Reference)

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

WordNet Verb Relations (Reference)

Relation	**Definition**	**Example**
Hypernym	From events to superordinate events	*fly*^{9} $\rightarrow$ *travel*^{5}
Troponym	From events to subordinate event (often via specific manner)	*walk*^{1} $\rightarrow$ *stroll*^{1}
Entails	From verbs (events) to the verbs (events) they entail	*snore*^{1} $\rightarrow$ *sleep*^{1}
Antonym	Semantic opposition between lemmas	*increase*^{1} $\Longleftrightarrow$ *decrease*^{1}
Derivationally Related Form	Lemmas with same morphological root	*destroy*^{1} $\Longleftrightarrow$ *destruction*^{1}

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

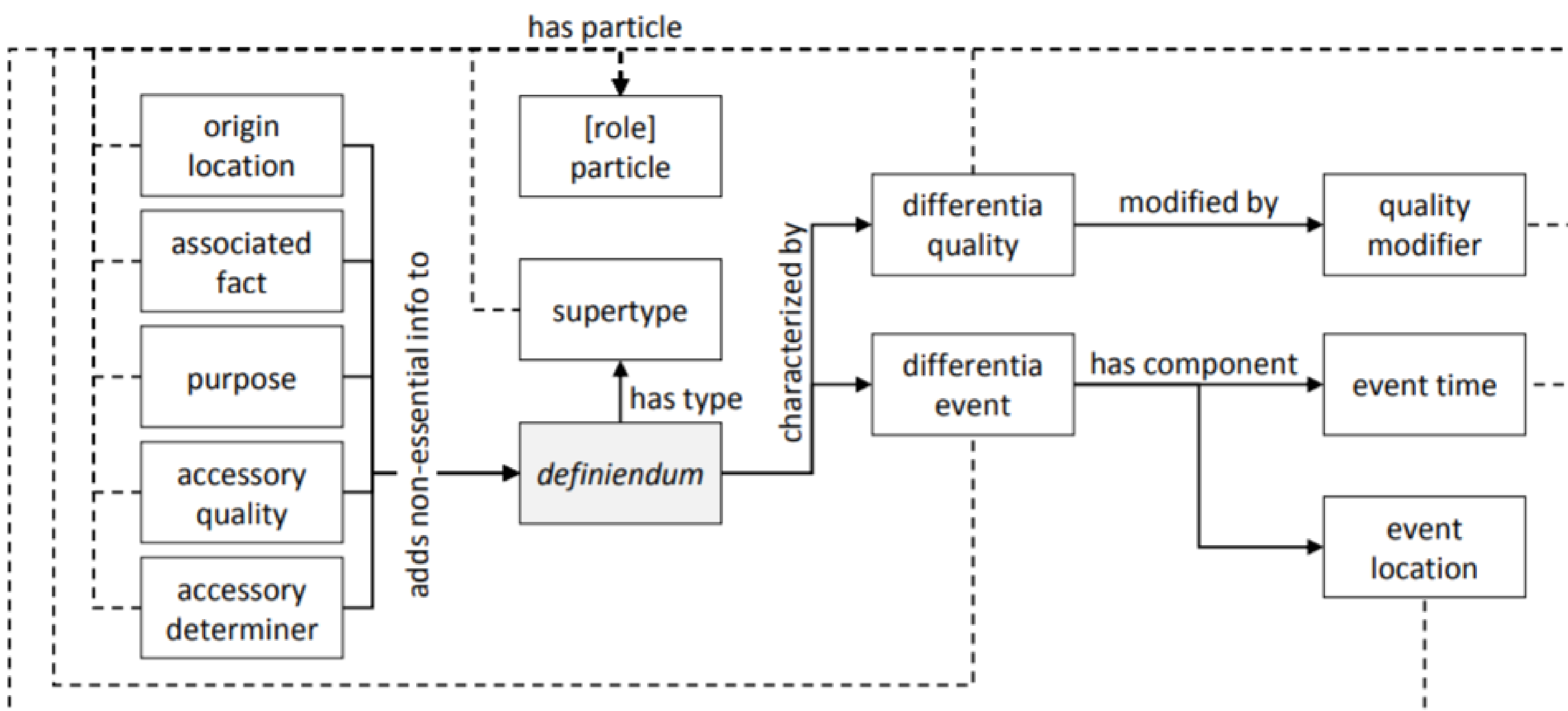
```
? - substance_of(water,X).
        X = [tear|_G407]
? - has_substance(water,X).
        X = [h2o|_G407]

?- part_of(leg,X).
X = [table|_G407]
?- has_part(leg,X).
X = [knee|_G407]
```

https://github.com/ekaf/wordnet-prolog

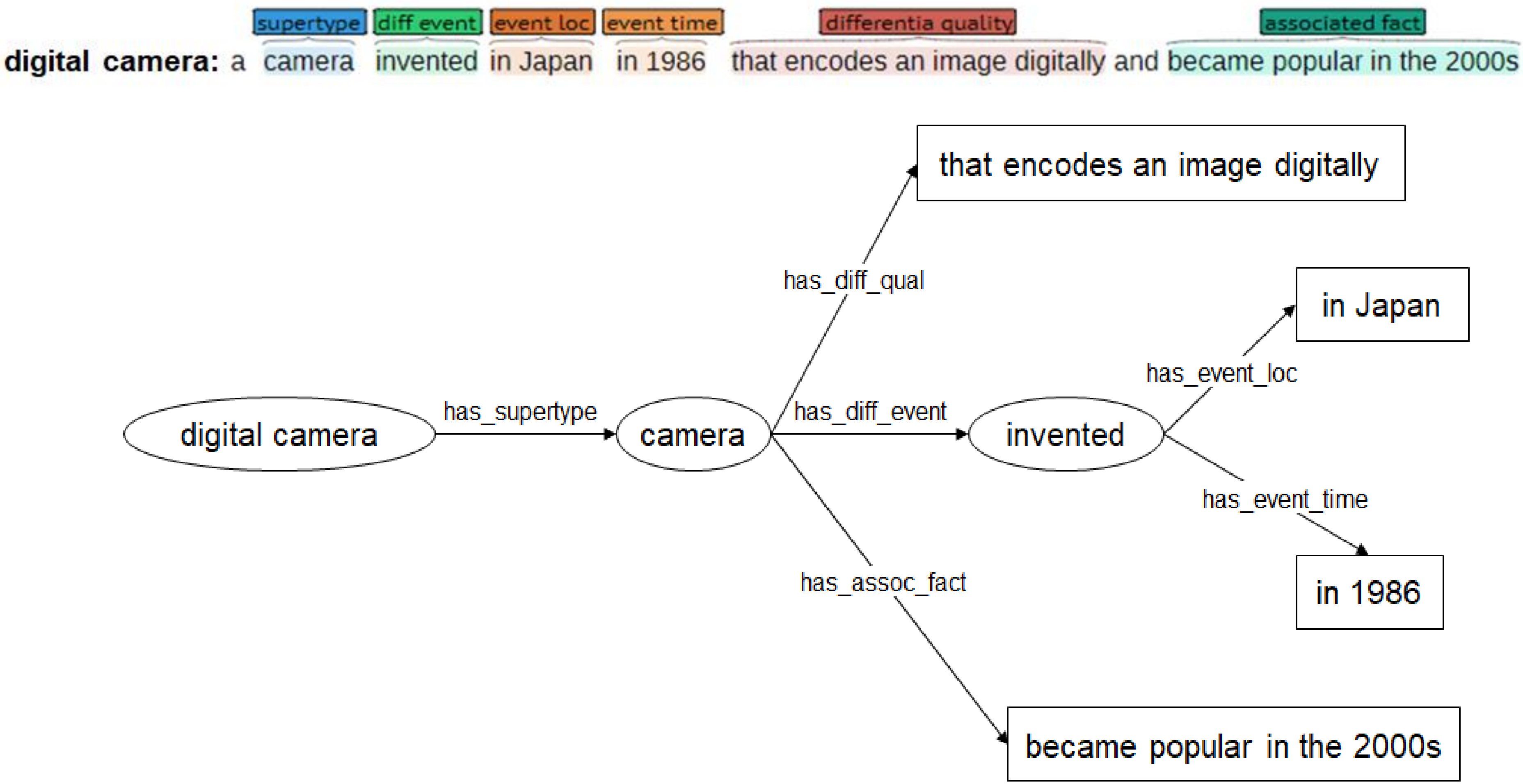
Sarah Witzig

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



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

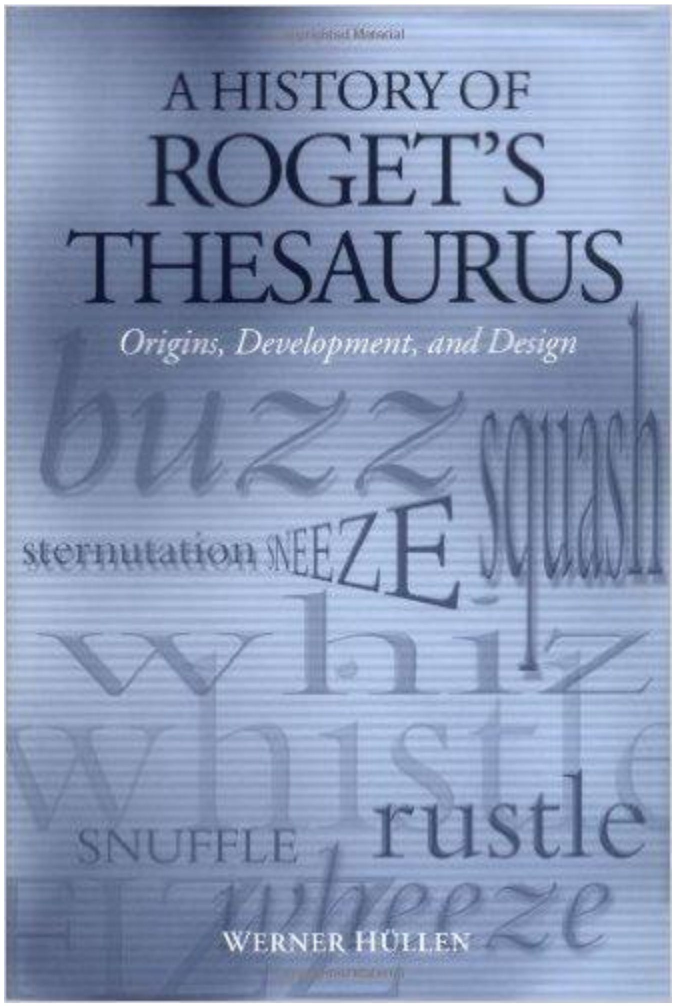
  Definition: The oxygen-carrying proteins of ERYTHROCYTES. They are found in all vertebrates and some invertebrates. The number of globin subunits in the hemoglobin quaternary structure differs between species. Structures range from monomeric to a variety of multimeric arrangements

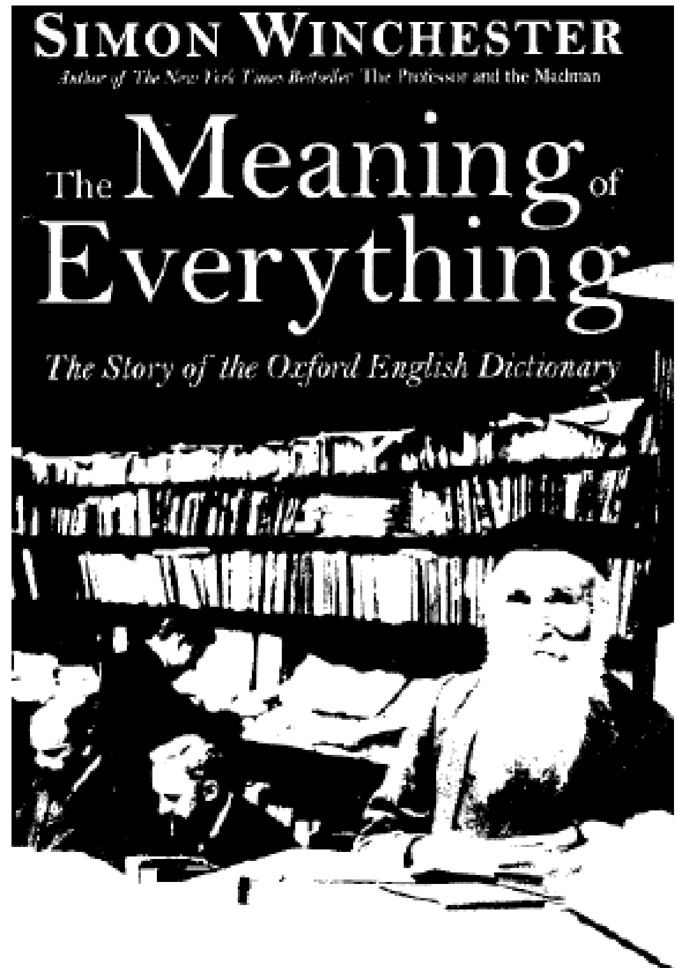
The MeSH Hierarchy

1. + Anatomy [A]
2. + Organisms [B]
3. + Diseases [C]
4. 1 − Chemicals and Drugs [D]
    - 2 Inorganic Chemicals [D01] +
    - Organic Chemicals [D02] +
    - 3 Heterocyclic Compounds [D03] +
    - 4 Polycyclic Compounds [D04] +
    - 5 Macromolecular Substances [D05] +
    - 6 Hormones, Hormone Substitutes, and
    - 7 Enzymes and Coenzymes [D08] +
    - 8 Carbohydrates [D09] +
    - 9 Lipids [D10] +
    - 10 Amino Acids, Peptides, and Proteins [
    - 11 Nucleic Acids, Nucleotides, and Nucleo
    - 12 Complex Mixtures [D20] +
    - 13 Biological Factors [D23] +
    - 14 Biomedical and Dental Materials [D25
    - Pharmaceutical Preparations [D26] +
    - 15 Chemical Actions and Uses [D27] +
5. 16 + 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]
                - Erythrocruorins [D12.776.124.400.220]

Curating Definitions: A Tour de Force





The theoretical distinction between <u>dictionaries</u> and <u>encyclopaedias</u> 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.

- <u>Componential analysis</u> or <u>semantic decomposition</u> approach:
  - word meaning is modelled in terms of <u>semantic features</u> or <u>primitives</u>.

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.

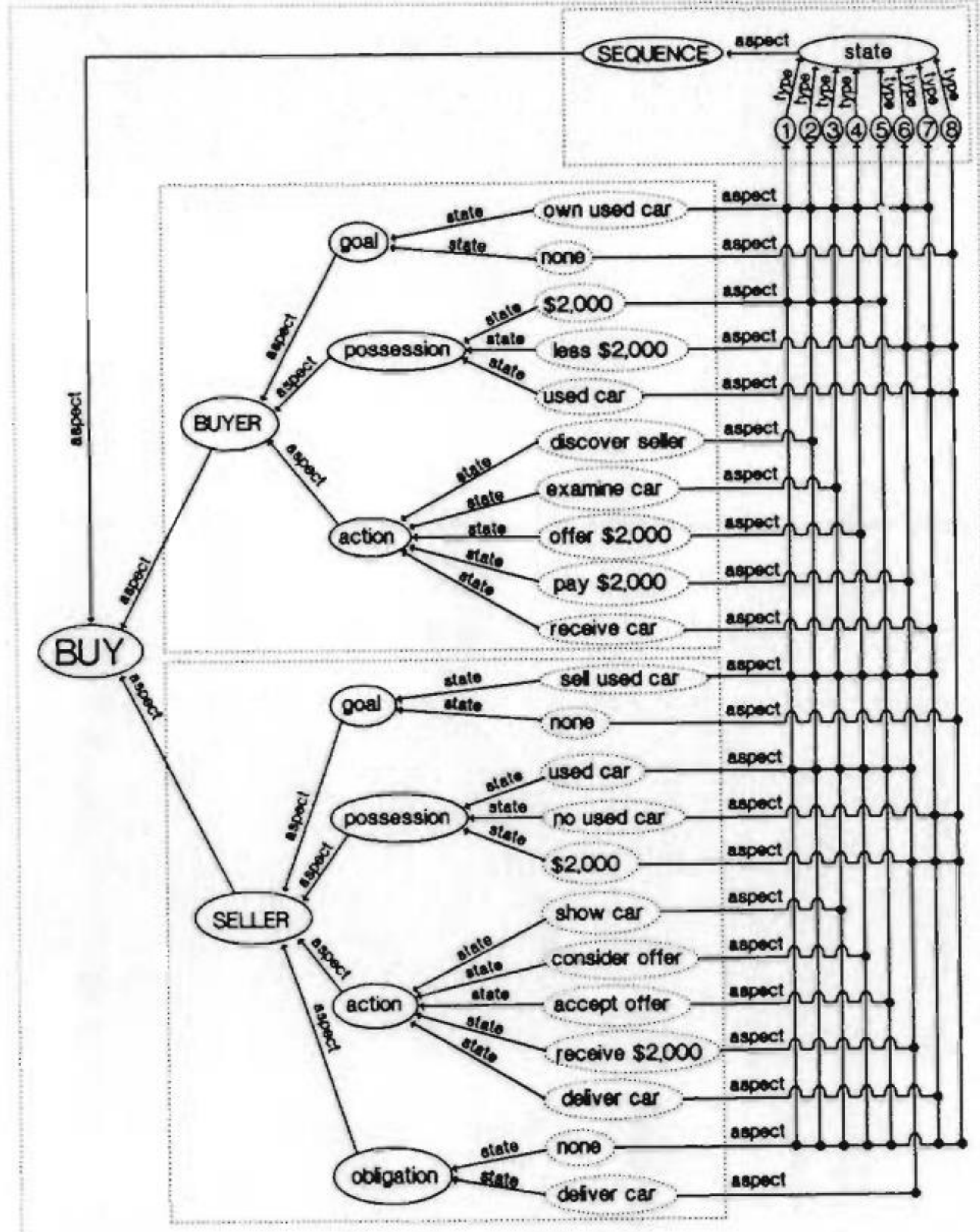
- not necessarily incompatible with feature theories
- fuzzy boundaries
- family resemblance

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

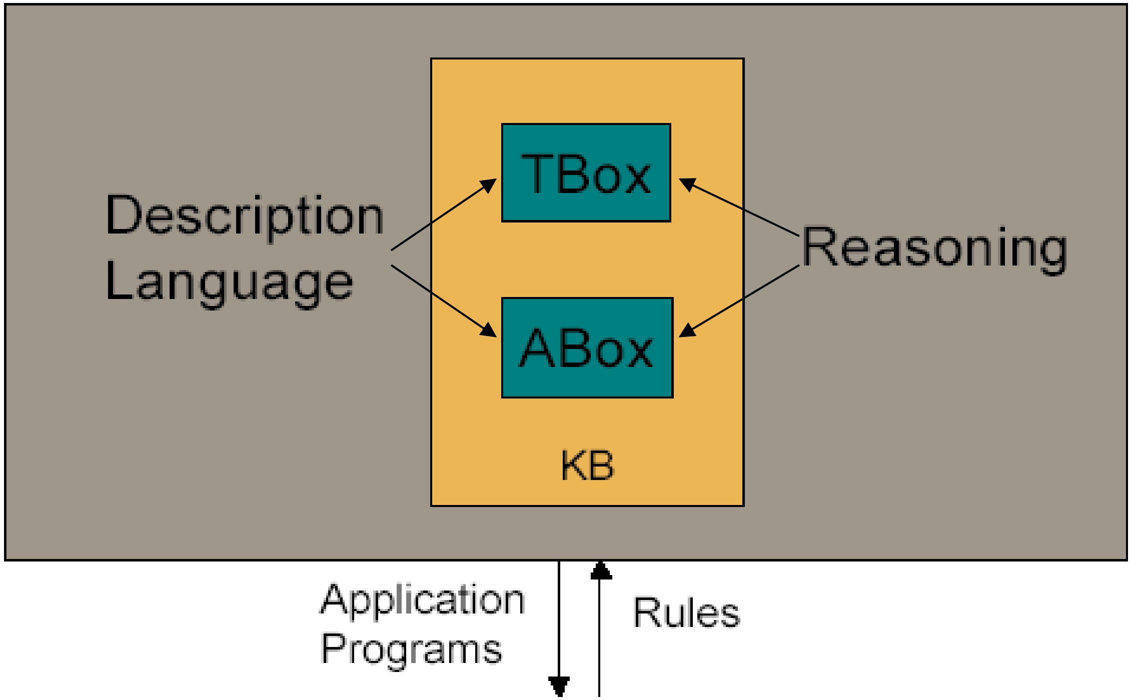
Horrocks

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).

Horrocks

Architecture of a DL System



Horrocks

Formulas

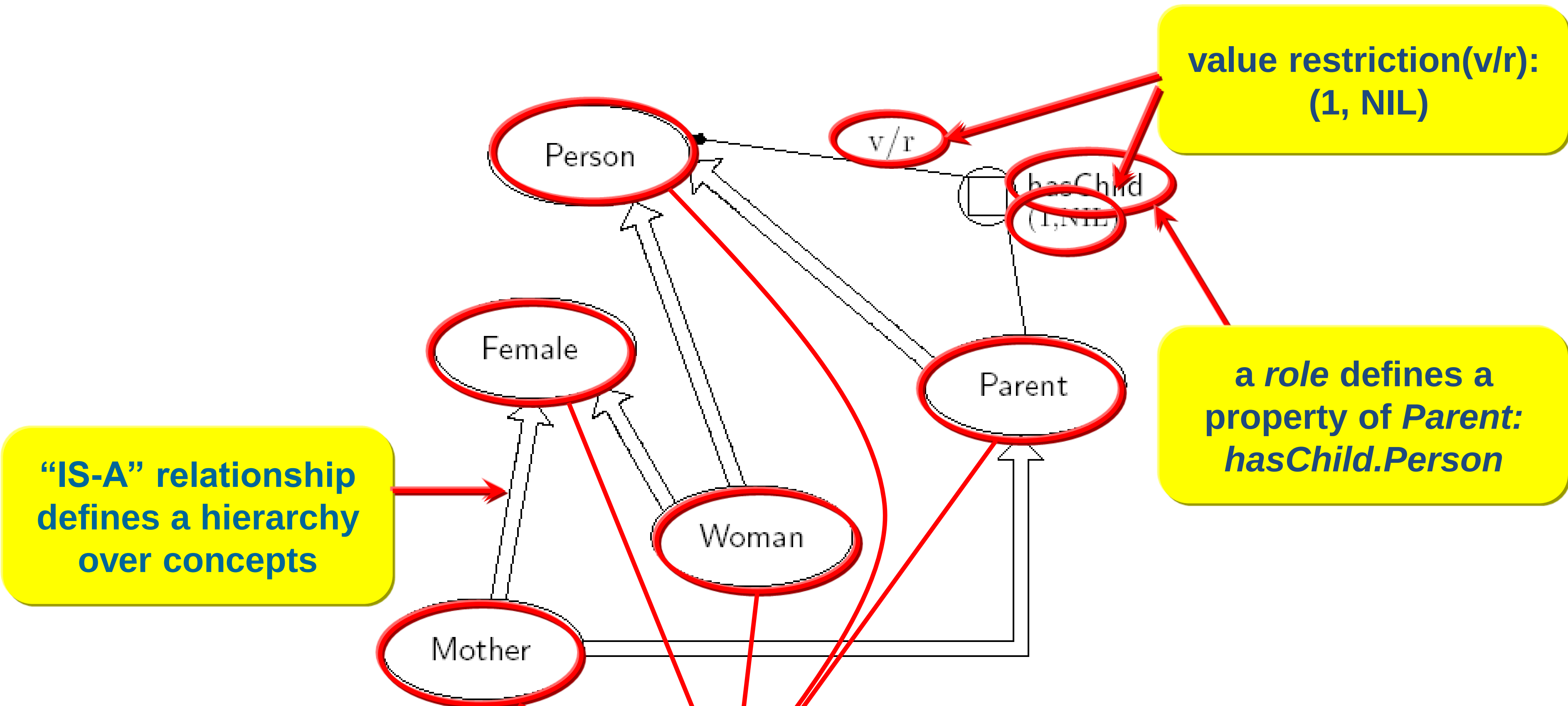
- Building blocks that allow complex descriptions of concepts and roles.
  - Example (we'll look at the syntax in more detail soon.)
    - A Woman is a Female Person
      - Woman = Person **u** Female
    - A Mother is a Woman and she has a child
      - Mother = Woman u hasChild.T

- The TBox can be used to assign names to complex descriptions.

  We will use the terms description and concept interchangeably.

Horrocks

An Example about Family Relationships



These are concepts

Examples in TBox:

Woman ´ Person u Female

Mother ´ Woman u hasChild.Person

Examples in ABox:

hasChild(MARY, PETER)

Father(PETER)

Name Symbols vs. Base Symbols

- Atomic concepts occurring in a TBox **T** can be divided into two sets, <u>name symbols N_T</u> (or defined concepts) and <u>base symbols B_T</u> (or primitive concepts, occur only on the right-hand side)

- A <u>base interpretation</u> for **T** only interprets the base symbols.

Base symbols

Name Symbol

$$
\begin{array}{rcl}
\mathsf{Woman} & \equiv & \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} & \equiv & \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} & \equiv & \mathsf{Woman} \sqcap \exists \mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Father} & \equiv & \mathsf{Man} \sqcap \exists \mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Parent} & \equiv & \mathsf{Father} \sqcup \mathsf{Mother} \\
\mathsf{Grandmother} & \equiv & \mathsf{Mother} \sqcap \exists \mathsf{hasChild}.\mathsf{Parent} \\
\mathsf{MotherWithManyChildren} & \equiv & \mathsf{Mother} \sqcap \geqslant 3\, \mathsf{hasChild} \\
\mathsf{MotherWithoutDaughter} & \equiv & \mathsf{Mother} \sqcap \forall \mathsf{hasChild}.\neg\mathsf{Woman} \\
\mathsf{Wife} & \equiv & \mathsf{Woman} \sqcap \exists \mathsf{hasHusband}.\mathsf{Man}
\end{array}
$$

Horrocks

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)

Horrocks

OWL Class Constructor

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

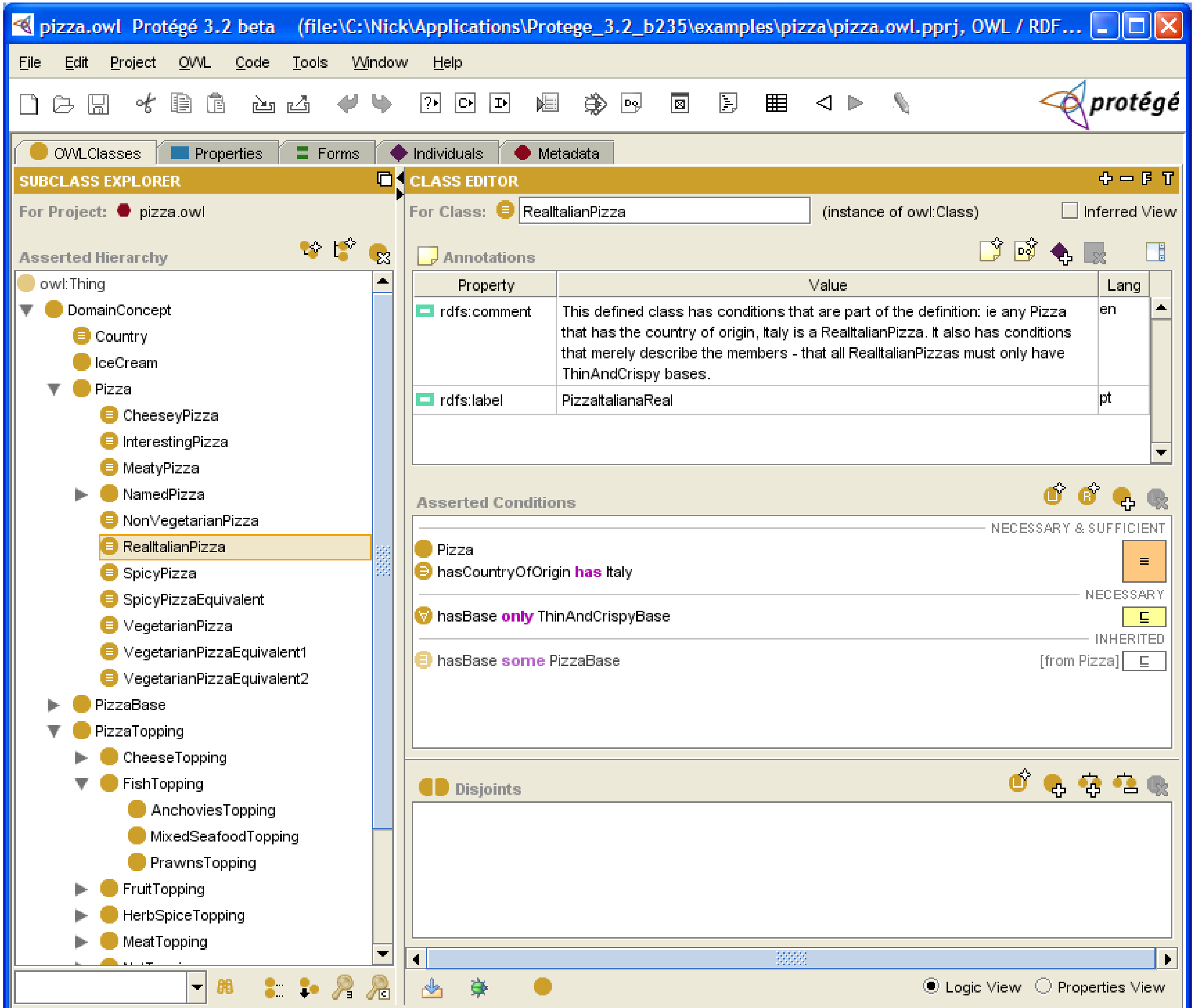
Horrocks

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	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant 1$hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant 1$hasSSN$^-$

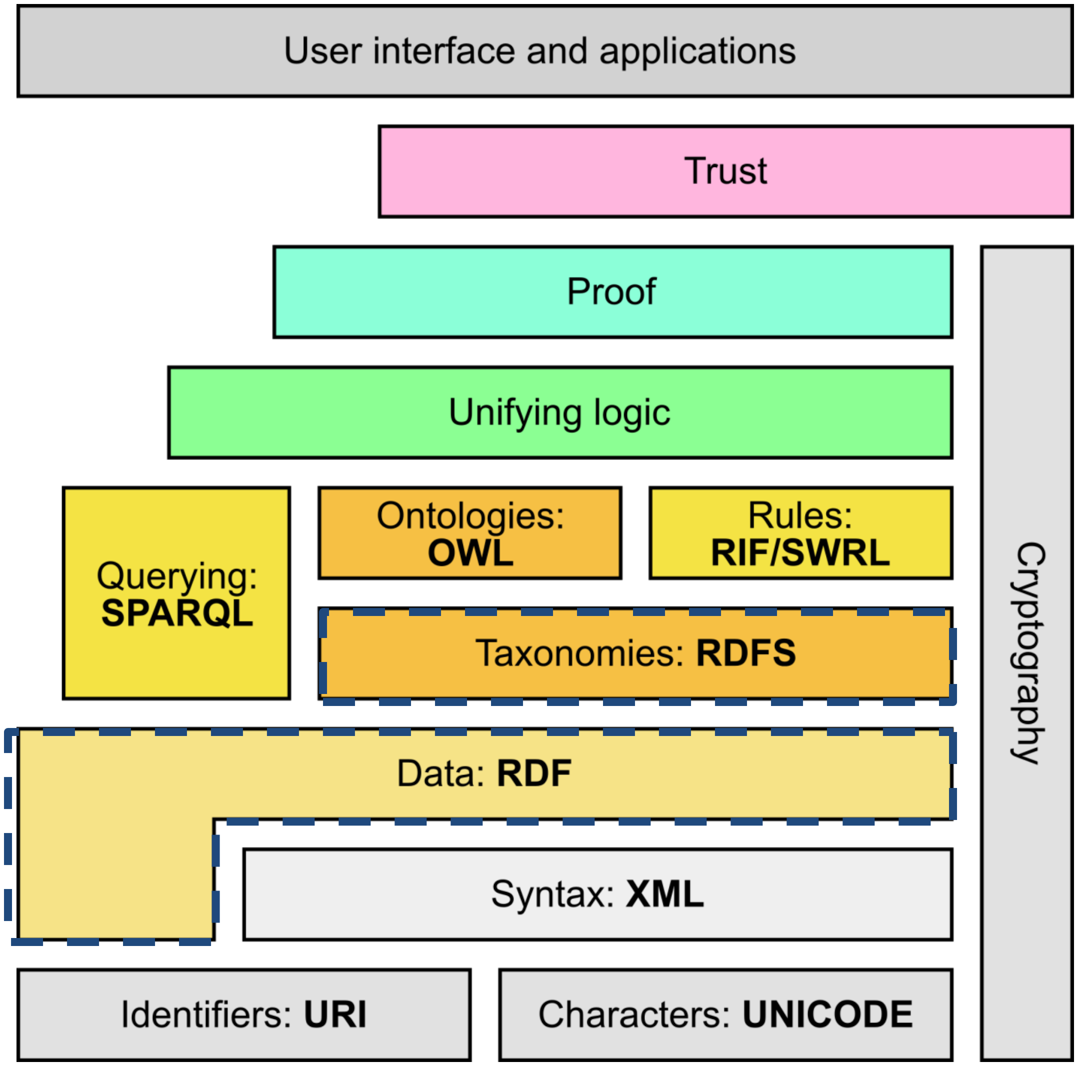
Horrocks

Ontology Editors



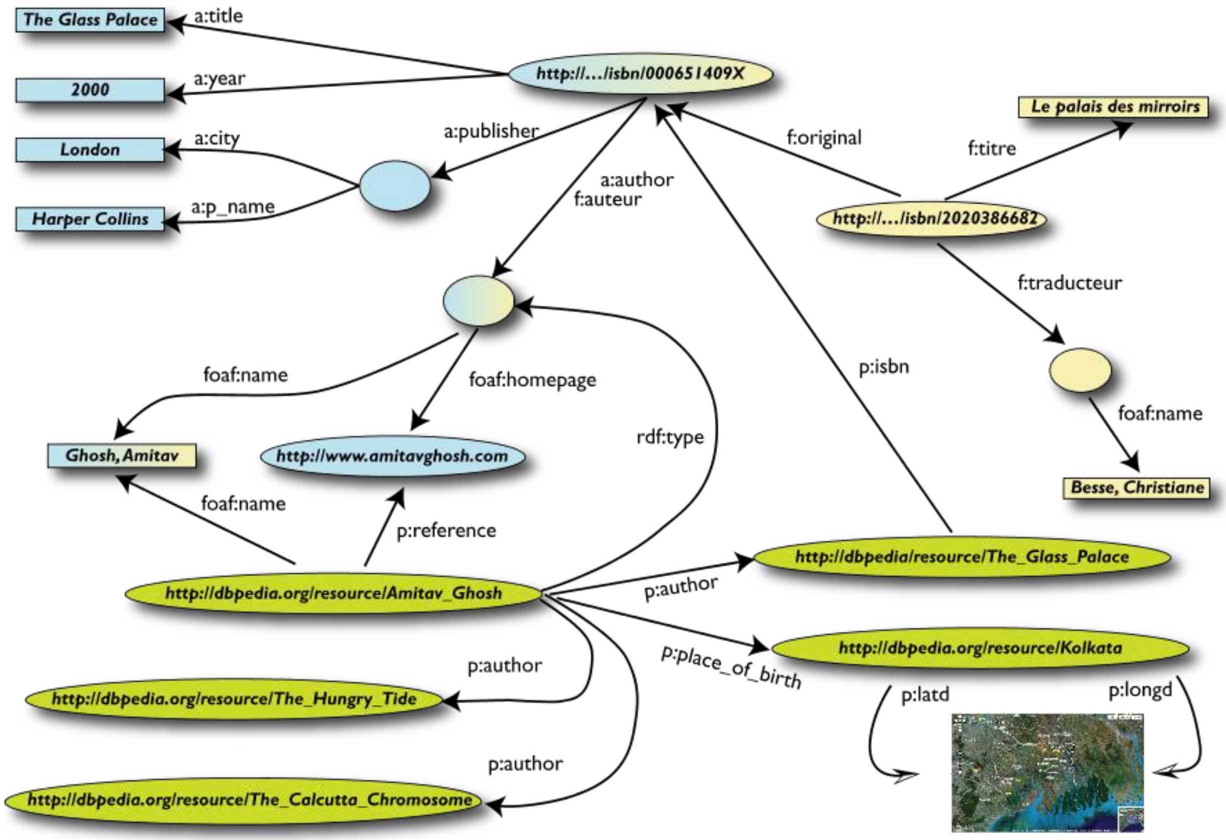
Semantic Web

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

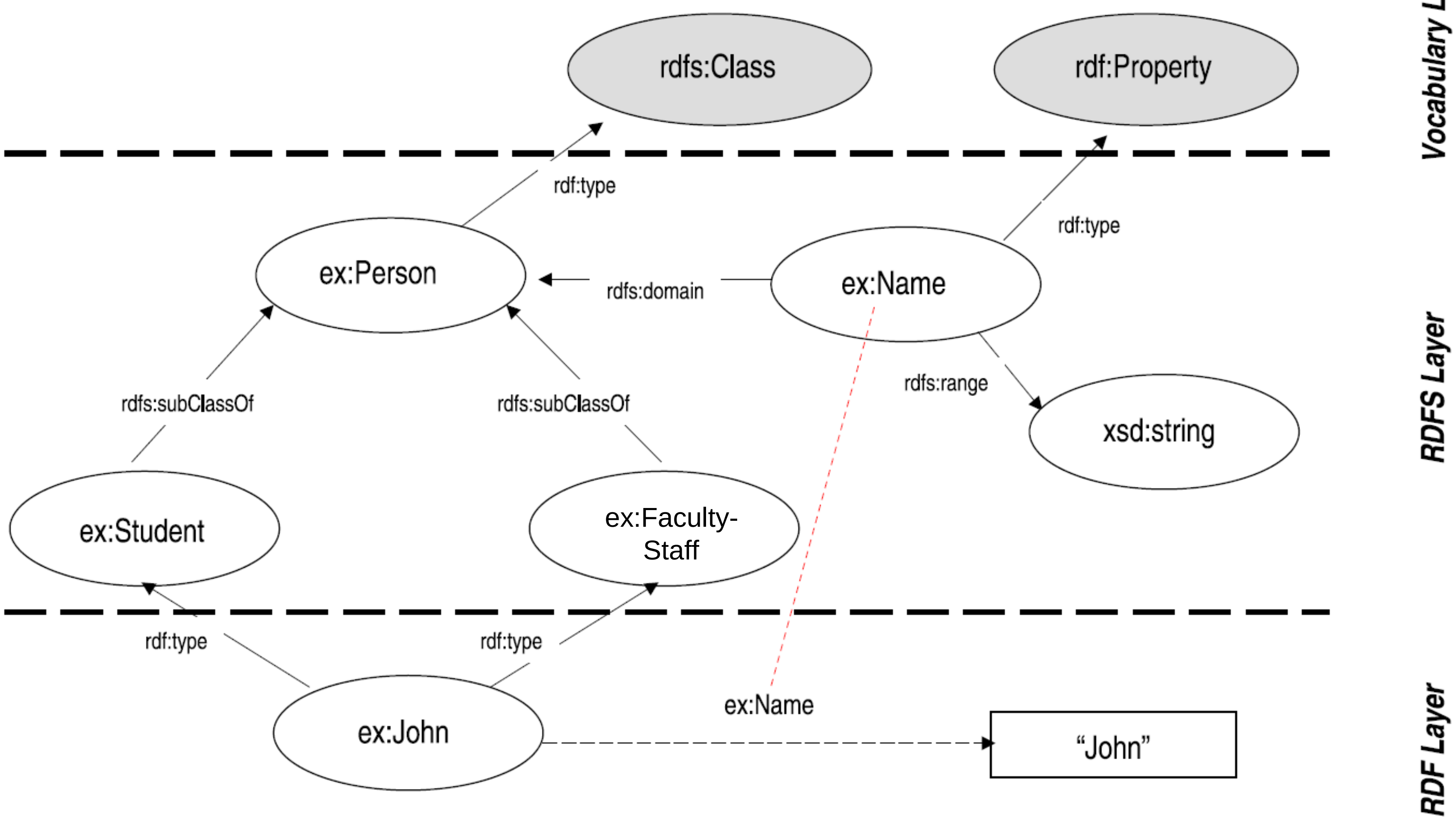
RDFS Classes

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

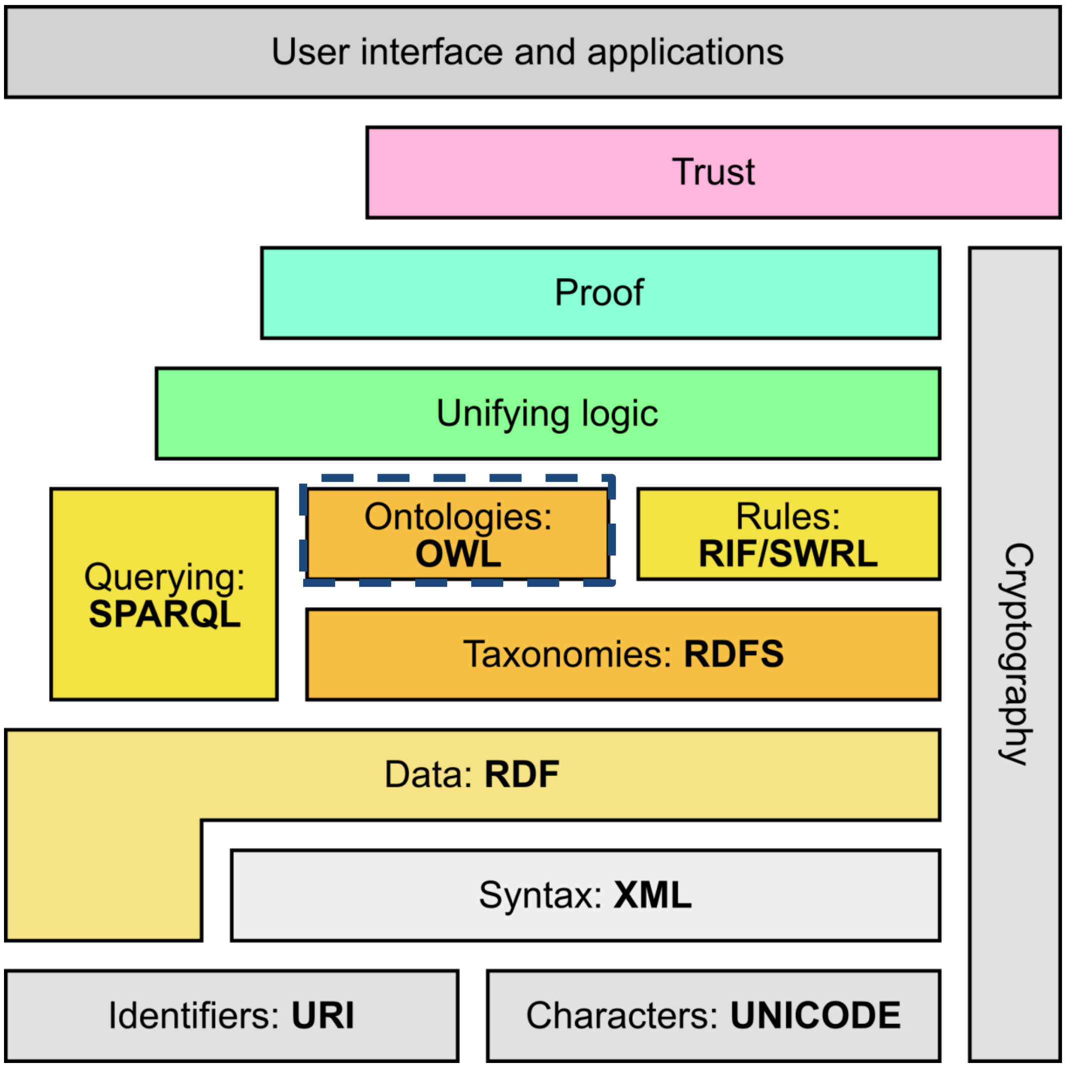
RDFS Properties

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

RDFS Example

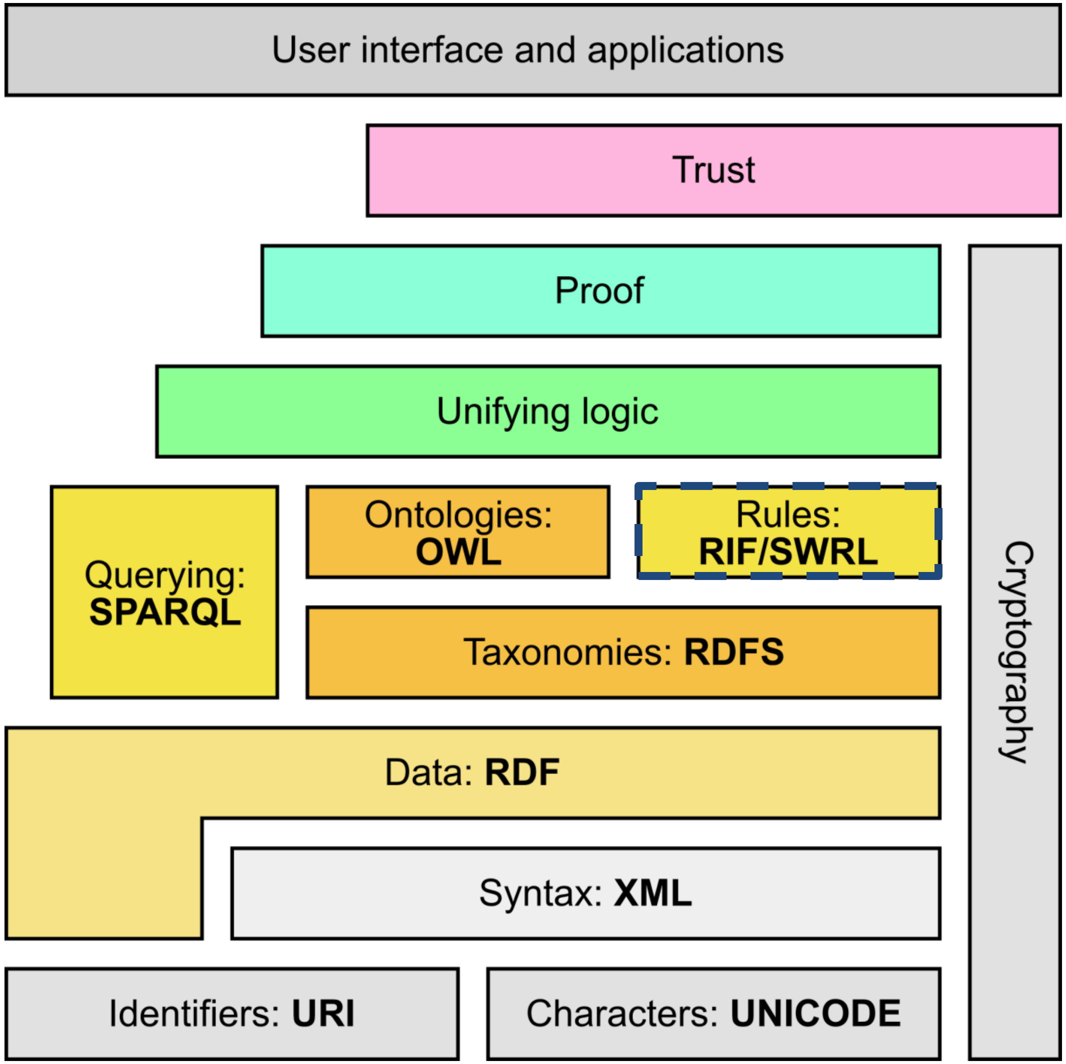


Semantic Web Stack



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

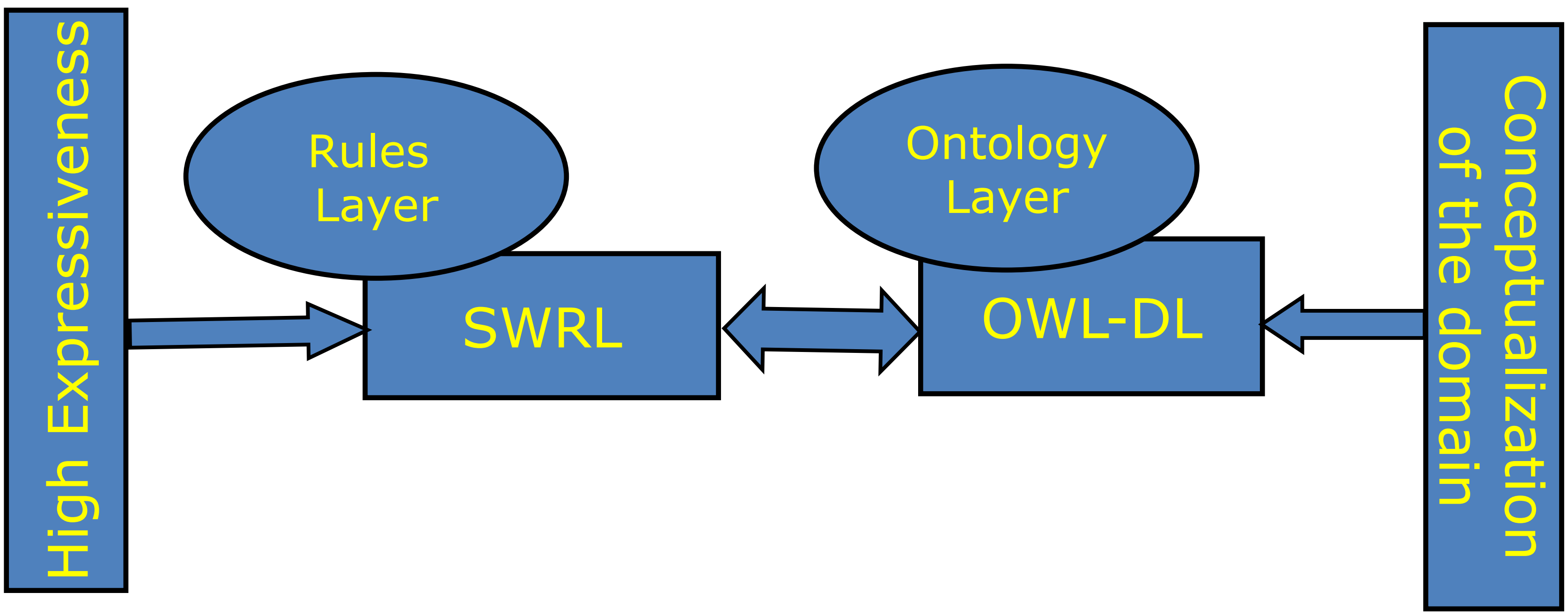
Semantic Web Stack



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

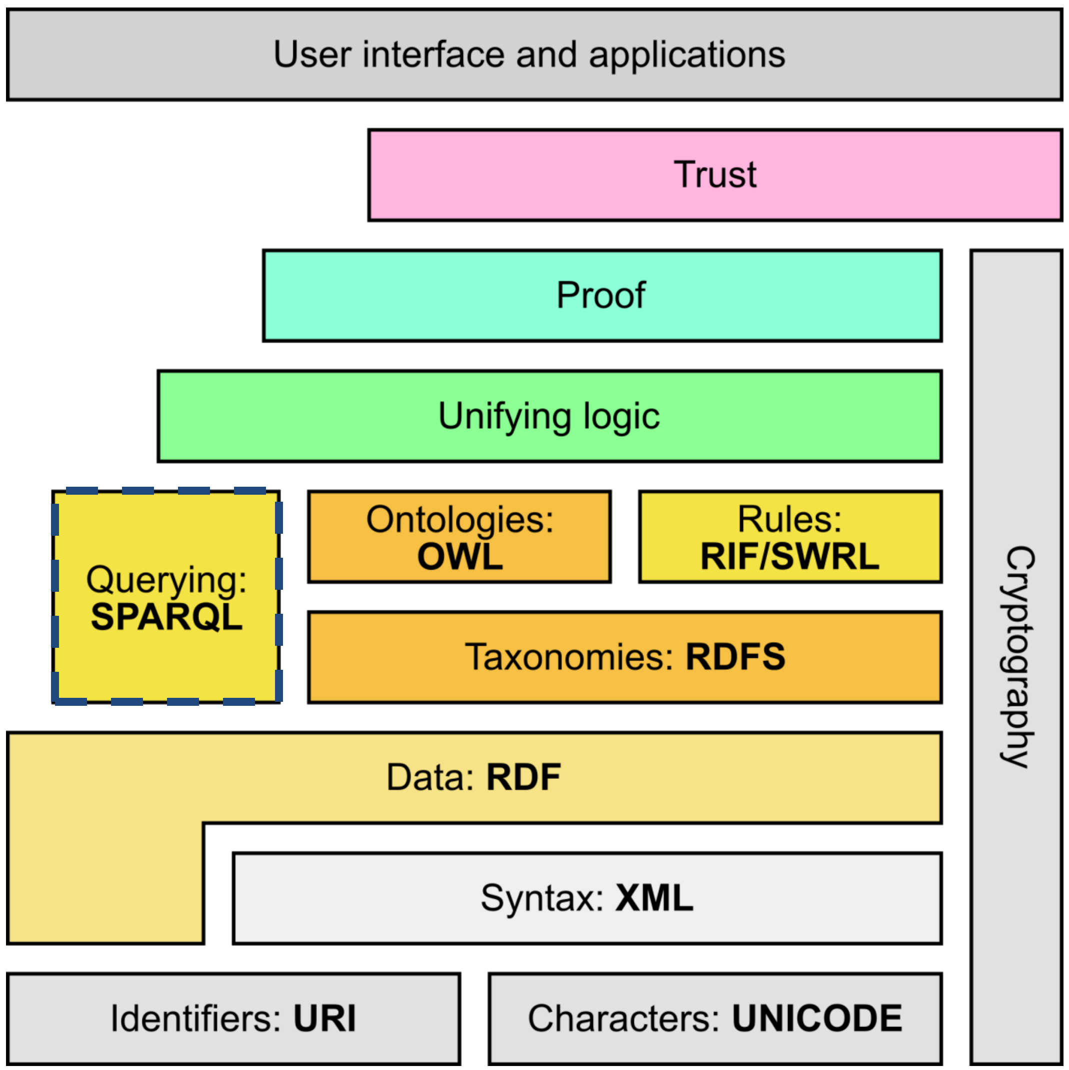
An atom is an expression of the form: ***P(arg1 arg2,...)***

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

Example SWRL Rule:

$$\underbrace{\textbf{Person(?p)} \wedge \textbf{hasSibling(?p,?s)} \wedge \textbf{Man(?s)}}_{\textbf{antecedent}} \rightarrow \underbrace{\textbf{hasBrother(?p,?s)}}_{\textbf{consequent}}$$

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

de.dbpedia.org/sparql

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

Query Text

```
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.

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

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

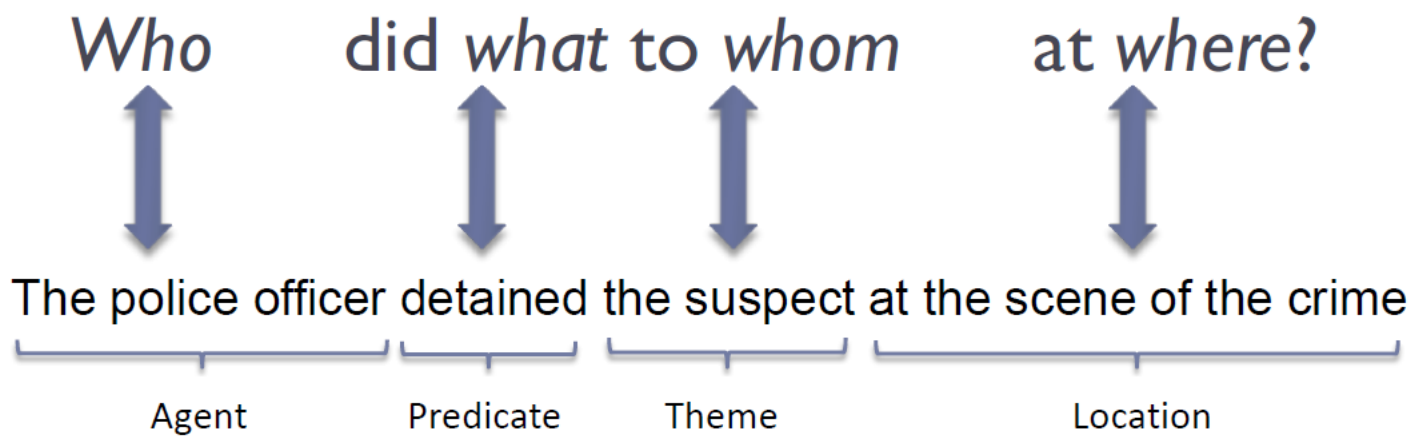
Representing Events

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

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*.

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

Verb Alternations (Diathesis Alternations)

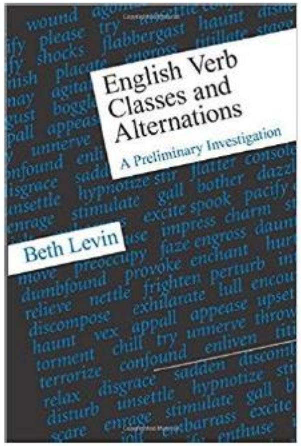
Doris gave the book to Cary.
AGENT THEME GOAL

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: Agreer
Arg1: Proposition
Arg2: Other entity agreeing

Ex1: [$_{\text{Arg0}}$ The group] *agreed* [$_{\text{Arg1}}$ it wouldn't make an offer].
Ex2: [$_{\text{ArgM-TMP}}$ Usually] [$_{\text{Arg0}}$ John] *agrees* [$_{\text{Arg2}}$ with Mary] [$_{\text{Arg1}}$ on everything].

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106

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

[$_{\text{Arg0}}$ Big Fruit Co.] increased [$_{\text{Arg1}}$ the price of bananas].
[$_{\text{Arg1}}$ The price of bananas] was increased again [$_{\text{Arg0}}$ by Big Fruit Co.]
[$_{\text{Arg1}}$ The price of bananas] increased [$_{\text{Arg2}}$ 5%].

Frege's Begriffsschrift for the Same Sentence

- Peirce's algebraic notation (1885):

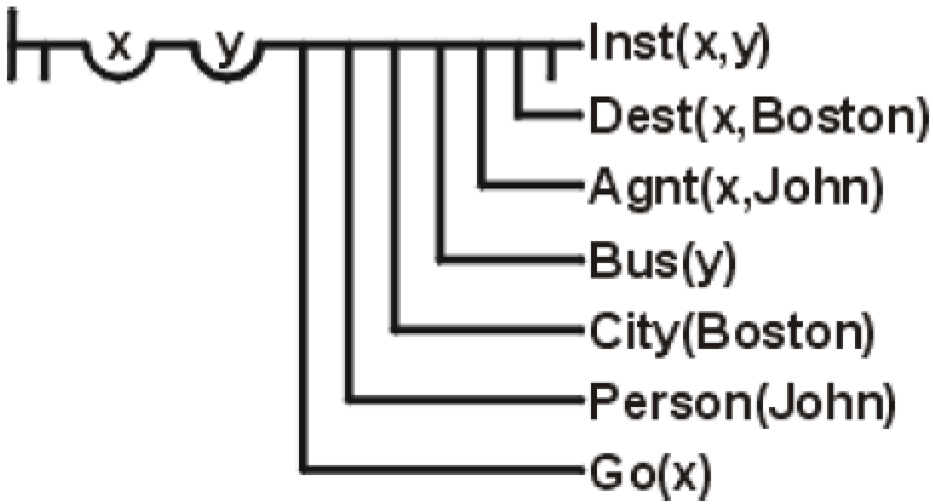
$$\Sigma_x \Sigma_y \, (\mathrm{Go}(x) \bullet \mathrm{Person(John)} \bullet \mathrm{City(Boston)} \bullet \mathrm{Bus}(y) \bullet \mathrm{Agnt}(x,\mathrm{John}) \bullet \mathrm{Dest}(x,\mathrm{Boston}) \bullet \mathrm{Inst}(x,y))$$

- With Peano's choice of symbols:

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

J. Sowa

Frege’s Begriffsschrift for the Same Sentence



- Translation to Peirce-Peano notation:

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

J. Sowa

Frege's Begriffsschrift for the Same Sentence

- Translation to Peirce-Peano notation:

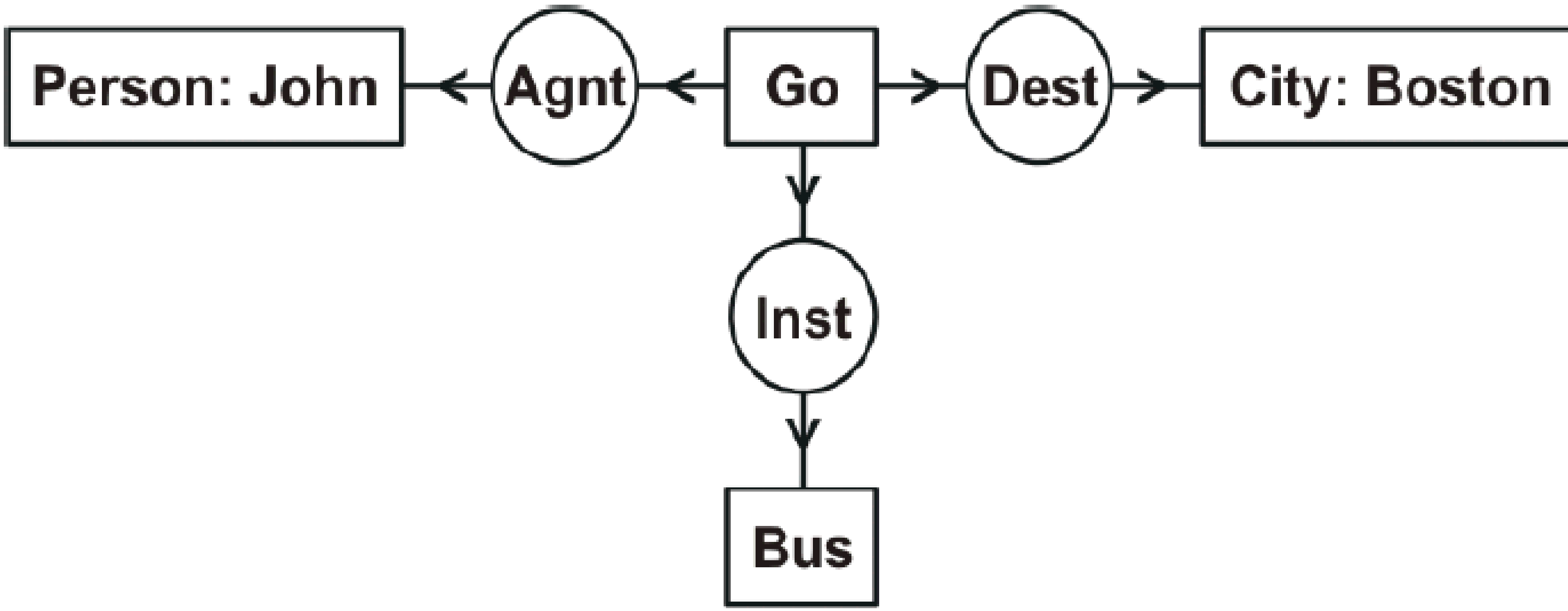
$$\sim(\forall x)(\forall y)(\mathrm{Go}(x) \supset (\mathrm{Person}(\mathrm{John}) \supset (\mathrm{City}(\mathrm{Boston}) \supset (\mathrm{Bus}(y) \supset (\mathrm{Agnt}(x,\mathrm{John}) \supset (\mathrm{Dest}(x,\mathrm{Boston}) \supset \sim\mathrm{Inst}(x,y)))))))$$

- 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*

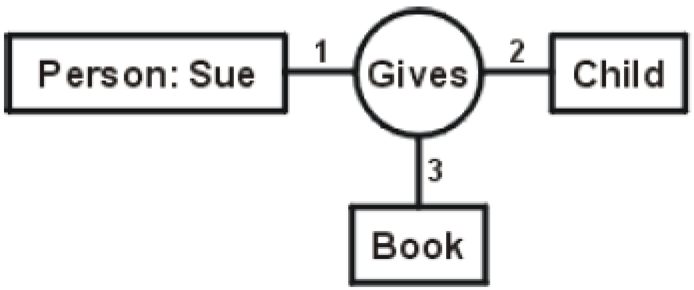


<u>Conceptual Graphs</u>

J. Sowa

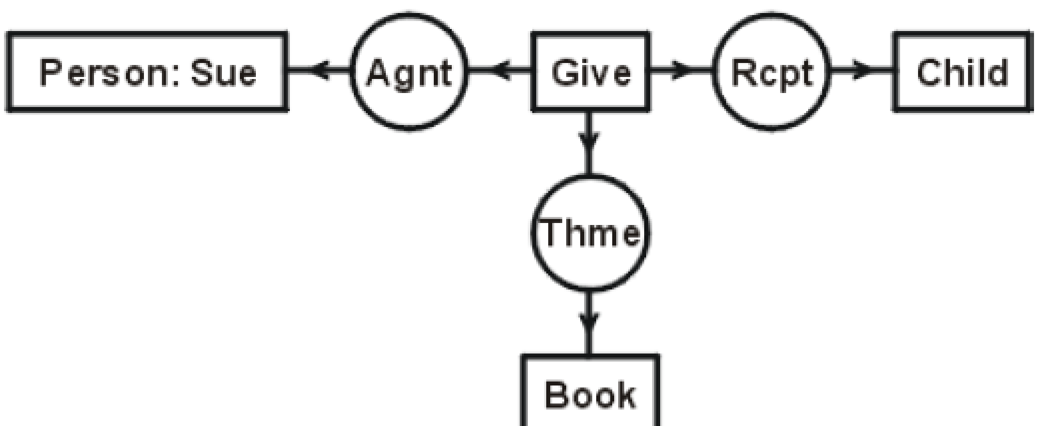
Existential Graphs

- Existential Graphs vs Conceptual Graphs



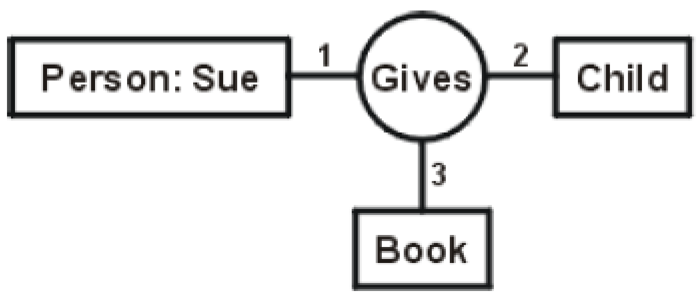
- The concept type **Give** is, in Peirce's terminology, a *hypostatic abstraction* of the relation type **Gives**.

- The idea of representing a verb by an entity that can be related by quantified variables is what Davidson called *event semantics*.



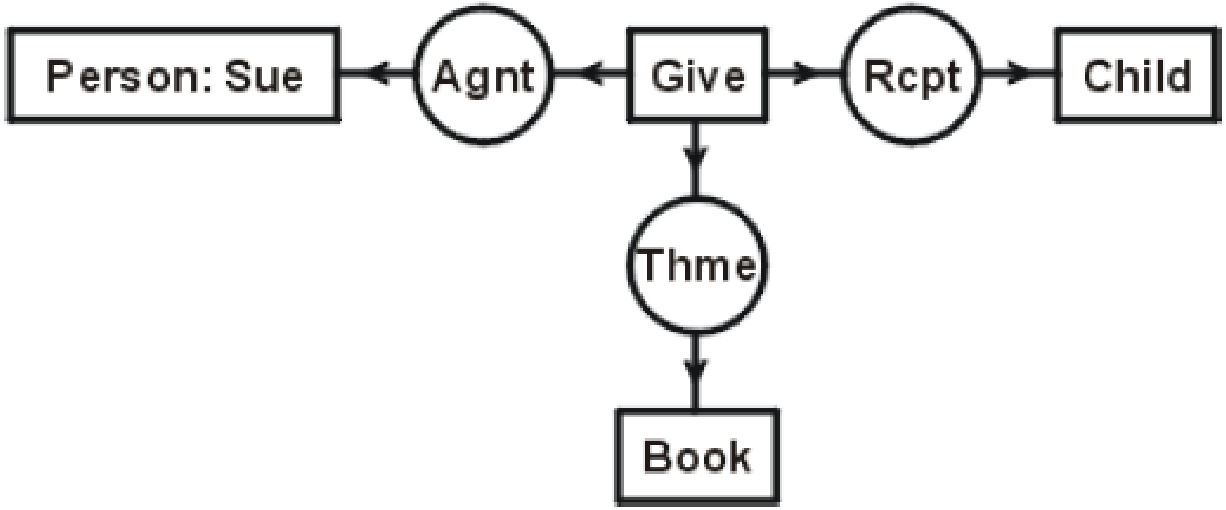
J. Sowa

- 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.



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

J. Sowa



- 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:

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

J. Sowa

EG & CG vs Algebraic Notation

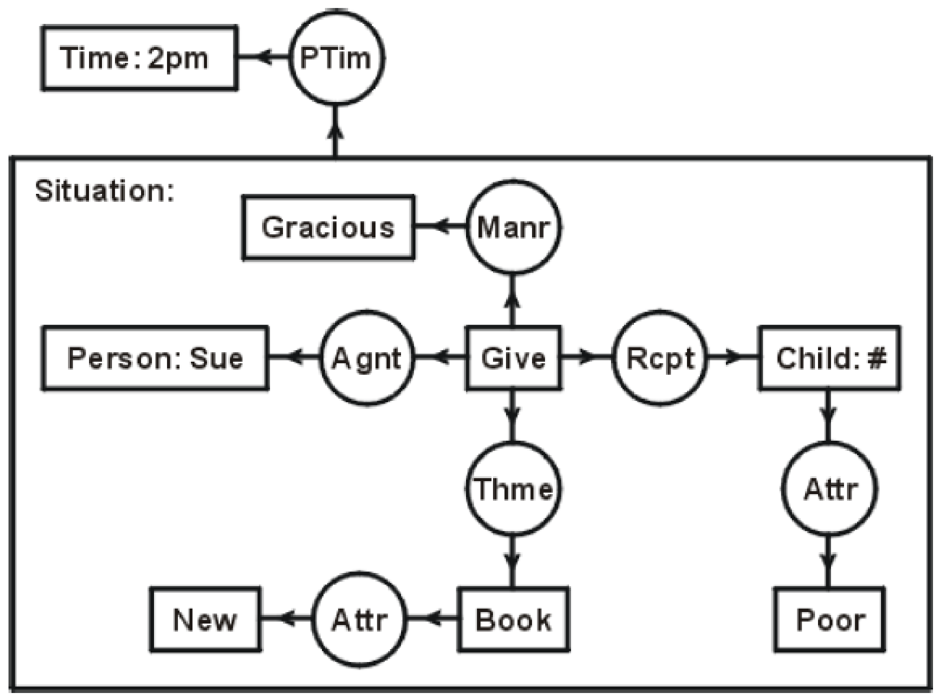
- As this example illustrates, the graph notation directly shows the *topology* of the logic, which is determined by the connectivity of the nodes and the cycles in the graph.

- 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.

J. Sowa

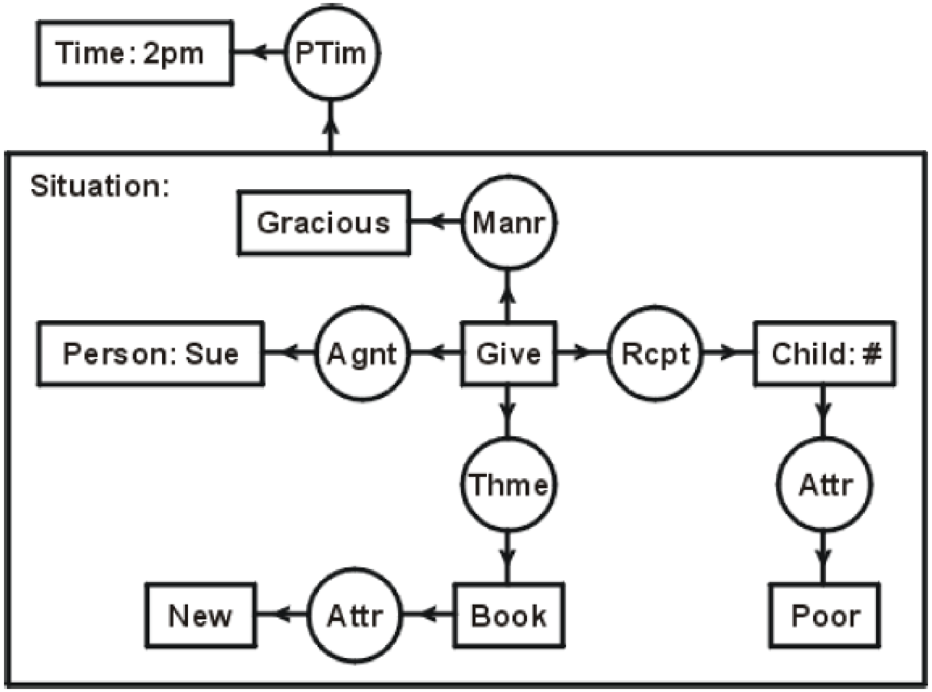
Sentence: *At 2 pm, Sue graciously gave the poor child a new book*.



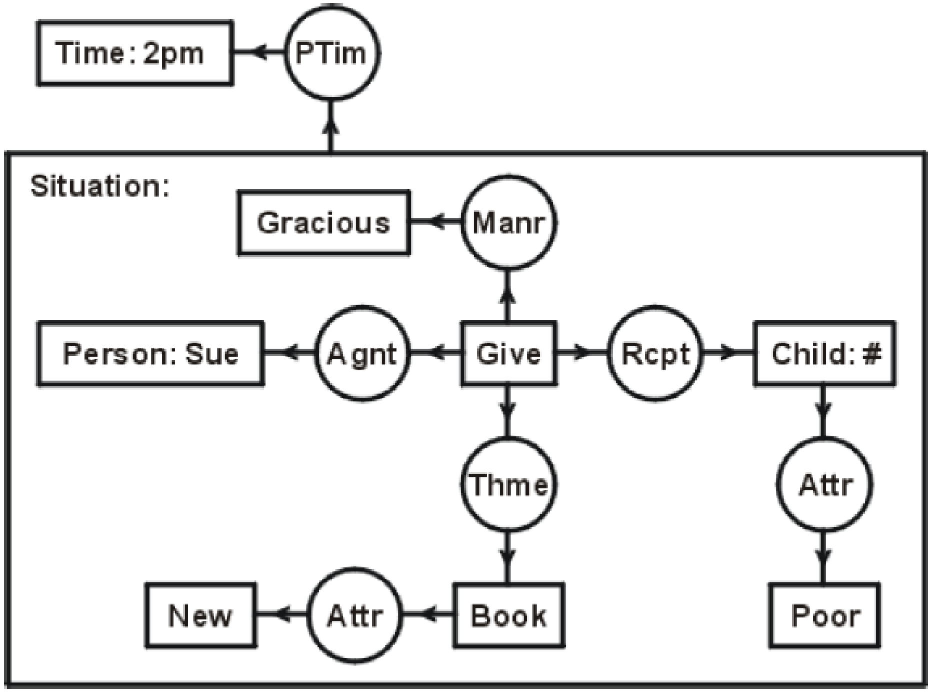
$$(\exists s)(\text{Situation}(s) \land \text{Time}(2pm) \land \text{PTim}(s,2pm) \land \text{dscr}(s, (\exists y)(\exists z)(\exists u)(\exists v)(\exists w)(\text{Person}(Sue) \land \text{Child}(Bob) \land \text{Book}(y) \land \text{Give}(z) \land \text{Gracious}(u) \land \text{Poor}(v) \land \text{New}(w)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

J. Sowa

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*.

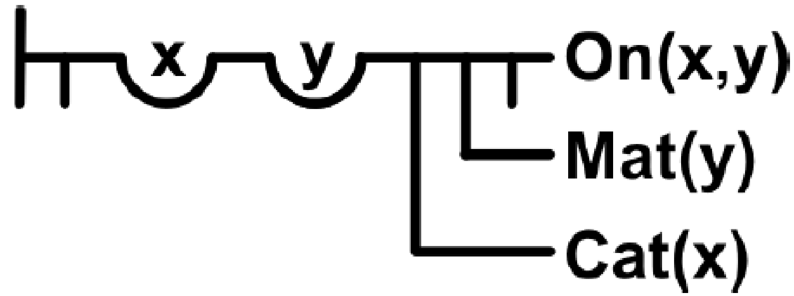


J. Sowa

- 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**

J. Sowa

J. Sowa

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

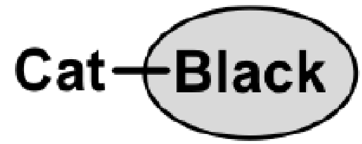
Some cat is not on a mat: Cat—On—Mat

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

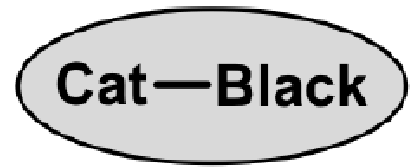
J. Sowa

The Scope of Quantifiers

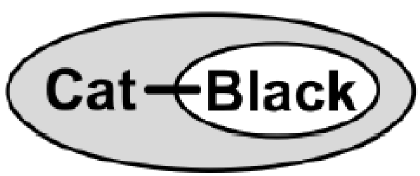
Some cat is black.



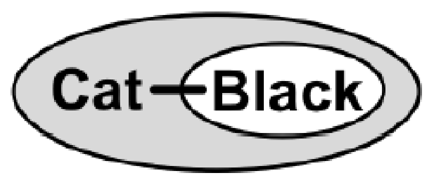
Some cat is not black.



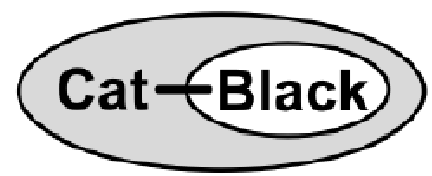
No cat is black.



It is false that some cat is not black.



If there is a cat, then it is 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.

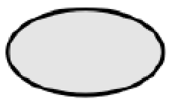
On some non-blue table, there is something red that is not a ball.

J. Sowa

Existential Graph Interchange Format

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

Existence: **—** [*x]

Negation:  ~[ ]

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

***A cat is on a mat*:** [*x] [*y] (Cat ?x) (On ?x ?y) (Mat ?y)

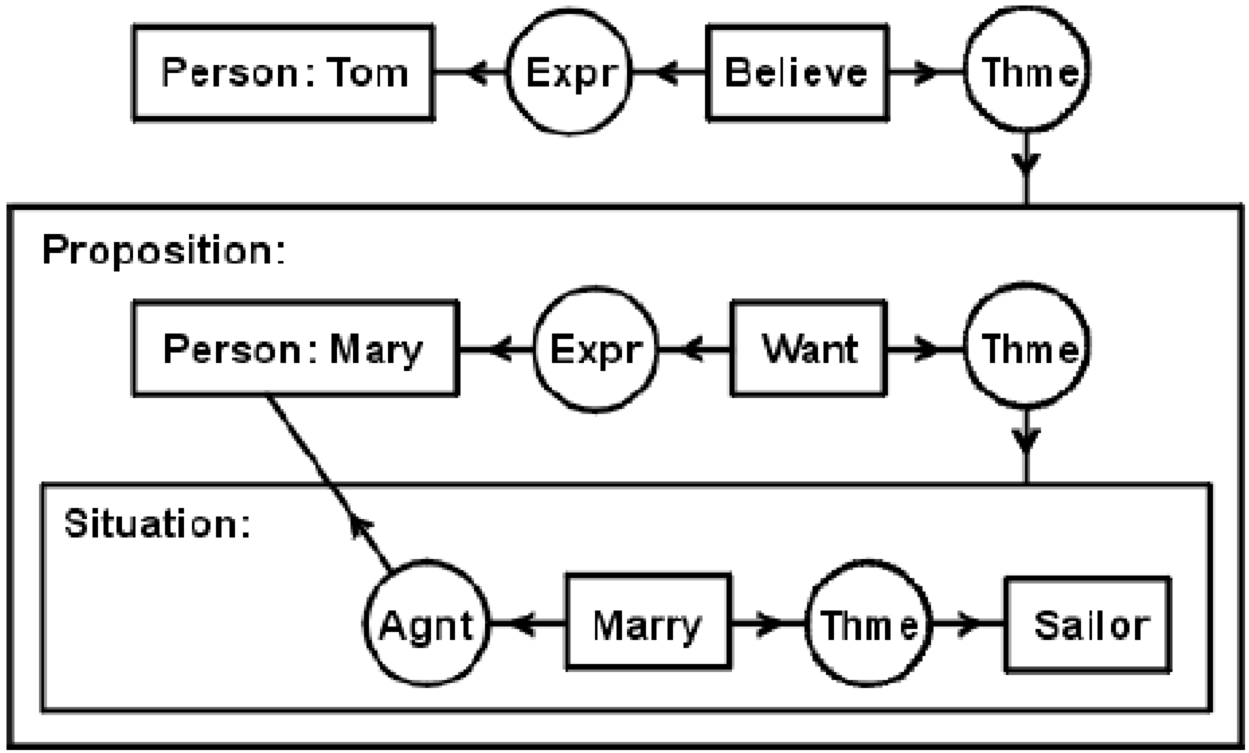
***Something is under a mat*:** [*x] [*y] (Under ?x ?y) (Mat ?y)

***Some cat is not on a mat*:** [*x] (Cat ?x) ~[[*y] (On ?x ?y) (Mat ?y)]

***Some cat is on something that is not a mat*:**
[*x] [*y] (Cat ?x) (On ?x ?y) ~[(Mat ?y)]

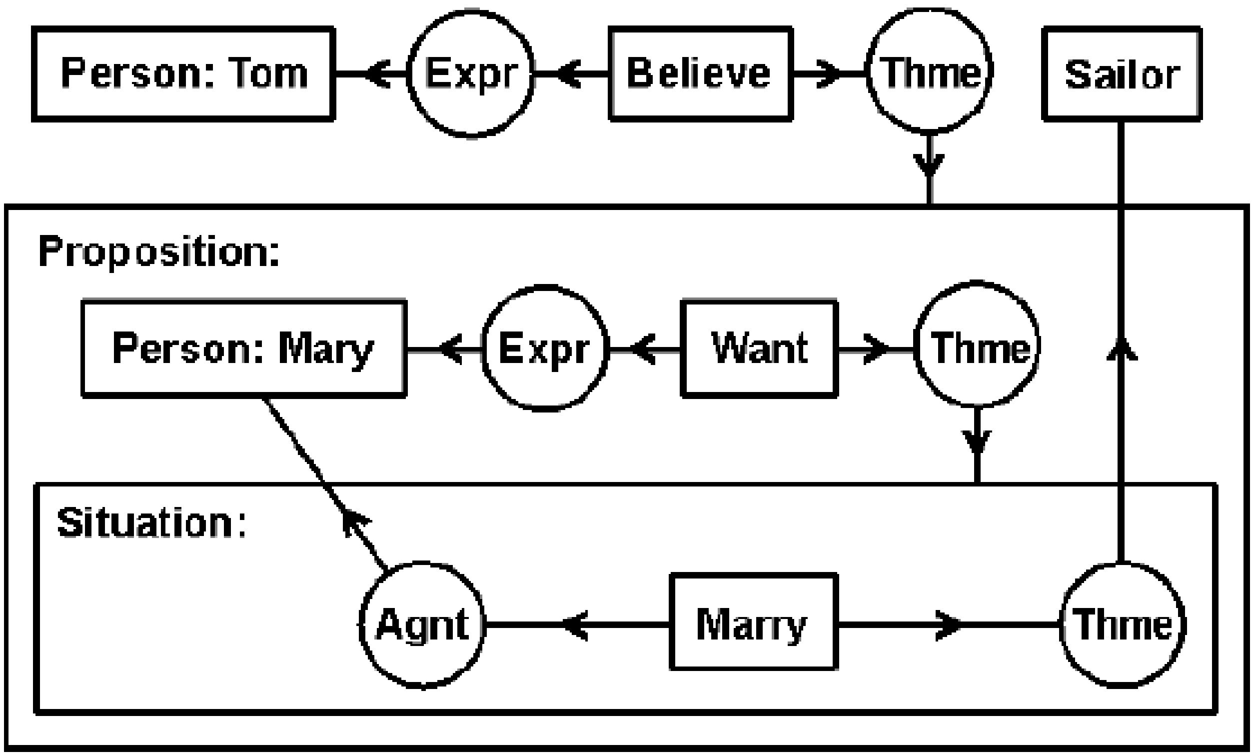
J. Sowa

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



J. Sowa

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



J. Sowa

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. **She** smokes.

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

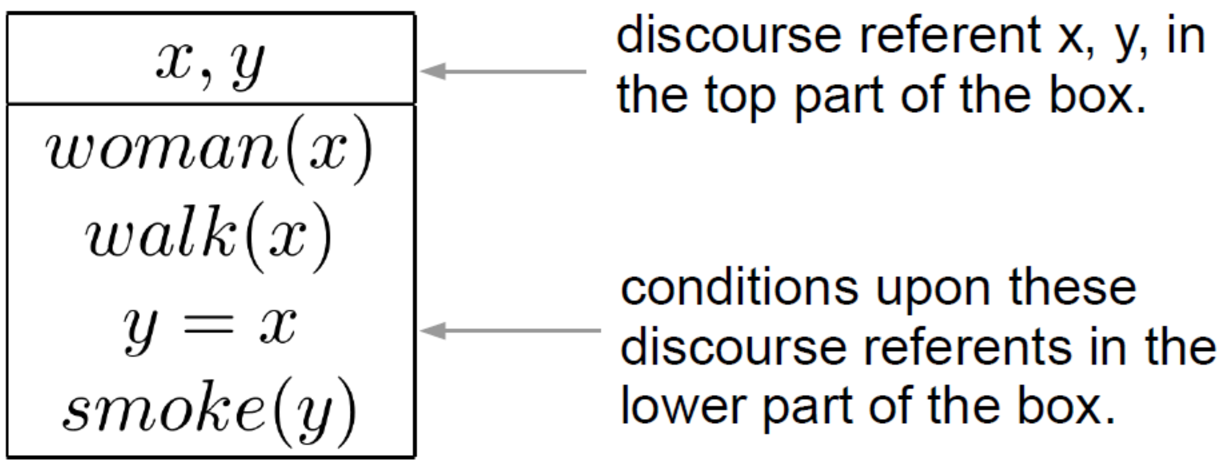
$$smoke(x)$$

$$\exists x(\underline{woman(x) \wedge walk(x) \wedge smoke(x)})$$

Need to Expand the scope of the existential quantifier.

A woman walks. **She** smokes.

- **A woman** walks. **She** smokes.



Discourse Structure and Accessibility

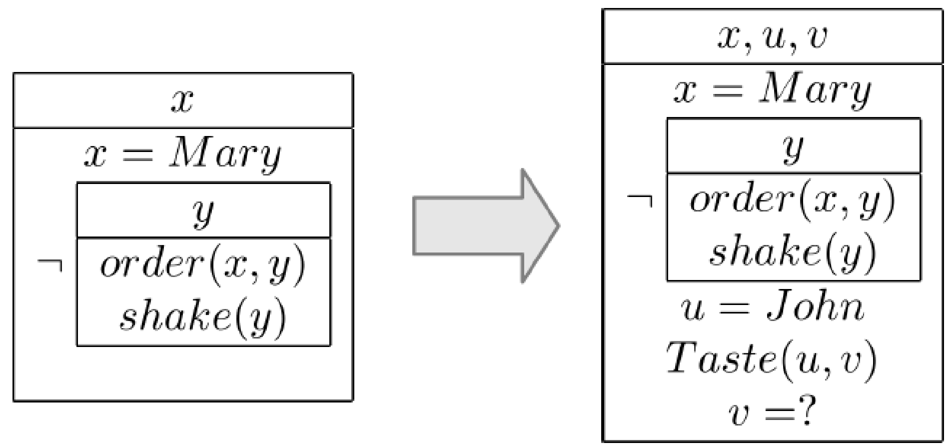
- Mary ordered a milk shake, John tasted **it**.

$$
\begin{array}{|c|}
\hline
x, y \\
\hline
x = Mary \\
order(x, y) \\
shake(y) \\
\hline
\end{array}
\quad \Rightarrow \quad
\begin{array}{|c|}
\hline
x, y, u, v \\
\hline
x = Mary \\
order(x, y) \\
shake(y) \\
u = John \\
taste(u, v) \\
v = y \\
\hline
\end{array}
$$

- 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.**

$$
\begin{array}{|c|}
\hline
\\
\hline
\begin{array}{|c|}
\hline
x, y \\
\hline
farmer(x) \\
donkey(y) \\
own(x, y) \\
\hline
\end{array}
\Rightarrow
\begin{array}{|c|}
\hline
z \\
\hline
beat(x, z) \\
z = y \\
\hline
\end{array}
\\
\\
\hline
\end{array}
$$

- 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

Taboada

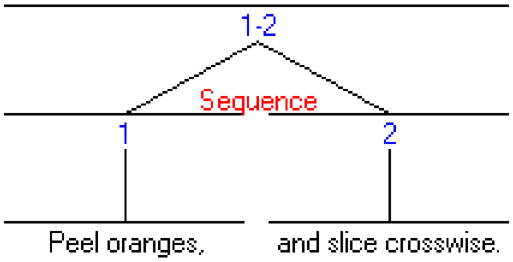
Components

- 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.

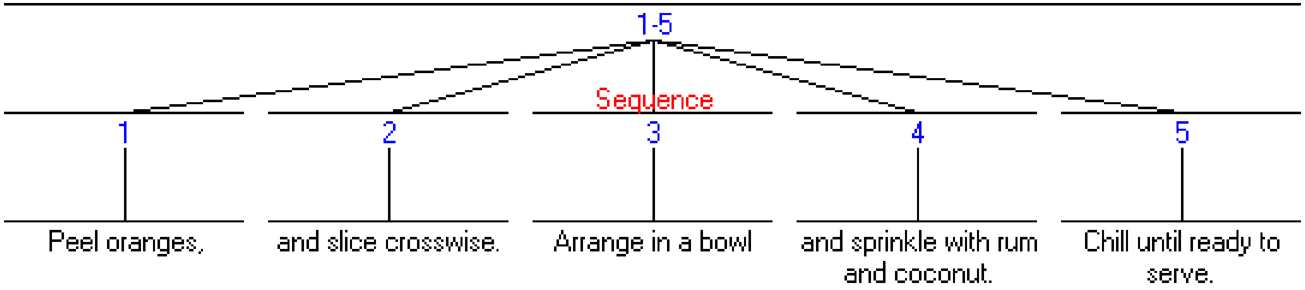
Taboada

Paratactic (coordinate)

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

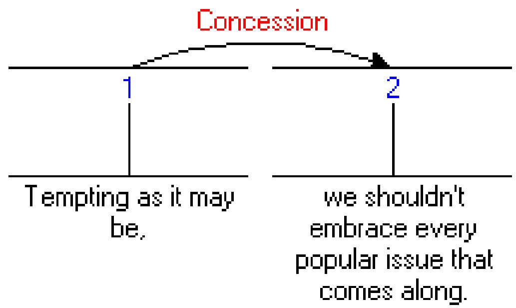


- But also across sentences
  - 1. Peel oranges, 2. and slice crosswise. 3. Arrange in a bowl 4. and sprinkle with rum and coconut. 5. Chill until ready to serve.

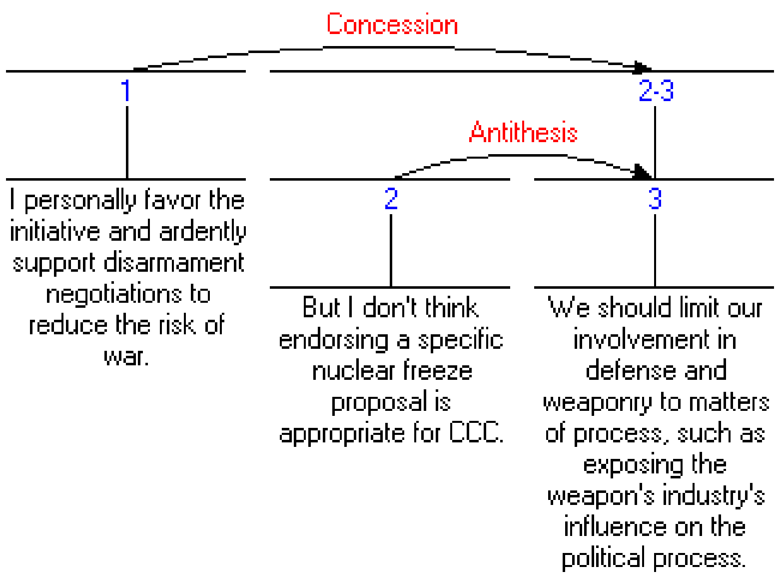


Hypotactic (subordinate)

- 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

Taboada

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

Taboada

Relation names (in M&T 1988)

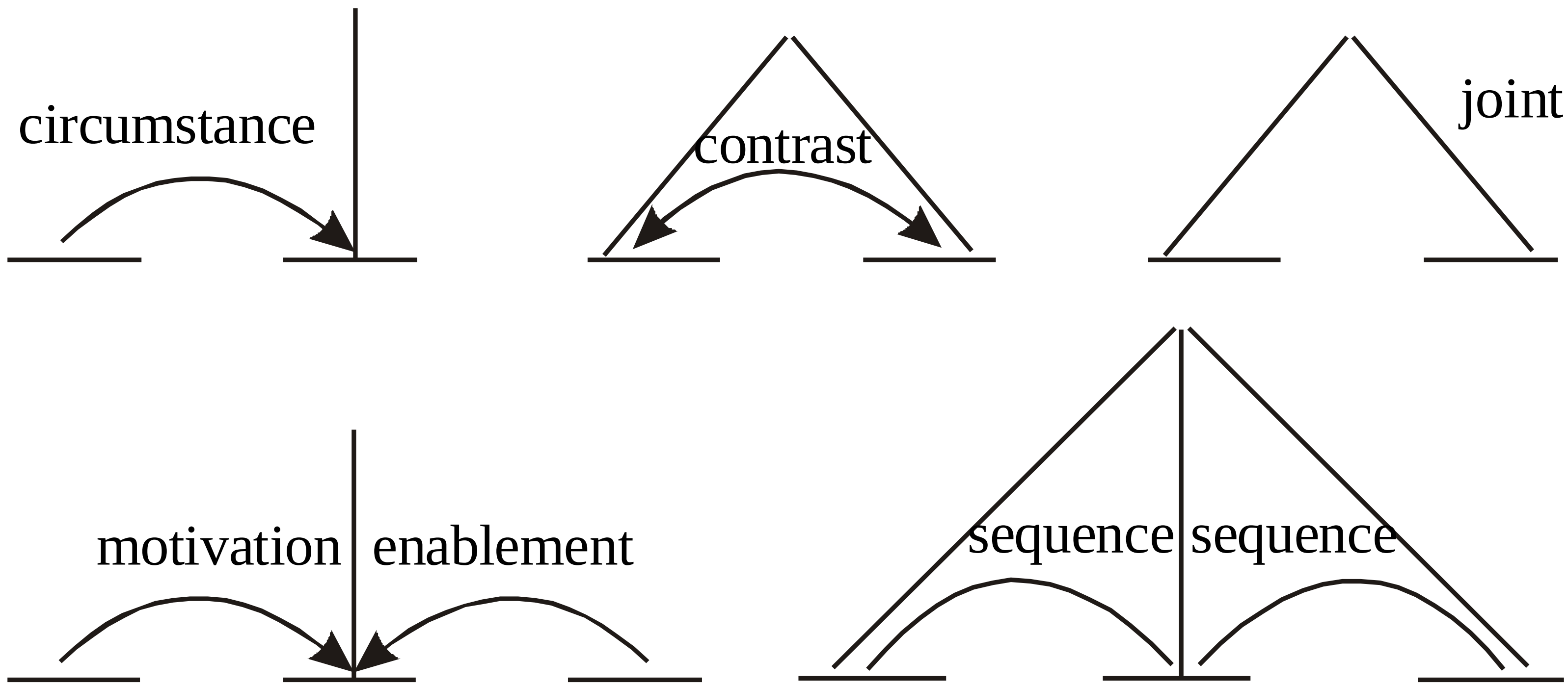
Circumstance	**Antithesis and Concession**
Solutionhood	**Antithesis**
Elaboration	**Concession**
Background	**Condition and Otherwise**
Enablement and Motivation	**Condition**
Enablement	**Otherwise**
Motivation	**Interpretation and Evaluation**
Evidence and Justify	**Interpretation**
Evidence	**Evaluation**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

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

Taboada

Schemas

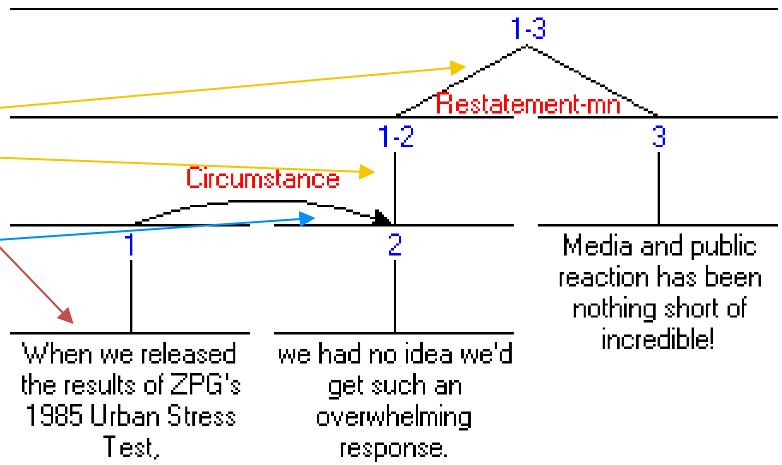
- They specify how spans of text can co-occur, determining possible RST text structures



Taboada

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



Taboada

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.

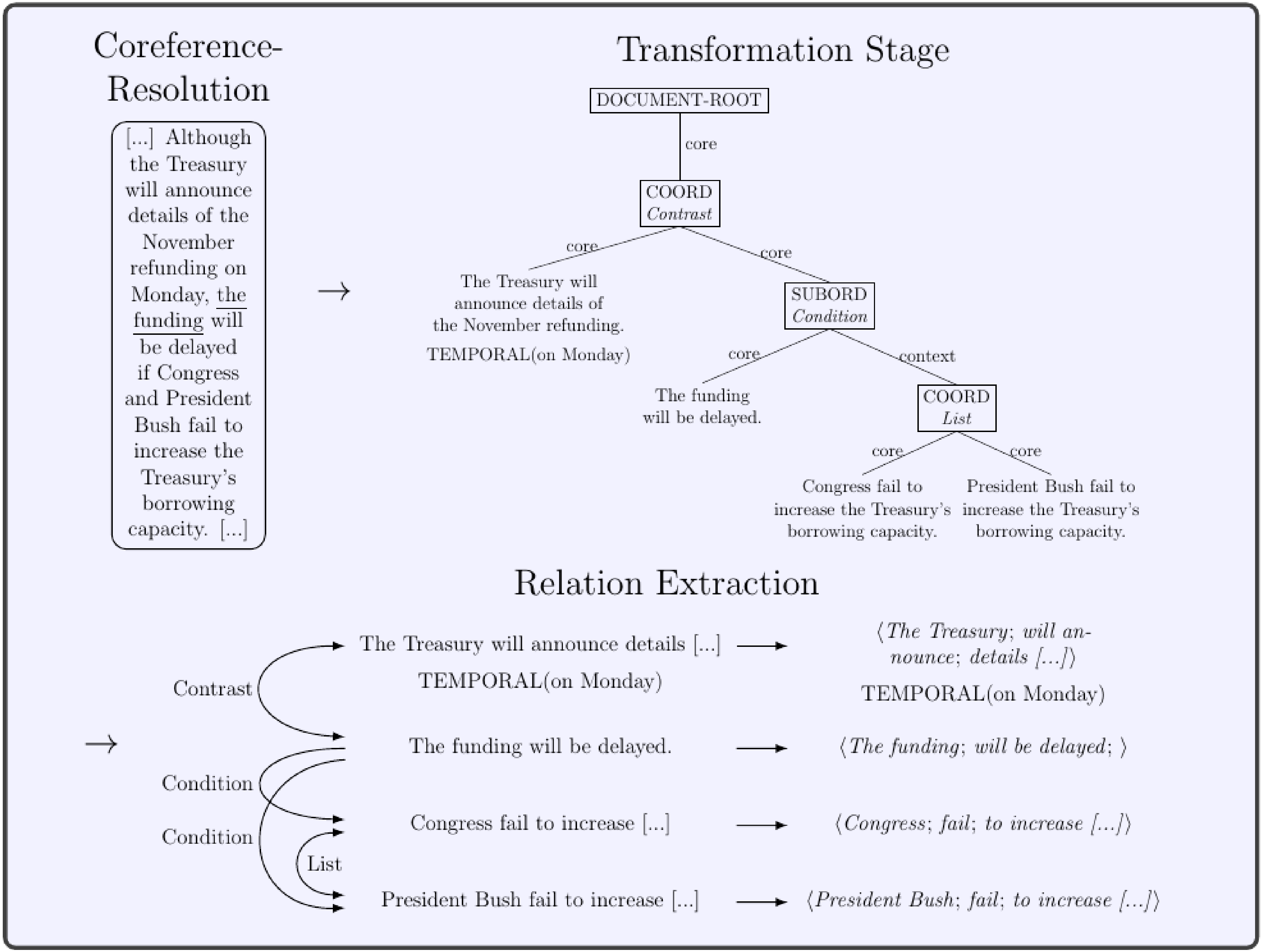
Taboada

Putting All Together

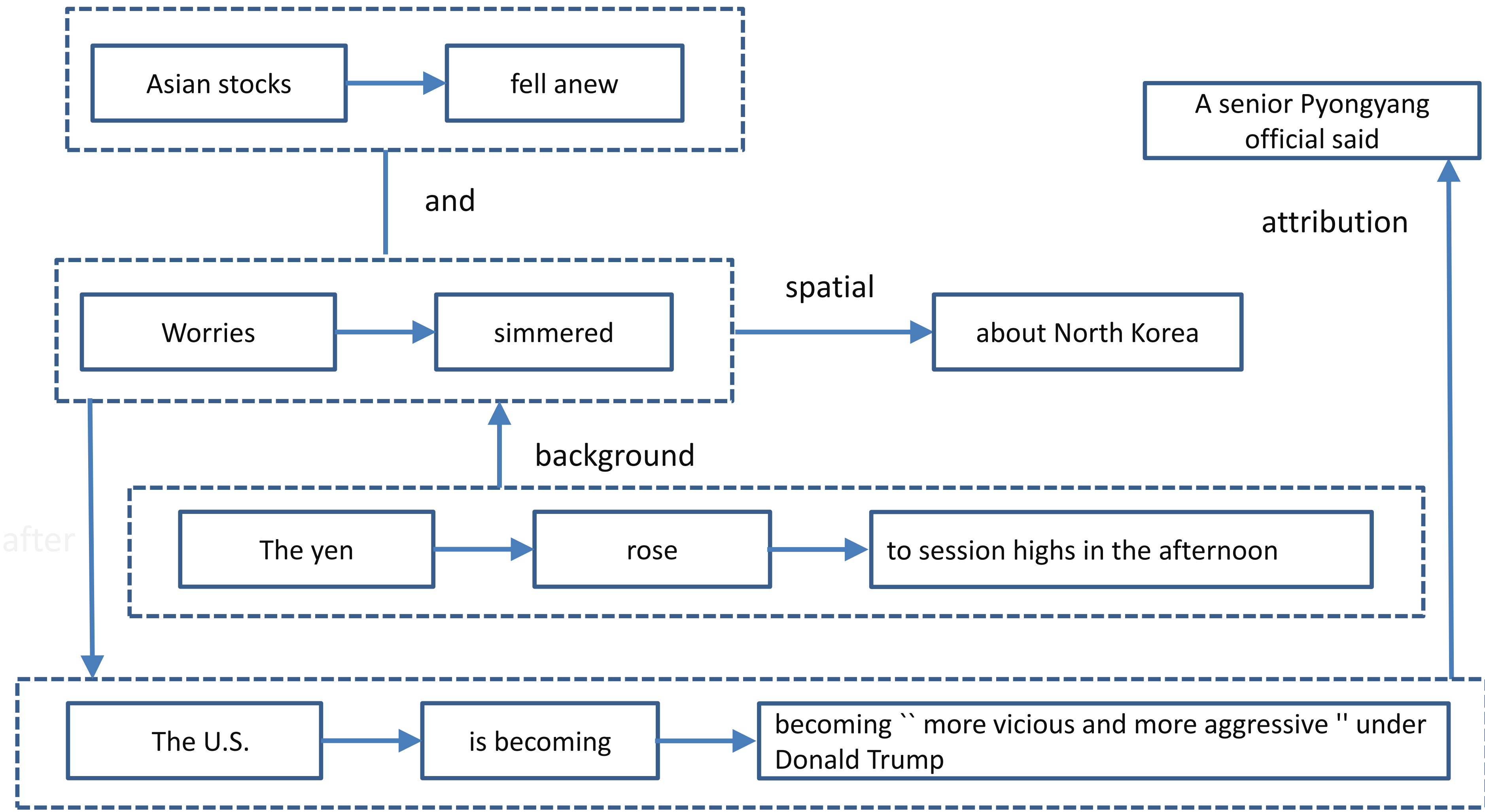
Software: Extracting Knowledge Graphs from Text

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
DOCUMENT-ROOT
core
COORD Contrast
core
core
The Treasury will announce details of the November refunding.
TEMPORAL(on Monday)
SUBORD Condition
core
context
The funding will be delayed.
COORD List
core
core
Congress fail to increase the Treasury's borrowing capacity.
President Bush fail to increase the Treasury's borrowing capacity.
Relation Extraction
Contrast
Condition
Condition
List
The Treasury will announce details [...]
TEMPORAL(on Monday)
The funding will be delayed.
Congress fail to increase [...]
President Bush fail to increase [...]
⟨The Treasury; will announce; details [...]⟩
TEMPORAL(on Monday)
⟨The funding; will be delayed; ⟩
⟨Congress; fail; to increase [...]⟩
⟨President Bush; fail; to increase [...]⟩

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 .

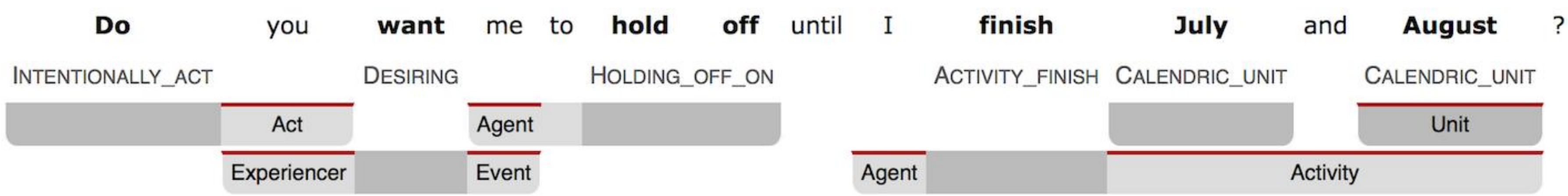
bacf06771e0f4fc5a8e68c30fc77c9c4 the Treasury will announce details of the November refunding
S:TEMPORAL on Monday .
L:CONTRAST 948eeebd73564adab7dee5c6f177b3b9

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

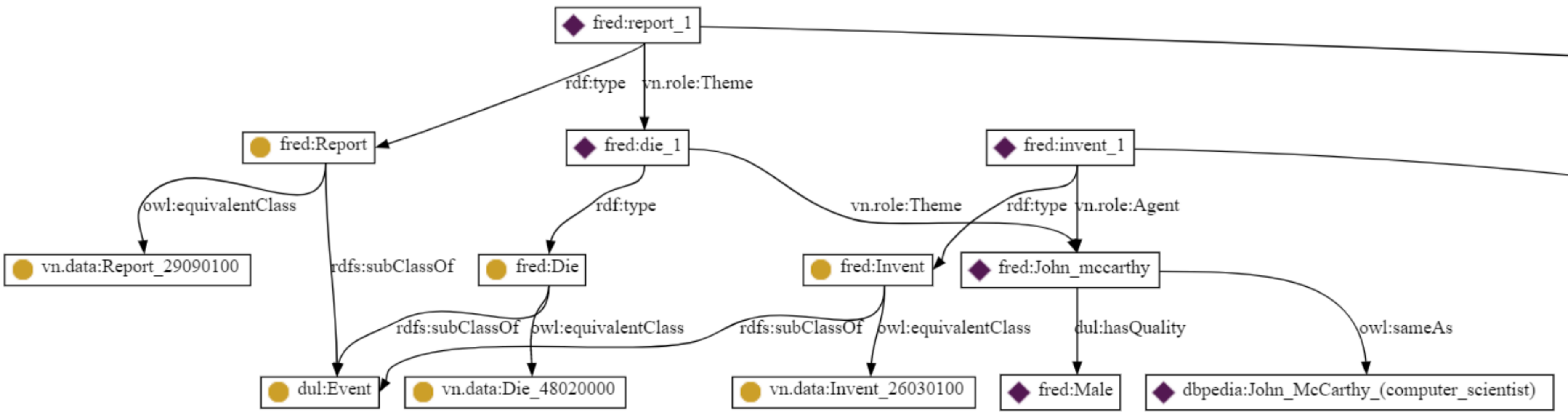
Semafor



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

Frame-based Extraction

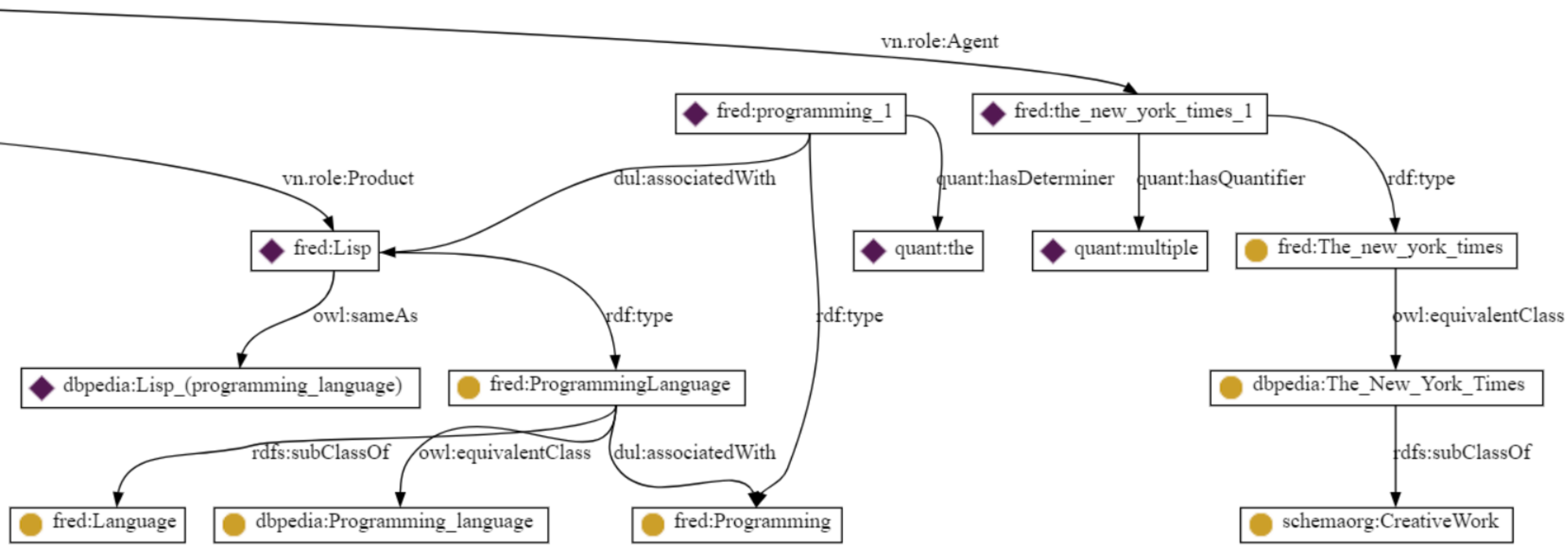
The New York Times reported that John McCarthy died. He invented the programming language LISP.



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

Frame-based Extraction

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



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

Software: FRED

FRED

Machine Reading for the Semantic Web

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

Gangemini 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

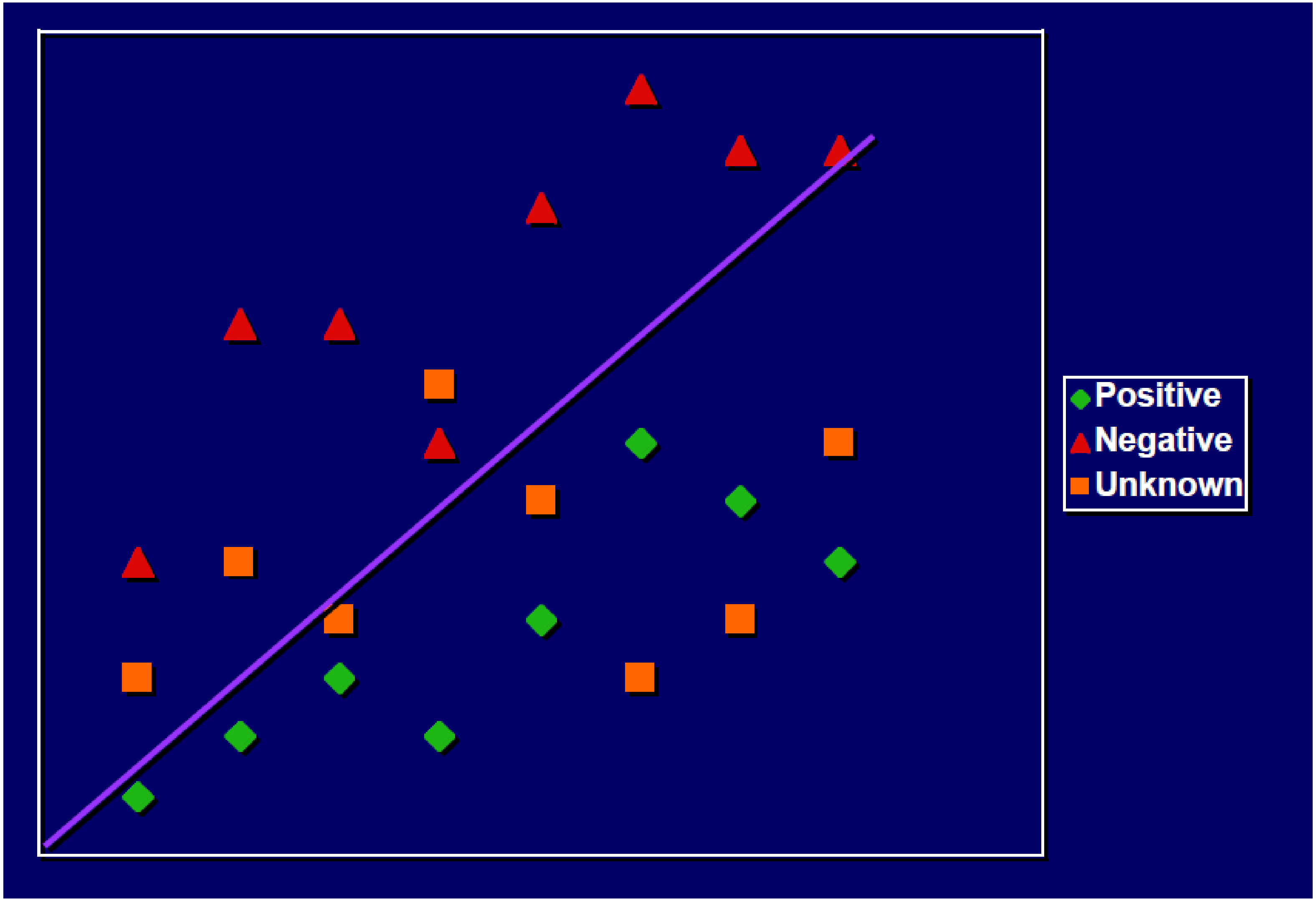
- These slides were based on the slides of:
  - Peter A. Flach, Rule induction tutorial, IDA Spring School 2001.
  - Anoop & Hector, Inductive Logic Programming (for Dummies).
  - Gabor Melli, Scribe Notes on FOIL and Inverted Deduction.
  - CS 5751 Machine Learning, Chapter 10 Learning Sets of Rules.

This Lecture

- Introduction to Inductive Logic Programming
- FOIL

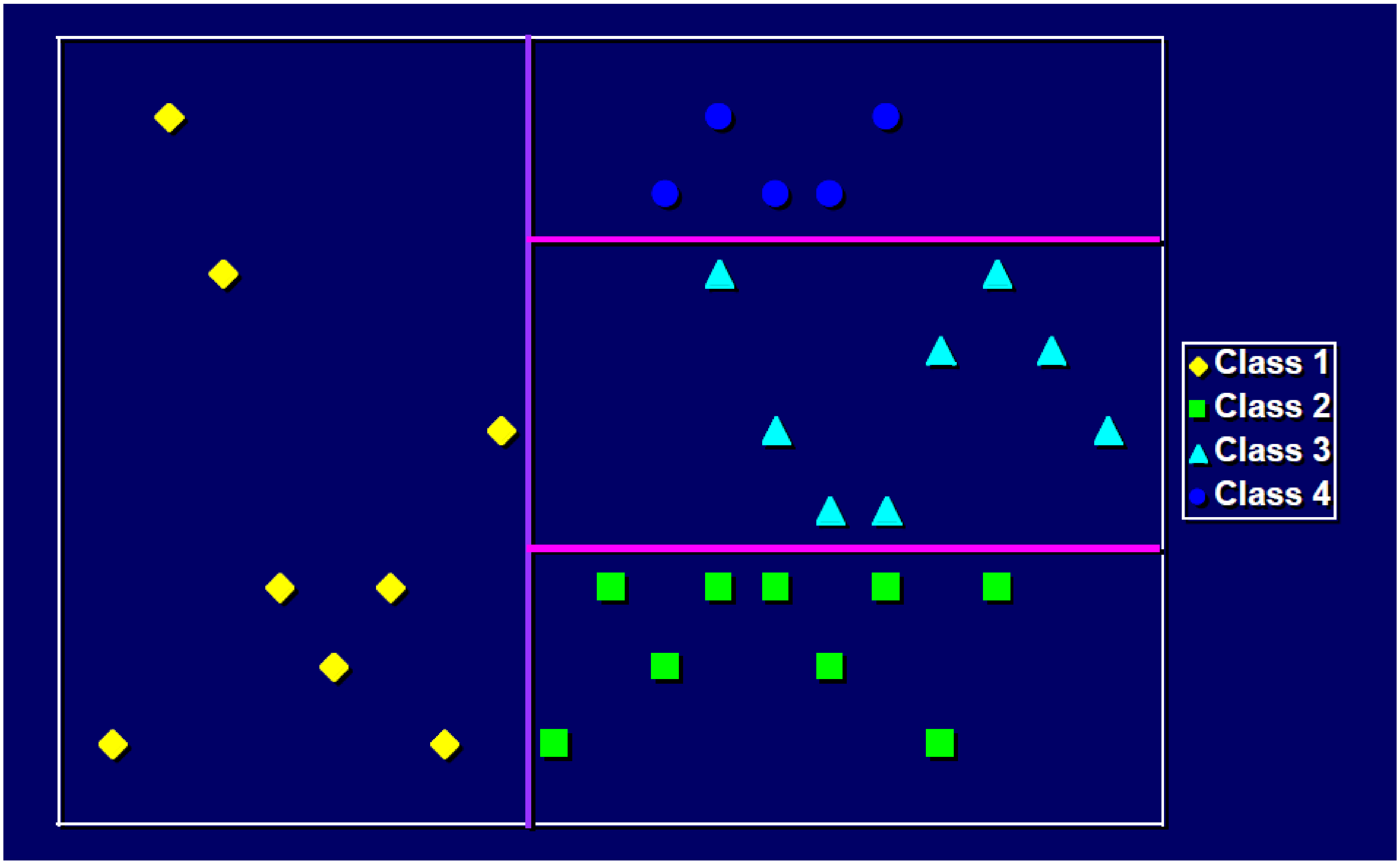
Flach 2001

Linear Classifier



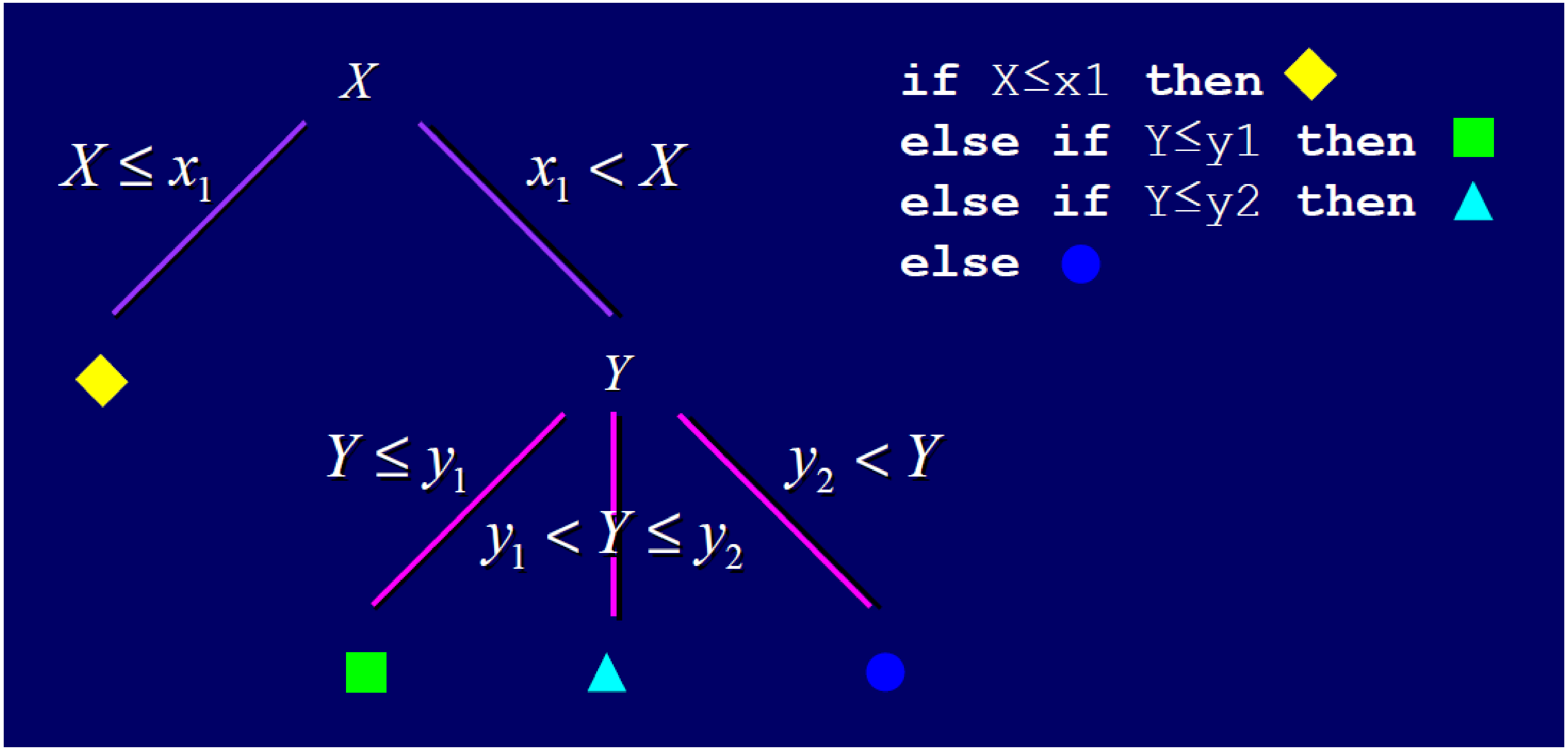
Flach 2001

Decision Trees



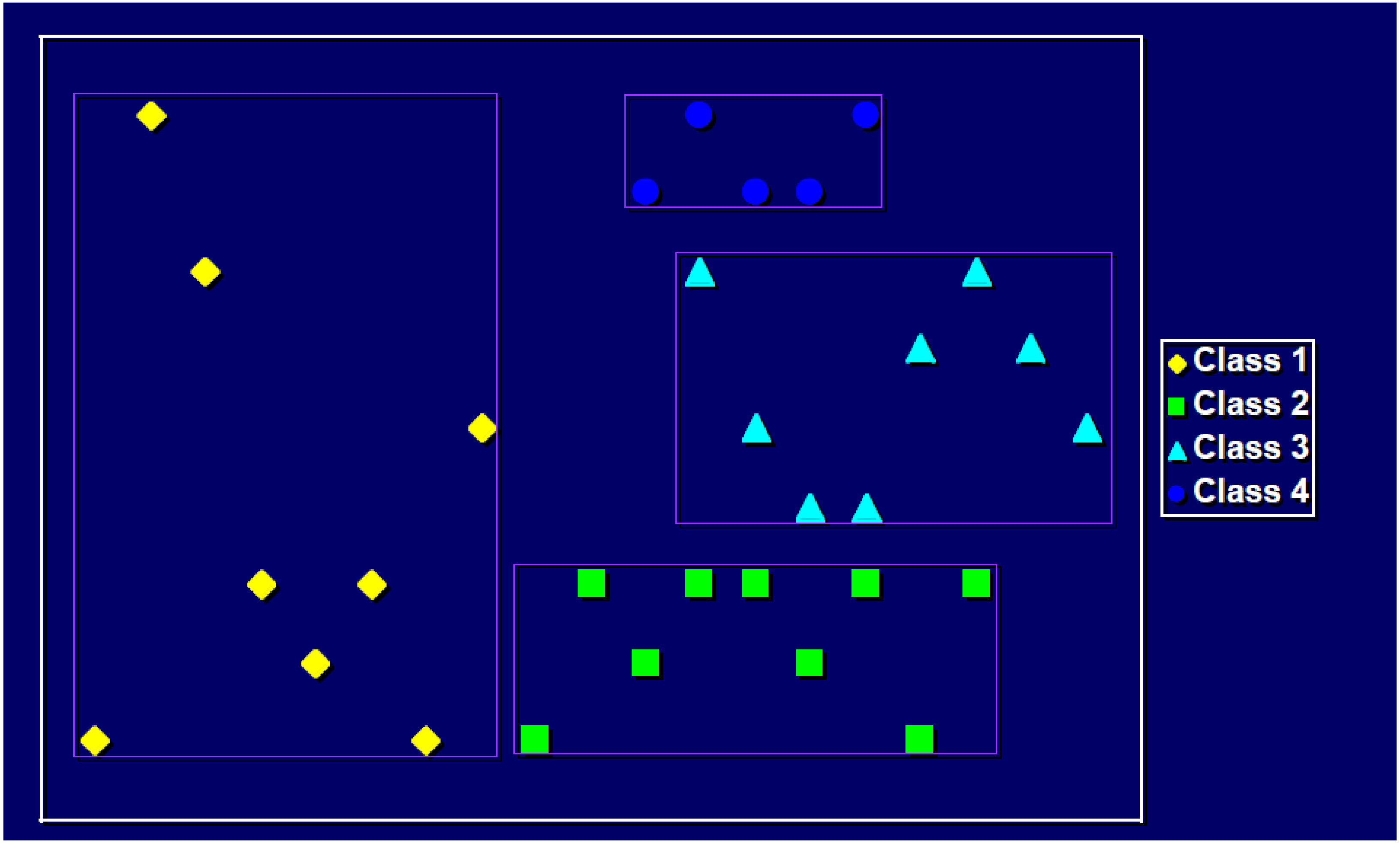
Flach 2001

Decision Trees



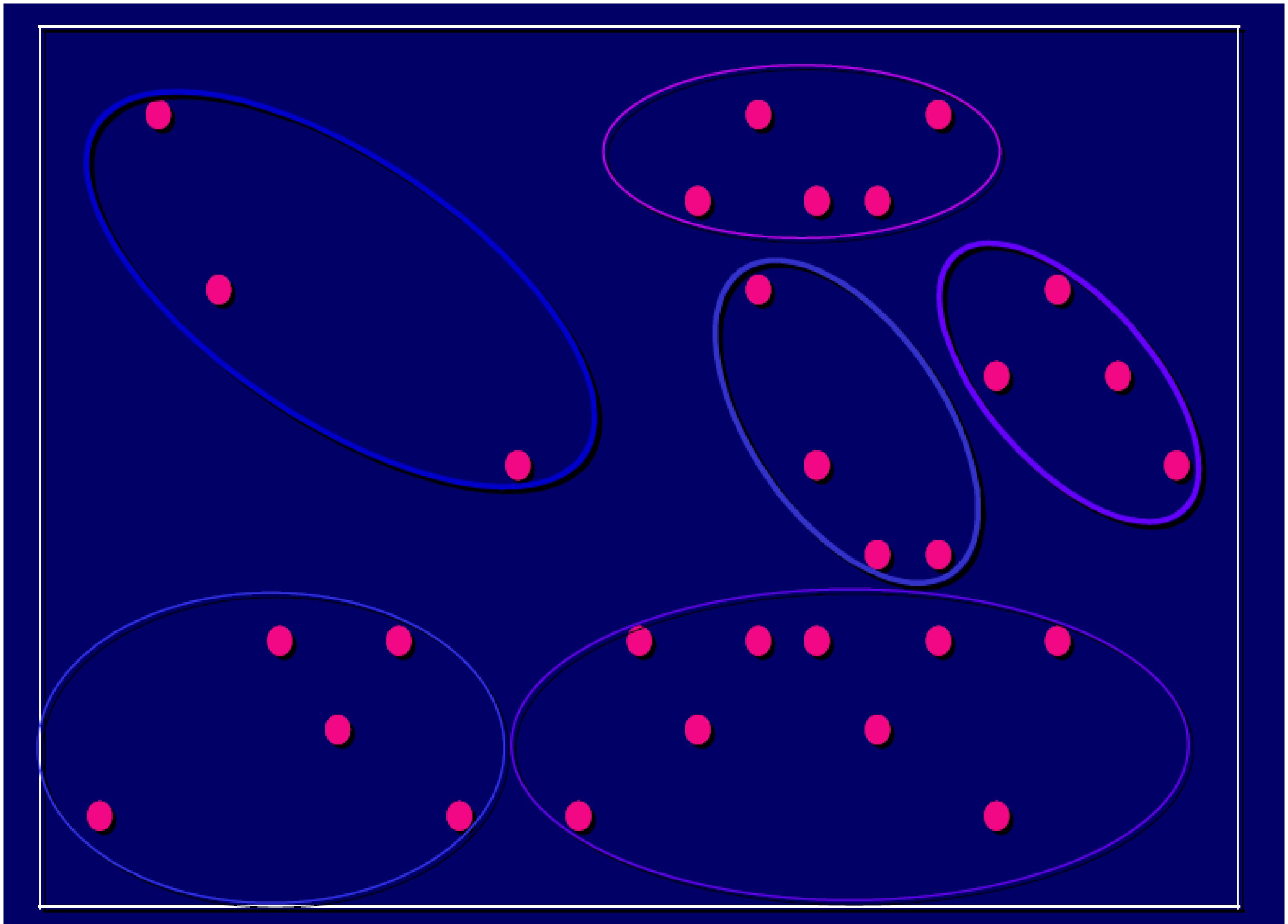
Flach 2001

Rules



Flach 2001

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.

Flach 2001

First-order Representations

- **<u>Propositional</u>** representations:
  - datacase is ***fixed-size vector of values***
  - features are those given in the dataset

- **<u>First-order</u>** 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

Flach 2001

Deductive Vs Inductive Reasoning

$T \cup B \rightarrow E$ **(deduce)**

T	B	E
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).

$E \cup B \rightarrow T$ **(induce)**

E	B	T
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

TRAIN

Visible

Father	
a_1	a_2
Sharon	Bob
Tom	Bob
Bob	Victor
…	…

Female
a_1
Sharon
…

GrandDaughter	
a_1	a_2
Victor	Sharon
…	…

TEST

Visible

Father	
a_1	a_2
Julie	John
John	Howard

Female
a_1
Julie

Hidden

GrandDaughter	
a_1	a_2
Howard	Julie

Melli

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
Father	***Child***	***Child is Fem.***	***Has Gdaugh***
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
Father	*Child*	*Child is Fem.*	*Has Gdaugh*
Bob	Sharon	TRUE	FALSE
Victor	Bob	FALSE	TRUE
…	…	…	…

Melli

Propositional Leaner 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 <u>to transform the data so that the required relations appear as attributes in the data</u>.

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

Predictors					Target
Father	***Child***	***Child is Fem.***	***Child's Child***	***C's C is Fem.***	***Has Gdaugh***
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

Melli

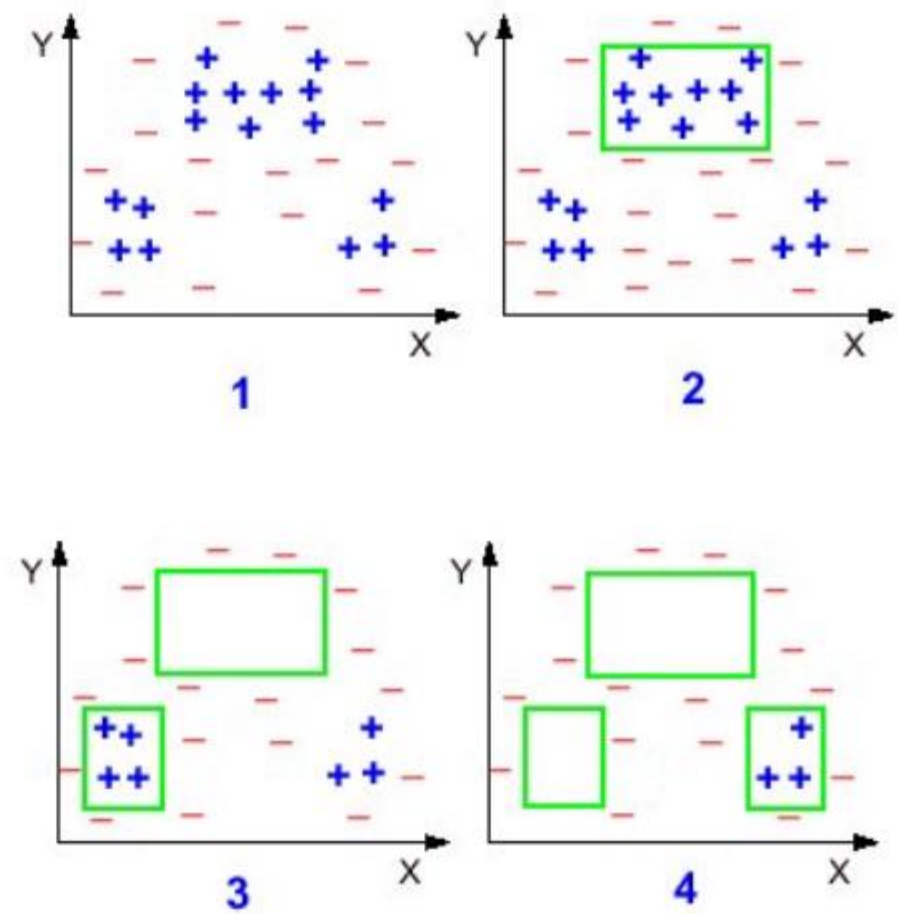
Propositional Sequential Covering

- 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:

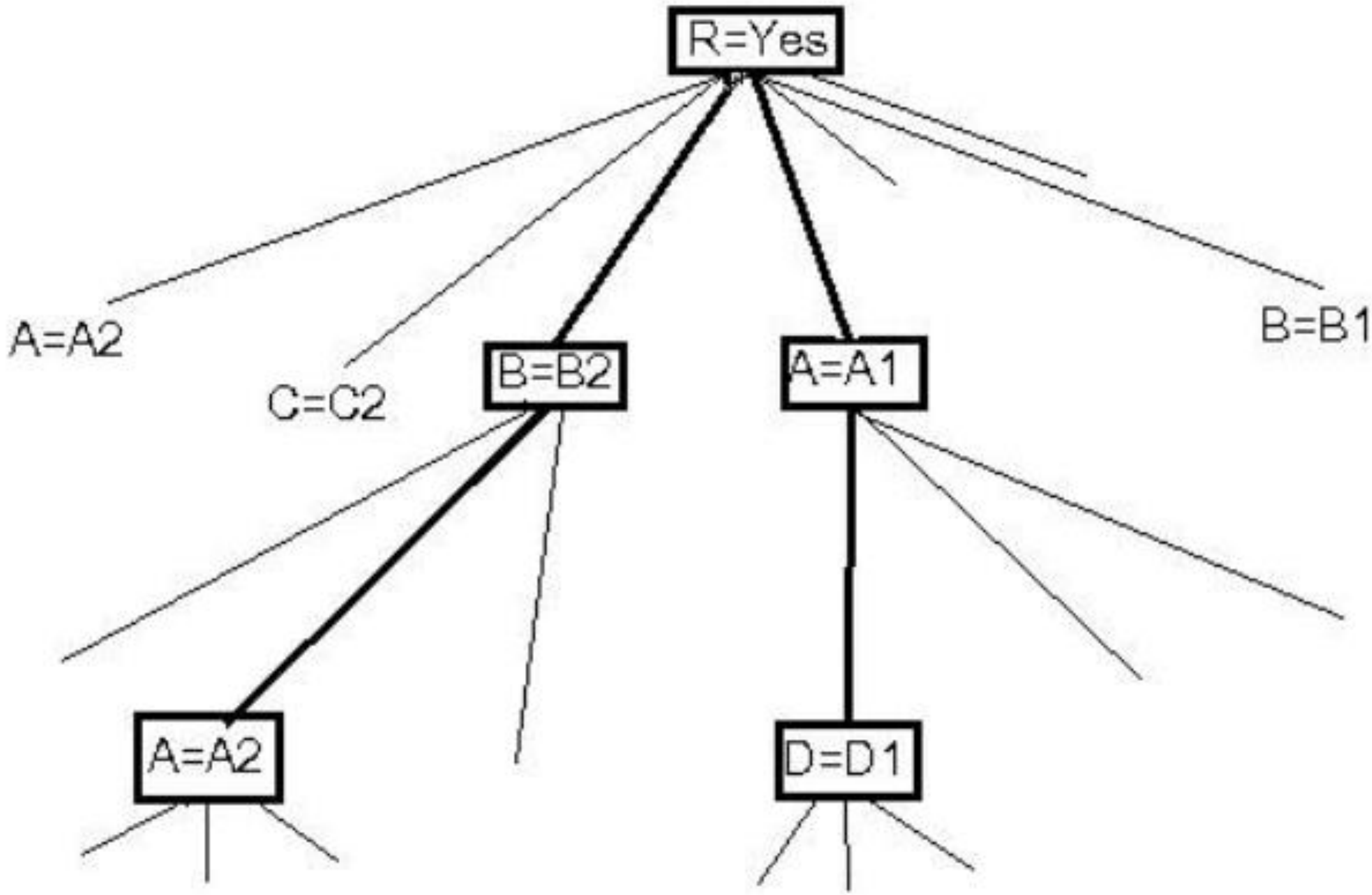
Colour=Green & Shape=Square	(specialising Colour=Green or Shape=Square)
Colour=Blue & Weight=Heavy	(specialising Colour=Blue or Weight=Heavy)

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



Melli

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.

Melli

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).

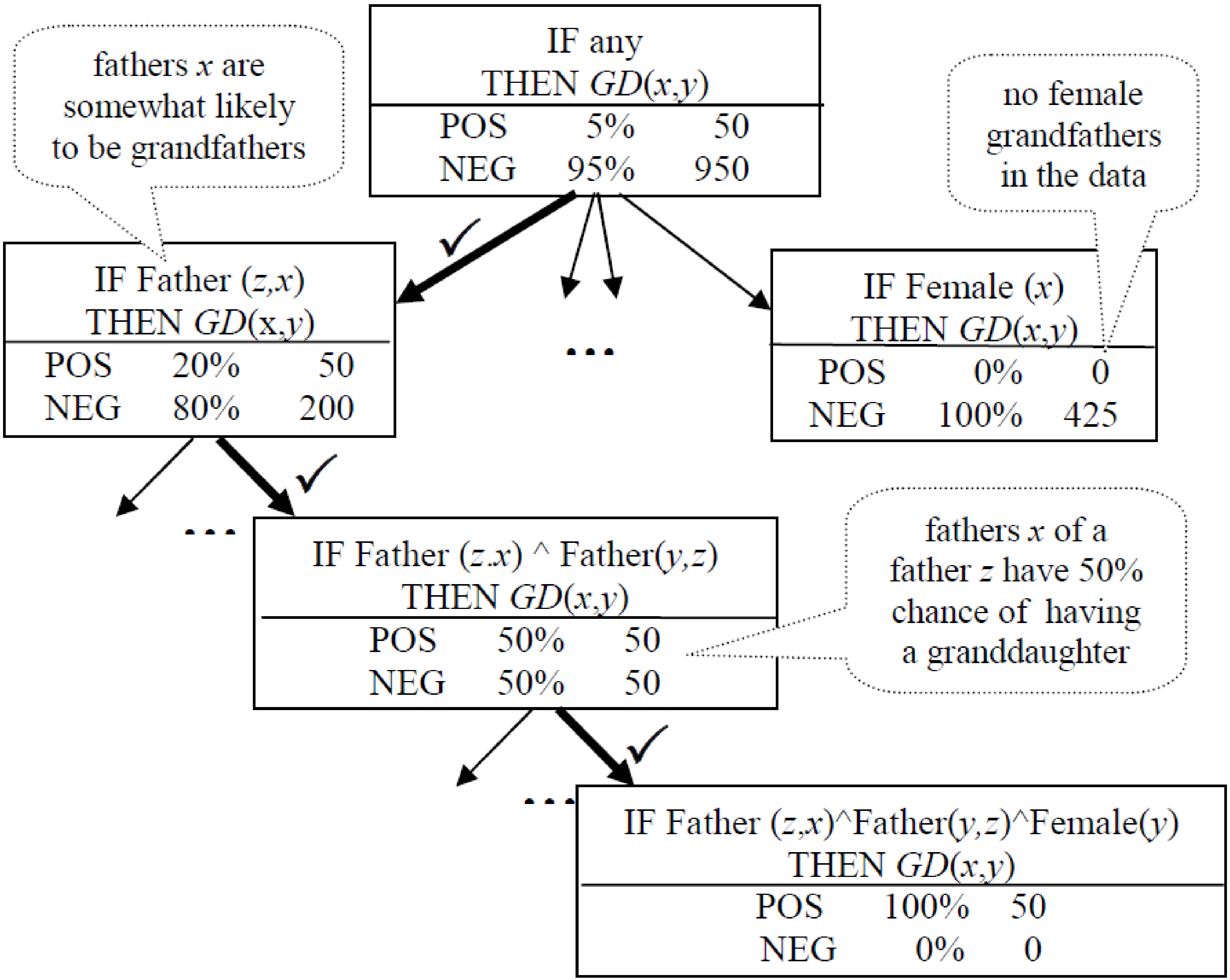
Melli

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.

Melli

Back to the Example


IF any
THEN GD(x,y)
POS 5% 50
NEG 95% 950
fathers x are somewhat likely to be grandfathers
IF Father (z,x)
THEN GD(x,y)
POS 20% 50
NEG 80% 200
no female grandfathers in the data
IF Female (x)
THEN GD(x,y)
POS 0% 0
NEG 100% 425
IF Father (z.x) ^ Father(y,z)
THEN GD(x,y)
POS 50% 50
NEG 50% 50
fathers x of a father z have 50% chance of having a granddaughter
IF Father (z,x)^Father(y,z)^Female(y)
THEN GD(x,y)
POS 100% 50
NEG 0% 0

Melli

FOIL

FOIL(*Target_predicate*,*Predicates*,*Examples*)
Pos ← positive *Examples*
Neg ← negative *Examples*
while *Pos* do (*Learn a New Rule*)
    NewRule ← most general rule possible
    NegExamplesCovered ← *Neg*
    while *NegExamplesCovered* do
        Add a new literal to specialize *NewRule*
        1. *Candidate_literals* ← generate candidates
        2. *Best_literal* ← $\text{argmax}_{L \in candidate\_literal}$ FOIL_GAIN(*L*,*NewRule*)
        3. Add *Best_literal* to *NewRule* preconditions
        4. *NegExamplesCovered* ← subset of *NegExamplesCovered* that satistifies *NewRule* preconditions
    Learned_rules ← *Learned_rules* + *NewRule*
    Pos ← *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-CONVERING algorithm, the inner to FIND-A-RULE

Specialising Rules in FOIL

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

Candidate specializations add new literal of form:

- $Q(v_1, \ldots, v_r)$, 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) $\leftarrow$
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

Flach 2001

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.

Flach 2001

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 <u>sname</u>, string major, string minor)
  - Course(string <u>cname</u>, string prof, string cred)
  - Enrolled(string <u>sname,</u> string <u>cname</u>)

- 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

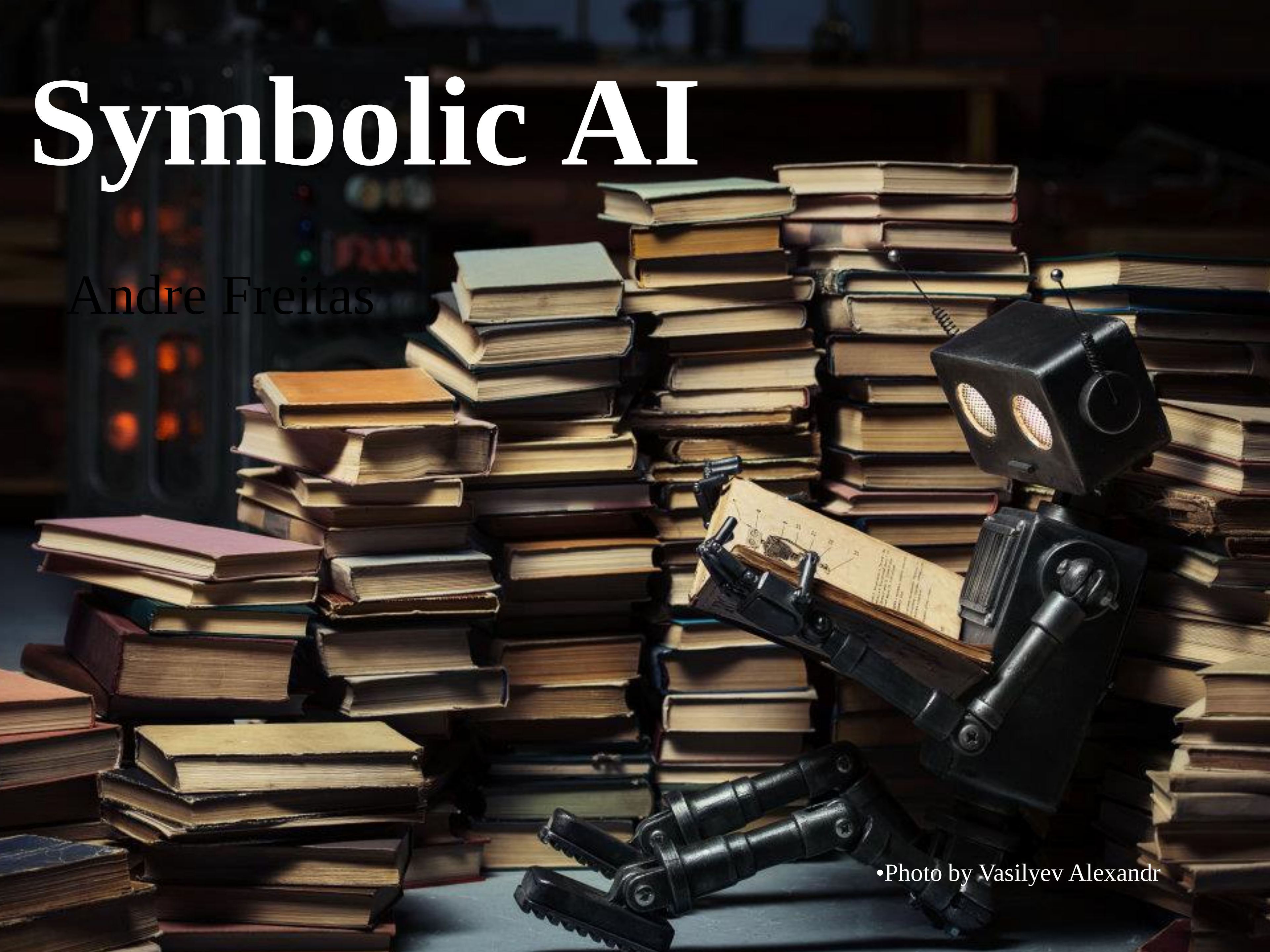
Qiang Zeng
University of Wisconsin–Madison
qzeng@cs.wisc.edu

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
        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: $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow$ 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.

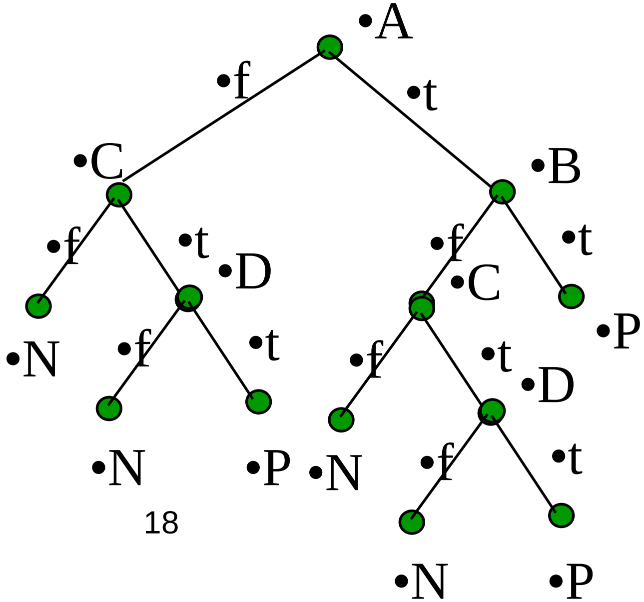
Rule Learning Issues

- 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

- **<u>Sequential covering</u>:**
  - 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 earch precondition for each rule

- **<u>Simultaneous covering</u>:**
  - 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.

$$(\forall < x_i, f(x_i) > \in D)(B \wedge h \wedge x_i) \vdash f(x_i)$$

- 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.

C_2: KnowMaterial v ~Study

C_1: PassExam v $\boxed{\text{\textasciitilde KnowMaterial}}$

C: PassExam v ~Study

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

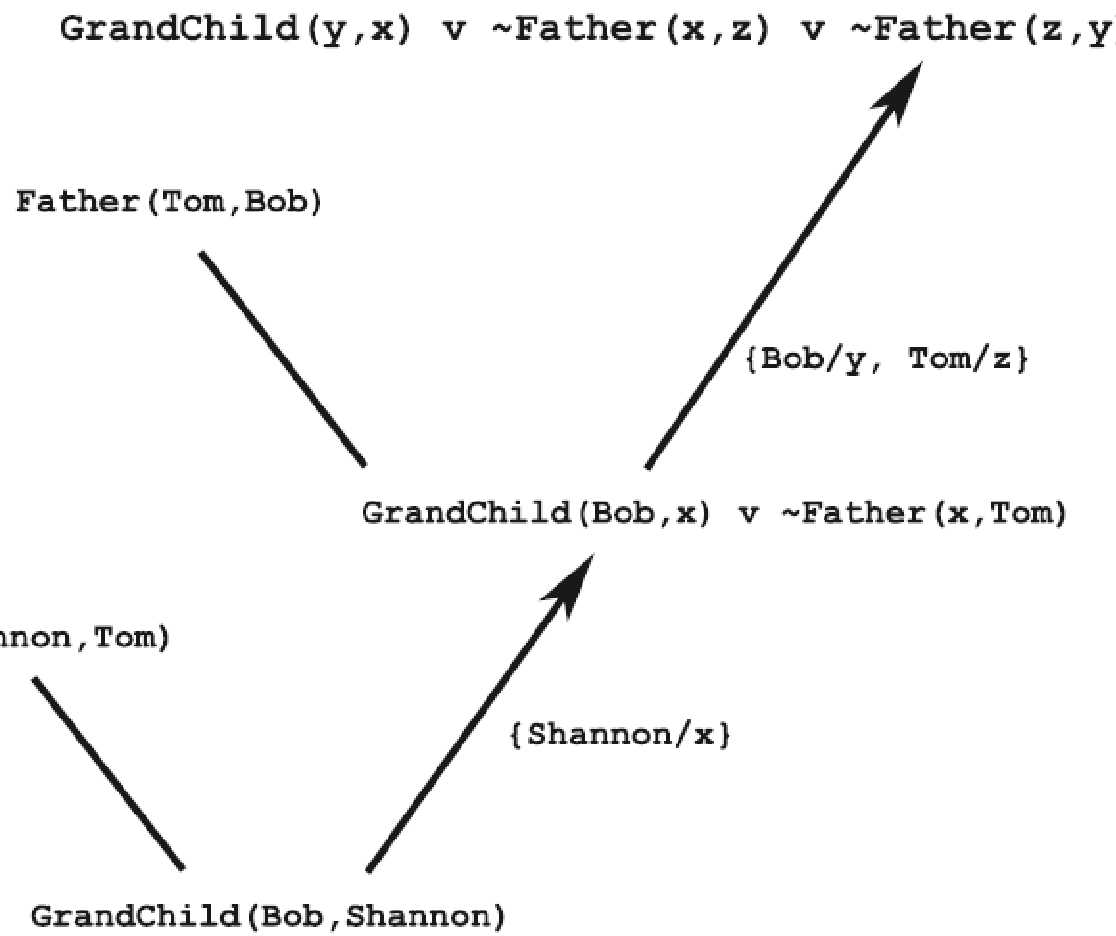
$$C_2 = (C - (C_1 - \{L\})) \cup \{L\}$$

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

C_2: $\boxed{\texttt{KnowMaterial}}$ `v ~Study`

C_1: `PassExam v ~KnowMaterial`

C: `PassExam v ~Study`



$$D = \{GrandChild(Bob, Shannon)\}$$

$$B = \{Father(Shannon, Tom), Father(Tom, Bob)\}$$

Generalization, θ-Subsumption, Entailment

interesting to consider the relationship between the $more\_general\_than$ relation and inverse entailment

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

$\theta - subsumption$: Consider two clauses C_j and C_k, both of the form $H \vee L_1 \vee ... \vee L_n$, where H is a positive literal and the L_i are arbitrary literals. Clause C_j is said to $\theta - subsume$ 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.
  - `path(X,Y) :- edge(X,Z), 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 `:[Head | Tail]`.

- Since FOIL cannot handle functions, this is re-represented as a predicate:

  `components(List, Head, 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: bird(penguin)

bird(eagle)

bird(crow)

bird(ostrich)

Negative: bird(carp)

bird(bat)

bird(horse)

Background knowledge:

lays_eggs(penguin).
lays_eggs(crow).
lays_eggs(eagle).
lays_eggs(ostrich).
lays_eggs(carp).

flies(eagle).
flies(crow).
flies(bat).

swims(carp).
swims(penguin).

runs(horse).
runs(ostrich).

fish(X) :- has_scales(X), swims(X).
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)

bird(X):- *lays_eggs(X), flies(X), has(X, feathers), has(X, beak), has(X, talons), makes_nest(X), eats(X,rodents).*

bird(eagle) bird(crow)

bird(ostrich)

bird(X):- *lays_eggs(X), has(X, feathers), has(X, beak), has(X, talons), makes_nest(X), eats(X,Y), validate_food(X,Y).*

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(sally,jim).
aunt_of(judy,jim).
```

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

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

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

Learn

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

Lambda Calculus

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

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

[Church 1932]

Lambda Calculus

Base Cases

- Logical constant
- Variable
- Literal
- Lambda term

Lambda Calculus
Logical Constants

- Represent objects in the world

$$NYC, CA, RAINIER, LEFT, \ldots$$
$$located\_in, depart\_date, \ldots$$

Lambda Calculus
Variables

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

$$x, y, z, \ldots$$

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

$$\lambda x.city(x)$$
$$\lambda x.\lambda y.located\_in(x, y)$$

Lambda Calculus
Lambda Terms

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

$$\lambda x.city(x)$$

$$\lambda x.\underbrace{\lambda}_{\text{Lambda operator}}\underbrace{y}_{\text{Variable}}.\underbrace{located\_in(x, y)}_{\text{Body}}$$

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
Wrangel	AK
Si	

Show me mountains in states bordering Texas

[Zettlemoyer and Collins 2005]

Capturing Meaning with Lambda Calculus

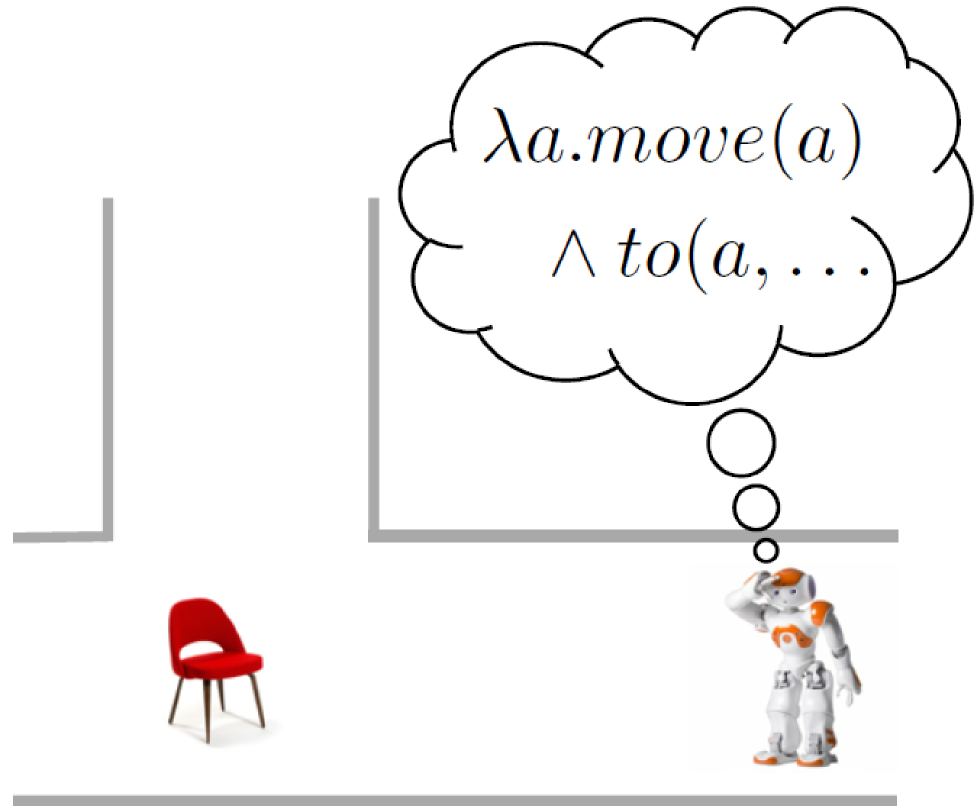
SYSTEM	how can I help you ?
USER	i ' d like to fly to new york
SYSTEM	flying to new york . leaving what city ?
USER	from boston on june seven with american airlines
SYSTEM	flying to new york . what date would you like to depart boston ?
USER	june seventh
SYSTEM	do you have a preferred airline ?
USER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

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

Combinatory Categorial Grammars

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

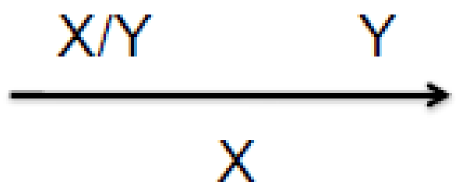
Combinatory Categorial Grammars

$$
\begin{array}{cccc}
\text{CCG} & \text{is} & \text{fun} & \\
\hline
NP & S\backslash NP/ADJ & ADJ & \\
CCG & \lambda f.\lambda x.f(x) & \lambda x.fun(x) & \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP} & > \\
 & \multicolumn{2}{c}{\lambda x.fun(x)} & \\
\hline
\multicolumn{3}{c}{S} & < \\
\multicolumn{3}{c}{fun(CCG)} &
\end{array}
$$

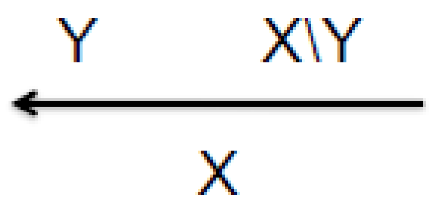
[Steedman 1996, 2000]

Formalism

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

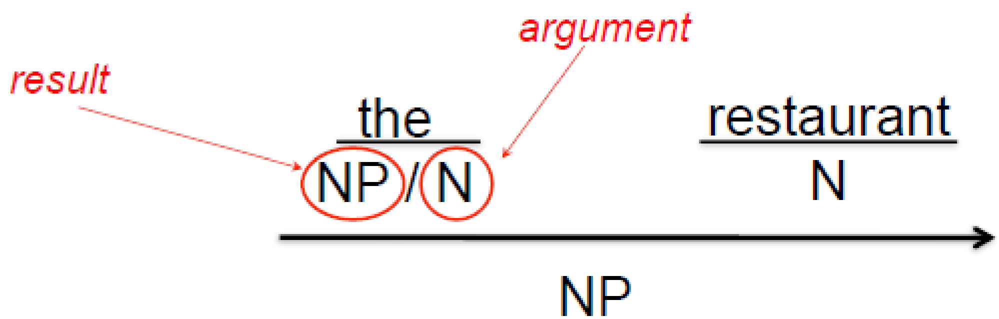


- X\Y: kind of word or phrase that combines with a **preceding** Y to form an X.



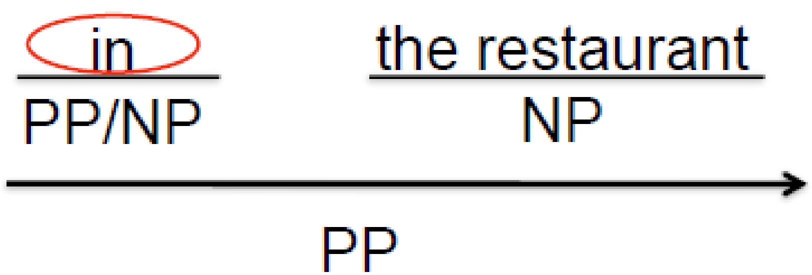
Determiners

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

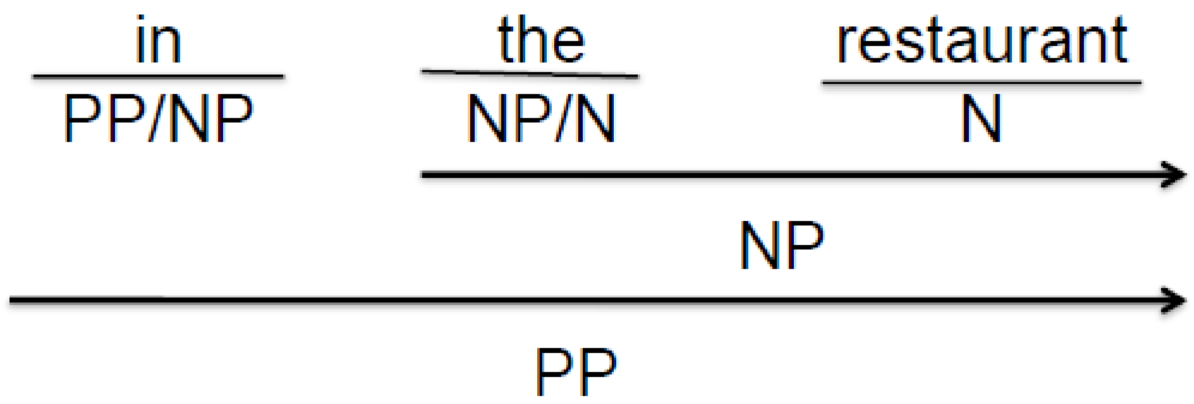


Prepositions

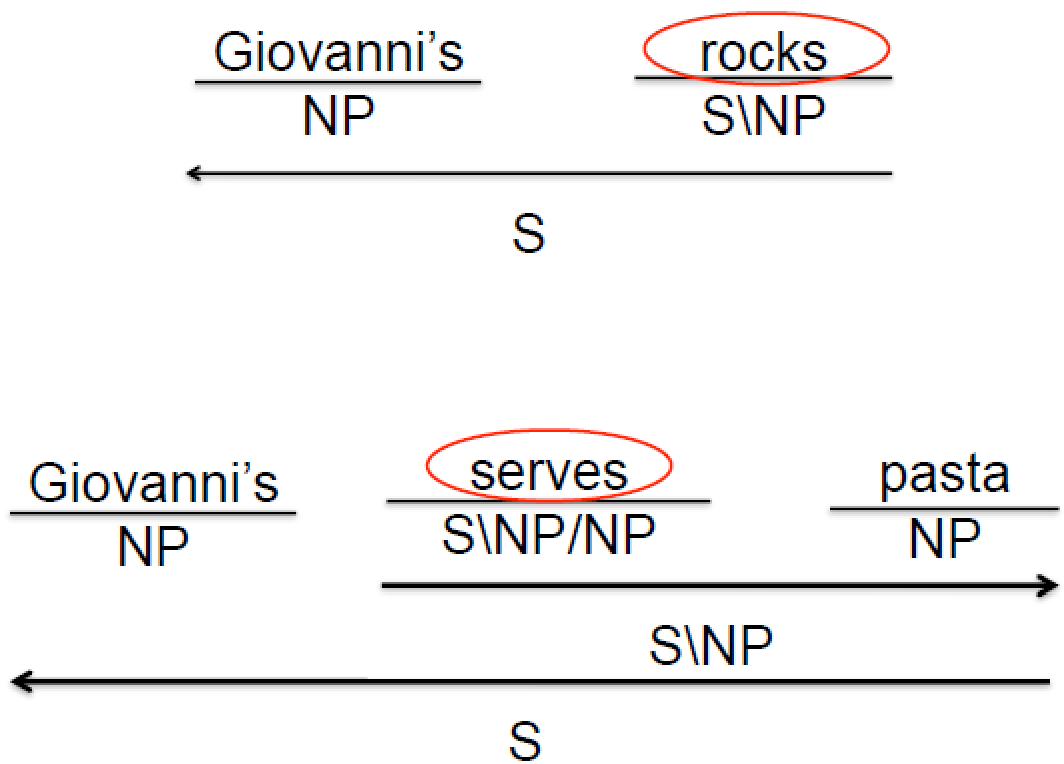
- **Preposition:** word that combines with a following NP to give a PP, i.e., a PP/NP.



Derivation



Verbs


Giovanni's
rocks
NP
S\NP
S
Giovanni's
serves
pasta
NP
S\NP/NP
NP
S\NP
S

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

$$\underbrace{\boxed{ADJ}}_{\text{Syntax}} : \lambda x.fun(x)$$

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

$$NP : CCG$$

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

CCG Categories

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

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

$$NP : CCG$$

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

CCG Lexical Entries

$$\text{fun} \ \vdash ADJ : \lambda x.fun(x)$$

CCG Lexical Entries

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

Natural Language

CCG Category

CCG Lexicons

$$\text{fun} \vdash ADJ : \lambda x.fun(x)$$

$$\text{is} \vdash (S\backslash NP)/ADJ : \lambda f.\lambda x.f(x)$$

$$\text{CCG} \vdash NP : CCG$$

Parsing with CCGs

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

CCG Operations

Application

$$\overbrace{B : g}^{\text{Argument}} \quad \overbrace{A \backslash B : f}^{\text{Function}} \Rightarrow \overbrace{A : f(g)}^{\text{Result}} \qquad (<)$$

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

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

Parsing with CCGs

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

Parsing with CCGs

$$
\begin{array}{ccc}
\text{CCG} & \text{is} & \text{fun} \\
\hline
NP & S\backslash NP/ADJ & ADJ \\
CCG & \lambda f.\lambda x.f(x) & \lambda x.fun(x) \\
 & \multicolumn{2}{c}{\overline{\hspace{10em}}{>}} \\
 & \multicolumn{2}{c}{S\backslash NP} \\
 & \multicolumn{2}{c}{\lambda x.fun(x)}
\end{array}
$$

Combine categories using operators

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

Parsing with CCGs

$$
\begin{array}{ccc}
\text{CCG} & \text{is} & \text{fun} \\
\hline
NP & S\backslash NP/ADJ & ADJ \\
CCG & \lambda f.\lambda x.f(x) & \lambda x.fun(x) \\
\cline{2-3}
 & \multicolumn{2}{c}{>} \\
 & \multicolumn{2}{c}{S\backslash NP} \\
 & \multicolumn{2}{c}{\lambda x.fun(x)} \\
\hline
\multicolumn{3}{c}{<} \\
\multicolumn{3}{c}{S} \\
\multicolumn{3}{c}{fun(CCG)}
\end{array}
$$

Combine categories using operators

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

CCG Operations
Composition

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

$$\underbrace{A/B : f}_{f} \quad \underbrace{B/C : g}_{g} \Rightarrow \underbrace{A/C : \lambda x.f(g(x))}_{f \circ g} \qquad (> B)$$

$$B\backslash C : g \quad A\backslash B : f \Rightarrow A\backslash C : \lambda x.f(g(x)) \qquad (< B)$$

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
Bo	
Elbe	

[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

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

$$\frac{\text{Washington}}{NP} \quad WA$$

$$\frac{\text{The Sunshine State}}{NP} \quad FL$$

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

$$
\begin{array}{ccc}
\underline{\text{Nevada}} & \underline{\text{borders}} & \underline{\text{California}} \\
NP & S\backslash NP/NP & NP \\
NV & \lambda x.\lambda y.border(y,x) & CA \\
 & \multicolumn{2}{c}{\overline{\hphantom{\lambda x.\lambda y.border(y,x)\quad CA}}>} \\
 & \multicolumn{2}{c}{S\backslash NP} \\
 & \multicolumn{2}{c}{\lambda y.border(y,CA)} \\
\multicolumn{3}{c}{\overline{\hphantom{NV\quad \lambda x.\lambda y.border(y,x)\quad CA}}<} \\
\multicolumn{3}{c}{S} \\
\multicolumn{3}{c}{border(NV,CA)}
\end{array}
$$

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

$\lambda x.state(x)$

$\{$ WA , AL , AK $, \ldots\}$

$e \rightarrow t$

functions define sets

mountain

$\lambda x.mountain(x)$

$\{$ BIANCA , ANTERO $, \ldots\}$

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

$$\frac{\textbf{state}}{N \atop \lambda x.state(x)}$$

$$\frac{\textbf{mountain}}{N \atop \lambda x.mountain(x)}$$

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

$$\lambda x.mountain(x) \wedge in(x, CO)$$

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

Prepositions

State

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

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

$$
\xrightarrow{\quad >}\quad \begin{array}{c} PP \\ \lambda x.in(x, CO) \end{array}
$$

$$
\begin{array}{c} N\backslash N \\ \lambda f.\lambda x.f(x) \wedge in(x, CO) \end{array}
$$

$$
\xrightarrow{\quad <}\quad \begin{array}{c} N \\ \lambda x.mountain(x) \wedge in(x, CO) \end{array}
$$

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

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

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

Function Words

State

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

$$
\begin{array}{cccc}
\text{state} & \text{that} & \text{borders} & \text{California} \\
\hline
N & PP/(S\backslash NP) & S\backslash NP/NP & NP \\
NV & \lambda f.f & \lambda x.\lambda y.border(y,x) & CA
\end{array}
$$

$$\xrightarrow{\quad\quad\quad\quad\quad} >$$

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

$$\xrightarrow{\quad\quad\quad\quad\quad} >$$

$$PP \quad \lambda y.border(y, CA)$$

$$N\backslash N \quad \lambda f.\lambda y.f(y) \wedge border(y, CA)$$

$$\xrightarrow{\quad\quad\quad\quad\quad} <$$

$$N \quad \lambda x.state(x) \wedge (x, CA)$$

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

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

mountain in Washington

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

$$\{ \text{RAINIER} \}$$

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

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

$$\{\text{RAINIER}\} \rightarrow \text{RAINIER}$$

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 Colorado

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

$\{$ BIANCA , ANTERO $\}$

No information to disambiguate

Definite Determiners

State

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

$$\cfrac{\cfrac{\text{the}}{\begin{array}{c}NP/N\\ \lambda f.\iota x.f(x)\end{array}} \qquad \cfrac{\begin{array}{c}\text{mountain in Colorado}\\ \vdots\end{array}}{\begin{array}{c}N\\ \lambda x.mountain(x)\wedge in(x,CO)\end{array}}}{\begin{array}{c}NP\\ \iota x.mountain(x)\wedge in(x,CO)\end{array}}>$$

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

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

[Steedman 2011; Artzi and Zettlemoyer 2013b]

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

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

Min or max

... over this set

... according to this measure

$\{$ WA, AL, AK, $\ldots\}$

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

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

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

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

$$\xrightarrow{\hspace{12em}} >$$

$$\begin{array}{c} NP \\ argmax(\lambda x.state(x), \lambda y.pop(y)) \end{array}$$

Superlatives

State

Abbr.	Capi
AL	Montgo
AK	June
AZ	Phoe
WA	Olym
NY	Alba
IL	Springf

$$
\begin{array}{ccc}
\text{the most} & \text{populated} & \text{state} \\
\hline
NP/N/N & N & N \\
\lambda g.\lambda f.argmax(\lambda x.f(x), \lambda y.g(y)) & \lambda x.pop(x) & \lambda x.state(x) \\
\hline
\multicolumn{2}{c}{NP/N} & \\
\multicolumn{2}{c}{\lambda f.argmax(\lambda x.f(x), \lambda y.pop(y))} & \\
\hline
\multicolumn{3}{c}{NP} \\
\multicolumn{3}{c}{argmax(\lambda x.state(x), \lambda y.pop(y))}
\end{array}
$$

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

Border

State1	State2
WA	OR
WA	ID
CA	OR

Mountains

Name	State
Bianca	CO
Antero	CO
Rainier	WA

Which mountains are in Arizona?

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

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?

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

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

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

[Parsons 1990]

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