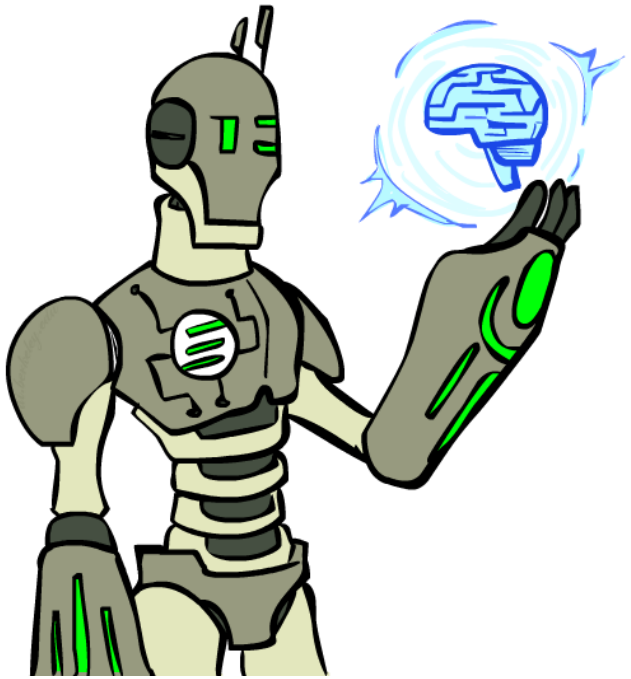
Chapter 2

Knowledge

Chapter Outline

- Intro to knowledge
- Knowledge representation
  - Ontological Engineering
  - Semantic Network
  - Frame
  - Knowledge Graph
- Representation using logic

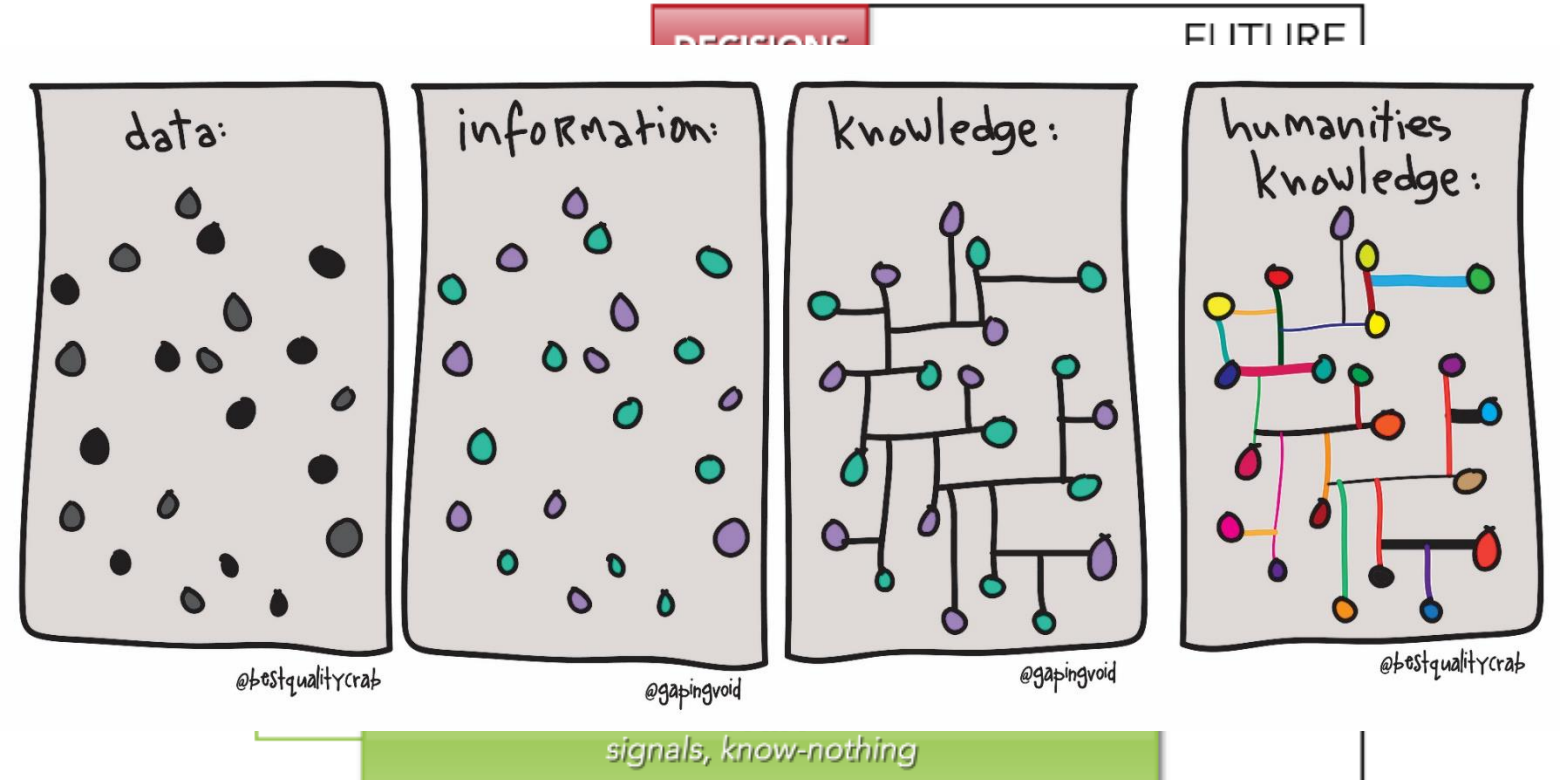


Intro to knowledge

Concepts

- **Data**
  - The measures and representations of the world.
  - As fact, signal, or symbol.
- **Information**
  - Produced by assigning meaning to data.
  - Structural vs. functional, subjective vs. objective
- **Knowledge**
  - Defined with reference to information.
  - As processed (加工的), procedural (过程的), or propositional (命题的).
- **Wisdom**
  - The experience to make decisions and judgments.
  - As “know-why” (知因), “know-how” (知然), or “why-do” (因何).

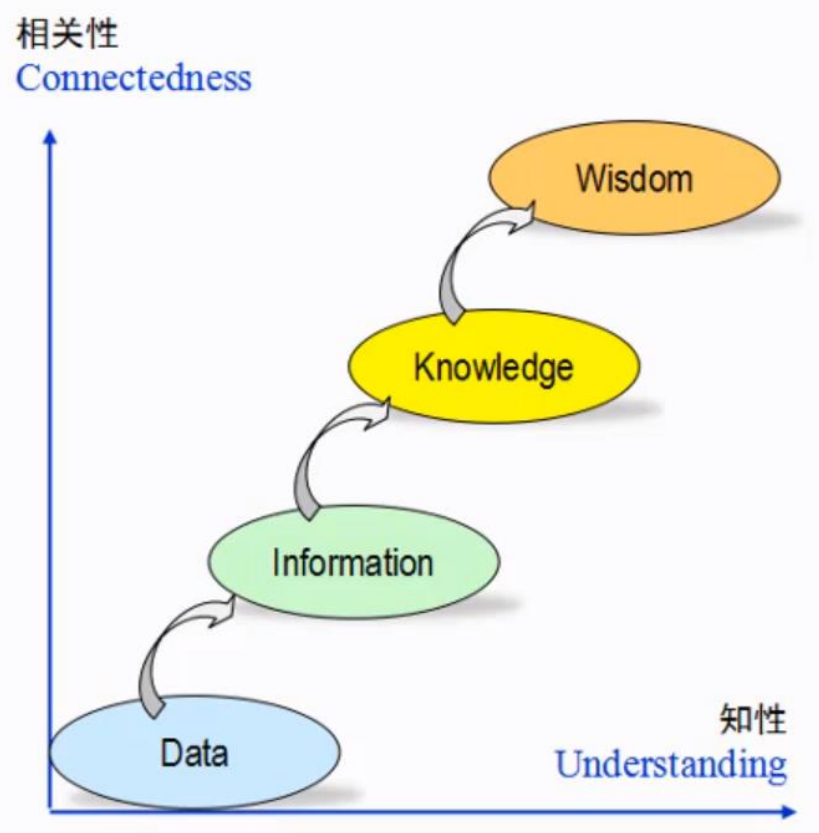
Concepts



https://figshare.com/articles/data_information_knowledge_humanities/1397453
https://www.linkedin.com/pulse/handy-concept-better-dikwd-pyramid-peter-j-korsten/

Example: Bank

- **Data**:
  - The numbers 100 or 5
  - Out of context
- **Information**:
  - Principal amount $100
  - Interest rate 5%
- **Knowledge**:
  - At the end of year I get $105
- **Wisdom**:
  - How to invest?



Knowledge

- **Explicit knowledge**
  - Can be articulated into formal language, including grammatical statements, etc.
  - Can be readily transmitted to others.
  - Can be easily represented using computer languages, decision trees and rules.
- **Tacit knowledge**
  - Individual experience and intangible factors, such as perspective, etc.
  - Hard to articulate with formal language.
  - Neural network offers the method to represent tacit knowledge.

Knowledge Types

Types	Features
Static knowledge	Unlikely to change
Dynamic knowledge	Records in a database
Surface knowledge	Accumulated through experience
Deep knowledge	Theories/Proofs/Problem Specifics
Procedural knowledge	Describes how a problem is solved
Declarative knowledge	Describes what is known about a problem
Meta knowledge	Describes knowledge about knowledge
Heuristic knowledge	A rule of thumb that guide the reasoning process

Knowledge Base

- The term **knowledge base** (KB) was to distinguish from the more common widely used term data base (DB).
  - KB is used to store complex structured and unstructured knowledge. It consists of a set of sentences, each one is expressed in a language called a **knowledge representation language** and represents some **assertion** about the world.
- A **KB system** (KBS) consists of a KB and an **inference engine**, where, KB represents facts about the world, inference engine can reason about those facts.
- **Knowledge Engineering** (KE) refers to all technical, scientific and social aspects involved in building, maintain and using KB systems.
- **Knowledge-based Engineering** (KBE) is the application of knowledge-based systems technology to the domain of manufacturing design and production.

Applications

- Search Engine
  - Google
  - Wikipedia
  - WolframAlpha
- QA systems

Knowledge Representation

Knowledge Representation

- What is knowledge representation?
  - Focus on designing **computer representations that capture knowledge** about the world that can be used to **solve complex problems**.
  - Make complex software **easier to define and maintain** than procedural code and can be used in expert systems.

- Why use knowledge representation?
  - Conventional procedural code is not the best formalism to solve complex problems.

Core Issues

- **Primitive** (原语)
  - The underlying framework used to represent knowledge, e.g., semantic network, first-order logic, *etc*.
- **Meta-representation** (元表示)
  - The knowledge representation language is itself expressed in that language, e.g., in Frame based environments, all frames would be instances of a frame.
- **Incompleteness** (不完备性)
  - To associate certainty factors with rules and conclusions, e.g., “Socrates is human with confidence 50%”.
- **Universals vs. Facts** (共性与事实)
  - Universals: general statements about the world, e.g., “All humans are mortal”.
  - Facts: specific examples of universals, e.g., “Socrates is a human and therefore mortal”.
- **Expressive adequacy** (表现的充分性)
  - How expressive they intend their representation to be.

Knowledge Representation

Ontological Engineering

Semantic Network

Frame

Knowledge Graph

What is Ontology?

- Ontology is a term used by philosophers to mean the most "**essential existence**" in the universe.
- For example, if A can be derived from or produced by B, A is **NOT** essential. Here B can be a single "existence" or a set of "existences".
- An existence may not be an "object". In fact, a group of philosophers believe that all existences are produced by a "spirit". These philosophers are called "spiritualists".
- On the contrary, some philosophers believe that the universe are made by various materials or objects, and they are called "materialists".

What is Ontology?

- In ontology engineering, we do not care what is the "origin" of the universe, but care about the **"true meanings" of concepts**. With the true meanings, we can define knowledge that can be shared, understood, and used by anybody in the world.

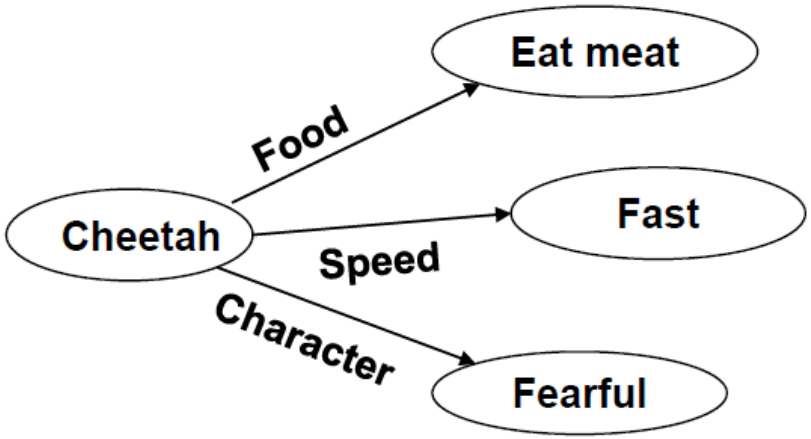
- For this purpose, researchers are trying to **represent knowledge graphically**, and open it to the world. People can then improve the **knowledge graph** or **ontology graph** from different angles.

Basic Idea

- In ontology engineering, **knowledge** is represented using a **graph**.
  - Each **node** of the graph represents a "concept", and the **edge** between two nodes represents the relation between the two concepts.
- A **concept** defines a group of existences. The form (shape or outlook), properties, states, *etc*. of these existences are memorized in the corresponding node.
- The **relation** between two concepts may include:
  - causal relation (因果关系), inclusion relation (包含关系), membership relation (成员关系), etc.
- Using an ontology graph, other people can understand the corresponding knowledge easily, and can improve freely.

Example: A Concept

- The concept “cheetah” has the following attributes and corresponding values:
  - A1=food; Value=meat;
  - A2=speed; Value=fast;
  - A3=character; Value=fearful;
- Fuzzy values are used to make them more understandable.



Classification vs. Descriptive

- **Classification ontologies**
  - They are used to classify things, such as books, documents, web pages, *etc*.; the purpose is to provide domain specific terminology and organize individuals accordingly. Such ontologies usually take the form of classifications with (BT\NT\RT) or without explicit relations.

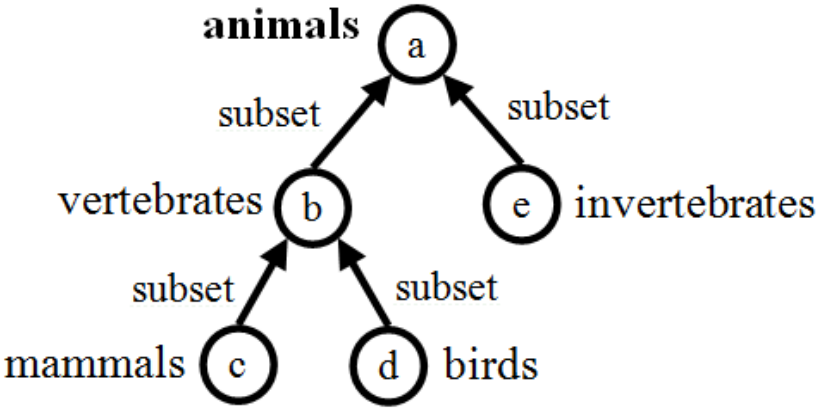
- **Descriptive ontologies**
  - They are used to describe a piece of world, such as the Gene ontology, Industry ontology, *etc*.; the purpose is to offer an unambiguous description of the world. Relations are typically explicit (e.g. is-a) and can be of any kind.

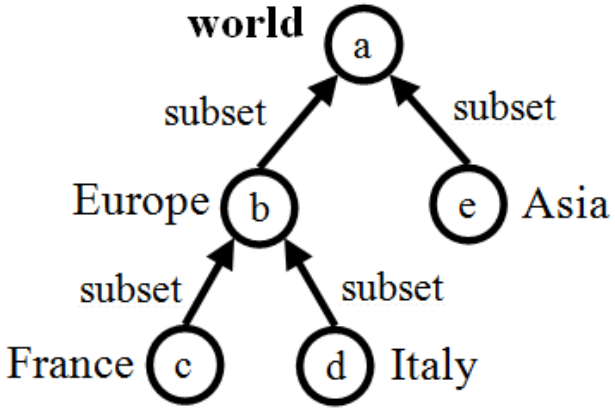
Classification vs. Descriptive

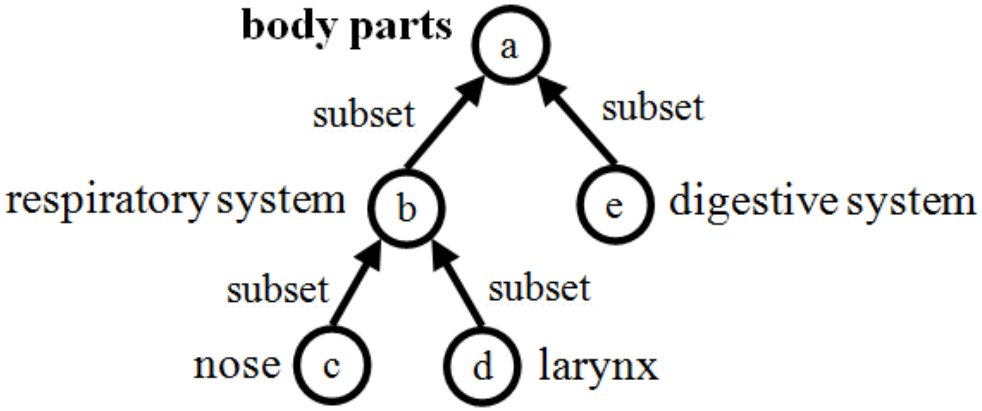
- **Classification ontologies** are in classification semantics
  - In classification ontologies, **the extension of each concept (label of a node) is the set of documents about the entities or individual objects described by the label of the concept**.
  - For example, the extension of the concept animal is "the set of documents about animals" of any kind.

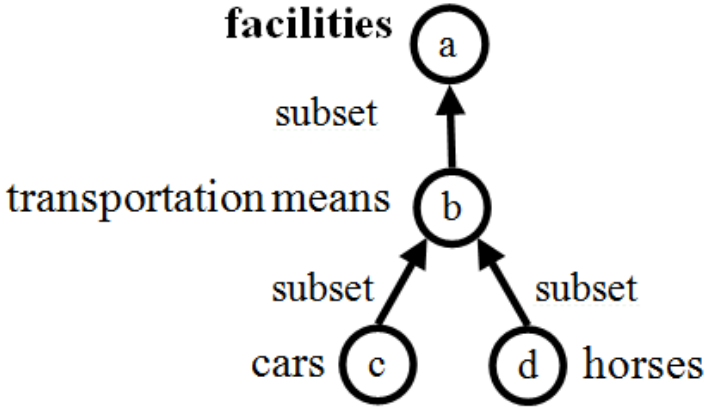
- **Descriptive ontologies** are in real world semantics
  - **Concepts represent real world entities**.
  - For example, the extension of the concept animal is the set of real world animals, which can be connected via relations of the proper kind.

Classification Ontologies

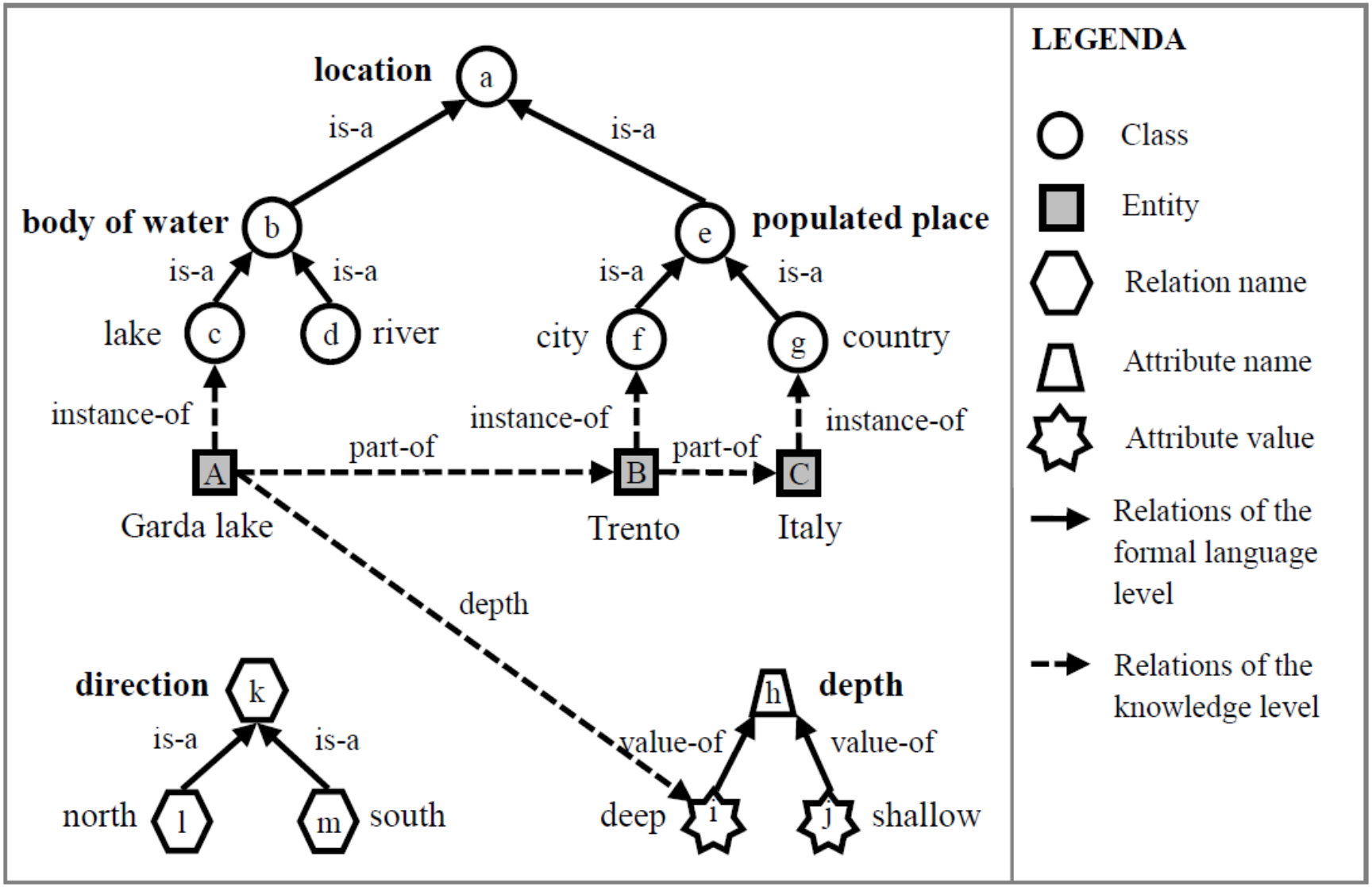








Descriptive Ontologies



Knowledge Representation

Ontological Engineering

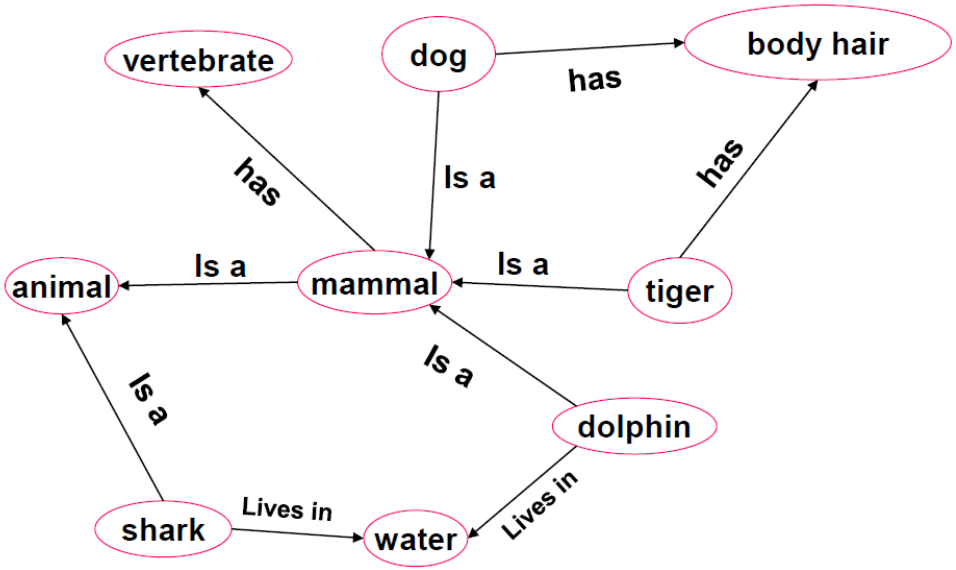
Semantic Network

Frame

Knowledge Graph

Semantic Network

- A network which represents semantic relations between concepts.
- It is a directed or undirected **graph** consisting of nodes and arcs, where
  - **nodes** represent concepts
  - **arcs** represent semantic relations between concepts



Example: Lisp

- A Semantic Network in Lisp:
  - Using an association list

```
(defun a-knowledge-base ()
        ((canary  (is-a bird)
                  (color yellow)
                  (size small))
        (penguin (is-a bird)
                  (movement swim))
        (bird      (is-a vertebrate)
                  (has-part wings)
                  (reproduction egg-laying))))
```

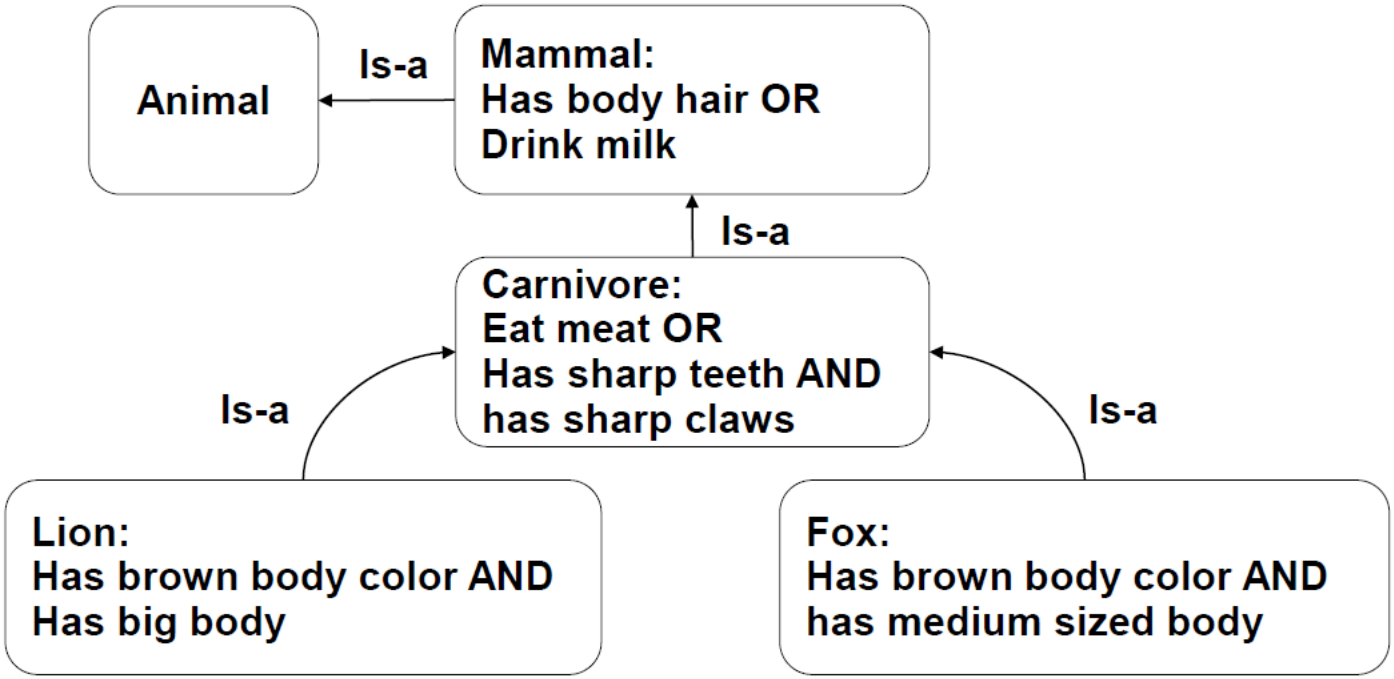
  - Use “assoc” function with a key of “canary” to extract all information about “canary” type.

Basics of Semantic Network

- Semantic Network is **cognitively** based.
- Organized into a **taxonomic hierarchy**.
- Used when one has knowledge that is best understood as a set of **concepts** that are related to one another.

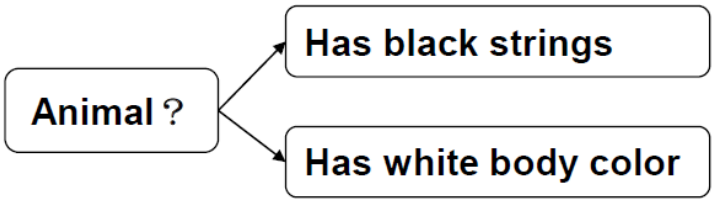
- However,
  - It is intractable for large domains, and can not represent performance or meta-knowledge very well.
  - Some properties are not easily expressed, e.g.,
    - Negation, disjunction, or general non-taxonomic knowledge.

Reasoning on Semantic Network



Reasoning on Semantic Network

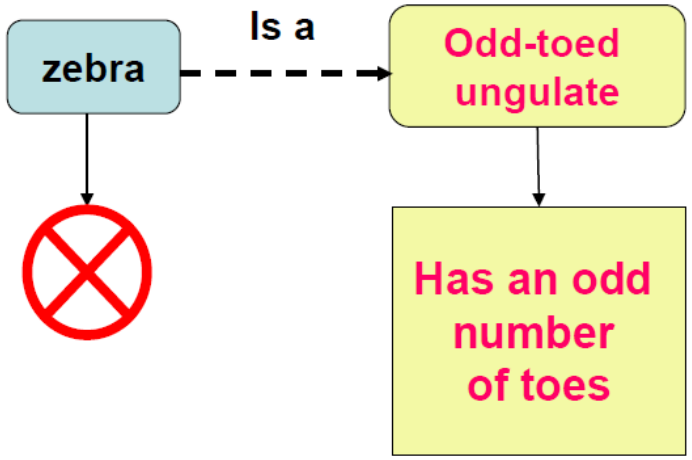
- Questions to ask:
  - Does lion has a big body?
  - Is fox an ungulate?
  - Does deer ruminate?
  - Is swallow black?
  - Does zebra has an odd number of toes?
  - What is the type of an animal who has
    - black strings and
    - a white body color?



Reasoning is equivalent to **graph matching**. A question network is generated for each question. We get an answer when the question network matches to a sub-graph of the semantic network.

Reasoning on Semantic Network

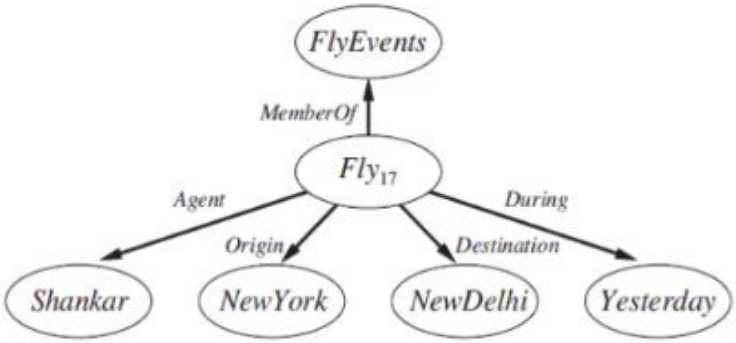
- Does zebra has an odd number of toes?



- Although the concept "zebra" does not contain the property mentioned in the question, its upper concept (super-set) contains the needed property. Therefore, the answer should be YES.

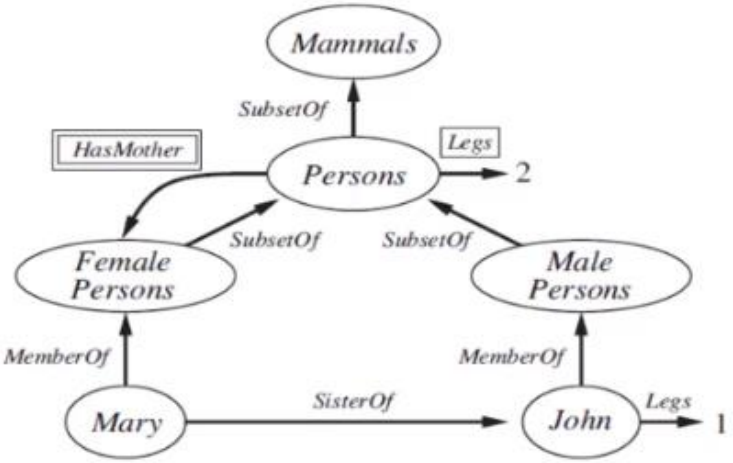
A sub-concept inherits all properties of its upper (parent) concept.

Example: Assertion/Objects



A semantic net of a logical assertion

Fly(*Shankar*, *New York*, *New Delhi*, *Yesterday*)



A semantic net with categories and objects

Mary ∈ *FemalePerson*, *John* ∈ *MalePersons*
SisterOf(*Mary*, *John*)

$\forall x\ x \in Persons \Rightarrow [\forall y\ HasMother(x,y) \Rightarrow y \in FemalePersons]$

Knowledge Representation

Ontological Engineering
Semantic Network
Frame
Knowledge Graph

From Semantic Network to Frame

- Frame was proposed by Minsky in 1975 as an extension of the semantic network.
- Frame is **more powerful** for knowledge representation.

Slot	Value	Type
Cheetah		ID (name) of this frame
Is-a	Carnivore	Pointer to the parent frame
Food	meat	Attribute and value
Speed	Fast	Attribute and value
Character	Fearful	Attribute and value

Frames

Living being	
is-a	*Thing*
Attribute	Live
Attribute	Re-producible
Attribute	Die

Animal	
is-a	*Living being*
Attribute	Move
Attribute	Eat
Attribute	Breath

Poch	
is-a	*Dog*
Popular	1
Size	Small
Origin	Japan

Cheetah	
is-a	*Animal*
Food	Meat
Character	Fearful
Speed	Fast

Dog	
is-a	*Animal*
Food	Omnivorous
Character	Friendly
Speed	Medium

Sahha	
is-a	*Cheetah*
Popular	2
Size	Medium
Origin	Sahara

Frames

Animal	
is-a	*Living being*
Attribute	*Head*
Attribute	*Body*
Attribute	*Move*

Head	
Front	*Face*
Tow sides	*Ears*
Upper side	*Forehead*
Lower side	*Neck*

Face	
Upper side	*Eyes*
Lower side	*Jaw*
Middle	*Nose*

- Difference frames are connected via "**hyper-links**".
- Different relations can be defined naturally.

Reasoning on Frames

- Similar to semantic network, to answer a question, we try to match the question to the frames.
  - An answer can be obtained directly if a frame **matches** the question.
  - In case there is no frame that can match the given question completely, we may try to look at the **parent** frames, and get the answer based on **inheritance**.
- To improve reasoning efficiency, some heuristics are need to select the frame to match. For example, we may start reasoning from the “most satisfied frame”.

Functional Frames

- In addition to static concepts, **procedures** (functions or methods) for conducting certain tasks can also be embedded into a frame.
- That is, frame can be used to represent both **declarative knowledge** and **procedural knowledge**.

Know what to do and how to do!

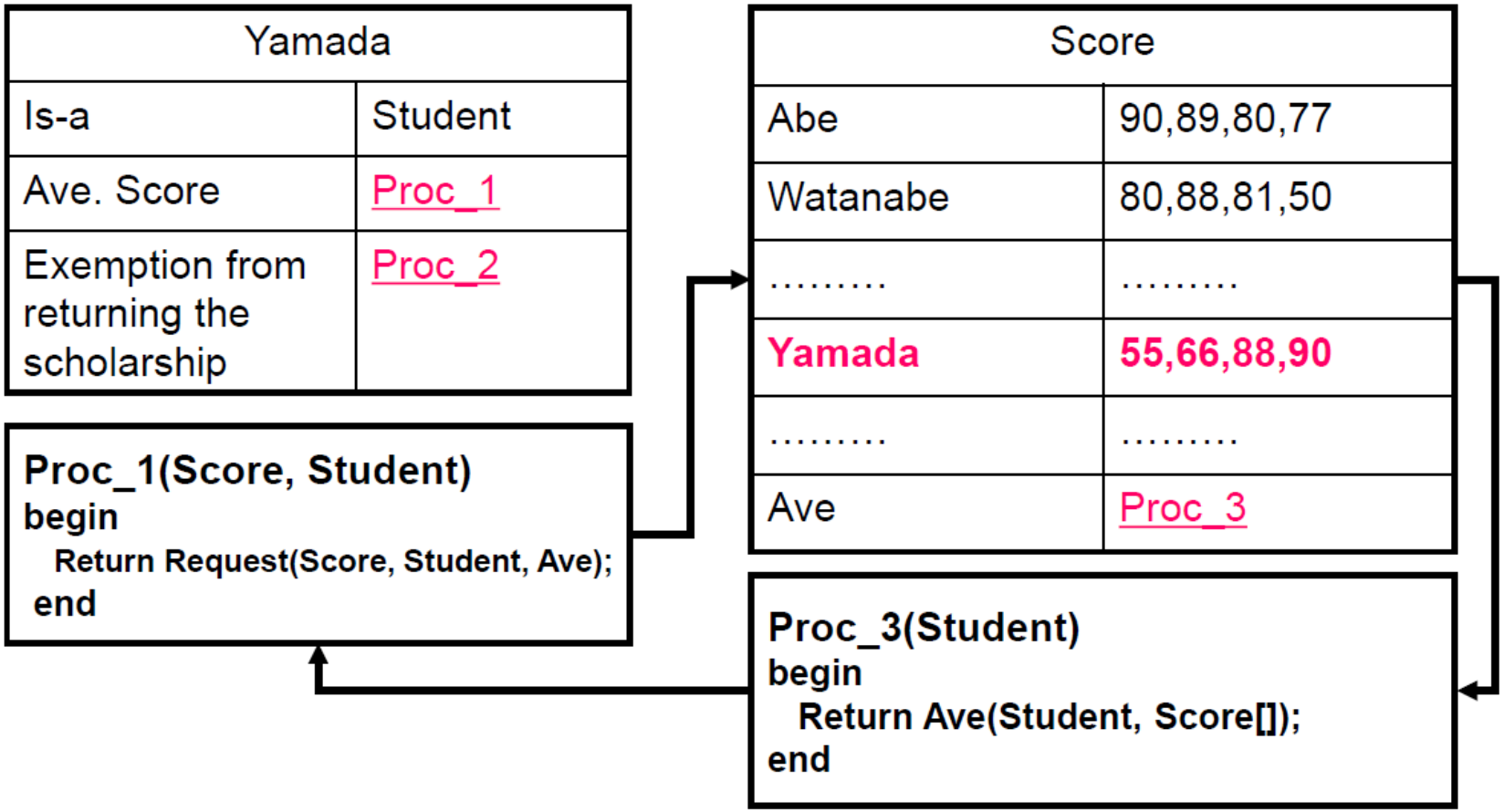
Demon or Agent

- The procedures are often called demons or agents.
- They are activated automatically when certain conditions are satisfied.
- Example of conditions:
  - If needed,
  - if added,
  - if removed,
  - if modified,
  - *etc*.

Yamada	
Is-a	student
GPA	**If-needed:** Proc_1
Earlier graduation	**If-needed:** Proc_2

- Example:
  - If the GPA of a student is needed, and the value does not exist yet, a procedure can be called to find the value.

Example


Yamada
Is-a
Student
Ave. Score
Proc_1
Exemption from returning the scholarship
Proc_2
Score
Abe
90,89,80,77
Watanabe
80,88,81,50
.....
.....
Yamada
55,66,88,90
.....
.....
Ave
Proc_3
Proc_1(Score, Student)
begin
Return Request(Score, Student, Ave);
end
Proc_3(Student)
begin
Return Ave(Student, Score[]);
end

Example

- For the question “what is the average score of Yamada”,
  - The system tries to find the frame of “Yamada” and then his average score;
  - If the average score exists, return the value and stop.
  - Otherwise, call Proc_1;
  - Proc_1 sends a request to the frame “Score” with a student name and the name of the slot (or attribute);
  - Proc_3 is activated because it is needed;
  - Proc_3 finds the average value of the specified student and replies to Proc_1 (of the specified student);
  - Proc_1 receives the value from Proc_3, and return it to the user.

Properties of Frame

- A frame system has the same ability as a semantic network for **representing declarative knowledge**.
- It can also represent **procedural knowledge** by using demons (agents, functions, or methods).
- That is, a frame system **can not only reason with existing knowledge, but also produce new knowledge when needed**.
- In other word, the concept represented by a frame (or node if we consider the frame system a knowledge graph) can be **dynamically modified** based on new data (or new situation).

Knowledge Representation

Ontological Engineering
Semantic Network
Frame
Knowledge Graph

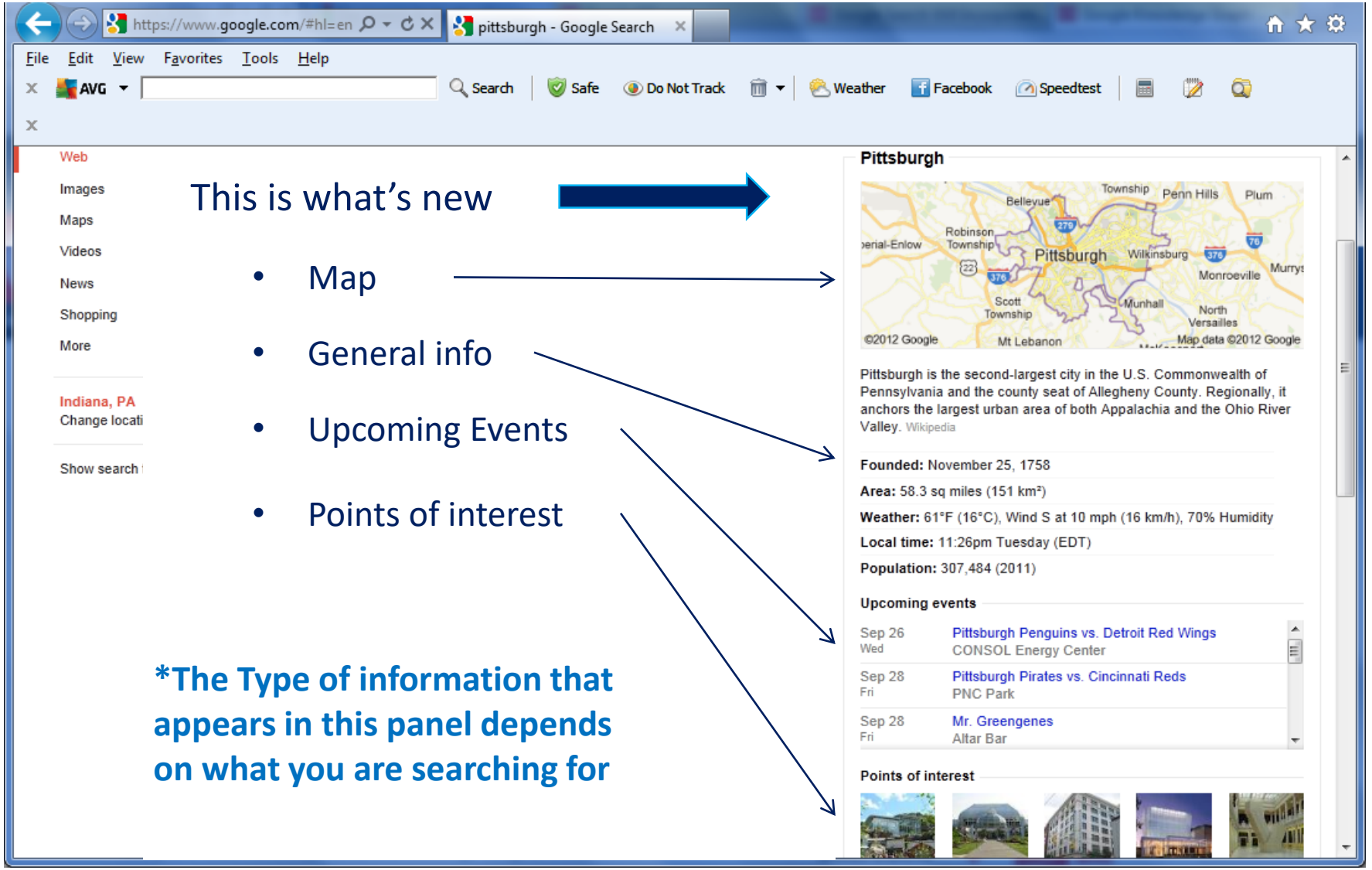
Before the Knowledge Graph

- How does Google work?
  - Crawling – searching the web
  - Page Ranking – Based on linkage
  - Word Indexing – indexes keywords on different pages
  - When user enters search looks for instances where a majority of the words they entered are grouped in proximity to each other
  - Displays these based on ranking

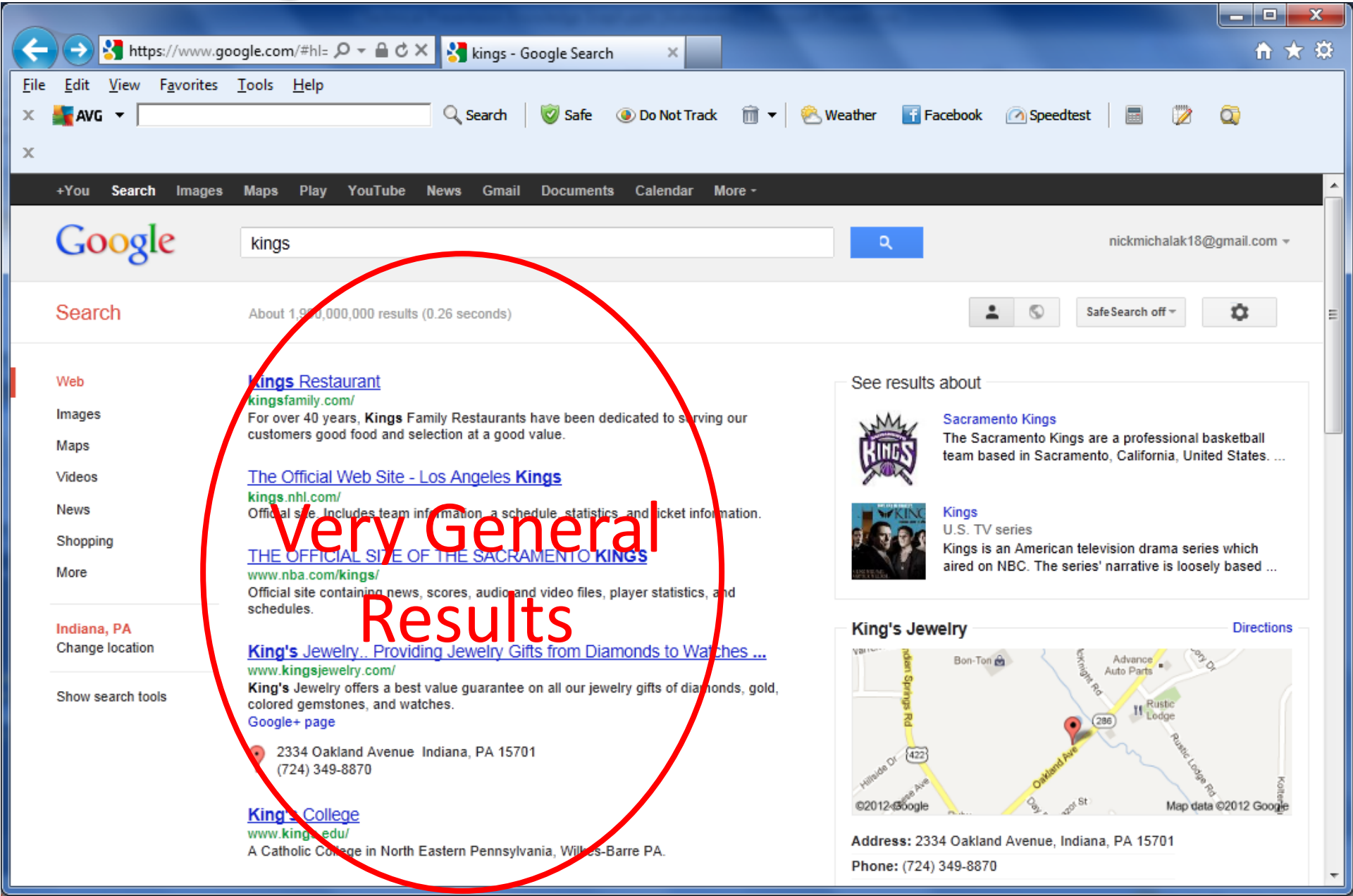
Knowledge Graph

- First step towards providing users with answers, rather than a collection of search results.
- Attempt to turn ambiguous words into actual concepts understandable by search engine.
- The knowledge graph only appears when you are searching for data already in its continuously growing database.

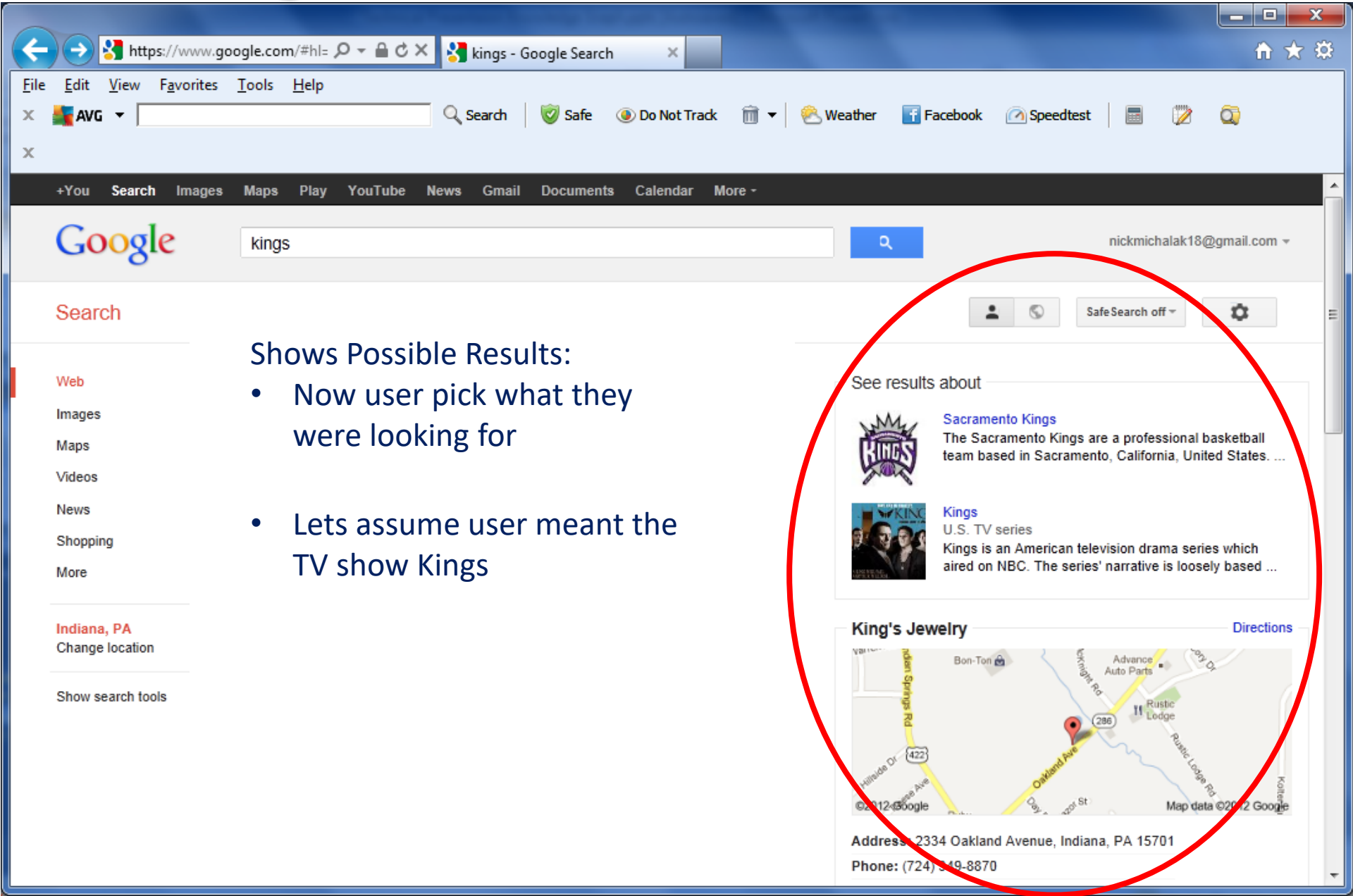
Google


https://www.google.com/#hl=en
pittsburgh - Google Search
File Edit View Favorites Tools Help
AVG
Search
Safe
Do Not Track
Weather
Facebook
Speedtest
Web
Images
Maps
Videos
News
Shopping
More
Indiana, PA
Change locati
Show search
This is what's new
Map
General info
Upcoming Events
Points of interest
*The Type of information that appears in this panel depends on what you are searching for
Pittsburgh
Bellevue
Township
Penn Hills
Plum
Robinson Township
Pittsburgh
Wilkinsburg
Monroeville
Scott Township
Munhall
North Versailles
Mt Lebanon
©2012 Google
Map data ©2012 Google
Pittsburgh is the second-largest city in the U.S. Commonwealth of Pennsylvania and the county seat of Allegheny County. Regionally, it anchors the largest urban area of both Appalachia and the Ohio River Valley. Wikipedia
Founded: November 25, 1758
Area: 58.3 sq miles (151 km²)
Weather: 61°F (16°C), Wind S at 10 mph (16 km/h), 70% Humidity
Local time: 11:26pm Tuesday (EDT)
Population: 307,484 (2011)
Upcoming events
Sep 26 Wed
Pittsburgh Penguins vs. Detroit Red Wings
CONSOL Energy Center
Sep 28 Fri
Pittsburgh Pirates vs. Cincinnati Reds
PNC Park
Sep 28 Fri
Mr. Greengenes
Altar Bar
Points of interest

Google


https://www.google.com/#hl=
kings - Google Search
File Edit View Favorites Tools Help
AVG
Search
Safe
Do Not Track
Weather
Facebook
Speedtest
+You Search Images Maps Play YouTube News Gmail Documents Calendar More
Google
kings
nickmichalak18@gmail.com
Search
About 1,960,000,000 results (0.26 seconds)
Safe Search off
Web
Images
Maps
Videos
News
Shopping
More
Indiana, PA
Change location
Show search tools
Kings Restaurant
kingsfamily.com/
For over 40 years, Kings Family Restaurants have been dedicated to serving our customers good food and selection at a good value.
The Official Web Site - Los Angeles Kings
kings.nhl.com/
Official site. Includes team information, a schedule, statistics, and ticket information.
THE OFFICIAL SITE OF THE SACRAMENTO KINGS
www.nba.com/kings/
Official site containing news, scores, audio and video files, player statistics, and schedules.
King's Jewelry.. Providing Jewelry Gifts from Diamonds to Watches ...
www.kingsjewelry.com/
King's Jewelry offers a best value guarantee on all our jewelry gifts of diamonds, gold, colored gemstones, and watches.
Google+ page
2334 Oakland Avenue Indiana, PA 15701
(724) 349-8870
King's College
www.kings.edu/
A Catholic College in North Eastern Pennsylvania, Wilkes-Barre PA.
Very General
Results
See results about
Sacramento Kings
The Sacramento Kings are a professional basketball team based in Sacramento, California, United States. ...
Kings
U.S. TV series
Kings is an American television drama series which aired on NBC. The series' narrative is loosely based ...
King's Jewelry
Directions
Bon-Ton
Advance Auto Parts
Rustic Lodge
Oakland Ave
286
422
©2012 Google
Map data ©2012 Google
Address: 2334 Oakland Avenue, Indiana, PA 15701
Phone: (724) 349-8870

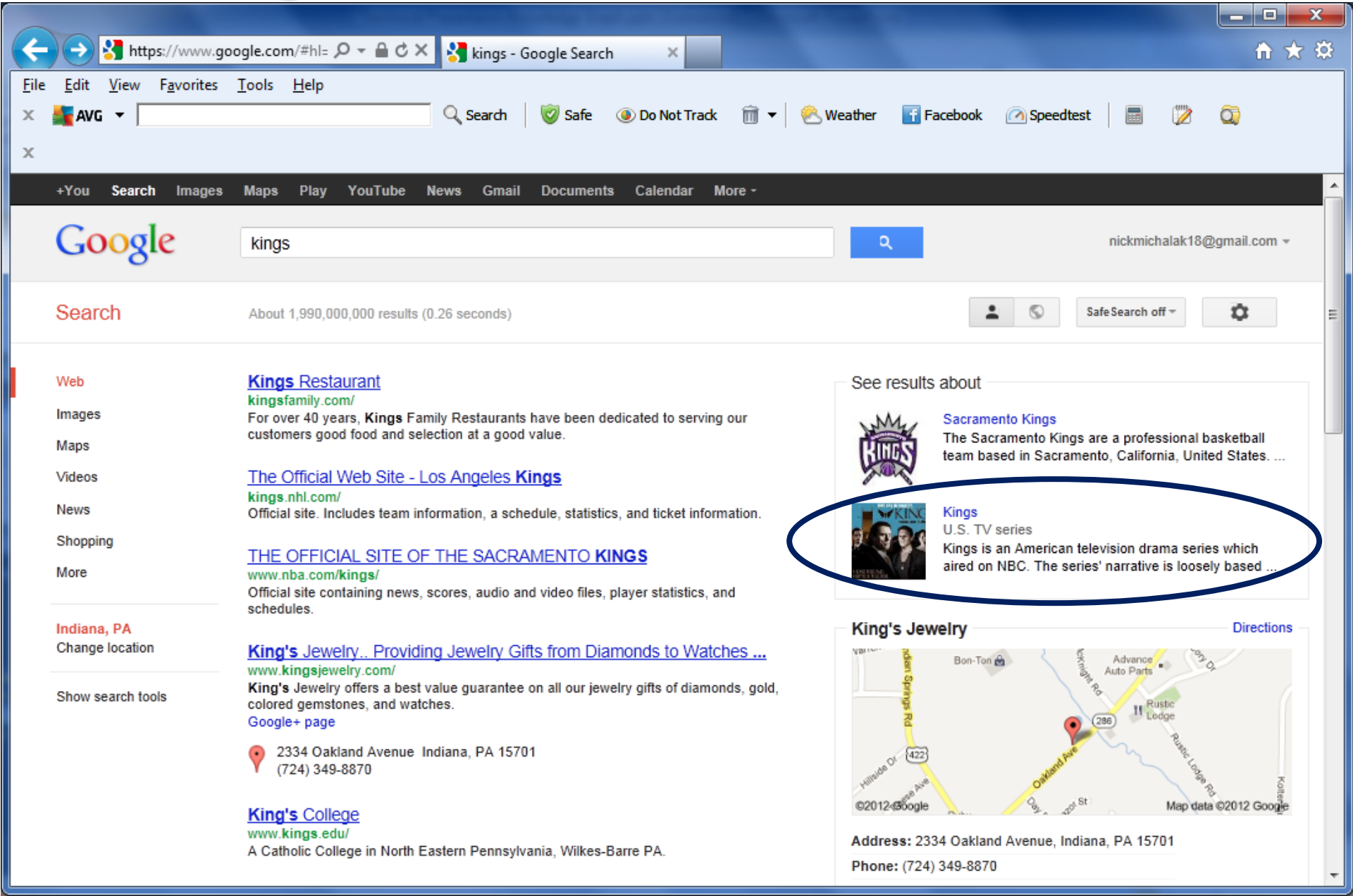
Google



Shows Possible Results:

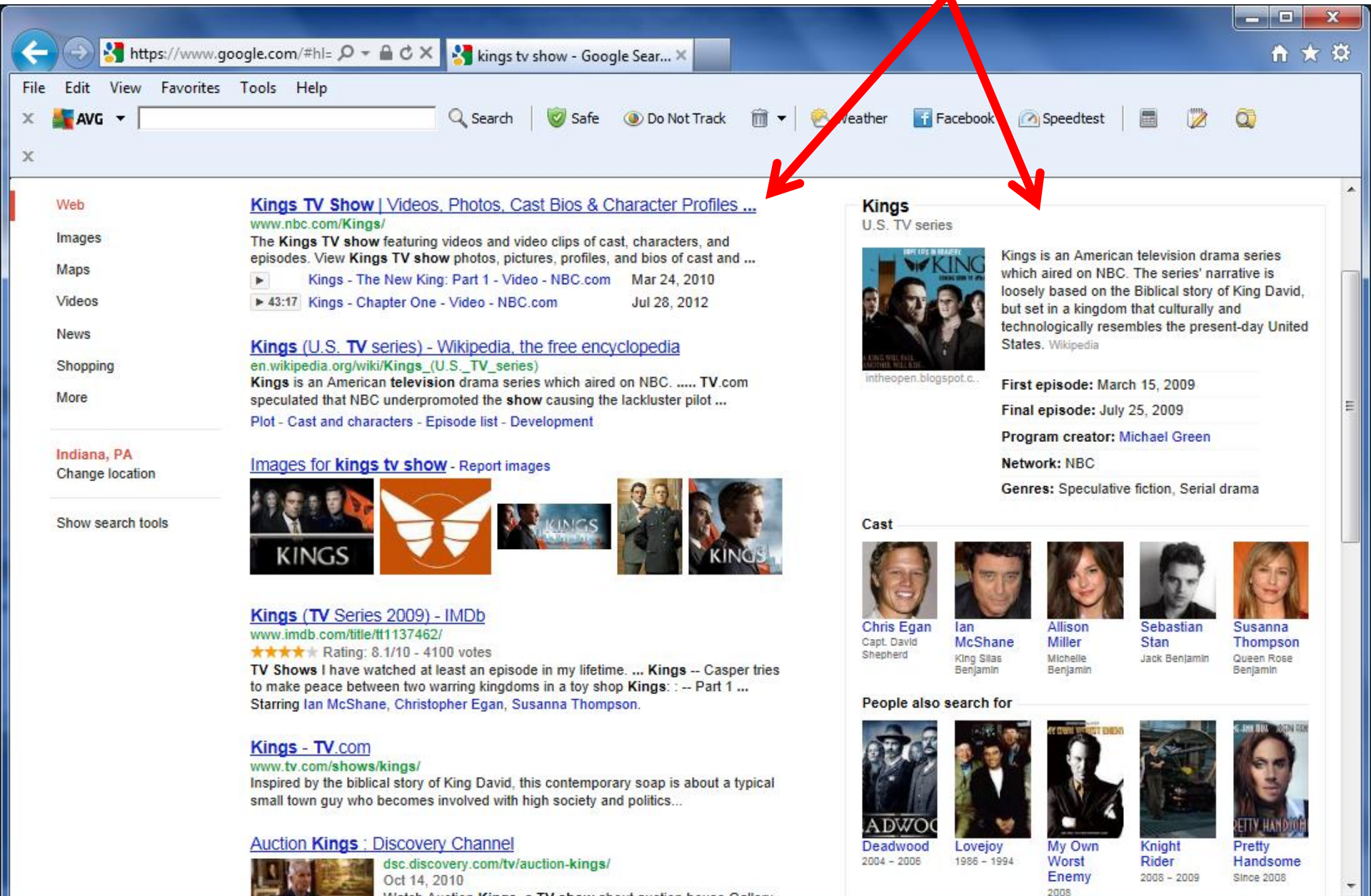
- Now user pick what they were looking for
- Lets assume user meant the TV show Kings

Google


https://www.google.com/#hl=
kings - Google Search
File Edit View Favorites Tools Help
AVG
Search
Safe
Do Not Track
Weather
Facebook
Speedtest
+You Search Images Maps Play YouTube News Gmail Documents Calendar More
Google
kings
nickmichalak18@gmail.com
Search
About 1,990,000,000 results (0.26 seconds)
Safe Search off
Web
Images
Maps
Videos
News
Shopping
More
Indiana, PA
Change location
Show search tools
Kings Restaurant
kingsfamily.com/
For over 40 years, Kings Family Restaurants have been dedicated to serving our customers good food and selection at a good value.
The Official Web Site - Los Angeles Kings
kings.nhl.com/
Official site. Includes team information, a schedule, statistics, and ticket information.
THE OFFICIAL SITE OF THE SACRAMENTO KINGS
www.nba.com/kings/
Official site containing news, scores, audio and video files, player statistics, and schedules.
King's Jewelry.. Providing Jewelry Gifts from Diamonds to Watches ...
www.kingsjewelry.com/
King's Jewelry offers a best value guarantee on all our jewelry gifts of diamonds, gold, colored gemstones, and watches.
Google+ page
2334 Oakland Avenue Indiana, PA 15701
(724) 349-8870
King's College
www.kings.edu/
A Catholic College in North Eastern Pennsylvania, Wilkes-Barre PA.
See results about
Sacramento Kings
The Sacramento Kings are a professional basketball team based in Sacramento, California, United States. ...
Kings
U.S. TV series
Kings is an American television drama series which aired on NBC. The series' narrative is loosely based ...
King's Jewelry
Directions
Bon-Ton
Advance Auto Parts
Rustic Lodge
Oakland Ave
286
422
©2012 Google
Map data ©2012 Google
Address: 2334 Oakland Avenue, Indiana, PA 15701
Phone: (724) 349-8870

Google

Notice both changed



Representation Using Logic

Procedural vs. Declarative

- **Procedural approaches**
  - Use procedural languages, such as
    - C/C++/C#/Java
    - Lisp
    - Python

- **Declarative approaches**
  - Use declarative languages, such as
    - Propositional logic
    - First-order logic
    - Temporal logic

5 Different Logics

Formal Language 形式语言	Ontological Commitment 本体论约定	Epistemological Commitment 认识论约定
Propositional logic 命题逻辑	Facts	True/false/unknown
First-order logic 一阶逻辑	Facts, objects, relations	True/false/unknown
Temporal logic 时序逻辑	Facts, objects, relations, times	True/false/unknown
Probability theory 概率论	Facts	Degree of belief $\in$ [0,1]
Fuzzy logic 模糊逻辑	Facts with degree of truth $\in$ [0,1]	Known interval value

Logical Symbols

Category 类别	Symbol 符号	Mean 含义	
Connectives 连接词	¬	not	非
	∧	and	与
	∨	or	或
	⇒	implies	蕴含
	⇔	if and only if (≡)	当且仅当
	⊨	entailment	导出
	⊭		
Quantifiers 限量词	∀	for all	所有
	∃	there exist	存在
Equality 等量词	=	equal	等于

Propositional logic

- **Propositional logic** (命题逻辑)
  - Also known as **propositional calculus**
  - Use of logical **connectives** (连接词), deal with simple declarative propositions (if they are true or false)
  - To describe **facts**

- Example:
  - $\neg P$, $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$, $P \leftrightarrow Q$
  - Truth table.

Interpretation

- When we see a logic formula, we do not care about the original meaning of each proposition. Rather, we care about the **logic relation between the propositions**.

- The **main concern** is to know if a logic formula is “true” or “false” when the atomic formulas contained in this formula are given.

- The process for determining the “true/false” of a logic formula based on the “true/false” of the atomic formulas is called “**interpretation**”.

Example

- For example, for the formula $P \wedge Q \Rightarrow R$, we can interpret it using the truth table.

P	Q	R	$P \wedge Q \Rightarrow R$
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

- Here, we have used the following fact:

$$P \wedge Q \Rightarrow R = \neg (P \wedge Q) \vee R$$

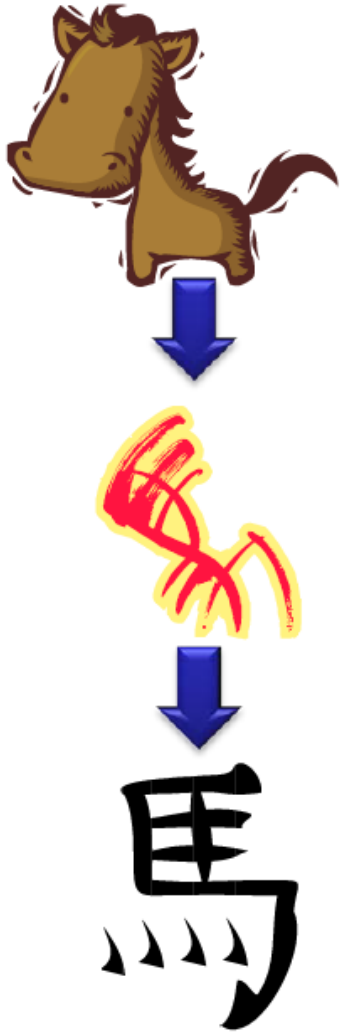
Laws of Propositional Logic

$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity Laws	$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination Law	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent Laws	$\neg (p \wedge q) \equiv \neg p \vee \neg q$ $\neg (p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$\neg(\neg p) \equiv p$	Double negation law	$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative Laws	$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Negation lows
$p \rightarrow q \equiv \neg p \vee q$	Definition of Implication	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$	Definition of Biconditional

Logic Formula Conversion

- Any logic formula can be converted to a clausal form as follows.

1. Remove the equivalence and implication symbols:
   - $P \Leftrightarrow Q = (P \Rightarrow Q) \wedge (Q \Rightarrow P)$
   - $P \Rightarrow Q = \neg P \vee Q$
2. Put the negation symbol just before the literals:
   - $\neg (\neg P) = P$
   - $\neg (P \wedge Q) = \neg P \vee \neg Q$
   - $\neg (P \vee Q) = \neg P \wedge \neg Q$
3. Adopt the distributive law:
   - $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$



Example

$$\neg P \Rightarrow ((Q \Rightarrow \neg R) \wedge \neg (R \Rightarrow \neg Q))$$

$$= P \vee ((\neg Q \vee \neg R) \wedge \neg (\neg R \vee \neg Q))$$

$$= P \vee ((\neg Q \vee \neg R) \wedge (R \wedge Q))$$

$$= (P \vee (\neg Q \vee \neg R)) \wedge (P \vee (R \wedge Q))$$

$$= (P \vee \neg Q \vee \neg R) \wedge ((P \vee R) \wedge (P \vee Q))$$

$$= (P \vee \neg Q \vee \neg R) \wedge (P \vee R) \wedge (P \vee Q)$$

First-order Logic

- **First-order logic** (一阶逻辑)
  - Also known as **first-order predicate calculus**
  - Additionally, use **quantifiers** (限量词), **equality** (等量词), and **predicates** (谓词) (often associated with sets)
  - To describe **objects** and **relationships** among objects

- Example:
  - $\forall x\ H(x) \Rightarrow D(x)$:
    - For any x, if x is a human, x dies.
  - $\exists x\ D(x) \Rightarrow H(x)$:
    - For some x, x dies, x is a human.

Syntax with BNF

- BNF: Backus-Naur Form (巴克斯诺尔范式)
- Propositional logic

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{True} \mid \textit{False} \mid P \mid Q \mid R \mid \ldots \\
\textit{ComplexSentence} & \rightarrow & (\textit{Sentence}) \mid [\textit{Sentence}] \\
& \mid & \neg \textit{Sentence} \\
& \mid & \textit{Sentence} \wedge \textit{Sentence} \\
& \mid & \textit{Sentence} \vee \textit{Sentence} \\
& \mid & \textit{Sentence} \Rightarrow \textit{Sentence} \\
& \mid & \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
\textit{OPERATOR PRECEDENCE} & : & \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{array}
$$

Syntax with BNF

- First-order logic

Sentence	→	*AtomicSentence* | *ComplexSentence*
AtomicSentence	→	*Predicate* | *Predicate*(*Term*, ...) | *Term* = *Term*
ComplexSentence	→	< *Sentence* > | [*Sentence*] | ¬ *Sentence* | *Sentence* ∧ *Sentence*
	|	*Sentence* ∨ *Sentence* | *Sentence* ⇒ *Sentence* | *Sentence* ⇔ *Sentence*
	|	*Quantifier Variable*, ... *Sentence*
Term	→	*Function*(*Term*, ...) | *Constant* | *Variable*
Quantifier	→	∀ | ∃
Constant	→	*A* | X_1 | *John* | ...
Variable	→	*a* | *x* | *s* | ...
Predicate	→	*True* | *False* | *After* | *Loves* | *Raining* | ...
Function	→	*Mother* | *LeftLeg* | ...
OPERATOR PRECEDENCE	:	¬, =, ∧, ∨, ⇒, ⇔

Terminologies

- **Individual constant**
  - a, b, c, Taro, Hanako
  - Used in the same way as in propositional logic
- **Individual variable**
  - x, y, z
  - Used to represent something not yet fixed.
- **Domain or universe of discourse**
  - Sets of students, human, living being.
  - Sets of prime numbers, integers, real numbers.
  - Sets of machines, robots, systems.

Terminologies

- **Functional symbol**
  - Map from *D* to *D*: from individual(*s*) to individual(*s*).
  - Ex1: *y*=mother(*x*): Find the mother of *x*, and put the result into *y*.
  - Ex2: brother(*x*): Find the brother(*s*) of *x*.
- **Predicate symbol**
  - Map from *D* to {True, False}: from individuals to logic value
  - Ex1: Human(*x*): True if x is a human.
  - Ex2: Mother(*x*, *y*): True if *y* is the mother of *x*.
- Example
  - *a*="Emily", *b*="Jason"
  - Known: Emily is the mother of Jason
  - Mother(*b*,*a*) is True
  - mother(*b*) is *a*="Emily"

Formation Rules

- The formation rules define:
  - **Terms** (项)
  - **Formulas** (公式)

- The formation rules can be used to write a formal grammar for terms and formulas.

- Formation rules are generally **context-free**, i.e.,
  - Each production has a single symbol on the left side.

Formation Rules: Terms

- **Rule 1: Variables**
  - Any variable is a term.
- **Rule 2: Constants**
  - Any constant is also a term.
- **Rule 3: Functions**
  - Any expression $f(t_1, \ldots, t_n)$ of n arguments is a term, where each argument t_i is a term, and f is a function symbol of valence n (n价函数).
  - In particular, symbols denoting individual constants are 0-*ary* function symbols (零元符号), and are thus terms.

Formation Rules: Formulas

- **Predicate symbols** 谓词符号
  - If P is an n-ary predicate symbol and t_1, …, t_n are terms, then $P(t_1, \ldots, t_n)$ is a formula.
- **Equality** 等量
  - If the equality symbol is considered part of logic, and t_1 and t_2 are terms, then $t_1 = t_2$ is a formula.
- **Negation** 否定
  - If φ is a formula, then $\neg \varphi$ is a formula.
- **Binary connectives** 二元连接
  - If φ and Ψ are formulas, then $(\varphi \Rightarrow \Psi)$ is a formula.
  - Similar rules apply to other binary logical connectives.
- **Quantifiers** 限量
  - If φ is a formula and x is a variable, then $\forall x \varphi$ and $\exists x \varphi$ are formulas.

Example: Wumpus World

Performance measure:

- **Gold**: +1000
- **Death**: - 1000
- **Taking a step**: - 1
- **Using an arrow**: - 10

Environment:

- **Stenchy** when adjacent to Wumpus.
- **Breezy** when adjacent to pit.
- **Glitter** iff gold is in same square.
- **Shooting** fire an arrow in a **straight line** in the direction the agent is looking.
  - The arrow **continues** until it either hits and kills the wumpus or hits the wall.
  - Shooting uses up **only arrow**.
- **Grabbing** picks up gold if in same square.
- **Releasing** drops the gold in same square.

Sensors:

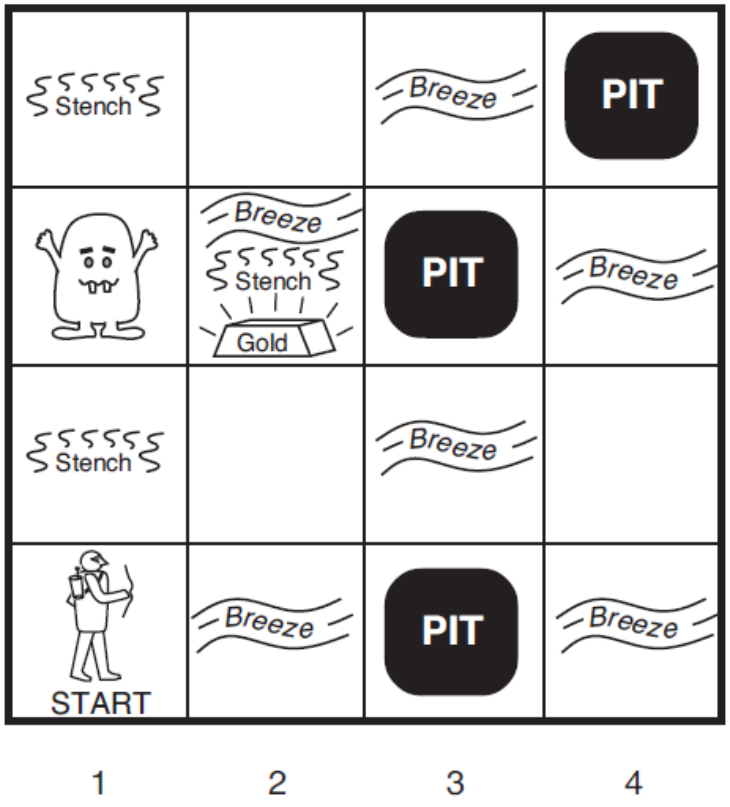
- **Stench**, **Breeze**, **Glitter**, **Bump**, **Scream**

Actuators:

- **Left turn**, **Right turn**, **Forward**, **Grab**, **Release**, **Shoot**

Goal:

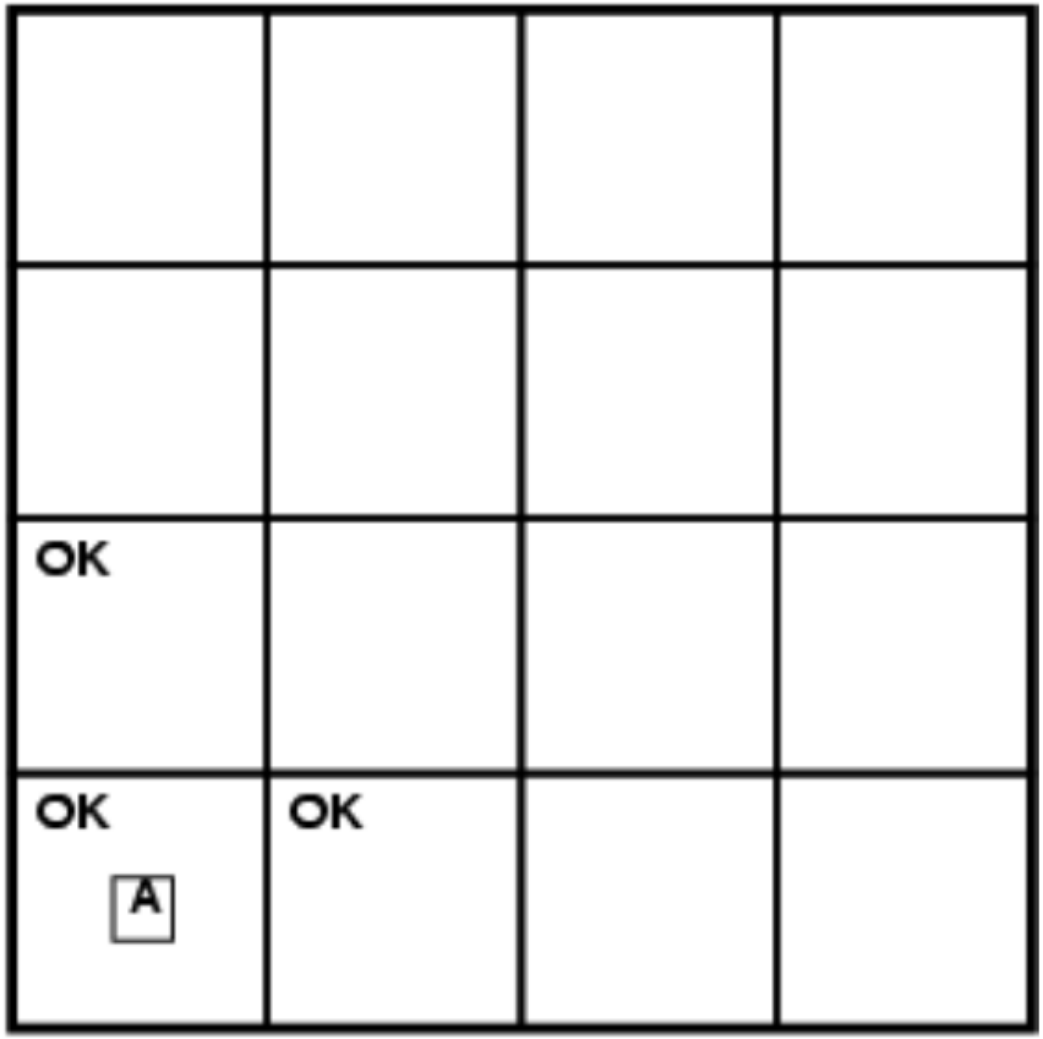
Bring back the gold with fewest cost, without getting killed.



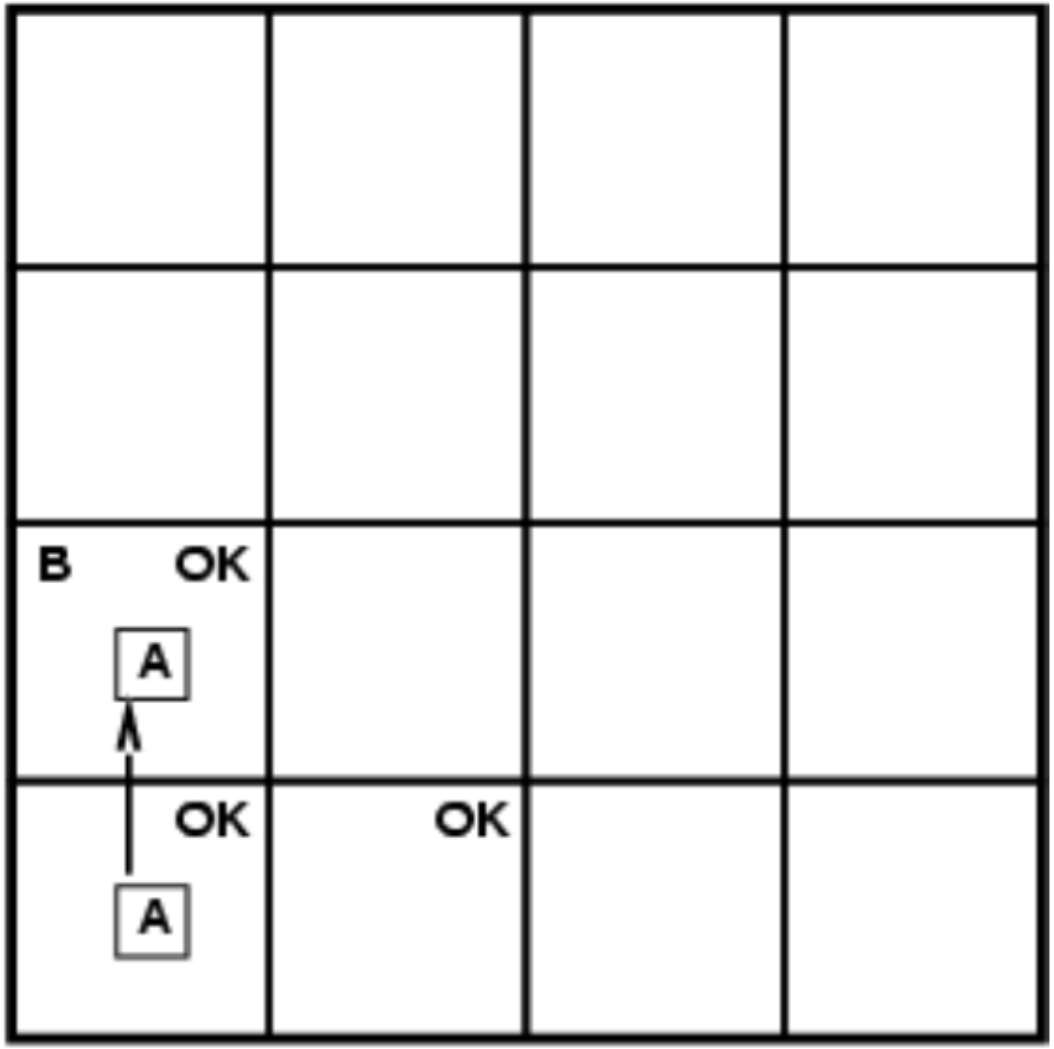
Example: Wumpus World

- Fully Observable?
  - No, only local perception.
- Deterministic?
  - Yes, outcomes are exactly specified.
- Episodic?
  - No, sequential at the level of actions.
- Static?
  - Yes, the Wumpus and pits do not move.
- Discrete?
  - Yes.
- Single-agent?
  - Yes, the Wumpus is essentially a natural feature.

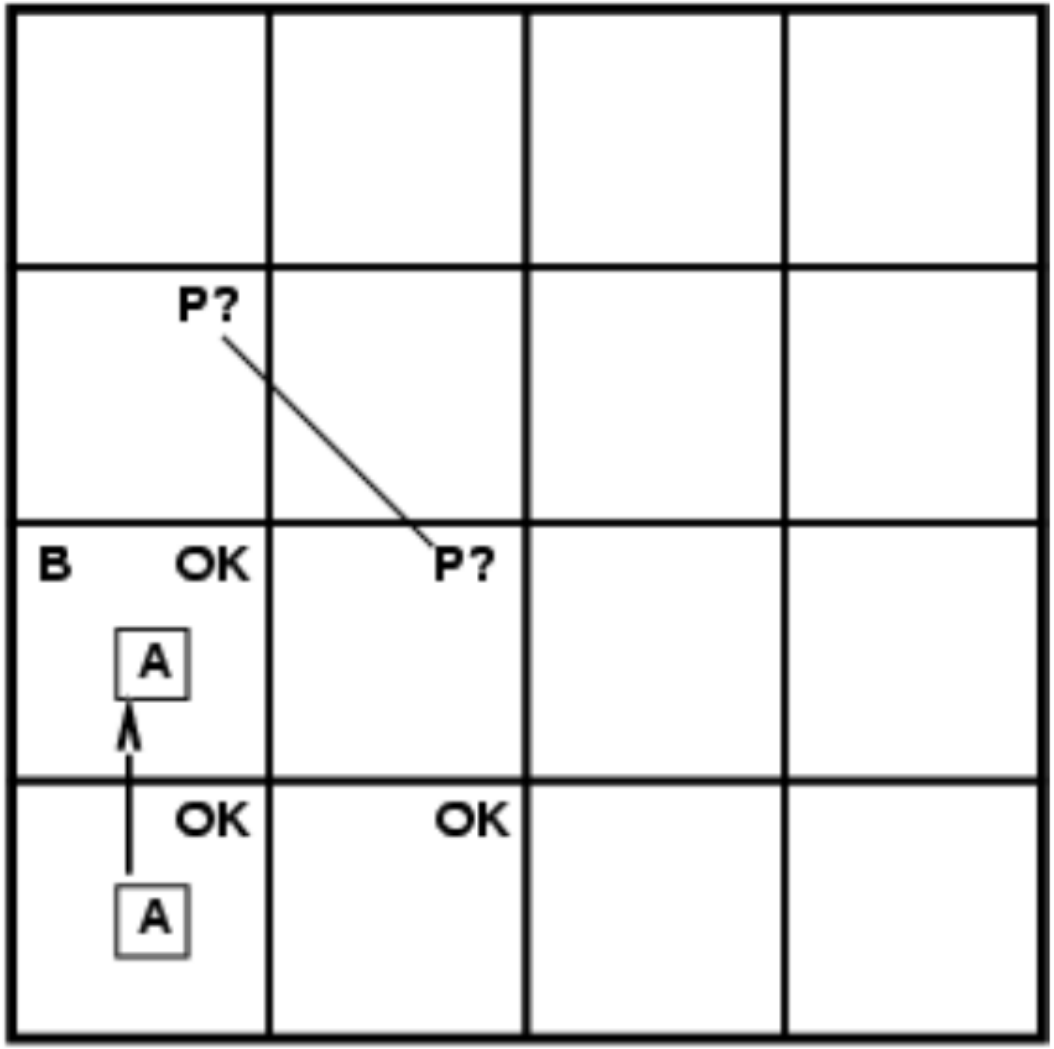
Example: Wumpus World



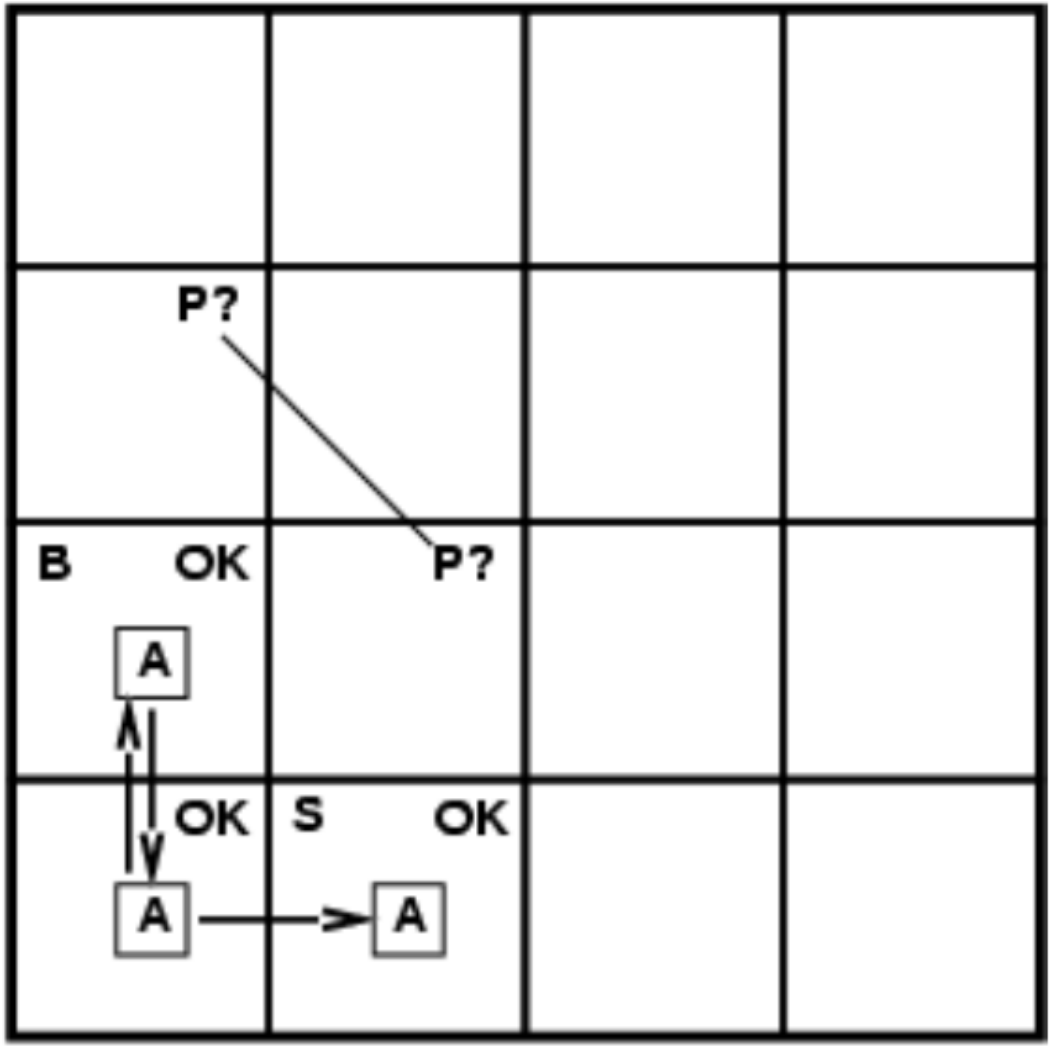
Example: Wumpus World



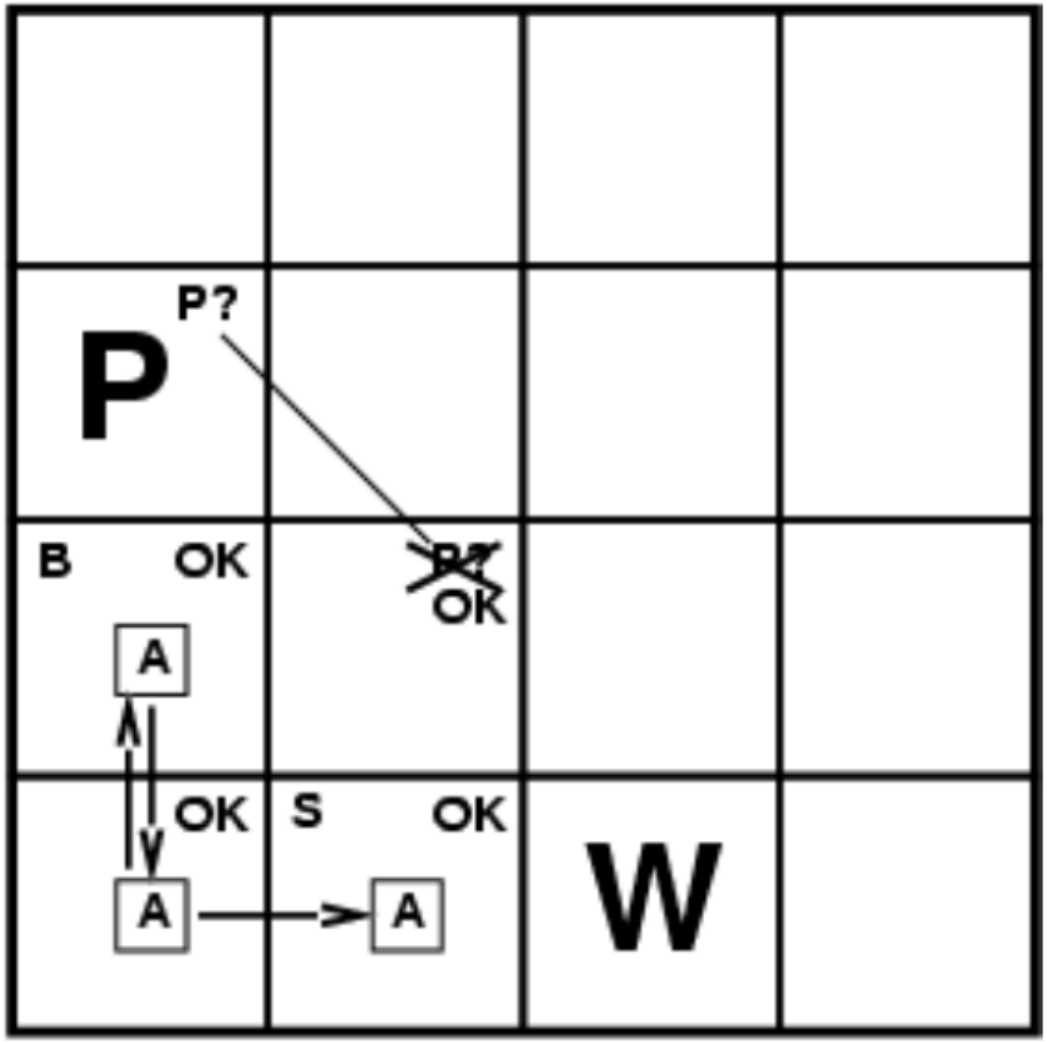
Example: Wumpus World



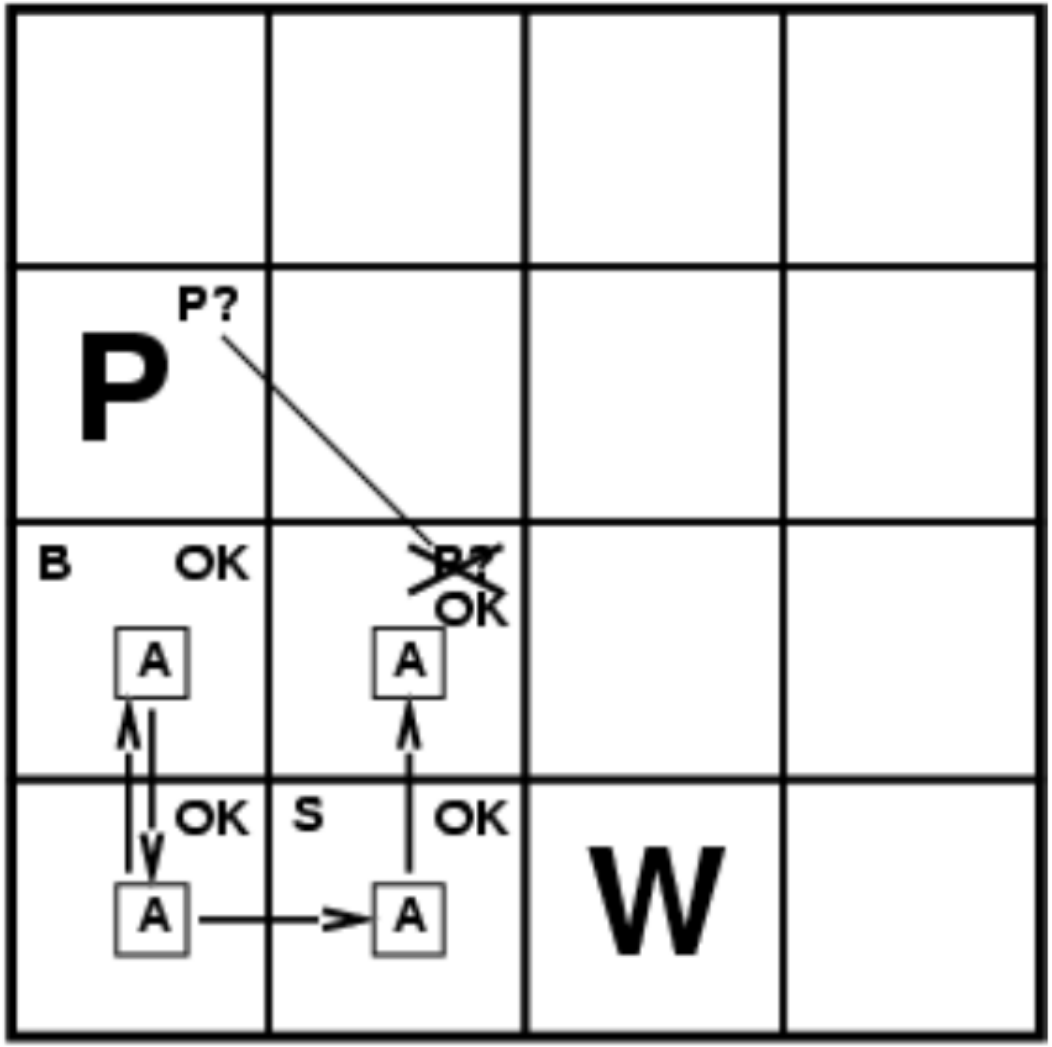
Example: Wumpus World



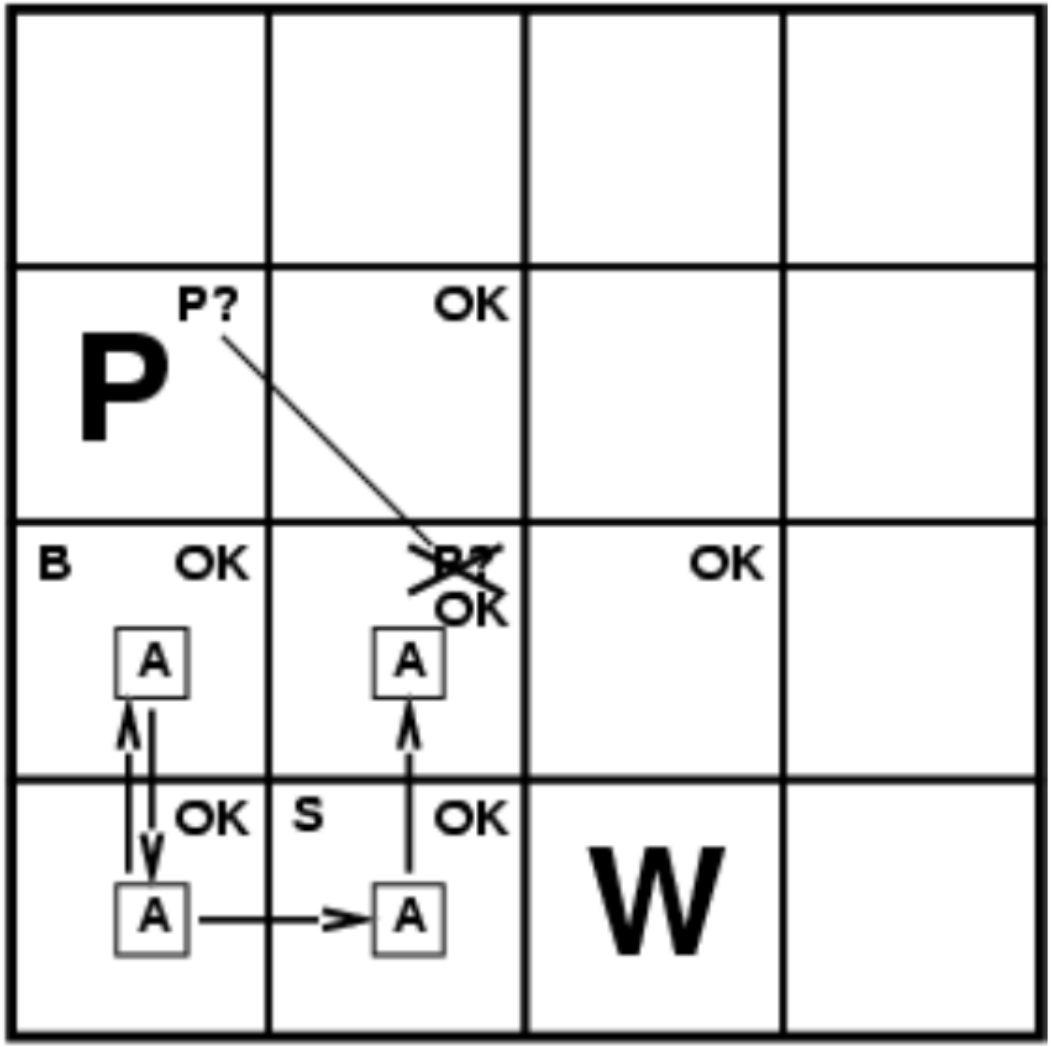
Example: Wumpus World



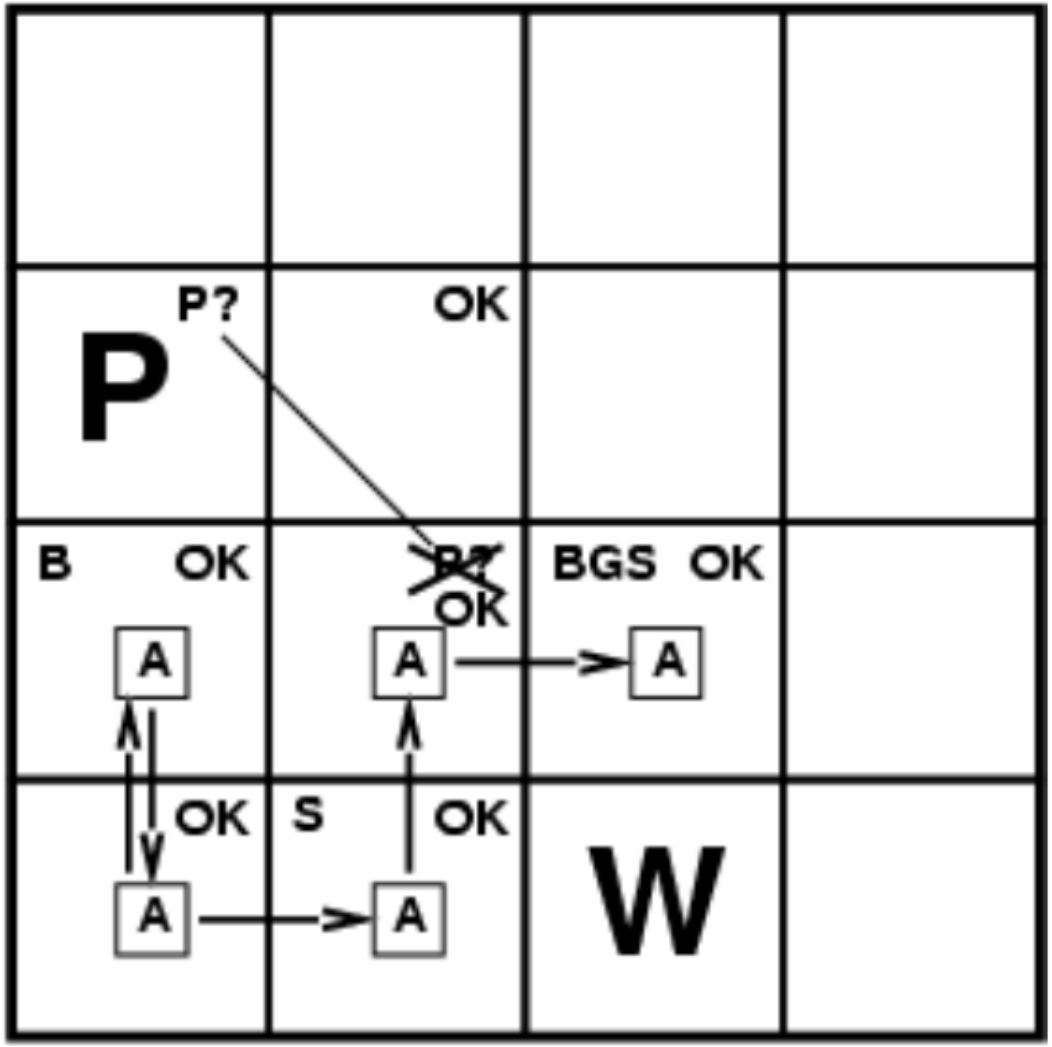
Example: Wumpus World



Example: Wumpus World



Example: Wumpus World

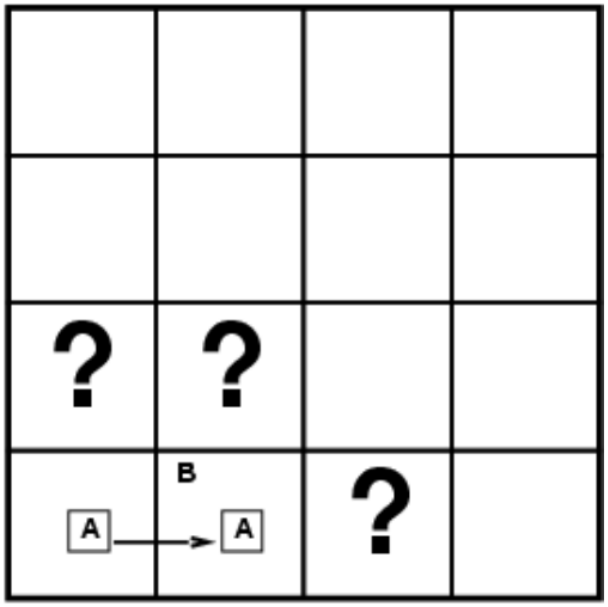


Models and Entailment

- A **model** assigns a value to all variables.
  - We can think of a model as a possible world.
  - In the semantics of arithmetic, $x + y = 4$:
    - is **true** in a world where x is 2 and y is 2,
    - but **false** in a world where x is 1 and y is 10.
  - We say m is a model of a sentence α if α is true in m.
  - $M(\alpha)$ is the set of all models of α.

- **Entailment** $\alpha \models \beta$
  - β follows logically from α.
  - In every model in which α is true in all worlds where β is true.
  - $\alpha \models \beta$ iff $M(\alpha) \subseteq M(\beta)$.

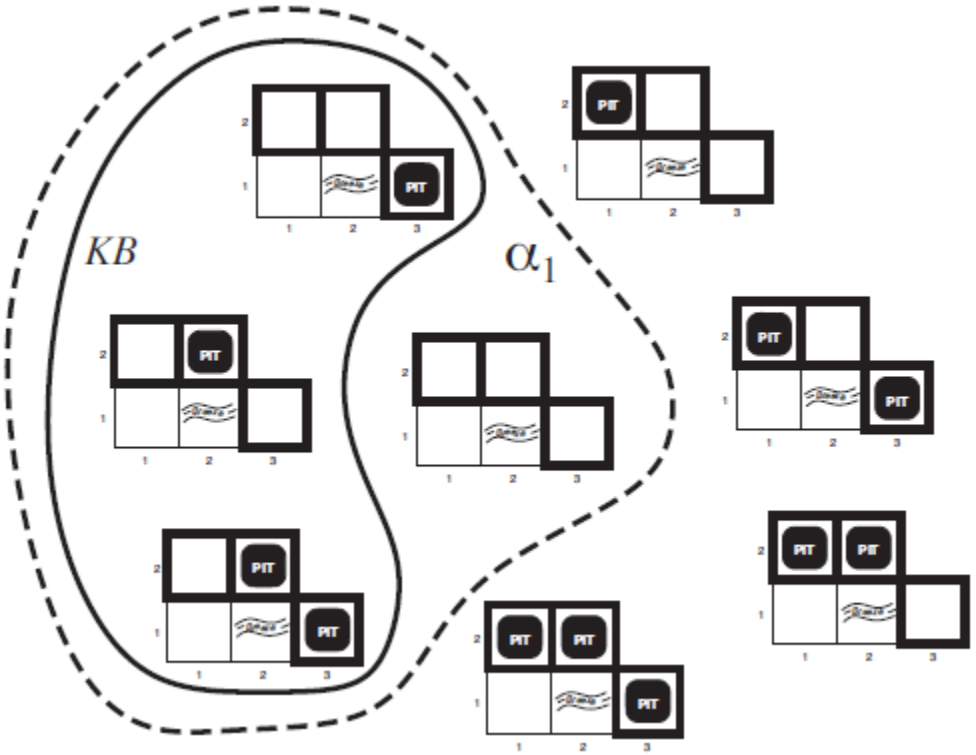
Entailment in Wumpus world



- You detect nothing in [1,1].
- You move right.
- You detect a breeze in [2,1].
- Consider possible models for KB assuming only pits:

  3 (Boolean) possibilities → 8 possible models.

Entailment in Wumpus world

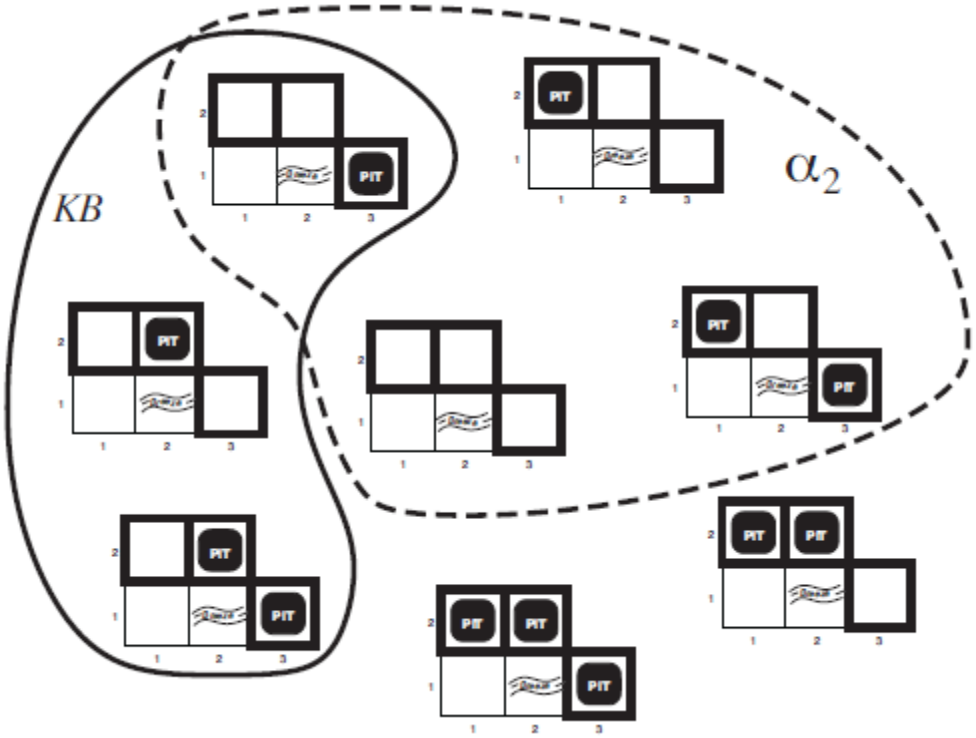


KB = Wumpus-world rules + observations

KB = Nothing in [1,1] and Breeze in [1,2]

α_1 = "[2,1] is safe." Does $KB \models \alpha_1$?

Entailment in Wumpus world



KB = Wumpus-world rules + observations

KB = Nothing in [1,1] and Breeze in [1,2]

α_2 = "[2,2] is safe." Does $KB \models \alpha_2$?

End of Chapter 1