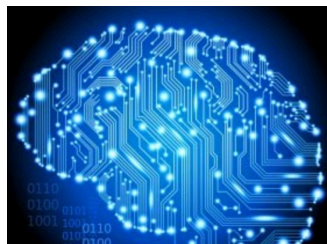# Lecture 8: Knowledge 3

http://cs.nju.edu.cn/yuy/course_ai18.ashx

Previously...

Propositional Logic

> PL-Forward chaining
> PL-Backward chaining
> PL-Resolution

First Order Logic (FOL)

> Instantiation
> FOL-Forward chaining
> FOL-Backward chaining
> FOL-Resolution

SAT problems

Propositional logic, CNF

literals: $x_1, x_2, \ldots, x_n$

clauses: $(x_1 \vee x_2 \vee x_5)$ $(\neg x_2 \vee x_3 \vee \neg x_7)$ **...**

problem: find an assignment to literals so that the conjunction of the clauses is true, or prove unsatisfiable

$$(x_1 \vee x_2 \vee x_5) \wedge (\neg x_2 \vee x_3 \vee \neg x_7) \wedge \ldots$$

2SAT: every clause has at most 2 literals
P-solvable

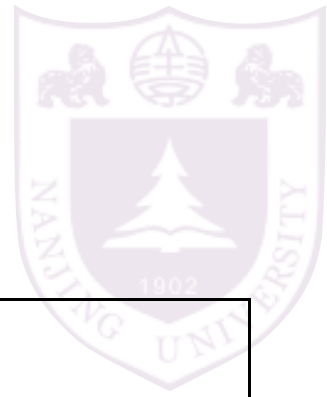3SAT: every clause has at most 3 literals
NP-hard

SAT solvers

SAT problems have many important applications

many SAT solvers are ready for use

DPLL

WalkSAT

DPLL

---

function DPLL-SATISFIABLE?(s) **returns** *true* or *false*
  inputs: s, a sentence in propositional logic

  $clauses \leftarrow$ the set of clauses in the CNF representation of s
  $symbols \leftarrow$ a list of the proposition symbols in s
  return DPLL($clauses, symbols, \{\ \}$)

---

function DPLL($clauses, symbols, model$) **returns** *true* or *false*

  if every clause in $clauses$ is true in $model$ **then return** *true*
  if some clause in $clauses$ is false in $model$ **then return** *false*
  $P, value \leftarrow$ FIND-PURE-SYMBOL($symbols, clauses, model$)
  if P is non-null **then return** DPLL($clauses, symbols - P, model \cup \{P{=}value\}$)
  $P, value \leftarrow$ FIND-UNIT-CLAUSE($clauses, model$)
  if P is non-null **then return** DPLL($clauses, symbols - P, model \cup \{P{=}value\}$)
  $P \leftarrow$ FIRST($symbols$); $rest \leftarrow$ REST($symbols$)
  return DPLL($clauses, rest, model \cup \{P{=}true\}$) **or**
      DPLL($clauses, rest, model \cup \{P{=}false\}$))

a deep-first search with heuristics

DPLL heuristics

Pure symbol heuristic: A **pure symbol** is a symbol that always appears with the same "sign" in all clauses.

$$(A \vee \neg B) \wedge (\neg B \vee \neg C) \wedge (C \vee A)$$

A and B is pure, but not C

Unit clause heuristic: A **unit clause** is a clause with just one literal.

$$(A \vee \neg B) \text{ with } A = \text{true}$$

is a unit clause

Other tricks

Component analysis : find disjoint subsets

Variable and value ordering : assign most frequent variable at first

Intelligent backtracking : remember conflicts

Random restart

Clever indexing

WalkSAT

a local search hill-climbing or others.

function WALKSAT(*clauses*, *p*, *max_flips*) **returns** a satisfying model or *failure*
    inputs: *clauses*, a set of clauses in propositional logic
            p, the probability of choosing to do a "random walk" move, typically around 0.5
            max_flips, number of flips allowed before giving up

    model ← a random assignment of *true*/*false* to the symbols in *clauses*
    for $i = 1$ **to** *max_flips* **do**
        if *model* satisfies *clauses* **then return** *model*
        clause ← a randomly selected clause from *clauses* that is false in *model*
        with probability *p* flip the value in *model* of a randomly selected symbol from *clause*
        else flip whichever symbol in *clause* maximizes the number of satisfied clauses
    return *failure*

failure ≠ unsatisfiable

The landscape of random SAT problems

Not all SAT instances are hard
under-constraint: a few clauses => easy to enumerate
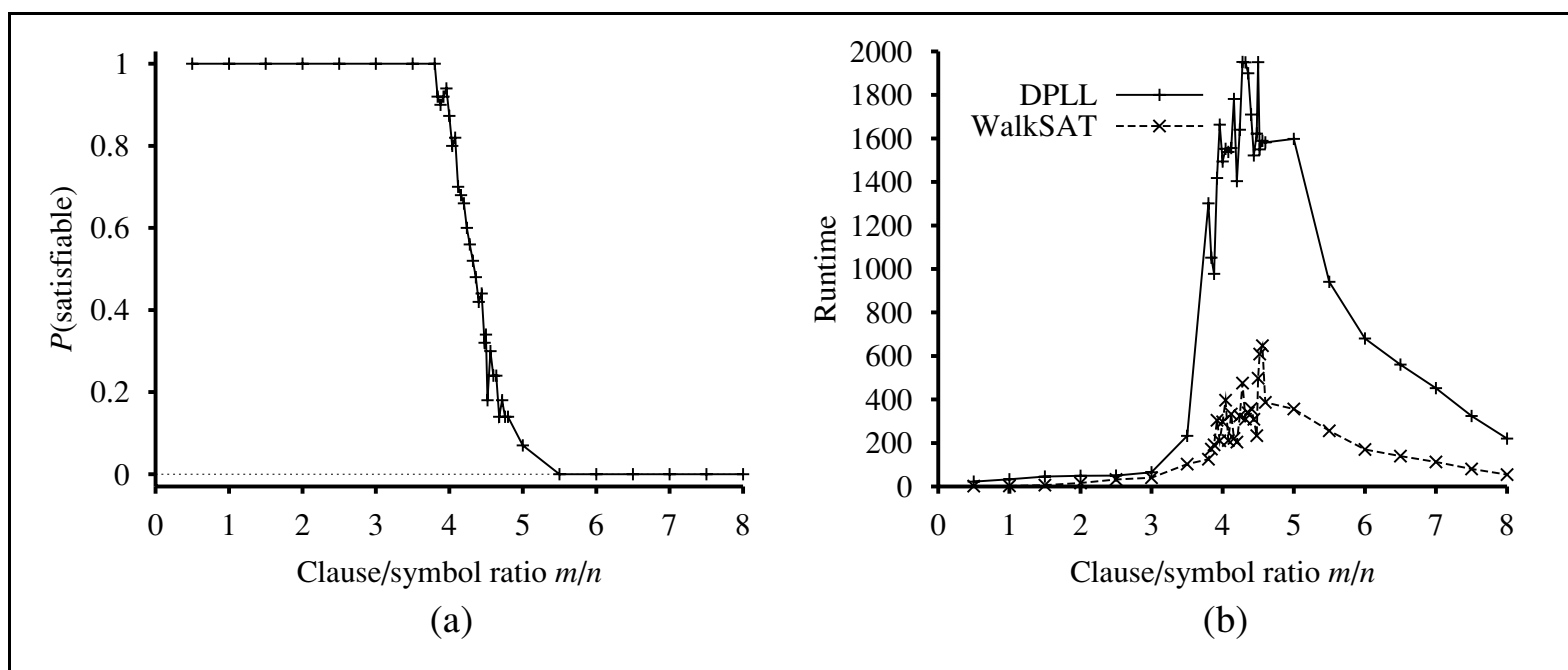over-constraint: too many clauses => unsatisfiable

Figure 7.19   (a) Graph showing the probability that a random 3-CNF sentence with $n = 50$ symbols is satisfiable, as a function of the clause/symbol ratio m/n. (b) Graph of the median run time (measured in number of recursive calls to DPLL, a good proxy) on random 3-CNF sentences. The most difficult problems have a clause/symbol ratio of about 4.3.

Planning

Language

There are many languages description the world
Planning Domain Definition Language
1.2, 2.1, 2.2, 3.0, 3.1

state s
Action(s)
Result(s,a)

$Action(Fly(p, from, to),$
$\quad \text{PRECOND:} At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$
$\quad \text{EFFECT:} \neg At(p, from) \wedge At(p, to))$

$Action(Fly(P_1, SFO, JFK),$
$\quad \text{PRECOND:} At(P_1, SFO) \wedge Plane(P_1) \wedge Airport(SFO) \wedge Airport(JFK)$
$\quad \text{EFFECT:} \neg At(P_1, SFO) \wedge At(P_1, JFK))$

Precondition

action **a** is **applicable** in state **s** if the preconditions are satisfied by **s**

$$(a \in \text{ACTIONS}(s)) \Leftrightarrow s \models \text{PRECOND}(a)$$

$$\forall p, from, to \ (Fly(p, from, to) \in \text{ACTIONS}(s)) \Leftrightarrow$$
$$s \models (At(p, from) \land Plane(p) \land Airport(from) \land Airport(to))$$

Result

removing the fluents that appear as negative literals in the action's effects (what we call the **delete list** or DEL(a)), and adding the fluents that are positive literals in the action's effects (what we call the **add list** or ADD(a))

$$\text{RESULT}(s, a) = (s - \text{DEL}(a)) \cup \text{ADD}(a) .$$

$Action(Fly(P_1, SFO, JFK),$
  $\text{PRECOND:} At(P_1, SFO) \wedge Plane(P_1) \wedge Airport(SFO) \wedge Airport(JFK)$
  $\text{EFFECT:} \neg At(P_1, SFO) \wedge At(P_1, JFK))$

Example

$Init(On(A, Table) \land On(B, Table) \land On(C, A)$
$\quad \land Block(A) \land Block(B) \land Block(C) \land Clear(B) \land Clear(C))$
$Goal(On(A, B) \land On(B, C))$
$Action(Move(b, x, y),$
$\quad$ PRECOND: $On(b, x) \land Clear(b) \land Clear(y) \land Block(b) \land Block(y) \land$
$\quad\quad (b{\neq}x) \land (b{\neq}y) \land (x{\neq}y),$
$\quad$ EFFECT: $On(b, y) \land Clear(x) \land \neg On(b, x) \land \neg Clear(y))$
$Action(MoveToTable(b, x),$
$\quad$ PRECOND: $On(b, x) \land Clear(b) \land Block(b) \land (b{\neq}x),$
$\quad$ EFFECT: $On(b, Table) \land Clear(x) \land \neg On(b, x))$

Figure 10.3   A planning problem in the blocks world: building a three-block tower. One solution is the sequence $[MoveToTable(C, A), Move(B, Table, C), Move(A, Table, B)]$.
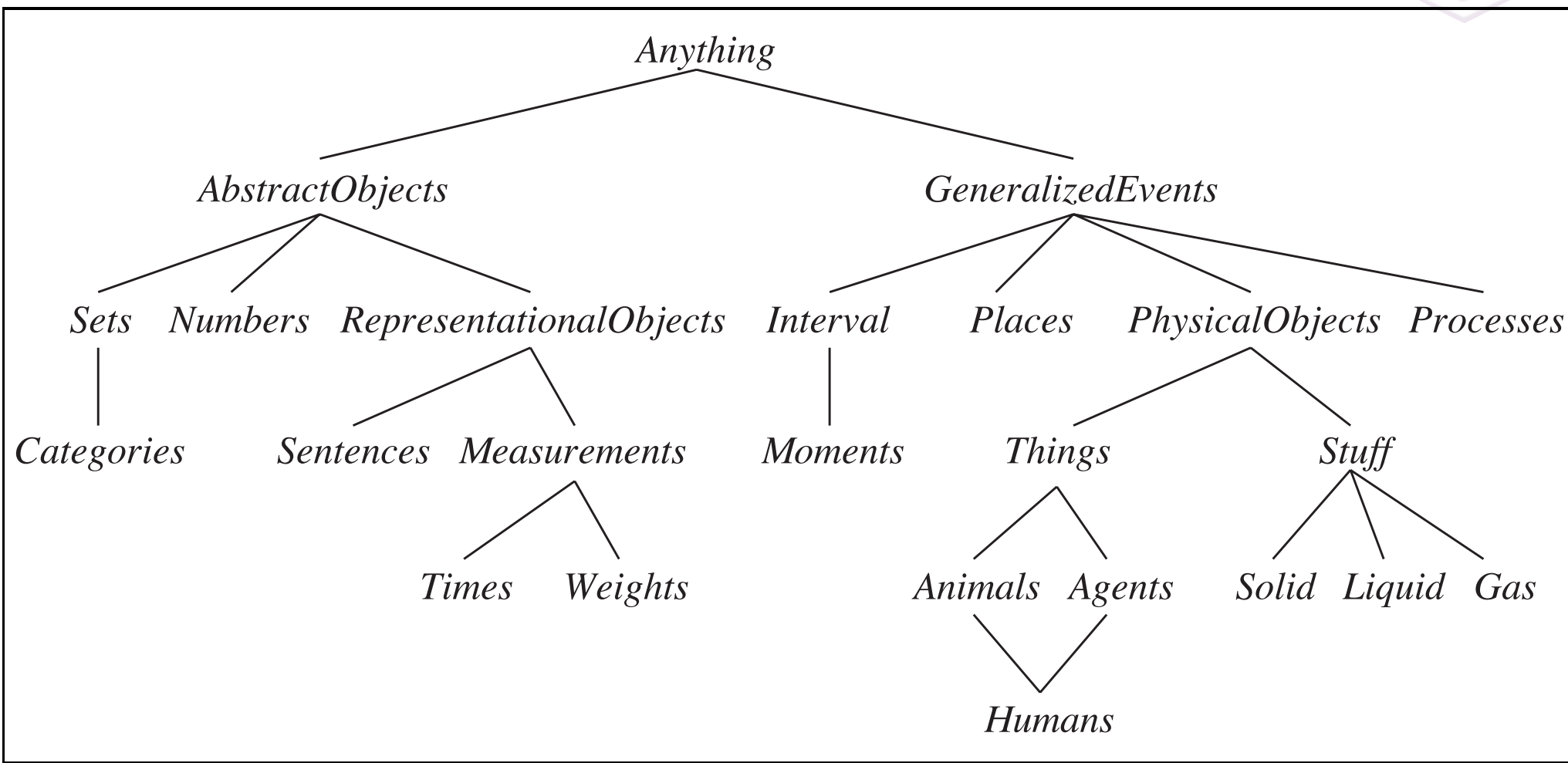

Start State

Goal State

Figure 10.4   Diagram of the blocks-world problem in Figure 10.3.

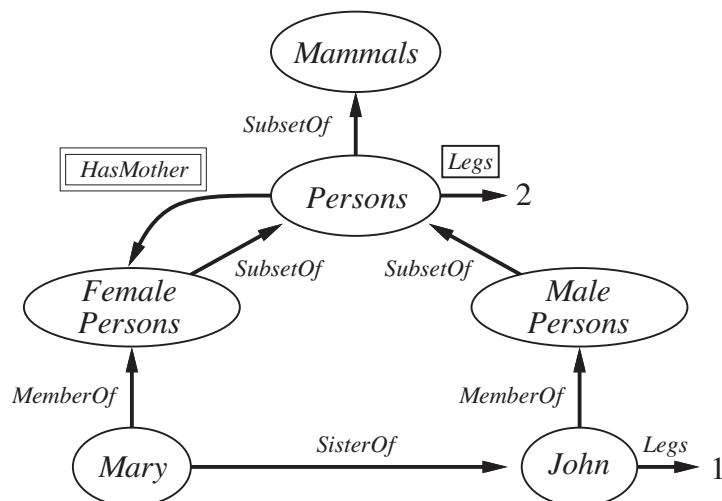# Ontology and Semantic Web

Up ontology

Domain ontology

Figure 12.5 A semantic network with four objects (John, Mary, 1, and 2) and four categories. Relations are denoted by labeled links.
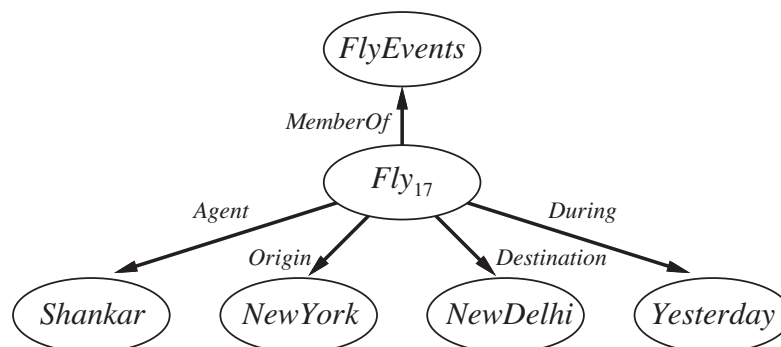

Figure 12.6 A fragment of a semantic network showing the representation of the logical assertion $Fly(Shankar, NewYork, NewDelhi, Yesterday)$.

Example: Wordnet

Hamburger

- Hamburger (an inhabitant of Hamburg)
    - direct hypernym:
        - German (a person of German nationality)
    - sister term
        - German (a person of German nationality)
            - East German (a native/inhabitant of the former GDR)
            - Bavarian (a native/inhabitant of Bavaria)
    - derivationally related form
        - Hamburg (a port city in northern Germany on the Elbe
          River that was founded by Chalemagne in the…)

[from wikipedia]

Semantic web

- handling complex and heterogeneous information resources
- retrieving documents based on a set of relationships that are external to these documents
- providing multiple search options for richer investigation
- targeting and sifting results more efficiently
- using authoritative information resources more effectively as guides to searching

An RDF graph

Freebase

Freebase API (Deprecated)
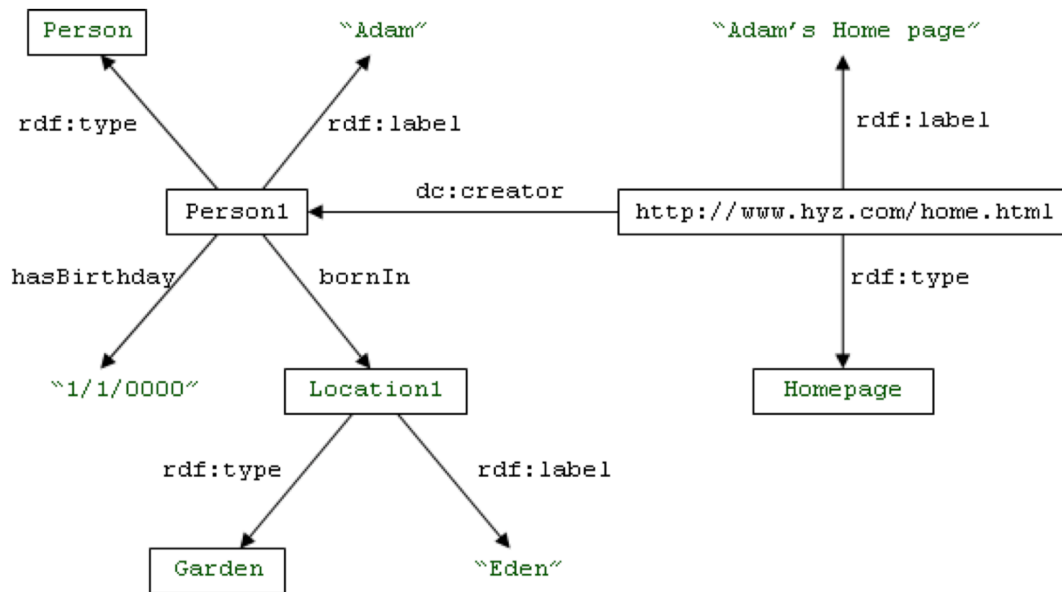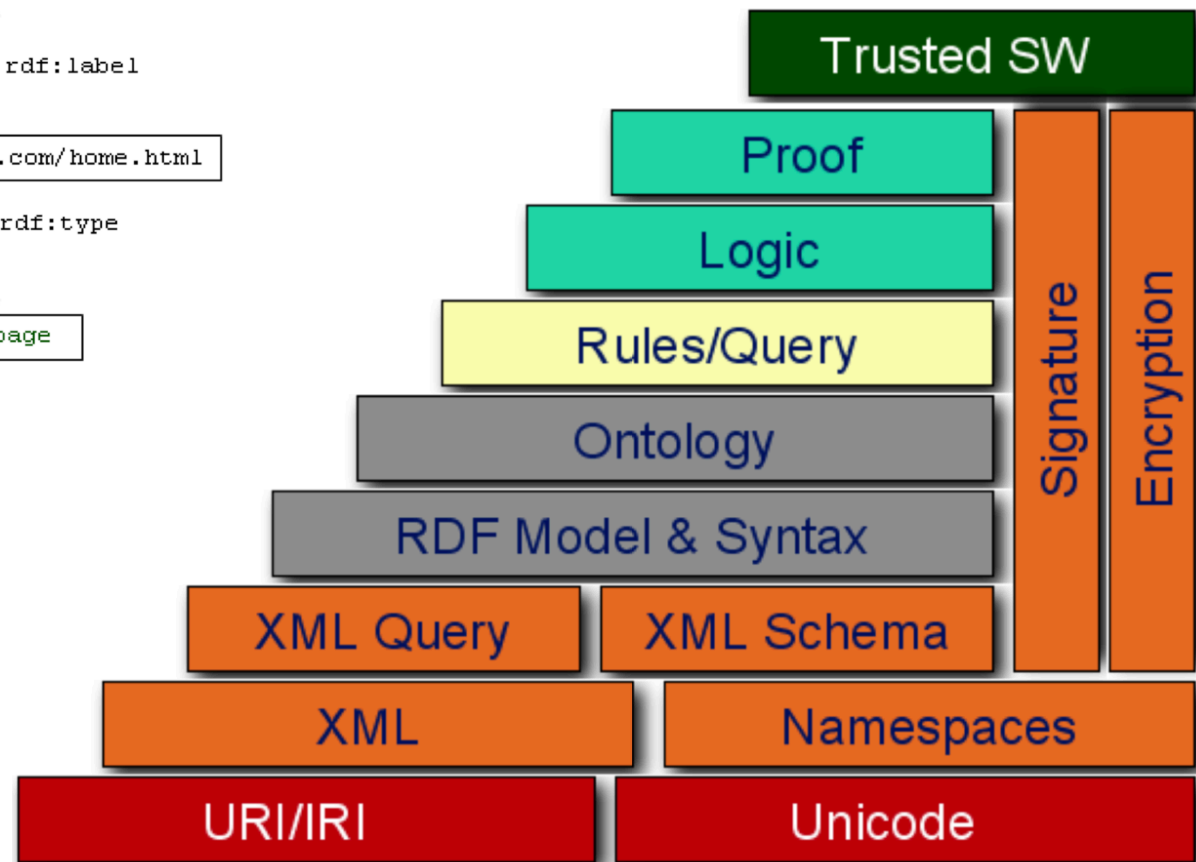
Hey there! Are you maybe looking for **Firebase** instead?

Data Dumps

⚠️ The Freebase API will be completely shut-down on Aug 31 2016. This page provides access to the last available data dump. Read more.

Data Dumps are a downloadable version of the data in Freebase. They constitute a snapshot of the data stored in Freebase and the Schema that structures it, and are provided under the same CC-BY license. The Freebase/Wikidata mappings are provided under the CC0 license.

Freebase Triples

This dataset contains every fact currently in Freebase.	**Total triples:** 1.9 billion	**22 GB** gzip	⬇ DOWNLOAD
	Updated: Weekly	250 GB	
	Data Format: N-Triple		
	License: CC-BY		

The RDF data is serialized using the N-Triples format, encoded as U...

RDF

```
<http://rdf.freebase.com/ns/g.11vjz1ynm>  <http://rdf.fr...
<http://rdf.freebase.com/ns/g.11vjz1ynm>  <http://rdf.f...
<http://rdf.freebase.com/ns/g.11vjz1ynm>  <http://rdf.f...
<http://rdf.freebase.com/ns/g.11vjz1ynm>  <http://rdf.f...
<http://rdf.freebase.com/ns/g.11vjz1ynm>  <http://www.w3...
```

⭐ If you're writing your own code to parse the RDF dumps its often m... extracting the data first and then processing the uncompressed da...

```
<subject>  <predicate>  <object> .
```


Note: In Freebase, objects have MIDs that look like `/m/012rkqx`. In... Freebase schema like `/common/topic` are written as `common.topi...`

The *subject* is the ID of a Freebase object. It can be a Freebase MID... readable ID (ex. `common.topic`) for schema.

The *predicate* is always a human-readable ID for a Freebase proper... RDFS. Freebase foreign key namespaces are also used as predicates to make it easier to look up keys by namespace.

The object field may contain a Freebase MID for an object or a human-readable ID for schema from Freebase or other RDF vocabularies. It may also include literal values like strings, booleans and numeric values.
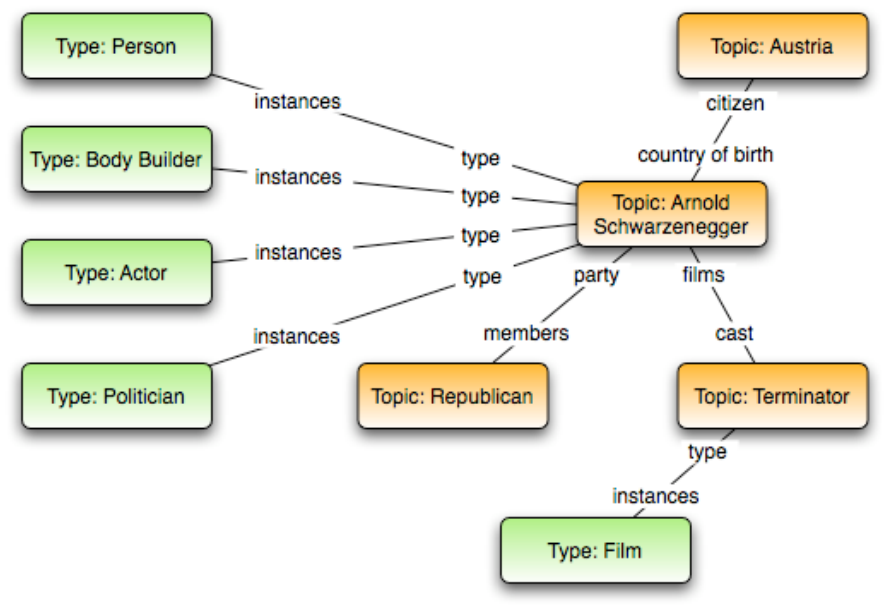
WikiData

WIKIDATA

Main page
Community portal
Project chat
Create a new item
Recent changes
Random item
Query Service
Nearby
Help
Donate

Print/export

Create a book
Download as PDF
Printable version

open

multilingual

free

collaborative

Welcome to Wikidata

the free knowledge base with 47,001,953 data items that anyone can edit.

Introduction · Project Chat · Community Portal · Help

Want to help translate? Translate the missing messages.

label ——— # Douglas Adams (Q42) ——— **item identifier**

description ——— English writer and humorist
Douglas Noël Adams | Douglas Noel Adams ——— **aliases**

▶ In more languages

Statements

property — educated at | **value** — St John's College

end time	1974
academic major	English literature
academic degree	Bachelor of Arts
start time	1971

qualifiers

rank —

▼ 2 references

stated in	Encyclopædia Britannica Online
reference URL	http://www.nndb.com/people/731/000023662/
original language of work	English
retrieved	7 December 2013
publisher	NNDB
title	Douglas Adams (English)

opened references

+ add reference

statement group

Brentwood School

end time	1970
start time	1959

▶ 0 references — **collapsed reference**

+ add (statement)

Welcome!

Wikidata is a free and open knowledge base that can be read and edited by both humans and machines.

Wikidata acts as central storage for the **structured data** of its Wikimedia sister projects including Wikipedia, Wikivoyage, Wikisource, and others.

Wikidata also provides support to many other sites and services beyond just Wikimedia projects! The content of Wikidata is available under a free license, exported using standard formats, and can be interlinked to other open data sets on the linked data web.

Get involved

Learn about Wikidata

- What is Wikidata? Read the **Wikidata introduction**.
- Explore Wikidata by looking at a featured showcase item for author **Douglas Adams**.
- Get started with Wikidata's **SPARQL query service**.

Contribute to Wikidata

- Learn to edit Wikidata: follow the **tutorials**.
- Work with other volunteers on a subject that interests you: **join a WikiProject**.
- Individuals and organisations can also **donate data**.

Meet the Wikidata community

- Visit the **community portal** or attend a **Wikidata event**.
- Create a **user account**.

Learn about data

New to the wonderful world of data? Develop and improve your data literacy through content designed to get you up to speed and feeling comfortable with the fundamentals in no time.

item: *Earth (Q2)*   property: *highest point (P610)*   custom value: *Mount Everest (Q513)*

Popular items

- 2018 Toronto van attack (Q52152274)
- 2018 Giro dell'Appennino (Q51687919)
- Liège–Bastogne–Liège for Women 2018 (Q42116955)
- Saleh Ali al-Sammad (Q19429078)
- Marguerite Rouvière (Q51954596)
- Karen Karapetyan (Q1979923) (pictured)
- Semiramis Hotel bombing (Q2086163)

Discover

Example application