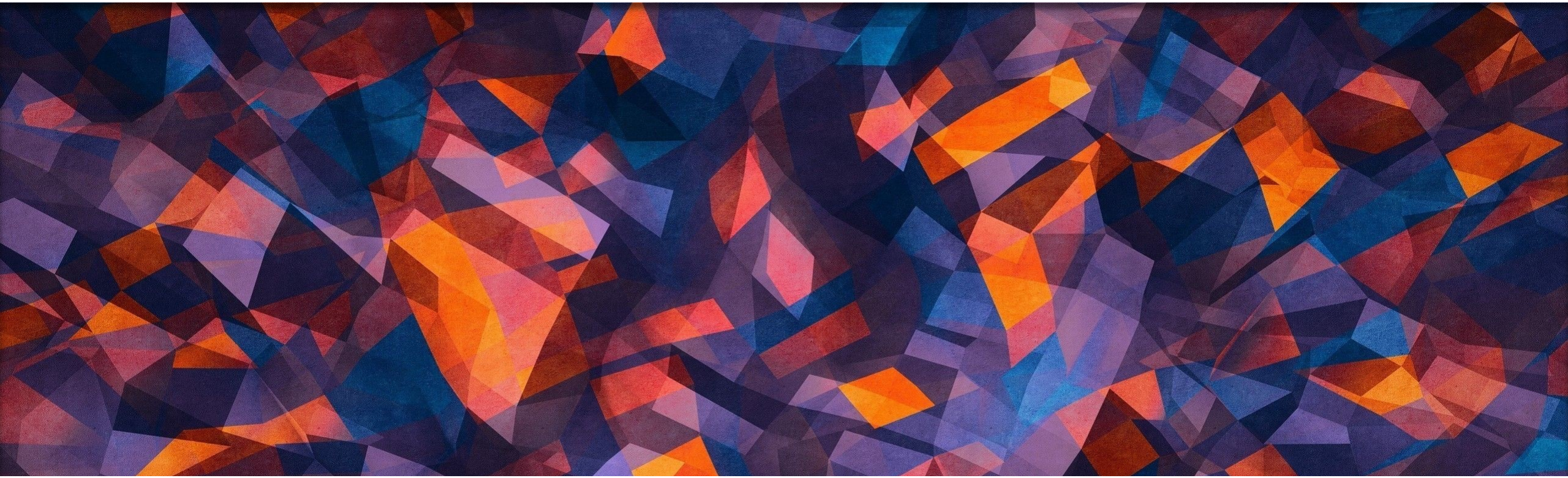
Universität St.Gallen



Web-based Autonomous Systems

Knowledge Representation and Reasoning for the Web

Chair for Interaction- and Communication-based Systems (ICS-HSG)

Our Journey

Universität St.Gallen

Prerequisites:
- **ASSE**
- **(…)**

Week 1:
Introduction

Week 2:
A Web for Machines

Week 3:
Knowledge Representation and Reasoning for the Web

Week 4:
Linked Data and Distributed Knowledge Graphs

Week 5:
Hypermedia Agents (Arch. and Programming)

Week 6 (Coordination I):
Agent Communication and Interaction

Week 7 (Coordination II):
Self-Organization and Stigmergy

Week 8 (Coordination III):
Normative MAS and Organizations

Week 9 (Coordination IV):
Trust & Reputation

Week 10:
Game Theory and Social Choice

Week 11:
Reinforcement Learning and Multi-Agent Learning

Week 12:
An Industry Perspective

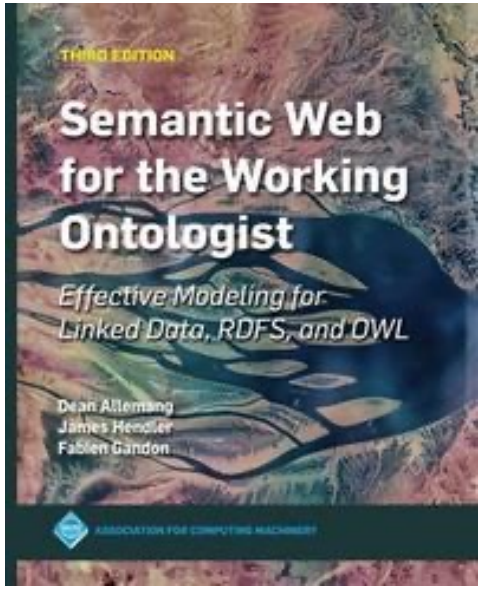
Exercises

Ex1: Writing Your First Agent(s)!
Ex2: Automated Planning
Ex3: Web Ontologies
Ex4: Deductive Reasoning on the Web
Ex5: Cognitive Agents in Hypermedia Env.
Ex6: Interacting Agents on the Web
Ex7: Ant Colony Optimization
Ex8: Organized Agent
Ex9: Trustworthy Agents
Ex10: Axelrod's Agents
Ex11: Reinforcement Learning Agents
Course Review and Q&A

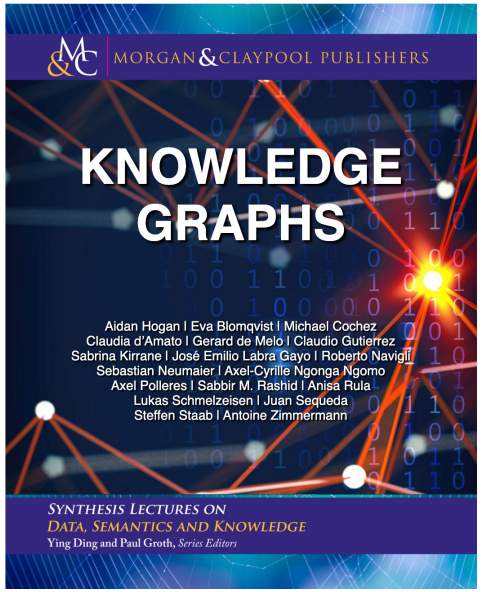
Universität St.Gallen

Today's Agenda

- Web Ontologies
  - Description Logics and OWL
  - Building Web Ontologies with Protégé (Tutorial)
- Knowledge Graphs
  - Graph Data Models
  - Querying Graph Datasets



[Allemang et al., 2020]



https://kgbook.org/

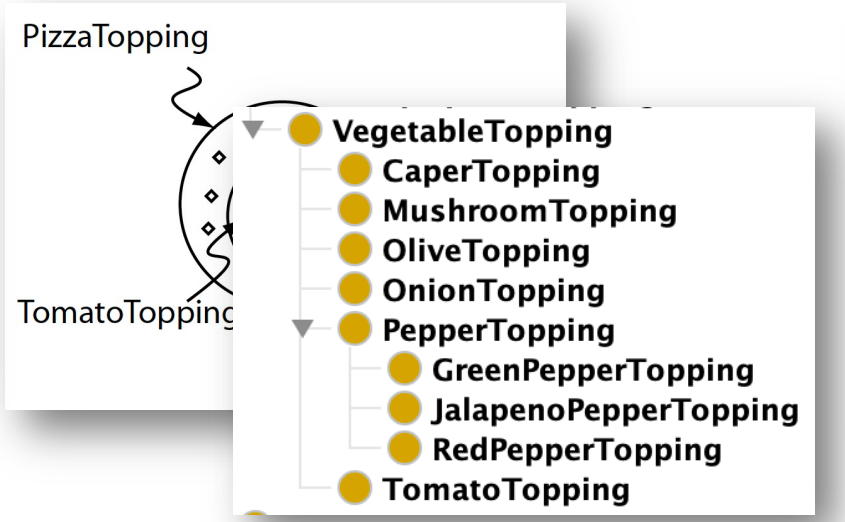
Chapter 1
Chapter 2
Chapter 4
(Chapter 5)

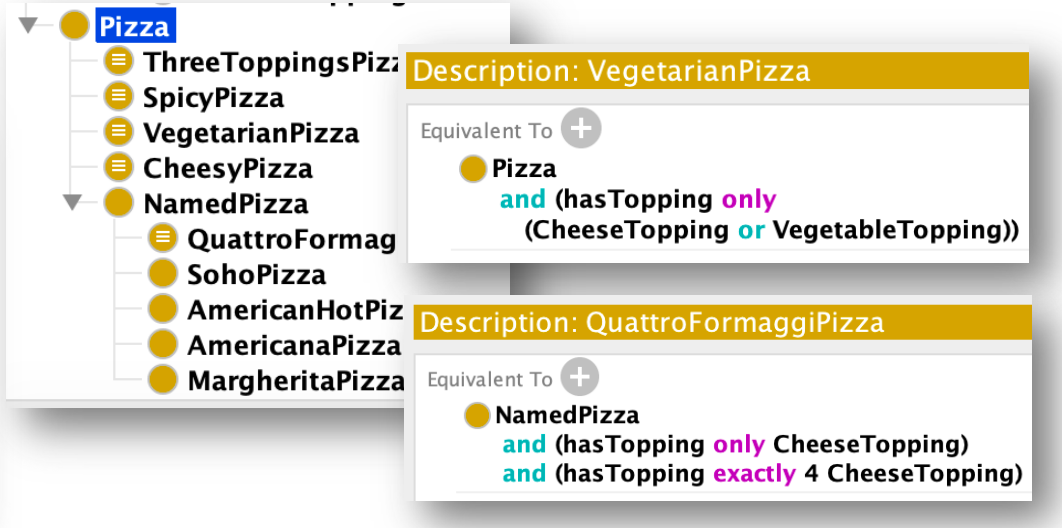
Aidan Hogan et al. Knowledge Graphs. 2021.
D. Allemang, J. Hendler, F. Gandon. Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS, and OWL (Third Edition), ACM Books, 2020.

Ontologies

An ontology is the **formal**, **explicit specification** of a **shared conceptualization** of an **area of interest** [Studer et al., 1998].

- **formal**: using a formal knowledge representation language, such as the **Web Ontology Language (OWL)**
- **explicit specification**: the conceptual model is represented explicitly
- **shared**: usually defined and adopted by a group of people / a community
- **conceptualization**: abstract simplified view (conceptual model)
- **area of interest**: ex., pizza





http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf

R. Studer, V.R. Benjamins, and D. Fensel. Knowledge Engineering: Principles and Methods, Data & Knowledge Engineering, 1998

Last Week on First-Order Logic…

A glimpse of reasoning in first-order logic

Universität St.Gallen

The Dreadbury Mansion Mystery

1. Someone who lives in Dreadbury Mansion killed Aunt Agatha.
2. Agatha, the butler, and Charles live in D
3. A killer always hates his victim, and is r
4. Charles hates no one that Aunt Agatha
5. Agatha hates everyone except the butl
6. The butler hates everyone not richer th
7. The butler hates everyone Aunt Agatha
8. No one hates everyone.
9. Agatha is not the butler.

Who killed Aunt Agatha?

Web-based Autonomous Systems — Institute of Computer Science (ICS-HSG

Universität St.Gallen

A Glimpse of Reasoning in First-Order Logic

A killer always hates his victim and is never richer than his victim.

- $\forall x, y(killed(x, y) \rightarrow hates(x, y))$
- $\forall x, y(killed(x, y) \rightarrow \neg richer(x, y))$

Charles hates no one that Aunt Agatha hate

- $\forall x(hates(charles, x) \rightarrow \neg hates(agatha,$

$$\frac{killed(x, y) \rightarrow hates(x, y}{killed(cha}$$

Agatha hates everyone except the butler.

- $\forall x(\neg hates(agatha, x) \leftrightarrow x = butler))$

$$\frac{killed(charles, y) \rightarrow \neg hates}{killed}$$

Agatha is not the butler.

- $\neg agatha = butler$

$$\frac{killed(charles, y)}{\neg k}$$

Web-based Autonomous Systems — Institute of Computer Science (ICS-HSG)

Syllogism:

max planck institut informatik

spass

WebSPASS - Interactive SPASS Input Form Submission

Go to: Upload Form | Interactive Input Form | Help

You are running 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.109 Safari/537.36' from '51.154.106.193'

Your WebSPASS Input Form Submission is now being processed...

```
----------SPASS-START----------
Input Problem:
1[0:Inp] ||  -> lives(skc1)*.
2[0:Inp] ||  -> lives(a)*.
3[0:Inp] ||  -> lives(b)*.
4[0:Inp] ||  -> lives(c)*.
```

```
SPASS V 3.9
SPASS beiseite: Proof found.
Problem: /tmp/webspass-webform_2022-03-08_00:47:02_17434l.txt
SPASS derived 44 clauses, backtracked 11 clauses, performed 2 splits and kept 51 clauses.
```

```
list_of_formulae(conjectures).

formula(killed(a,a)).

end_of_list.
```

First-order Logic is very **expressive**, so why not use it for the Semantic Web?

→ computationally expensive

https://spass-prover.org/

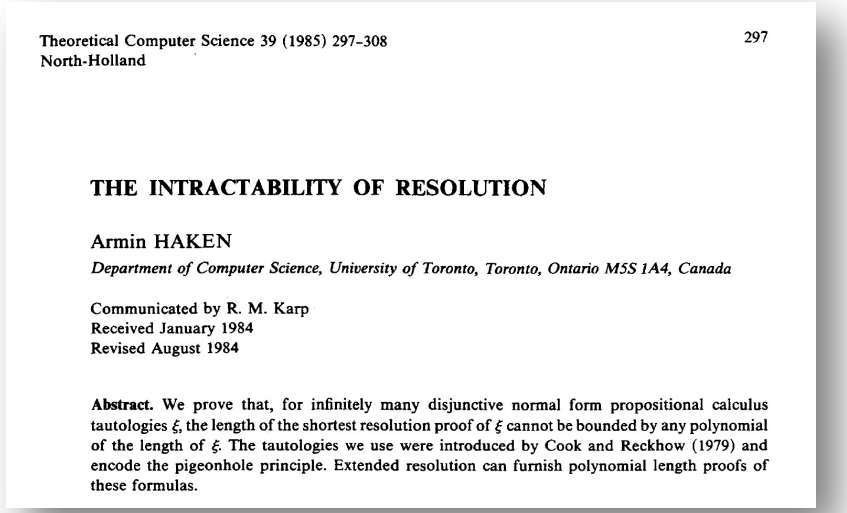
Universität St.Gallen

Description Logics and OWL

Reasoning in First-Order Logic is **semi-decidable** and **intractable**

There are algorithms that:

- can always prove that a true formula is true
- never say that a false formula is false
- BUT there will **always** be formulas that are false for which the algorithm does not answer (for instance, runs forever)



Theoretical Computer Science 39 (1985) 297–308
North-Holland

297

THE INTRACTABILITY OF RESOLUTION

Armin HAKEN
Department of Computer Science, University of Toronto, Toronto, Ontario M5S 1A4, Canada

Communicated by R. M. Karp
Received January 1984
Revised August 1984

Abstract. We prove that, for infinitely many disjunctive normal form propositional calculus tautologies ξ, the length of the shortest resolution proof of ξ cannot be bounded by any polynomial of the length of ξ. The tautologies we use were introduced by Cook and Reckhow (1979) and encode the pigeonhole principle. Extended resolution can furnish polynomial length proofs of these formulas.

Description Logics (DLs) are fragments of First-Order Logic

- designed to achieve ***favorable trade-offs*** between **expressivity** and **performance**

  ⇒ **decidability** and **tractability** depend on the **expressive power** of the DL at hand

more limited s.t. better cost

The **Web Ontology Language (OWL)** was strongly influenced by Description Logics

↳ DARPA: Agents that have to share logic

Armin Haken, The Intractability of Resolution, Theoretical Computer Science, Volume 39, 1985.
R.J. Brachman, H.J. Levesque. Knowledge Representation and Reasoning. Elsevier. 2004.

Universität St.Gallen

Description Logics

Description Logics (DLs) are a family of logics that represent knowledge of an application domain (the "world") in terms of:

Concepts or **classes** describe sets of things
- $FacultyMember, Employee, Person, University, Country, \ldots$

Roles or **properties** describe binary relations among things
- $employedBy, locatedIn, \ldots$

Individuals are instances of classes
- $Andrei, HSG, Switzerland, \ldots$

Terminological vocabulary

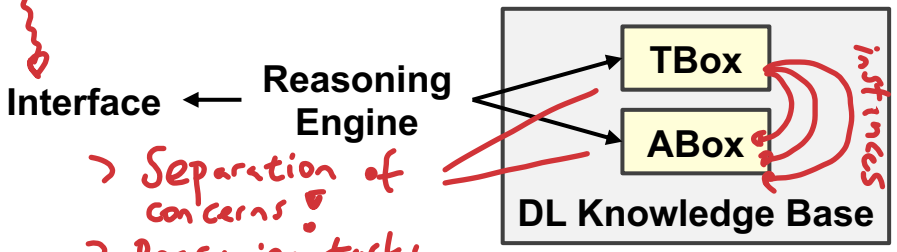
Assertional vocabulary

Three kinds of **formulas** or **axioms** are common to all DLs:

- *concept inclusion*: $C \sqsubseteq D$, where C and D are names of concepts
- *concept assertion*: $C(a)$, where C is a concept name and a is the name of an individual
- *role assertion*: $R(a, b)$, where R is a role name, and a and b are names of individuals

Terminological Knowledge (TBox)

Assertional Knowledge (ABox)



$$TBox = \{ FacultyMember \sqsubseteq Employee, Employee \sqsubseteq Person \}$$

$$ABox = \{ FacultyMember(Andrei), University(HSG), employedBy(Andrei, HSG) \}$$

A. Zimmermann, Knowledge Representation and Reasoning, MINES Saint-Étienne, 2021.
F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Universität St.Gallen

DLs: A Family of Logics

- [$\mathcal{EL}$] Existential Language
  - concept intersection ($C \sqcap D$), existential restrictions ($\exists R.C$)
- [$\mathcal{ALC}$] Attributive Language with Complement
  - concept intersection ($C \sqcap D$), concept union ($C \sqcup D$), concept negation ($\neg C$), universal restrictions ($\forall R.C$), and existential restrictions ($\exists R.C$)
  - [$\mathcal{S}$] extends $\mathcal{ALC}$ with transitive closure ($Trans(R)$)

The label of the logic encodes its expressivity: $\mathcal{ALCN}$, etc.

Extensions:

- [$\mathcal{H}$] adds role inclusion ($R \sqsubseteq S$)
  - [$\mathcal{R}$] adds (limited) complex role inclusion ($R_1 \circ R_2 \sqsubseteq S$), role reflexivity ($Ref(R)$), role irreflexivity ($Irref(R)$), role disjointness ($Disj(R,S)$), and the universal role (U)
- [$\mathcal{O}$] adds (limited) nominals ($\{a_1, \ldots, a_n\}$)
- [$\mathcal{I}$] adds inverse roles (R^-)
- [$\mathcal{F}$] adds (limited) functional roles ($Func(R)$)
  - [$\mathcal{N}$] adds (limited) number restrictions ($* R$, where $* \in \{\geq, \leq, =\}$)
    - [$\mathcal{Q}$] adds (limited) qualified number restrictions ($* R.C$, where $* \in \{\geq, \leq, =\}$)
- [$(\mathcal{D})$] adds datatype properties, data values, or data types

https://kgbook.org/#sssec-dls

Universität St.Gallen

TBox Examples in $\mathcal{ALC}$

Atomic concepts: **Person**, **Female**, **Dragon**

Atomic roles: **hasChild**

Concepts are constructed according to the following syntax rule:

$C, D ::= A \mid$ (atomic concept)

$\top \mid$ (top concept) — root

$\bot \mid$ (bottom concept) — nothing

$\neg C \mid$ (atomic negation) — subclass of all other concepts

$C \sqcap D \mid$ (intersection)

$C \sqcup D \mid$ (union)

$\forall R.C \mid$ (universal restriction) — has to be individual of that class

$\exists R.C \mid$ (existential restriction) — at least one

$* n\ R$, where $* \in \{\geq, \leq, =\}$ (number restrictions)

Woman ≡ **Person** ⊓ **Female** (equivalence axiom)

Parent ≡ **Person** ⊓ ∃ **hasChild**.⊤ — at least one — could be anything

PersonWithoutChildren ≡ **Person** ⊓ ¬ **Parent** — not

MotherOfOnlyDragons ≡ **Woman** ⊓ **Parent** ⊓ ∀**hasChild**.**Dragon**

MotherOfOnlyManyDragons ≡ **?** (at least 3)

We can't! We need to extend to $\mathcal{ALCN}$

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Universität St.Gallen

TBox Examples in $\mathcal{ALCN}$

Atomic concepts: **Person**, **Female**, **Dragon**

Atomic roles: **hasChild**

Concepts are constructed according to the following syntax rule:

$C, D ::= A \mid$ (atomic concept)

$\top \mid$ (top concept)

$\bot \mid$ (bottom concept)

$\neg C \mid$ (atomic negation)

$C \sqcap D \mid$ (intersection)

$C \sqcup D \mid$ (union)

$\forall R.C \mid$ (universal restriction)

$\exists R.C \mid$ (existential restriction)

$* \, n \, R$, where $* \in \{\geq, \leq, =\}$ (number restrictions)

constructors

Woman ≡ **Person** ⊓ **Female** (≡ denotes equivalence)

Parent ≡ **Person** ⊓ ∃ **hasChild**.⊤

PersonWithoutChildren ≡ **Person** ⊓ ¬ **Parent**

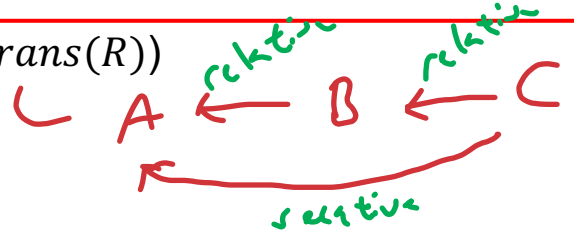
MotherOfOnlyDragons ≡ **Woman** ⊓ **Parent** ⊓ ∀**hasChild**.**Dragon**

MotherOfOnlyManyDragons ≡ **MotherOfOnlyDragons** ⊓ ≥ 3 **hasChild**

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

DLs: A Family of Logics

- [$\mathcal{EL}$] Existential Language
  - concept intersection ($C \sqcap D$), existential restrictions ($\exists R.C$)
- [$\mathcal{ALC}$] Attributive Language with Complement
  - concept intersection ($C \sqcap D$), concept union ($C \sqcup D$), concept negation ($\neg C$), universal restrictions ($\forall R.C$), and existential restrictions ($\exists R.C$)
  - [$\mathcal{S}$] extends $\mathcal{ALC}$ with transitive closure ($Trans(R)$)



OWL DL: $\mathcal{SHON}^{(\mathcal{D})}$
OWL 2 DL: $\mathcal{SROIQ}^{(\mathcal{D})}$

Extensions:

- [$\mathcal{H}$] adds role inclusion ($R \sqsubseteq S$)
  - [$\mathcal{R}$] adds (limited) complex role inclusion ($R_1 \circ R_2 \sqsubseteq S$), role reflexivity ($Ref(R)$), role irreflexivity ($Irref(R)$), role disjointness ($Disj(R,S)$), and the universal role (U)
- [$\mathcal{O}$] adds (limited) nomimals ($\{a_1, \ldots, a_n\}$)
- [$\mathcal{I}$] adds inverse roles (R^-)
- [$\mathcal{F}$] adds (limited) functional properties ($Func(R)$)
  - [$\mathcal{N}$] adds (limited) number restrictions ($* R$, where $* \in \{\geq, \leq, =\}$)
    - [$\mathcal{Q}$] adds (limited) qualified number restrictions ($* R.C$, where $* \in \{\geq, \leq, =\}$)
- [$(\mathcal{D})$] adds datatype properties, data values, or data types

The Pizza Ontology Revisited

An ontology is the **formal**, **explicit specification** of a **shared** **conceptualization** of an **area of interest** [Studer et al., 1998].

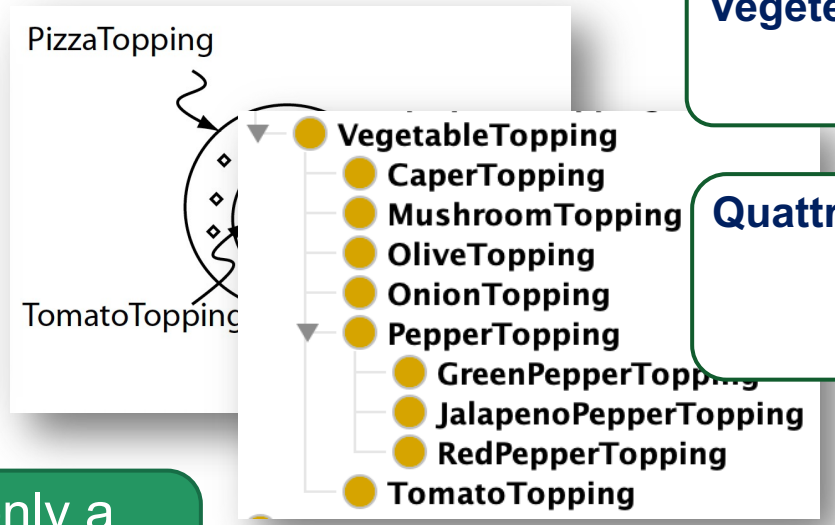
using a formal knowledge representation language, such as the **Web Ontology Language (OWL)**

the conceptual model is represented explicitly

usually defined and adopted by a group of people / a community

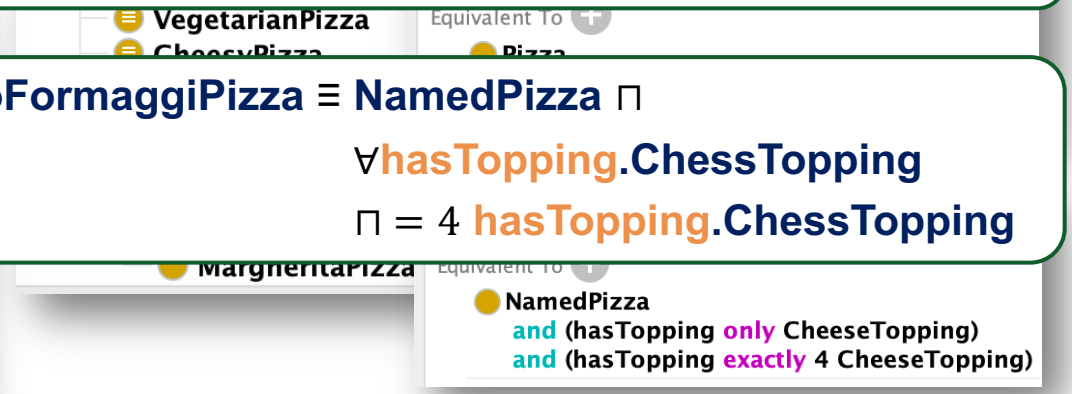
abstract simplified view (conceptual model)

ex., pizza



VegeterianPizza ≡ **Pizza** ⊓ ∀ **hasTopping**.(**CheeseTopping** ⊔ **VegetableTopping**)

QuattroFormaggiPizza ≡ **NamedPizza** ⊓
∀**hasTopping**.**ChessTopping**
⊓ = 4 **hasTopping**.**ChessTopping**



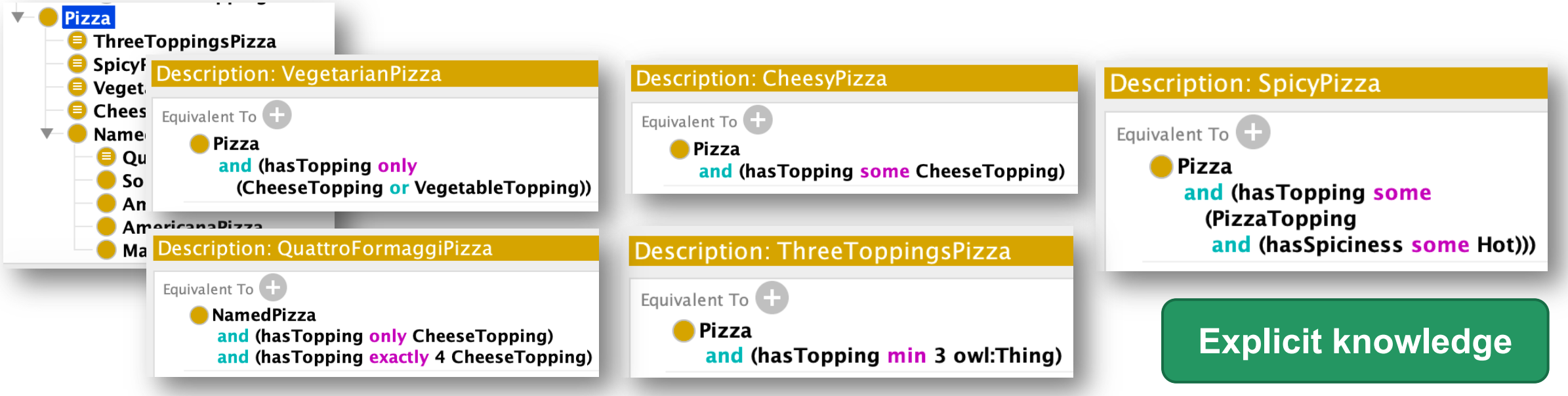
We can represent only a **finite number** of axioms

http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf

R. Studer, V.R. Benjamins, and D. Fensel. Knowledge Engineering: Principles and Methods, Data & Knowledge Engineering, 1998

Reasoning with the Pizza Ontology

Deductive reasoning is used to **bridge the gap** between what is **represented explicitly** and what is **known implicitly**

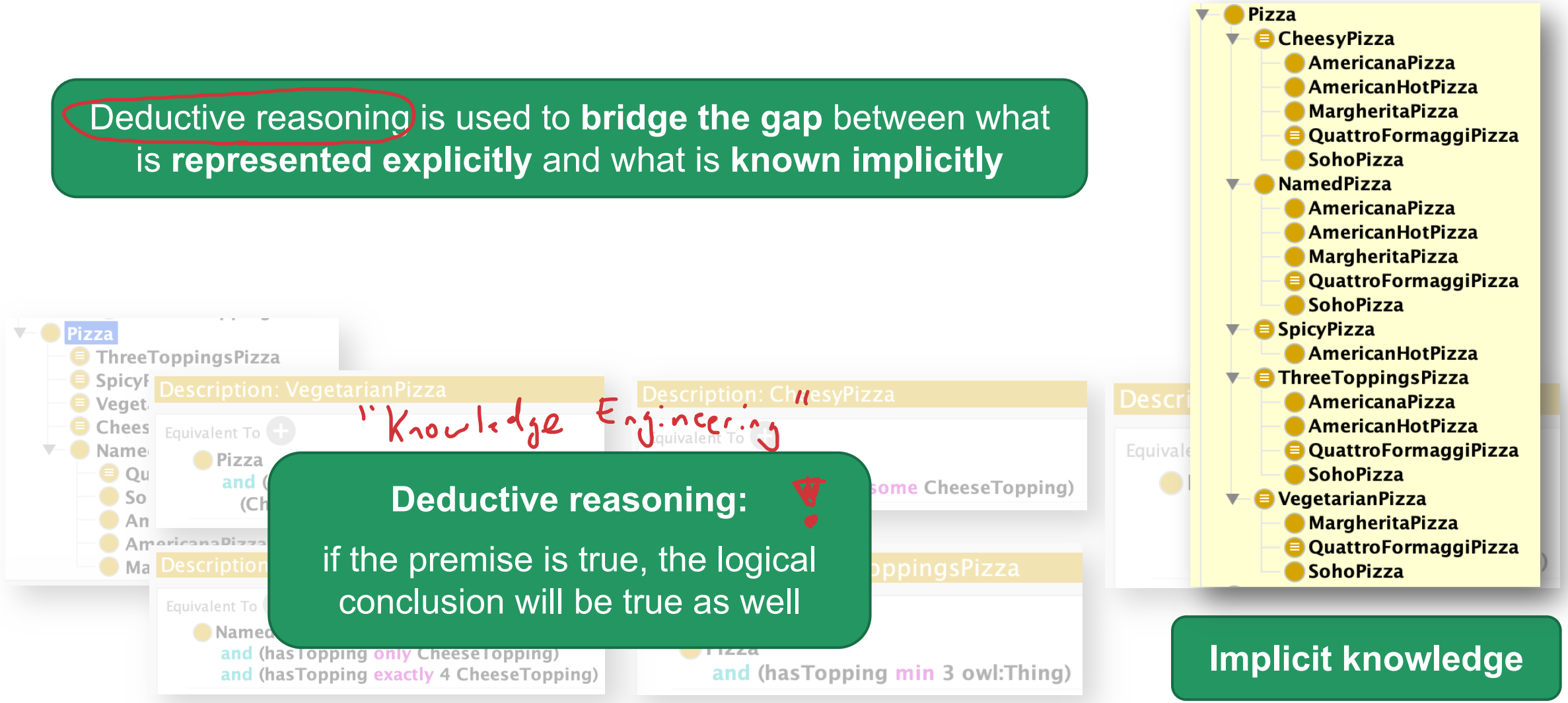


Explicit knowledge

http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf

Reasoning with the Pizza Ontology

Deductive reasoning is used to **bridge the gap** between what is **represented explicitly** and what is **known implicitly**



"Knowledge Engineering"

Deductive reasoning:

if the premise is true, the logical conclusion will be true as well

Implicit knowledge

http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf

Universität St.Gallen

Model-Theoretic Semantics

What we have seen so far, is **DL syntax**

– rules for constructing well-formed sequences of symbols

$$\textbf{VegeterianPizza} \equiv \textbf{Pizza} \sqcap \forall\, \textbf{hasTopping}.(\textbf{CheeseTopping} \sqcup \textbf{VegetableTopping})$$

DL semantics define rules for deriving the meaning of complex sequences of symbols from atomic symbols

– meaning is derived based on an **interpretation** within a **formal model** over a **domain**

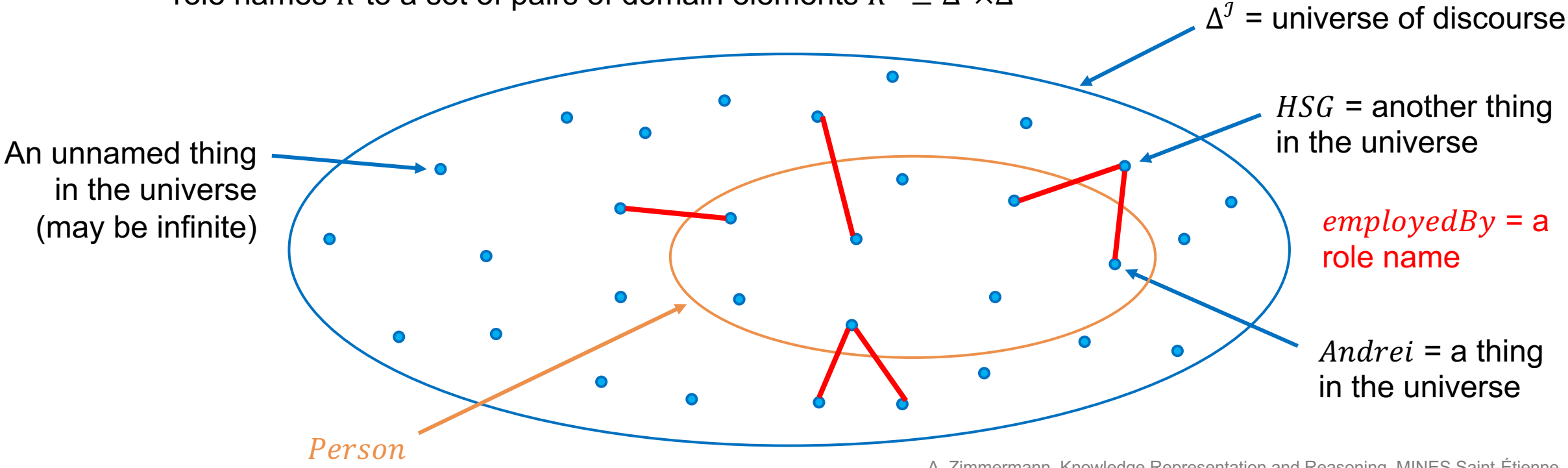
– there are usually **many possible interpretations** over the same domain

↳ need to remove ambiguity

DL Semantics: Interpretation

An **interpretation** is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that:

- $\Delta^{\mathcal{I}}$ is a non-empty set of individuals (**domain of interpretation** or **universe of discourse**)
- $\cdot^{\mathcal{I}}$ is an **interpretation function** that maps:
  - individual names a to domain elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
  - concept names C to a set of domain elements $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
  - role names R to a set of pairs of domain elements $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$



A. Zimmermann, Knowledge Representation and Reasoning, MINES Saint-Étienne, 2021.
F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Universität St.Gallen

$\mathcal{ALCN}$ Semantics: Interpretation

Syntax

Concepts are constructed according to the following syntax rule:

$C, D ::= A \mid$ (atomic concept)

$\top \mid$ (top concept)

$\bot \mid$ (bottom concept)

$\neg C \mid$ (atomic negation)

$C \sqcap D \mid$ (intersection)

$C \sqcup D \mid$ (union)

$\forall R.C \mid$ (universal restriction)

$\exists R.C \mid$ (existential restriction)

$* \, n \, R$, where $* \in \{\geq, \leq, =\}$ (number restrictions)

Semantics

Extending the interpretation function $\cdot^{\mathcal{I}}$ to concept definitions:

- $(A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}})$
- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
- $\bot^{\mathcal{I}} = \emptyset$
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\forall R.C)^{\mathcal{I}} = \{ x \in \Delta^{\mathcal{I}} \mid \forall y \, (x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}} \}$
- $(\exists R.C)^{\mathcal{I}} = \{ x \in \Delta^{\mathcal{I}} \mid \exists y \, (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}} \}$
- $(\geq nR)^{\mathcal{I}} = \{ x \in \Delta^{\mathcal{I}} \mid |\{ y \mid (x, y) \in R^{\mathcal{I}} \}| \geq n \}$

(similar definitions for the other number restrictions)

role

new concept: Set of x

for any

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

DL Semantics: Satisfiability

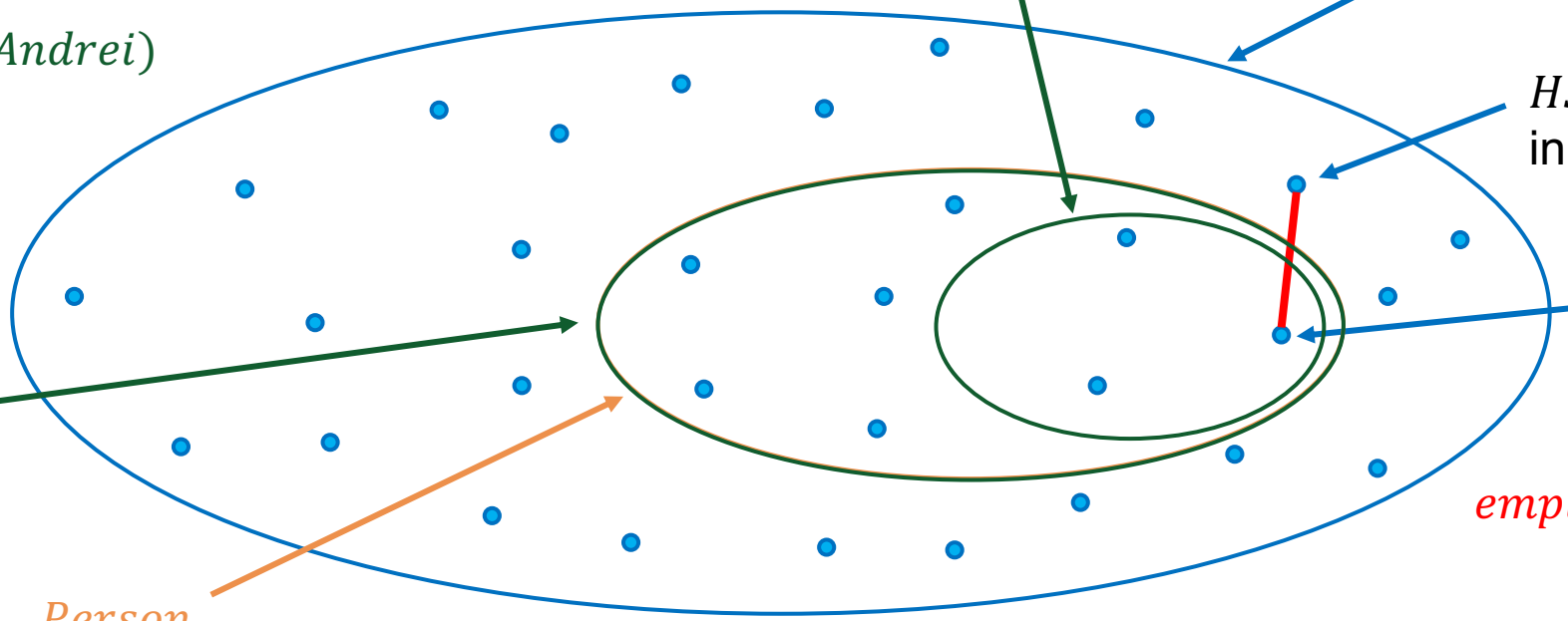
Given an interpretation $\mathcal{I}$, the **satisfaction** relation $\models$ is defined as follows:

- $\mathcal{I} \models C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (concept inclusion axiom)
- $\mathcal{I} \models C(a)$ if and only if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (concept assertion axiom)
- $\mathcal{I} \models R(a, b)$ if and only if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ (role assertion axiom)
- $\mathcal{I} \models C \equiv D$ if and only if $C^{\mathcal{I}} = D^{\mathcal{I}}$ (equivalence axiom)

$FacultyMember \sqsubseteq Person$

$FacultyMember(Andrei)$

$Human \equiv Person$



$\Delta^{\mathcal{I}}$= universe of discourse

HSG = another thing in the universe

$Andrei$ = a thing in the universe

$employedBy(Andrei, HSG)$

A. Zimmermann, Knowledge Representation and Reasoning, MINES Saint-Étienne, 2021.
F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Universität St.Gallen

DL Semantics: Satisfiability

Given an interpretation $\mathcal{I}$, the **satisfaction** relation $\models$ is defined as follows:

- $\mathcal{I} \models C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (concept inclusion axiom)
- $\mathcal{I} \models C(a)$ if and only if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (concept assertion axiom)
- $\mathcal{I} \models R(a, b)$ if and only if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ (role assertion axiom)
- $\mathcal{I} \models C \equiv D$ if and only if $C^{\mathcal{I}} = D^{\mathcal{I}}$ (equivalence axiom)

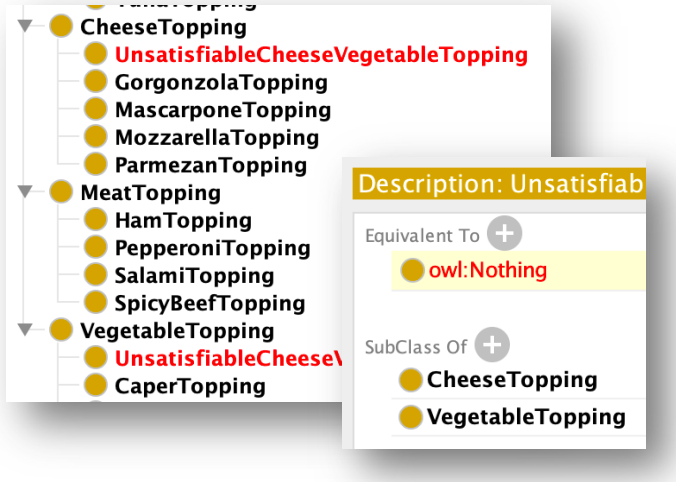
An interpretation $\mathcal{I}$ satisfies a TBox $\mathcal{T}$ if and only if $\mathcal{I}$ satisfies all axioms in $\mathcal{T}$.
($\mathcal{I}$ is a model of $\mathcal{T}$)

An interpretation $\mathcal{I}$ satisfies an ABox $\mathcal{A}$ if and only if $\mathcal{I}$ satisfies all axioms in $\mathcal{A}$.
($\mathcal{I}$ is a model of $\mathcal{A}$)

A. Zimmermann, Knowledge Representation and Reasoning, MINES Saint-Étienne, 2021.
F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Reasoning Tasks for the TBox

Satisfiability: A concept C is satisfiable with respect to a TBox T if there exists a model $\mathcal{I}$ of T such that $C^{\mathcal{I}}$ is nonempty ($\mathcal{I}$ is a model of C).



Concept Satisfiability [TBox]

(**CheeseTopping** and **VegetableTopping** are disjoint)

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Reasoning Tasks for the TBox

Satisfiability: A concept C is satisfiable with respect to a TBox T if there exists a model $\mathcal{I}$ of T such that $C^{\mathcal{I}}$ is nonempty ($\mathcal{I}$ is a model of C).

Subsumption: A concept C is subsumed by a concept D with respect to a TBox T if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of T.

interpret. of C

interpret. of C

subset of

Concept Subsumption [TBox]
(inferred classification)

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Universität St.Gallen

Reasoning Tasks for the TBox

Satisfiability: A concept C is satisfiable with respect to a TBox T if there exists a model $\mathcal{I}$ of T such that $C^{\mathcal{I}}$ is nonempty ($\mathcal{I}$ is a model of C).

Subsumption: A concept C is subsumed by a concept D with respect to a TBox T if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of T.

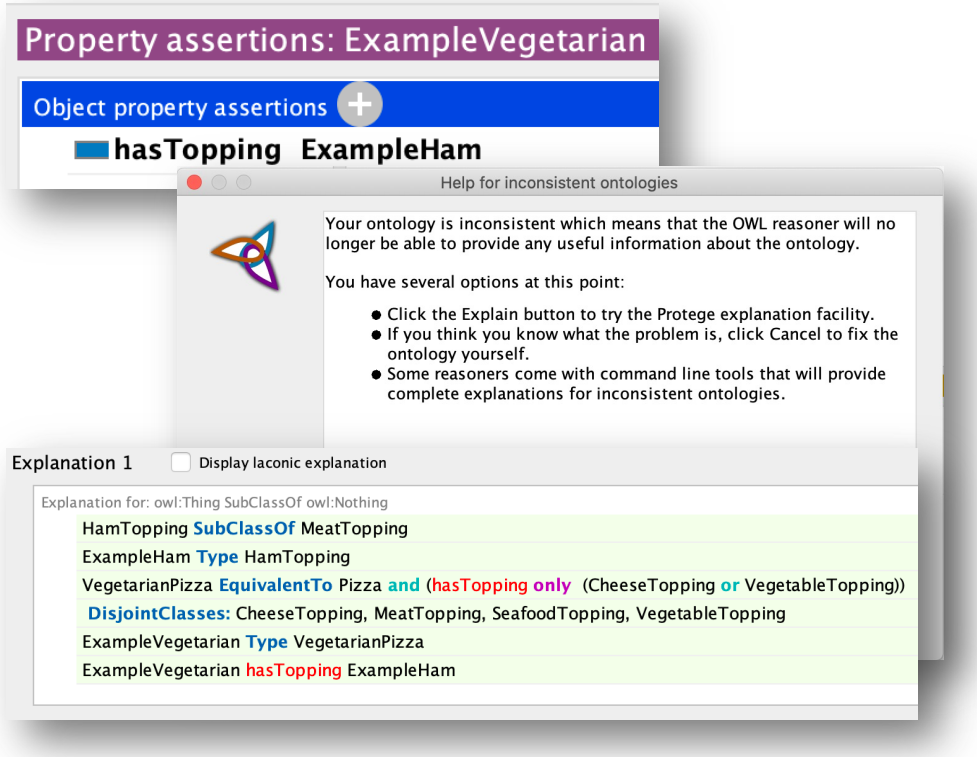
Equivalence: Two concepts C and D are equivalent with respect to a TBox T if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all models $\mathcal{I}$ of T.

Disjointness: Two concepts C and D are disjoint with respect to a TBox T if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for all models $\mathcal{I}$ of T.

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Reasoning Tasks for the ABox

Consistency: An ABox A is consistent with respect to a TBox T if there exists an interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a model of both A and T.



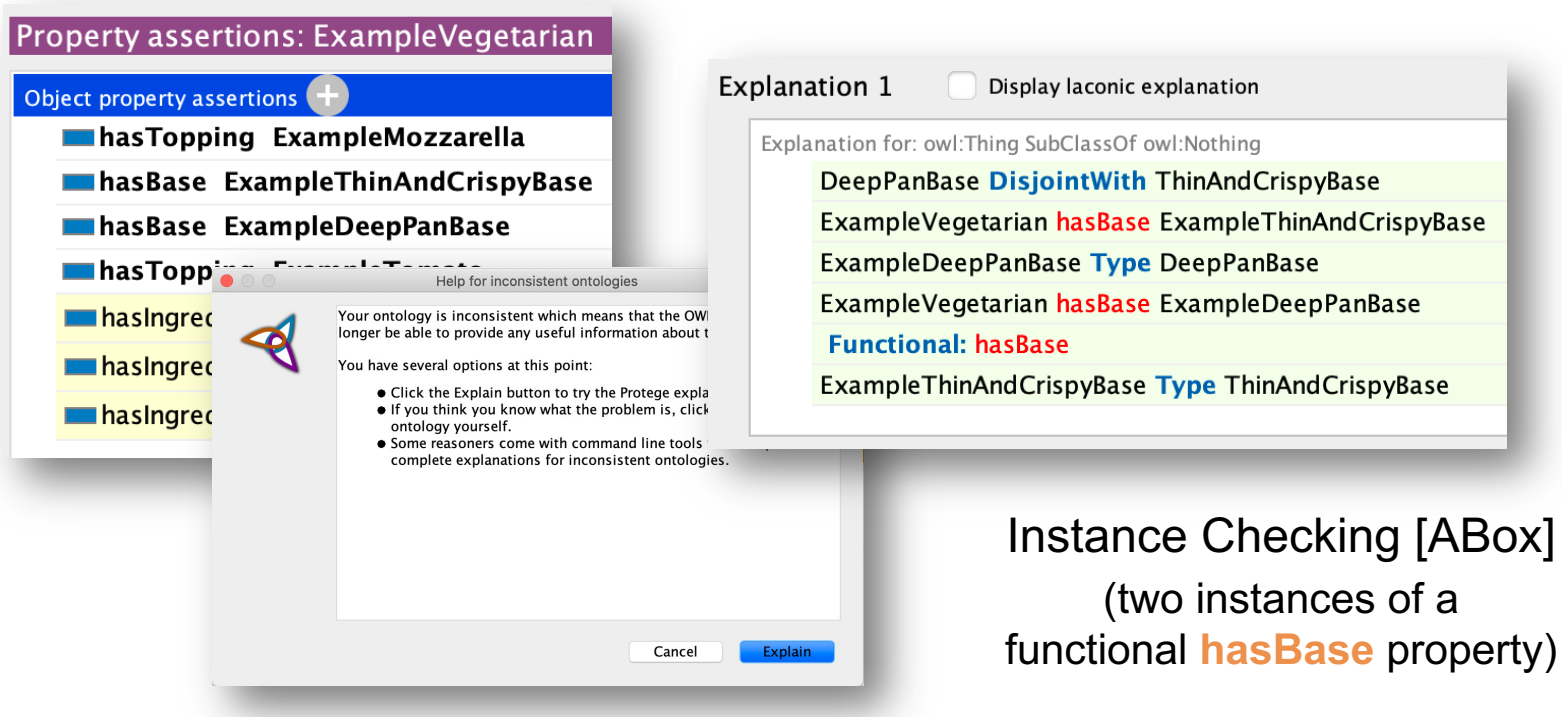
Consistency [ABox]
(creating a **VegetarianPizza** with ham)

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

Reasoning Tasks for the ABox

Consistency: An ABox A is consistent with respect to a TBox T if there exists an interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a model of both A and T.

Instance Checking: An assertion α is entailed by an ABox A and we write $A \models \alpha$ if all interpretations $\mathcal{I}$ that satisfy A, that is all models of A, also satisfy α.



Instance Checking [ABox]
(two instances of a functional **hasBase** property)

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

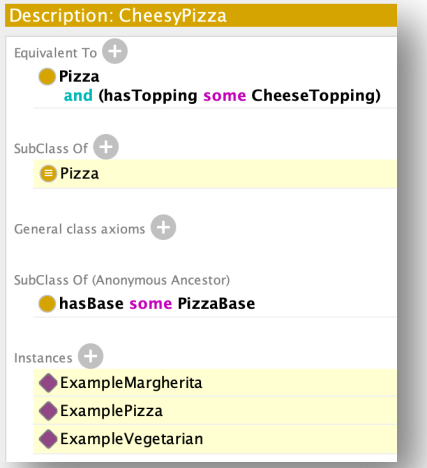
Universität St.Gallen

Reasoning Tasks for the ABox

Consistency: An ABox A is consistent with respect to a TBox T if there exists an interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a model of both A and T.

Instance Checking: An assertion α is entailed by an ABox A and we write $A \models \alpha$ if all interpretations $\mathcal{I}$ that satisfy A, that is all models of A, also satisfy α.

Retrieval Problem: Given an ABox A and a concept C, find all individuals a such that $A \models C(a)$.



Retrieval Problem [ABox]

(inferred instances of **CheesyPizza**)

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

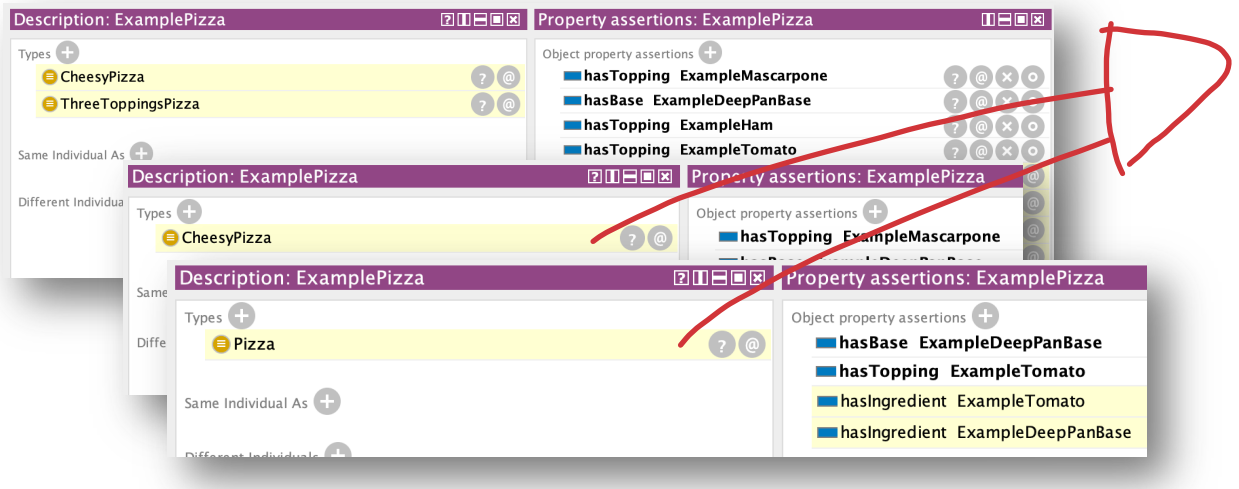
Reasoning Tasks for the ABox

Consistency: An ABox A is consistent with respect to a TBox T if there exists an interpretation $\mathcal{I}$ such that $\mathcal{I}$ is a model of both A and T.

Instance Checking: An assertion α is entailed by an ABox A and we write $A \models \alpha$ if all interpretations $\mathcal{I}$ that satisfy A, that is all models of A, also satisfy α.

Retrieval Problem: Given an ABox A and a concept C, find all individuals a such that $A \models C(a)$.

Realization Problem: Given an individual a and a set of concepts, find the most specific concepts C from the set such that $A \models C(a)$.



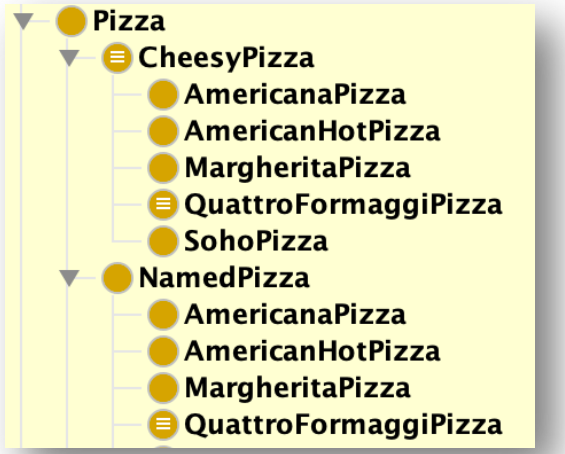
what is most restrictive type?

Realization Problem [ABox]

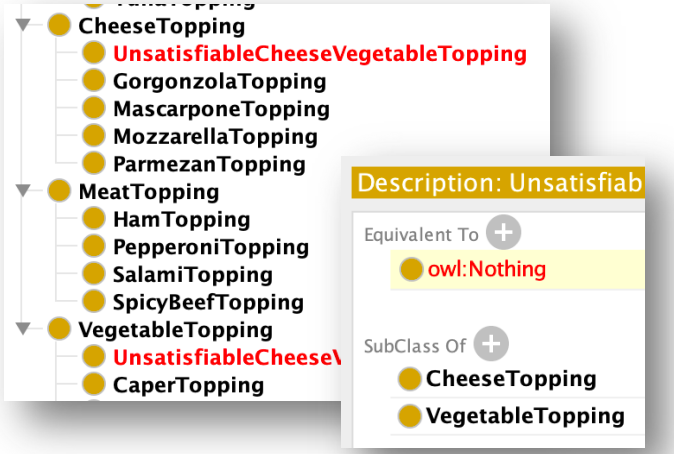
(removing toppings from a pizza results in less specific classifications)

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

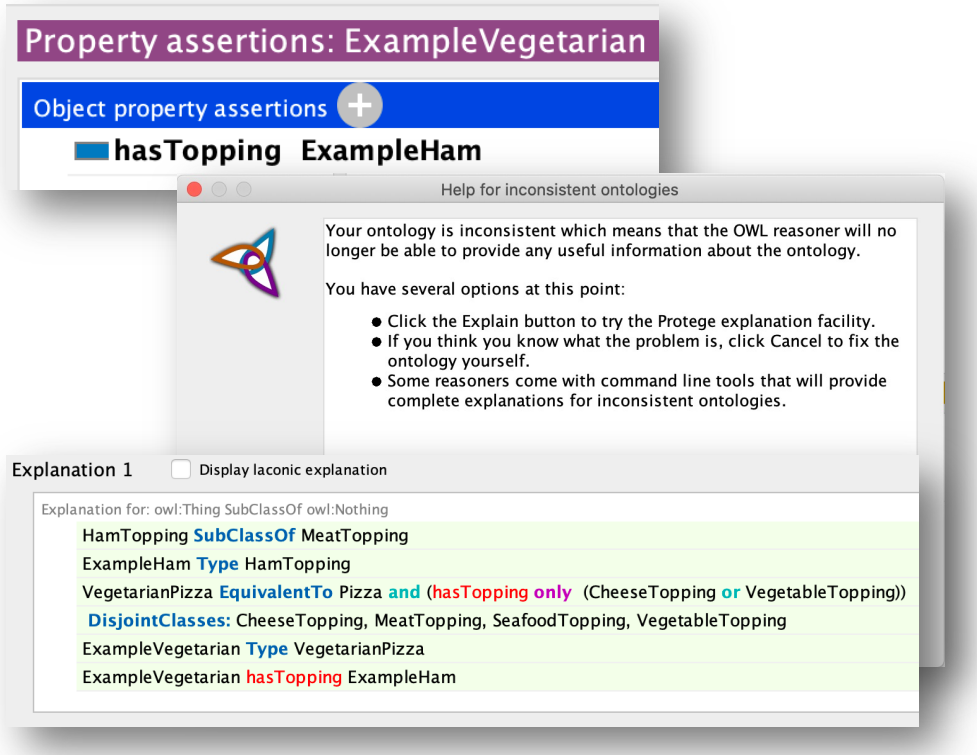
Reasoning Tasks with the Pizza Ontology



Concept Subsumption [TBox]
(inferred classification)



Concept Satisfiability [TBox]
(**CheeseTopping** and **VegetableTopping** are disjoint)



ABox Consistency
(creating a **VegetarianPizza** with ham)

And we saw live the other **reasoning tasks** discussed for the ABox (**instance checking**, **retrieval**, **realization**)

The Semantic Web Roadmap (1998)

Tim Berners-Lee
Date: September 1998. Last modified: $Date: 1998/10/14 20:17:13 $

Status: An attempt to give a high-level plan of the architecture of the Semantic WWW. Editing status: Draft. Comments welcome

Up to Design Issues

Semantic Web Road map

A road map for the future, an architectural plan untested by anything except thought experiments.

"This document is a plan for achieving a set of connected applications for data on the Web in such a way as to form a **consistent logical web of data** (**semantic web**)."

Sir Tim Berners-Lee, Semantic Web Road Map, 1998

Now this should have **depth**!

We'll discuss this next week, when we talk about **Linked Data**!

https://www.w3.org/DesignIssues/Semantic.html

Fabien Gandon. A Survey of the First 20 Years of Research on Semantic Web and Linked Data. 2018.

Universität St.Gallen

Closed- vs. Open-World Semantics

A database operates under the **Closed-World Assumption**

- what is not known to be true is necessarily false
  - ⇒ only one possible interpretation

A DL knowledge base operates under the **Open-World Assumption**

- what is not know to be true is simply not known (could be either true or false)
  - ⇒ many possible interpretations

many interpretations in description logic logic base

The Story of Oedipus (up to 2m10s): https://youtu.be/Jn9Gq5AkEx4

- well-known example in DL research [Baader & Nutt, 2003]

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

DL Reasoning: Open-World Semantics

Abox A

hasChild(Jocasta,Oedipus)
hasChild(Jocasta,Polyneikes)
hasChild(Oedipus,Polyneikes)
hasChild(Polyneikes,Thersandros)
Patricide(Oedipus)
¬Patricide(Thersandros)

Does Jocasta have a child who is a patricide and who themselves have a child who is not a patricide?

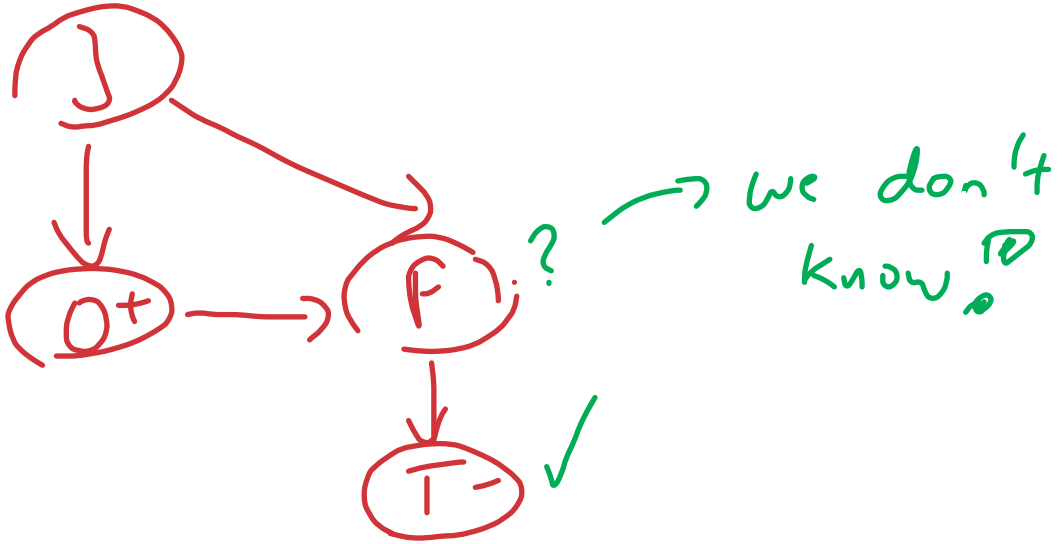
$$\alpha = (\exists \text{hasChild}.(\text{Patricide} \sqcap \exists \text{hasChild}.\neg\text{Patricide}))(\text{Jocasta})$$

concept definition

concept assertion axiom

⇒ DB: No!

⇒ Description Logic: Yes!



→ we don't know!

F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

DL Reasoning: Open-World Semantics

Abox A

hasChild(Jocasta,Oedipus)
hasChild(Jocasta,Polyneikes)
hasChild(Oedipus,Polyneikes)
hasChild(Polyneikes,Thersandros)
Patricide(Oedipus)
¬Patricide(Thersandros)

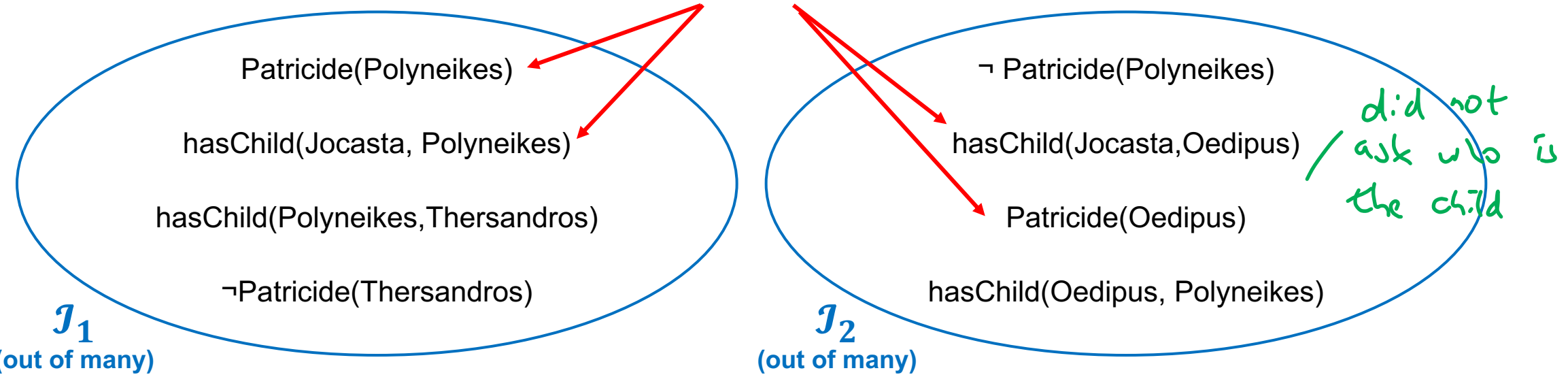
Does Jocasta have a child who is a patricide and who themselves have a child who is not a patricide?

$$\alpha = (\exists \text{hasChild}.(\text{Patricide} \sqcap \exists \text{hasChild}.\neg\text{Patricide}))(\text{Jocasta})$$

Do all possible interpretations $\mathcal{I}_x$ of A satisfy α?

Both $\mathcal{I}_1$ and $\mathcal{I}_2$ satisfy α (although different patricides)

Child of Jocasta who is a patricide and has a child who is not a patricide



F. Baader and W. Nutt, Basic Description Logics, The Description Logic Handbook: Theory, Implementation, and Applications. 2003.

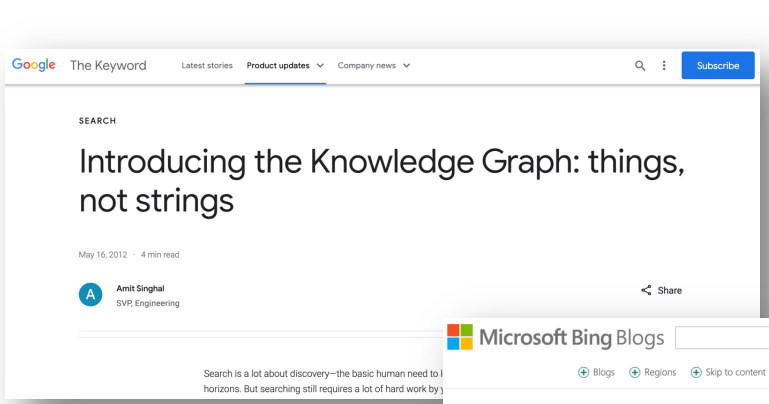
Universität St.Gallen

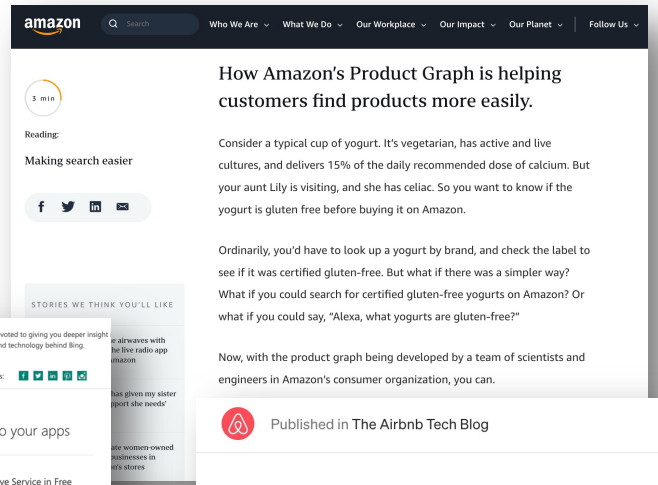
Today's Agenda

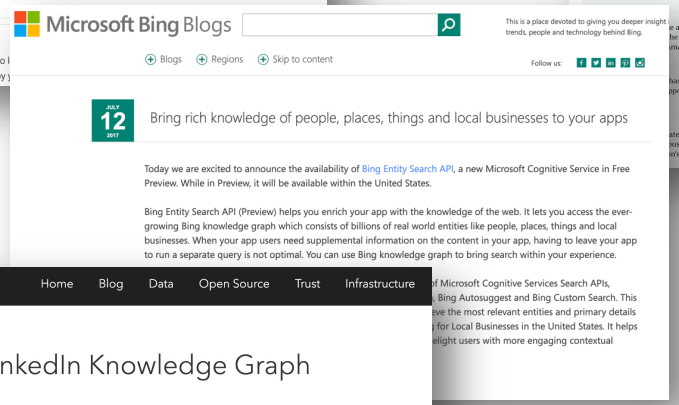
- Web Ontologies
  - Description Logics and OWL
  - Building Web Ontologies with Protégé (Tutorial)
- Knowledge Graphs
  - Graph Data Models
  - Querying Graph Datasets

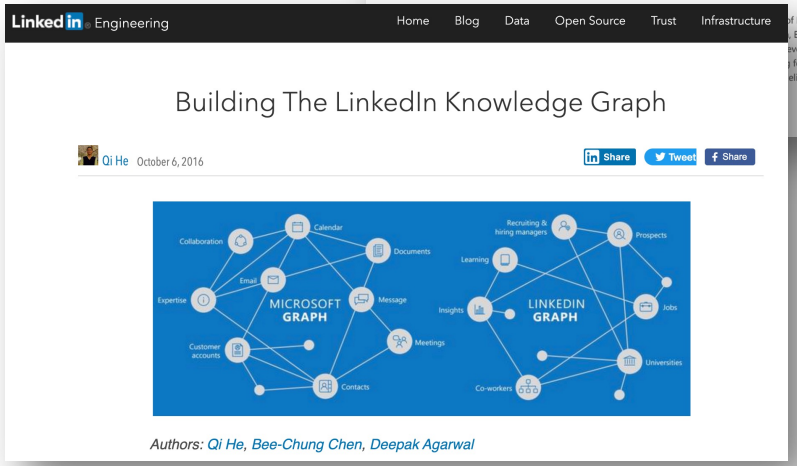
Universität St.Gallen

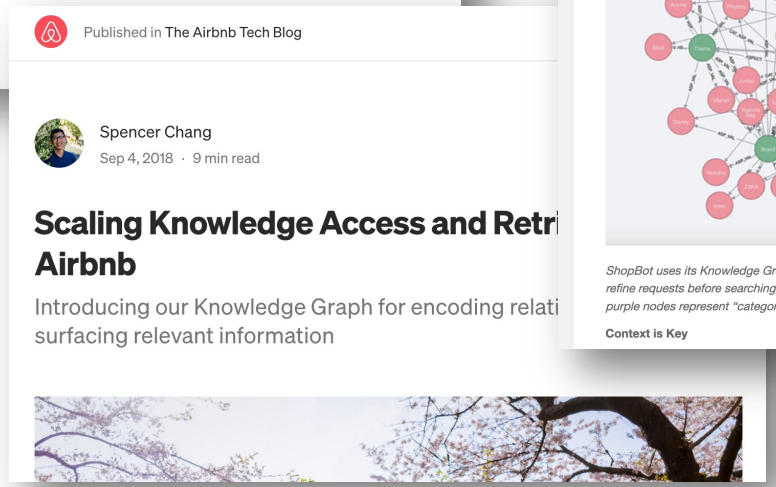
Knowledge Graphs

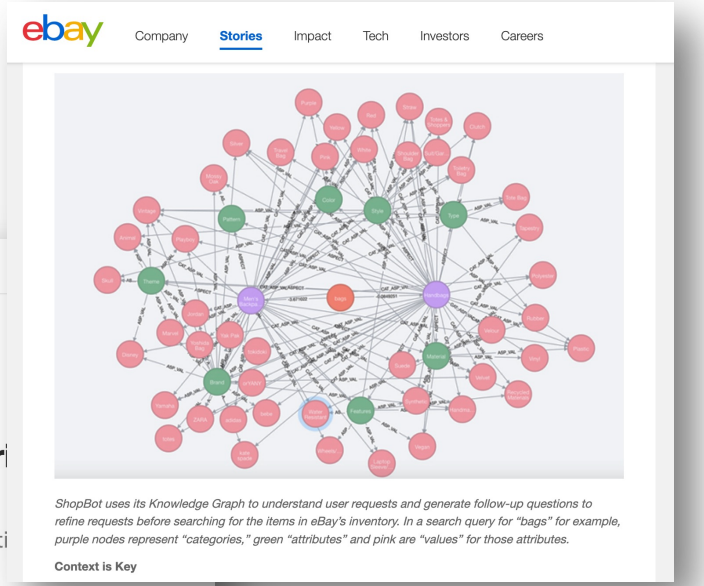






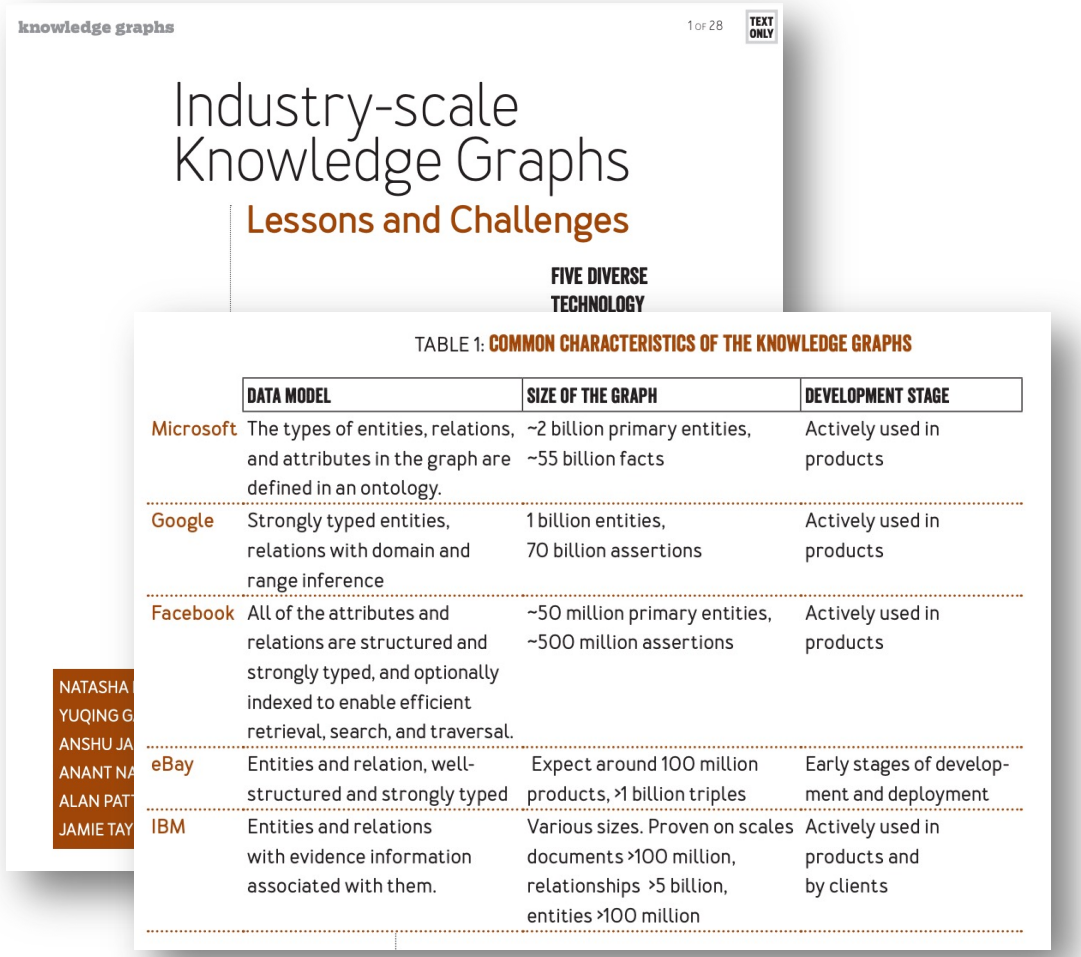






https://blog.google/products/search/introducing-knowledge-graph-things-not/
https://www.aboutamazon.com/news/innovation-at-amazon/making-search-easier
https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph
https://www.ebayinc.com/stories/news/cracking-the-code-on-conversational-commerce/
https://medium.com/airbnb-engineering/scaling-knowledge-access-and-retrieval-at-airbnb-665b6ba21e95
https://blogs.bing.com/search-quality-insights/2017-07/bring-rich-knowledge-of-people-places-things-and-local-businesses-to-your-apps

Knowledge Graphs



knowledge graphs

1 OF 28 TEXT ONLY

Industry-scale Knowledge Graphs

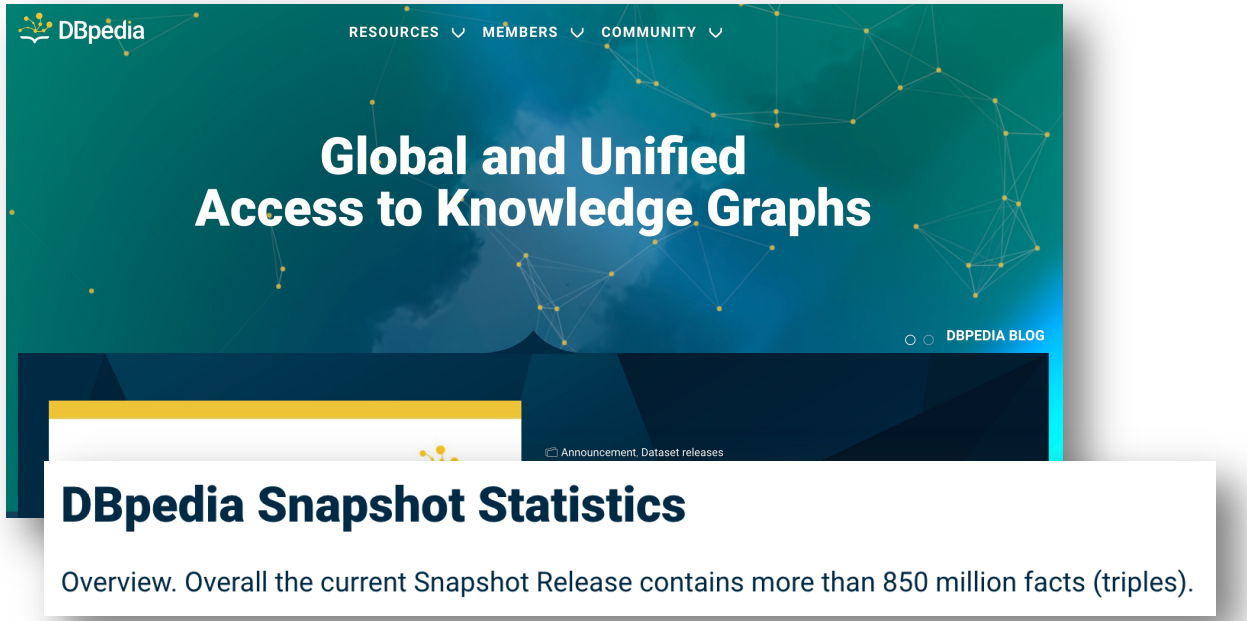
Lessons and Challenges

FIVE DIVERSE TECHNOLOGY

NATASHA
YUQING G
ANSHU JA
ANANT NA
ALAN PAT
JAMIE TAY

TABLE 1: COMMON CHARACTERISTICS OF THE KNOWLEDGE GRAPHS

	DATA MODEL	SIZE OF THE GRAPH	DEVELOPMENT STAGE
Microsoft	The types of entities, relations, and attributes in the graph are defined in an ontology.	~2 billion primary entities, ~55 billion facts	Actively used in products
Google	Strongly typed entities, relations with domain and range inference	1 billion entities, 70 billion assertions	Actively used in products
Facebook	All of the attributes and relations are structured and strongly typed, and optionally indexed to enable efficient retrieval, search, and traversal.	~50 million primary entities, ~500 million assertions	Actively used in products
eBay	Entities and relation, well-structured and strongly typed	Expect around 100 million products, >1 billion triples	Early stages of development and deployment
IBM	Entities and relations with evidence information associated with them.	Various sizes. Proven on scales documents >100 million, relationships >5 billion, entities >100 million	Actively used in products and by clients



DBpedia Snapshot Statistics

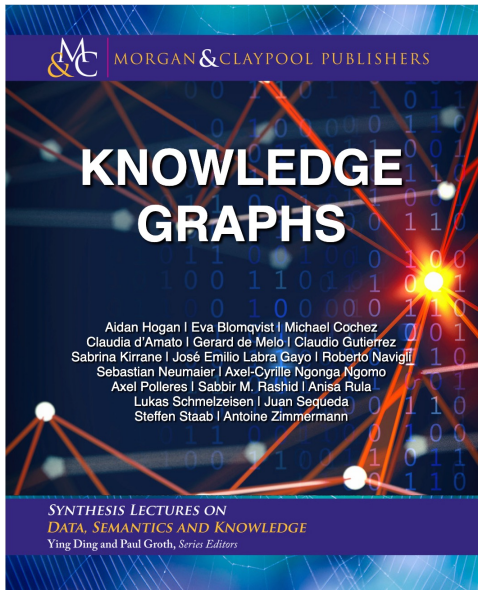
Overview. Overall the current Snapshot Release contains more than 850 million facts (triples).

https://www.dbpedia.org/

N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor. Industry-scale Knowledge Graphs: Lessons and Challenges — Five diverse technology companies show how it's done. Queue, 2019.

Knowledge Graphs

"A knowledge graph is a **graph of data** intended to accumulate and convey **knowledge of the real world**, whose **nodes** represent **entities of interest** and whose **edges** represent **relations between these entities**" [Hogan et al., 2021].



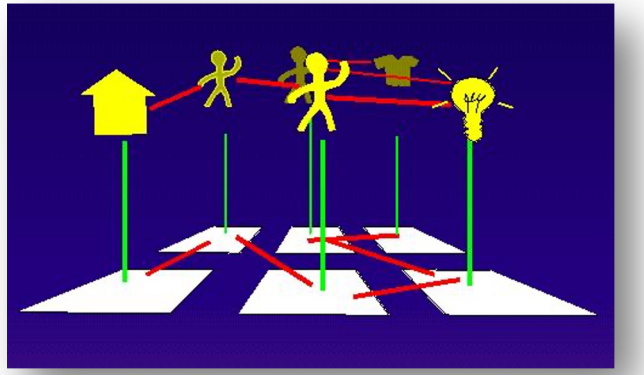
https://kgbook.org/

Aidan Hogan et al. Knowledge Graphs. 2021.
Antoine Zimmermann. *Knowledge Representation and Reasoning*, MINES Saint-Étienne, 2021.

Knowledge Graphs

"A knowledge graph is a **graph of data** intended to accumulate and convey **knowledge of the real world**, whose **nodes** represent **entities of interest** and whose **edges** represent **relations between these entities**" [Hogan et al., 2021].

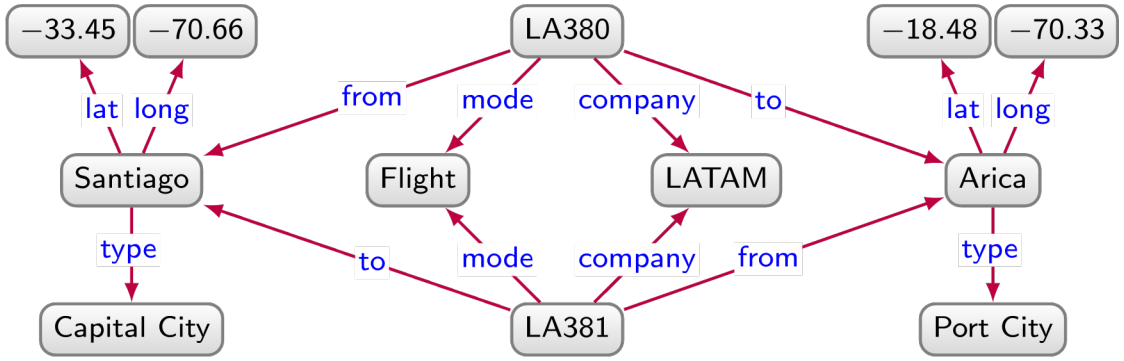
Syntax	Semantics
graph of data	knowledge of the real world



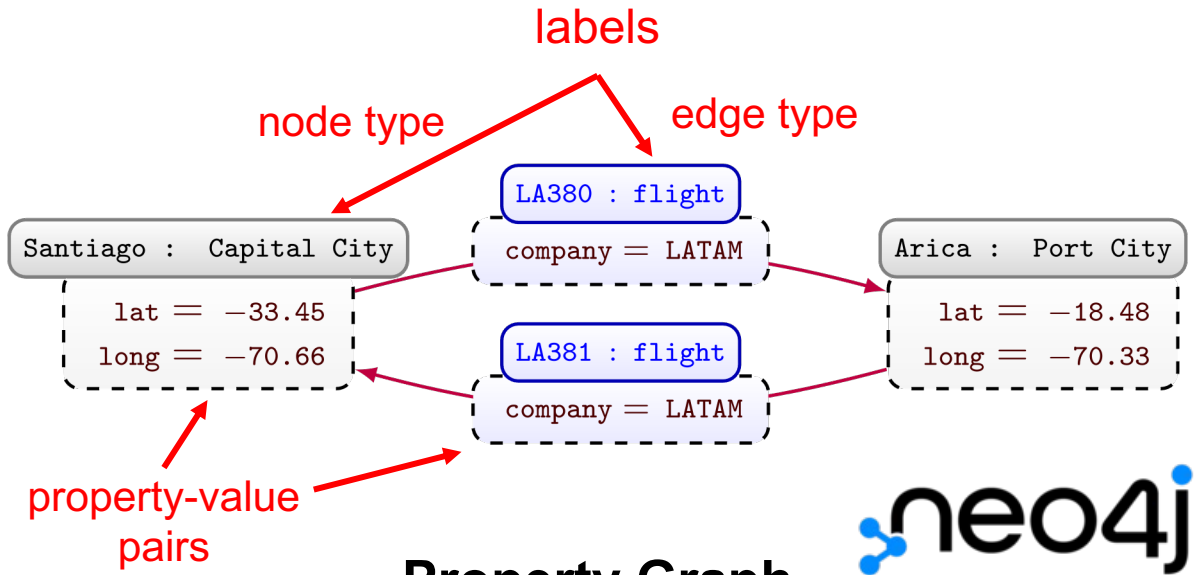
https://videos.cern.ch/record/2671957

Aidan Hogan et al. Knowledge Graphs. 2021.
Antoine Zimmermann. *Knowledge Representation and Reasoning*, MINES Saint-Étienne, 2021.

Data Models for Knowledge Graphs



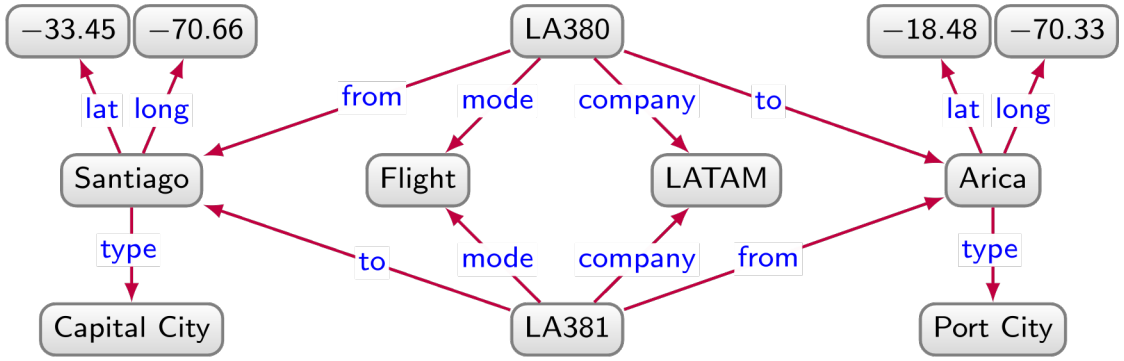
Directed Edge-Labelled Multigraph



Property Graph

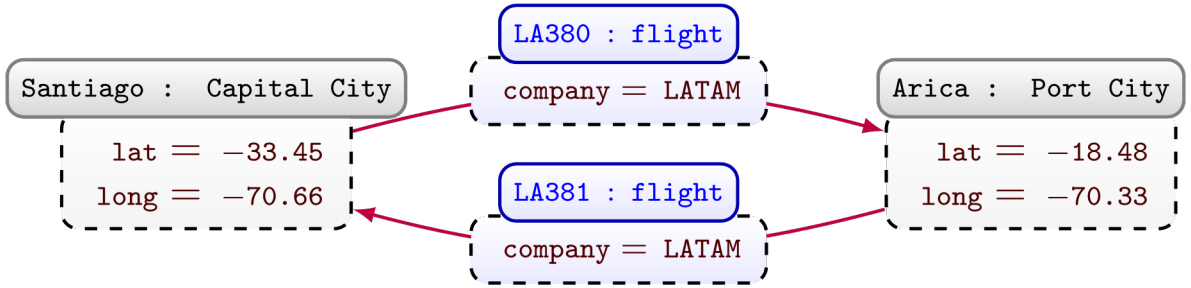
Aidan Hogan et al. Knowledge Graphs. 2021.

Data Models for Knowledge Graphs



Directed Edge-Labelled Multigraph

more **minimal** model



Property Graph

more **flexible** model

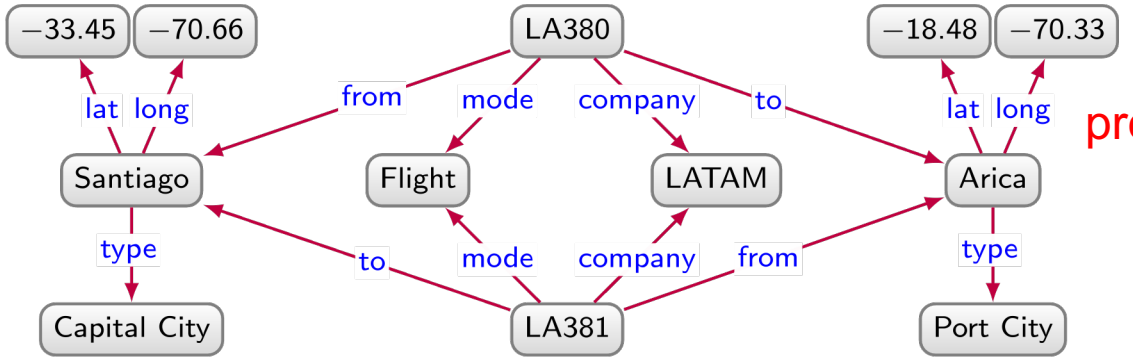
Property graphs can be converted to/from directed edge-labelled multigraphs [Angles et al., 2019].

The **Resource Description Framework (RDF)** is a standardized data model based on directed edge-labelled multigraphs.

Aidan Hogan et al. Knowledge Graphs. 2021.

R. Angles, H. Thakkar, and D. Tomaszuk. RDF and Property Graphs Interoperability: Status and Issues. CEUR Workshop Proceedings, vol. 2369. 2019.

Data Models for Knowledge Graphs



Directed Edge-Labelled Multigraph

more **minimal** model

RDF Graph (Turtle syntax)

subject

predicate

object

```
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

:LA380 :company :LATAM ;
       :mode    :Fligh
       :from    :Sant
       :to      :Aric

:LA381 :company :LATA
       :mode    :Flig
       :from    :Arica
       :to      :Sant

:Santiago rdf:type :C
       :lat  -33.45 ;
       :long -70.66 .

:Arica rdf:ty
       :lat
       :long
```

- **subjects** are URIs or blank nodes
- **predicates** are URIs
- **objects** are URIs, blank nodes, or literals (typed values)

An **RDF graph** is a **set of triples**:
subject predicate object

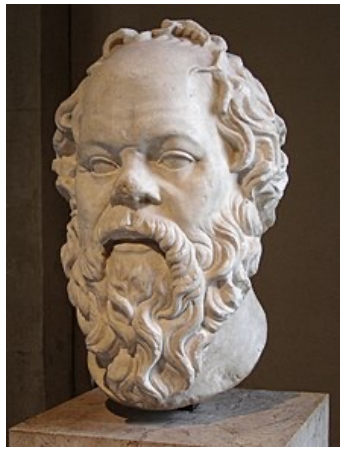
In **first-order logic**, a triple is just a binary predicate: **predicate(subject,object)**

Property graphs can be converted to/from directed edge-labelled multigraphs [Angles et al., 2019].

The **Resource Description Framework (RDF)** is a standardized data model based on directed edge-labelled multigraphs.

Aidan Hogan et al. Knowledge Graphs. 2021.
R. Angles, H. Thakkar, and D. Tomaszuk. RDF and Property Graphs Interoperability: Status and Issues. CEUR Workshop Proceedings, vol. 2369. 2019.

Deriving Knowledge from RDF Graphs



Premise: All men are mortal.
Premise: Socrates is a man.
Conclusion: Socrates is mortal.

First-Order Logic

$$\frac{\forall x\ (man(x) \rightarrow mortal(x)) \quad man(socrates)}{mortal(socrates)}$$

Modus Ponens:

$$\frac{P \rightarrow Q \quad P}{Q}$$

RDF (Turtle syntax)

```
prefix : <http://example.org/#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>


:Man rdfs:subClassOf :Mortal .   # premise
:Socrates rdf:type :Man .        # premise
:Socrates rdf:type :Mortal .     # conclusion (can be inferred)
```

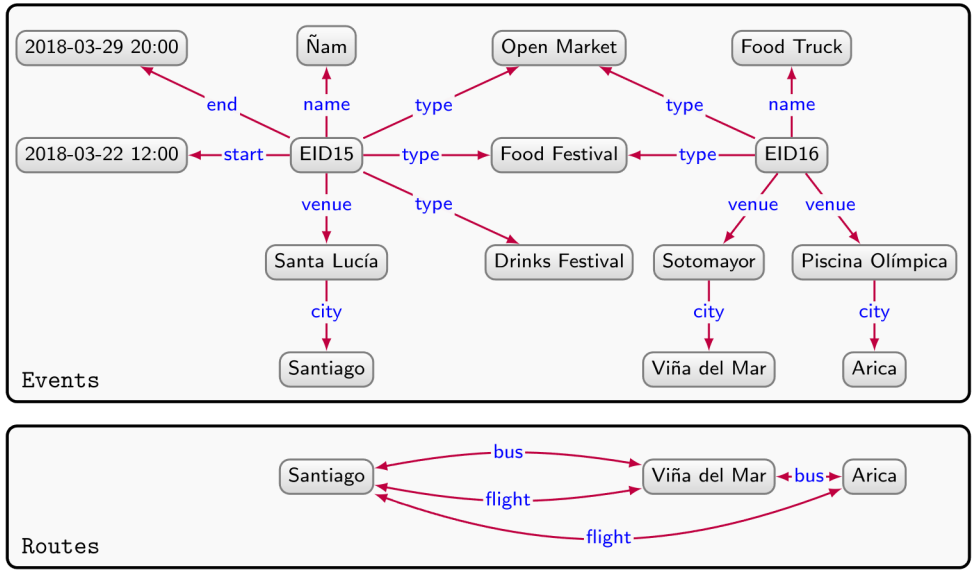
Tobias Käfer, Andreas Harth, and Lars Heling. Knowledge Graphs and Linked Data. AI4Industry Summer School, MINES Saint-Étienne, 2021.

Graph Datasets

A **graph dataset** is a **set of named graphs**

There is always a **default graph**

- sometimes the default graph is the union of all named graphs

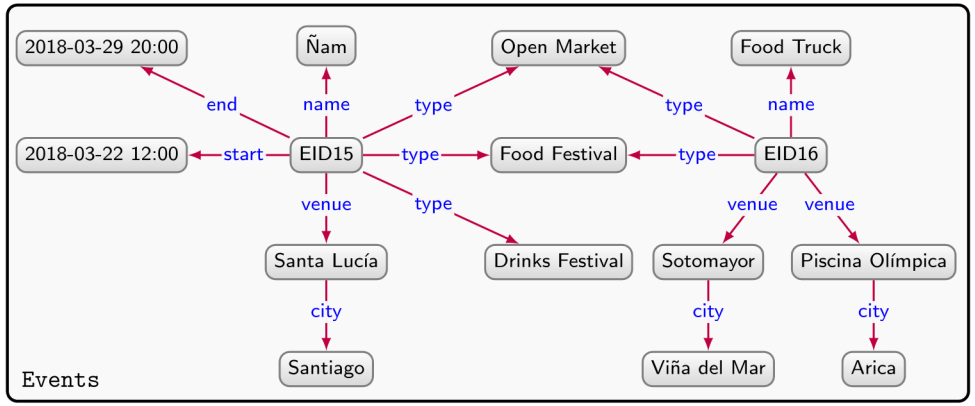


Aidan Hogan et al. Knowledge Graphs. 2021.

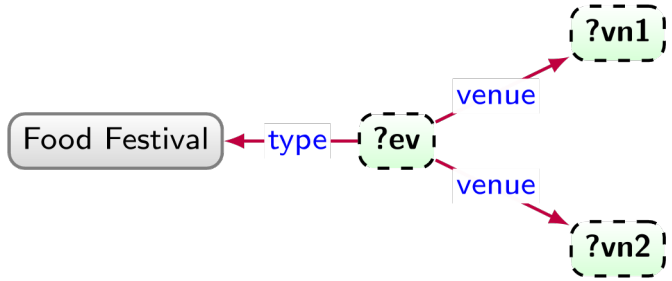
Graph Datasets and Querying

Graph query languages (SPARQL, Cypher, etc.) share 3 types of **graph patterns**:

- basic graph patterns



Q: "What are the venues of food festivals?"



?ev	?vn1	?vn2
EID16	Piscina Olímpica	Sotomayor
EID16	Sotomayor	Piscina Olímpica
EID16	Piscina Olímpica	Piscina Olímpica
EID16	Sotomayor	Sotomayor
EID15	Santa Lucía	Santa Lucía

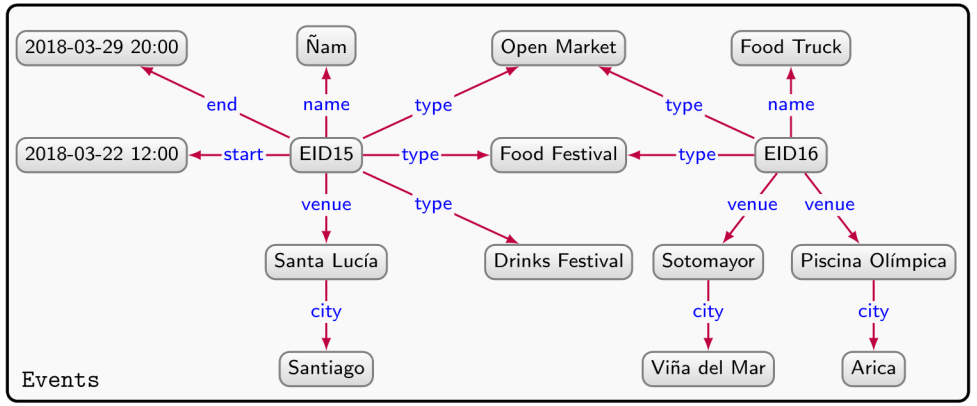
depends on semantics of query language

Aidan Hogan et al. Knowledge Graphs. 2021.

Graph Datasets and Querying

Graph query languages (SPARQL, Cypher, etc.) share 3 types of **graph patterns**:

- basic graph patterns



SPARQL Query

```
PREFIX : <http://example.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

FROM NAMED <http://example.org/events>

SELECT ?ev ?vn1 ?vn2 WHERE {
  ?ev rdf:type :FoodFestival .
  ?ev :venue   ?vn1 .
  ?ev :venue   ?vn2 .
}
```

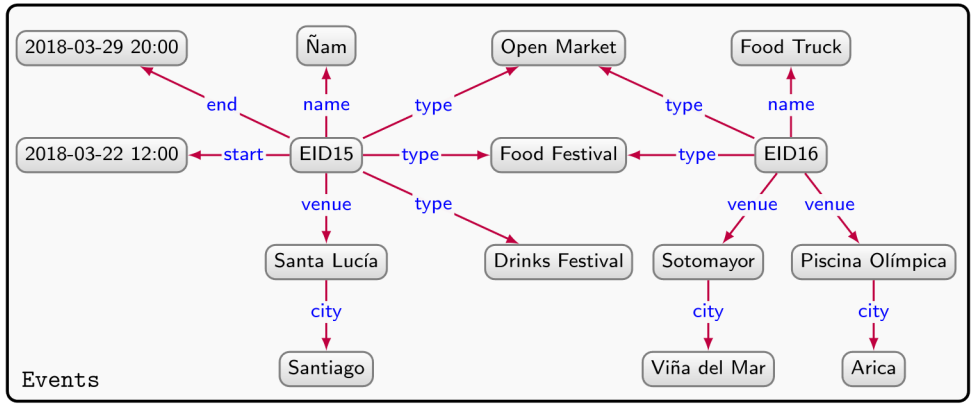
?ev	?vn1	?vn2
EID16	Piscina Olímpica	Sotomayor
EID16	Sotomayor	Piscina Olímpica
EID16	Piscina Olímpica	Piscina Olímpica
EID16	Sotomayor	Sotomayor
EID15	Santa Lucía	Santa Lucía

Aidan Hogan et al. Knowledge Graphs. 2021.

Graph Datasets and Querying

Graph query languages (SPARQL, Cypher, etc.) share 3 types of **graph patterns**:

- basic graph patterns
- complex graph patterns (relational algebra operators)



Q: "What are food festivals and drinks festivals not held in Santiago (optional: with start date and name)?"

Q_1: ?event —type→ Food Festival

Q_2: ?event —type→ Drinks Festival

Q_3: ?event —venue→ ?ven —city→ Santiago

Q_4: ?event —start→ ?start

Q_5: ?event —name→ ?name

$$Q := ((((Q_1 \cup Q_2) \triangleright Q_3) \bowtie Q_4) \bowtie Q_5)$$

union

minus (antijoin)

optional (left join)

?event	?start	?name
EID16		Food Truck

Aidan Hogan et al. Knowledge Graphs. 2021.

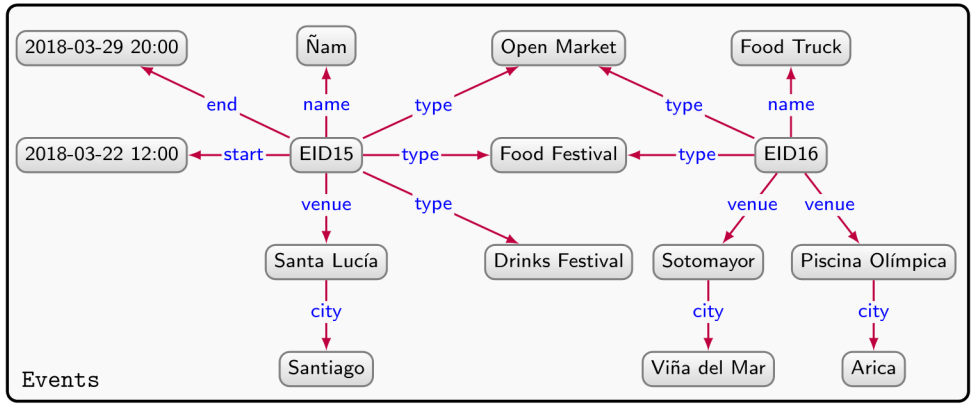
Graph Datasets and Querying

SPARQL Query

```
PREFIX : <http://example.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

FROM NAMED <http://example.org/events>

SELECT ?event ?name ?start WHERE {
  {
    {
      {
        ?event rdf:type :FoodFestival .          #Q1
      }
      UNION
      {
        ?event rdf:type :DrinksFestival .        #Q2
      }
      MINUS
      {
        ?event :venue ?venue .                   #Q3
        ?venue :city  :Santiago .
      }
    } OPTIONAL {
      ?event :start ?start .                     #Q4
    }
  } OPTIONAL {
    ?event :name ?name                           #Q5
  }
}
```



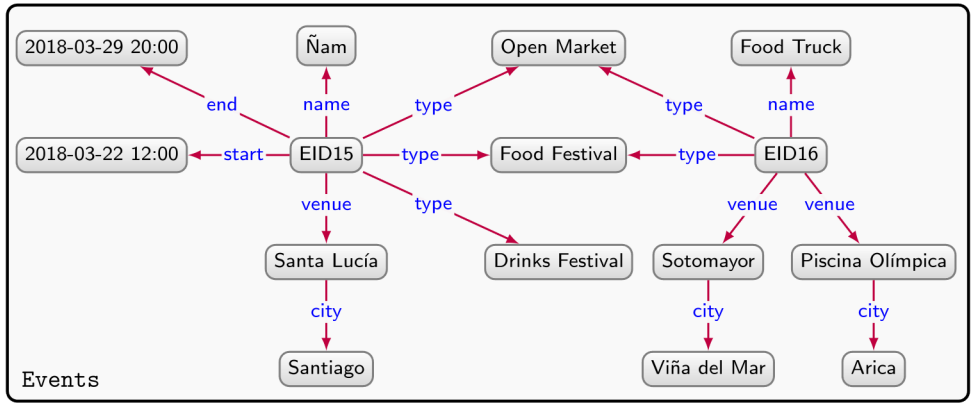
?event	?start	?name
EID16		Food Truck

Aidan Hogan et al. Knowledge Graphs. 2021.

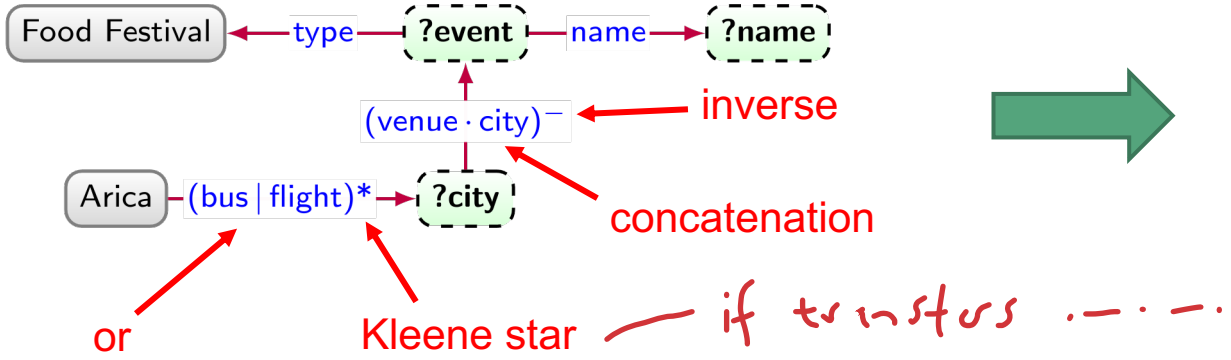
Graph Datasets and Querying

Graph query languages (SPARQL, Cypher, etc.) share 3 types of **graph patterns**:

- basic graph patterns
- complex graph patterns (relational algebra operators)
- navigational graph patterns (path expressions)



Q: "What are food festivals held in a city reachable from Arica by bus or flight?"



?event	?name	?city
EID15	Ñam	Santiago
EID16	Food Truck	Arica
EID16	Food Truck	Viña del Mar

Aidan Hogan et al. Knowledge Graphs. 2021.

Graph Datasets and Querying

Graph query languages (SPARQL, Cypher, etc.) share 3 types of **graph patterns**:

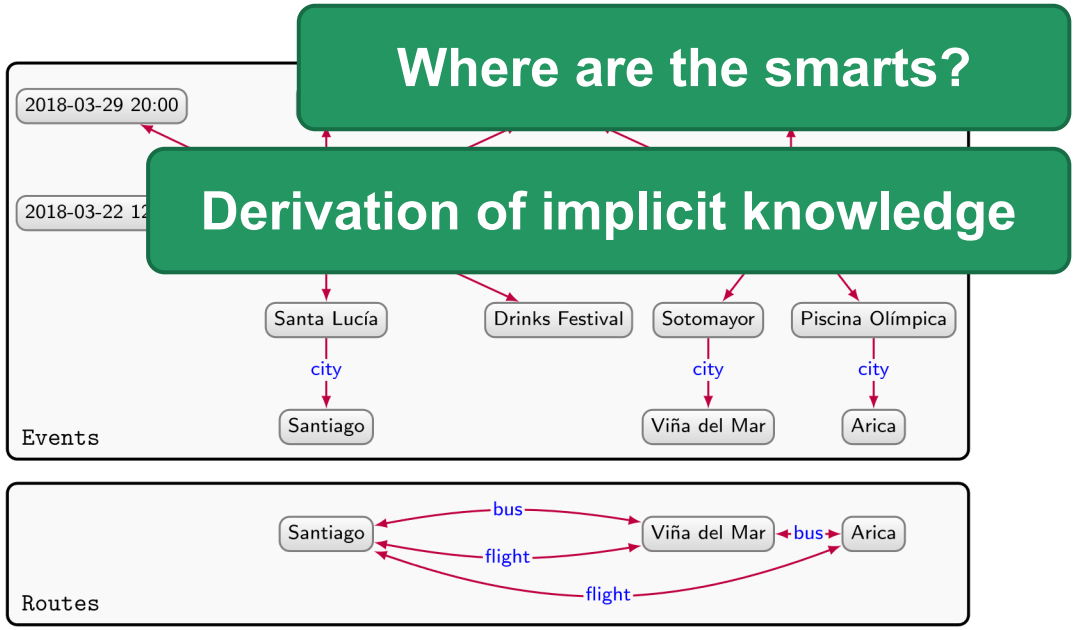
- basic graph patterns
- complex graph patterns (relational algebra operators)
- navigational graph patterns (path expressions)

SPARQL Query

```
PREFIX : <http://example.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

FROM NAMED <http://example.org/events>
FROM NAMED <http://example.org/routes>

SELECT ?event ?name ?city WHERE {
  ?event :name    ?name ;
         rdf:type :FoodFestival .
  :Arica (:bus|:flight)* ?city .
  ?city ^(:venue/:city) ?event .
}
```



?event	?name	?city
EID15	Ñam	Santiago
EID16	Food Truck	Arica
EID16	Food Truck	Viña del Mar

Aidan Hogan et al. Knowledge Graphs. 2021.

Universität St.Gallen

Our Journey

Prerequisites:
- **ASSE**
- **(…)**

Week 1:
Introduction

Week 2:
A Web for Machines

Week 3:
Knowledge Representation and Reasoning for the Web

Week 4:
Linked Data and Distributed Knowledge Graphs

Week 5:
Hypermedia Agents (Arch. and Programming)

Week 6 (Coordination I):
Agent Communication and Interaction

Week 7 (Coordination II):
Self-Organization and Stigmergy

Week 8 (Coordination III):
Normative MAS and Organizations

Week 9 (Coordination IV):
Trust & Reputation

Week 10:
Game Theory and Social Choice

Week 11:
Reinforcement Learning and Multi-Agent Learning

Week 12:
An Industry Perspective

Exercises

Ex1: Writing Your First Agent(s)!
Ex2: Automated Planning
Ex3: Web Ontologies
Ex4: Deductive Reasoning on the Web
Ex5: Cognitive Agents in Hypermedia Env.
Ex6: Interacting Agents on the Web
Ex7: Ant Colony Optimization
Ex8: Organized Agent
Ex9: Trustworthy Agents
Ex10: Axelrod's Agents
Ex11: Reinforcement Learning Agents
Course Review and Q&A

Universität St.Gallen

Any Questions / Comments / Doubts / Concerns?

Universität St.Gallen

Images

https://www.istockphoto.com/
https://freepik.com
https://www.bostondynamics.com/products/spot
https://billiards.colostate.edu/faq/cut/estimating-angle/
https://www.linearmotiontips.com/designing-linear-motion-tracks-robotic-positioning/
https://www.utas.edu.au/news/2017/6/7/301-a-day-in-the-life-of-a-typical-phd-student/