Methods of Artificial Intelligence: Lecture

6. Session: Knowledge Representation I

Kai-Uwe Kühnberger, Nohayr Muhammad

Winter Term 2022/2023

December 9th, 2022

Overview

- Philosophical Remarks
- Recap of Classical Logic
- Ontologies
- Representing Time
- Representing Space

Philosophical Remarks

What is Knowledge?

What is Knowledge?

- There is no generally accepted definition of what knowledge is
  - Different disciplines analyze knowledge differently
  - The common sense concept of knowledge is not a technical term

- Philosophical basis:
  - Traditionally the important question in the enlightenment era of philosophy was concerned with what can be known
  - In the 20th century, knowledge became a technical notion in different disciplines
  - New interest because of the rise of computer science

- Traditional usage in AI:
  - Content of the knowledge base of a system
  - Clearly, this approach is quite naïve, but it is probably the only possibility to avoid philosophical problems

What is Knowledge Representation?

- Knowledge Representation (KR)
  - Field of AI dedicated to represent information in a formal way
  - Goal: Allow automated reasoning on such information
  - It plays a key role in many AI problems
  - Several KR formalisms used
    - Predicate logic, Non-classical logic Prolog, statistical models, neural networks, diagrams
    - Key trade-off: Expressivity vs. Practicality

Document grammar

```
<!ELEMENT   table  (tr*)>
<!ELEMENT   tr     (td*)>
<!ELEMENT   td     (#PCDATA)>
```

	Hamburg	Berlin
Bielefeld	253.6	384.4
Giessen	442.0	468.2

```
<table>
  <tr>
    <td></td><td>Hamburg</td><td>Berlin</td>
  </tr>
  <tr>
    <td>Bielefeld</td><td>253.6</td><td>384.4</td>
  </tr>
  <tr>
    <td>Giessen</td><td>442.0</td><td>468.2</td>
  </tr>
</table>
```

Why Do We Need Knowledge Representation?

- Probably every application for which you can use a computer needs a knowledge representation formalism.
  - In symbolic approaches: relatively clear and transparent
  - In subsymbolic approaches: sometimes hidden

- Some examples:
  - Expert systems (IBM's Watson).
  - Background knowledge for semantic web applications.
    - Usually modeled in form of ontologies / terminological systems.
  - In linguistics, the formalism coding linguistic and world knowledge.
    - HPSG, transformational grammars, Montague semantics, DRT etc.
  - Cognitive modeling.
    - Find a representation for competence and performance issues.

Recap Classical Logic

The formal Basis of Reasoning

Repetition: Propositional Logic

- Formulas:
  - Given is a countable set of atomic propositions *AtProp* = {*p*,*q*,*r*,...}. The set of well-formed formulas *Form* of propositional logic is the smallest class such that it holds:
    - $\forall p \in \mathit{AtProp}$: $p \in \mathit{Form}$
    - $\forall \varphi, \psi \in \mathit{Form}$: $\varphi \wedge \psi \in \mathit{Form}$
    - $\forall \varphi, \psi \in \mathit{Form}$: $\varphi \vee \psi \in \mathit{Form}$
    - $\forall \varphi \in \mathit{Form}$: $\neg\varphi \in \mathit{Form}$

- Semantics:
  - Truth tables
    - A formula φ is valid if φ is true for all possible assignments of the atomic propositions occurring in φ.
    - A formula φ is satisfiable if φ is true for some assignment of the atomic propositions occurring in φ.
    - Models of propositional logic are specified by Boolean algebras.

Repetition: Predicate Logic

- We need signature
  - $\Sigma = (c_1,\ldots,c_n,f_1,\ldots,f_m,R_1,\ldots,R_l)$
- Defines the *vocabulary* of a logical language
  - $c_1,\ldots,c_n$ refer to constants / names
  - $f_1,\ldots,f_m$, refer to functions / function symbols (+ *arity*)
  - $R_1,\ldots,R_l$ refer to relations / relation symbols (+ *arity*)
- We can view *arity* as a function
  - *arity*: $\{f_1,\ldots,f_m,R_1,\ldots,R_l\} \rightarrow N$
- The set T_Σ of Terms is the smallest class such that:
  - A variable $x \in Var$ is a term (Var is a countable set of variables)
  - A constant $c_i \in \{c_1,\ldots,c_n\}$ is a term.
- If $f_i \in \{f_1,\ldots,f_n\}$ is a function symbol of arity r and $t_1,\ldots,t_r$ are terms, then $f_i(t_1,\ldots,t_r)$ is a term.

Repetition: Predicate Logic

- The set F_{Σ} of Formulas is the smallest class such that:
  - If R_j is a predicate symbol of arity r and $t_1,\ldots,t_r$ are terms, then $R_j(t_1,\ldots,t_r)$ is a formula (atomic formula or literal).
  - For all formulas φ and ψ: $\varphi \land \psi$, $\varphi \lor \psi$, $\neg\varphi$, $\varphi \rightarrow \psi$, $\varphi \leftrightarrow \psi$ are formulas.
  - If $x \in Var$ and φ is a formula, then $\forall x\varphi$ and $\exists x\varphi$ are formulas.

- Notice that “term” and “formula” are rather different concepts. Terms are used to define formulas and not vice versa.

Repetition: Predicate Logic

- Semantics (meaning) of FOL formulas
  - Expressions of FOL language of a given signature Σ are interpreted in structures of the same signature (or algebra)
  - $\mathcal{M} = (\mathcal{U}, (c'_1,\ldots,c'_n,f'_1,\ldots,f'_m,R'_1,\ldots,R'_l))$ where
  - c'_i is an element of the universe $\mathcal{U}$
  - f'_i is a function and R'_i is a relation
- An *interpretation function* [[.]] maps:
  - terms to elements of the universe: $[[\,.\,]]: T_\Sigma \to \mathcal{U}$
  - formulas to truth-values: $[[\,.\,]]: F_\Sigma \to \{\mathit{true}, \mathit{false}\}$
- Recursive definition for interpreting terms and formulas:
  - for $c \in \{c_1,\ldots,c_n\}$: $[[c]] = c'$, for $x \in Var$: $[[x]] \in \mathcal{U}$
  - $[[f_i(t_1,\ldots,t_r)]] = f_i'([[t_1]],\ldots,[[t_r]])$
  - $[[R(t_1,\ldots,t_r)]] = \text{true}$ iff $\langle[[t_1]],\ldots,[[t_r]]\rangle \in R'$
  - $[[\varphi \wedge \psi]] = \text{true}$ iff $[[\varphi]] = \text{true}$ and $[[\psi]] = \text{true}$
  - $[[\varphi \vee \psi]] = \text{true}$ iff $[[\varphi]] = \text{true}$ or $[[\psi]] = \text{true}$
  - $[[\neg\varphi]] = \text{true}$ iff $[[\varphi]] = \text{false}$
  - $[[\forall x \varphi(x)]] = \text{true}$ iff for all $d \in \mathcal{U}$: $[[\varphi(x)]]_{x|d} = \text{true}$
  - $[[\exists x \varphi(x)]] = \text{true}$ iff exists $d \in \mathcal{U}$: $[[\varphi(x)]]_{x|d} = \text{true}$

Ontologies

Representing Concepts as Hierarchies

Ontologies

- There is no generally accepted definition of an ontology
  - "Representation of the categories, properties and relations between concepts that can refer to one, many or all domains of discourse"

    (Adapted from Wikipedia)
  - "An ontology is an explicit specification of a conceptualization"

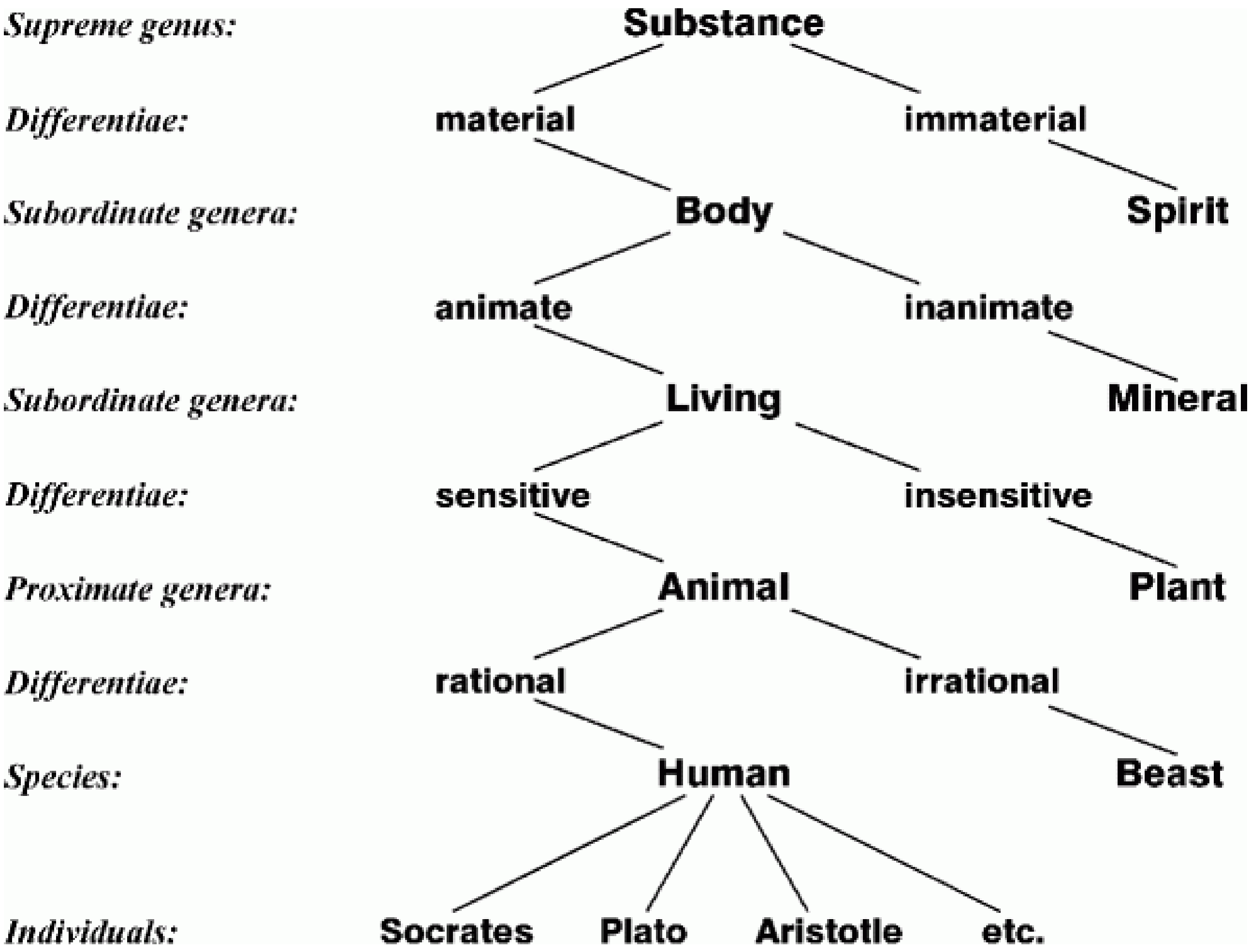
    (Gruber, 1993)
  - "An ontology is a shared understanding of some domain of interest"

    (Uschold & Gruninger, 1996)

- Formally: a tuple **O** = $\langle C$, `is_a`, $R, \sigma\rangle$ where:
  - C is a set of concepts,
  - `is_a` $\subseteq C \times C$ is a partial order relation on C,
  - R is a set of relations,
  - σ is a function that assigns to each relation an arity

    (Stumme & Maedche, 2001)

Ontologies: an Old Example



Tree of Porphyry

(III Century)

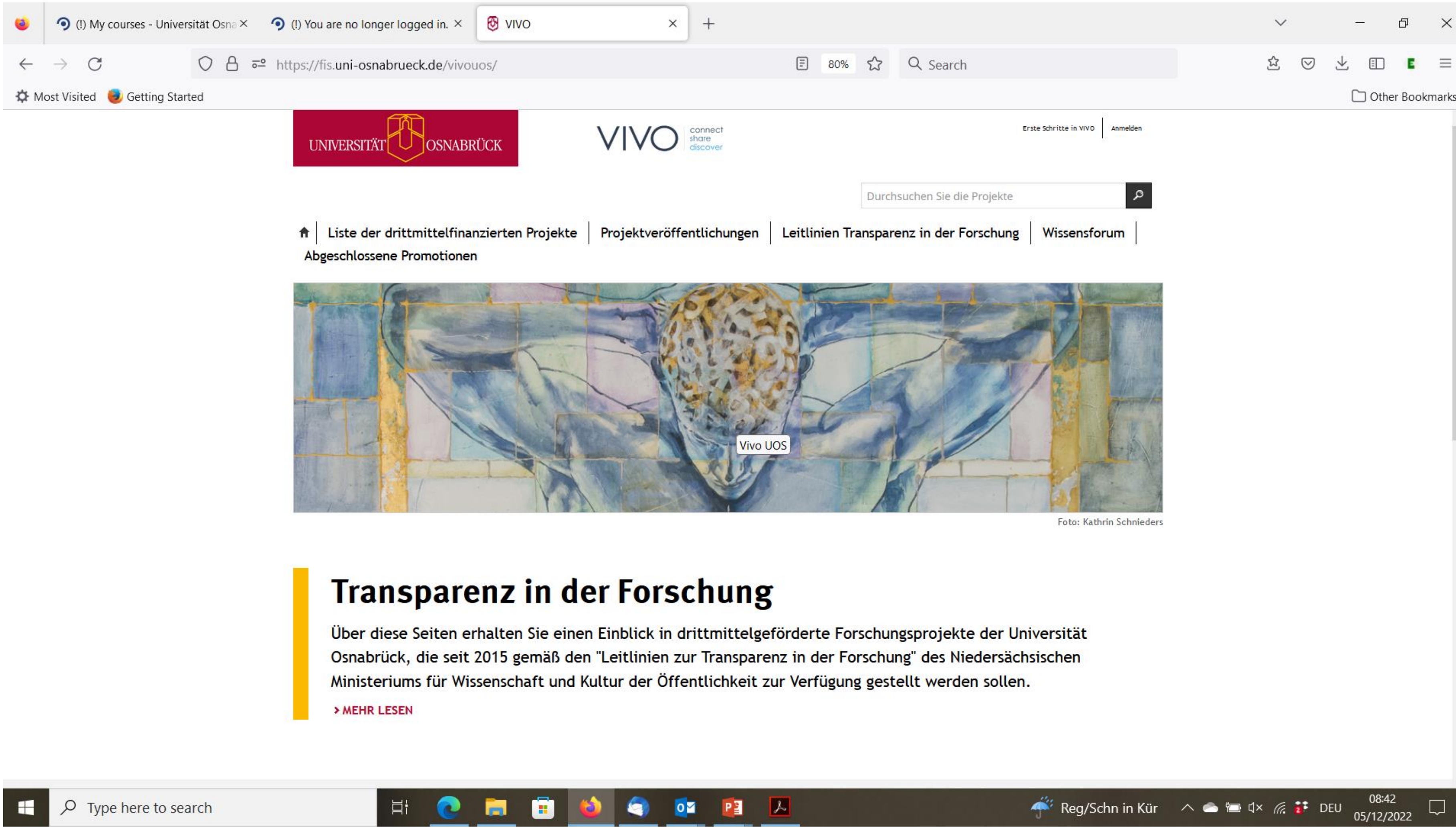
Domain-Specific vs. General-Purpose Ontologies

- Domain-specific ontologies
  - Domain ontologies refer to a specific part of the world
  - E.g., biology, politics…
  - Easier to create domain-specific ontologies
- Question: Is it possible to get to a general-purpose ontology?
  - Desirable properties
    1. Still applicable in any special-purpose domain
    2. Different areas of knowledge need to be unified
  - Reasoning might involve several areas simultaneously

Ontological Engineering

- Existing ontologies have been created along four routes:
  1. By a team of trained ontologists or logicians
     - E.g., CYC system, university’s FIS (research information system)
  2. By importing categories, attributes, and values from existing databases
     - E.g., DBpedia from Wikipedia
  3. By extracting information from text documents
     - E.g., TextRunner from a large corpus of web documents
  4. By enticing amateurs to enter commonsense knowledge
     - E.g., OpenMind built by volunteers who proposed facts in English

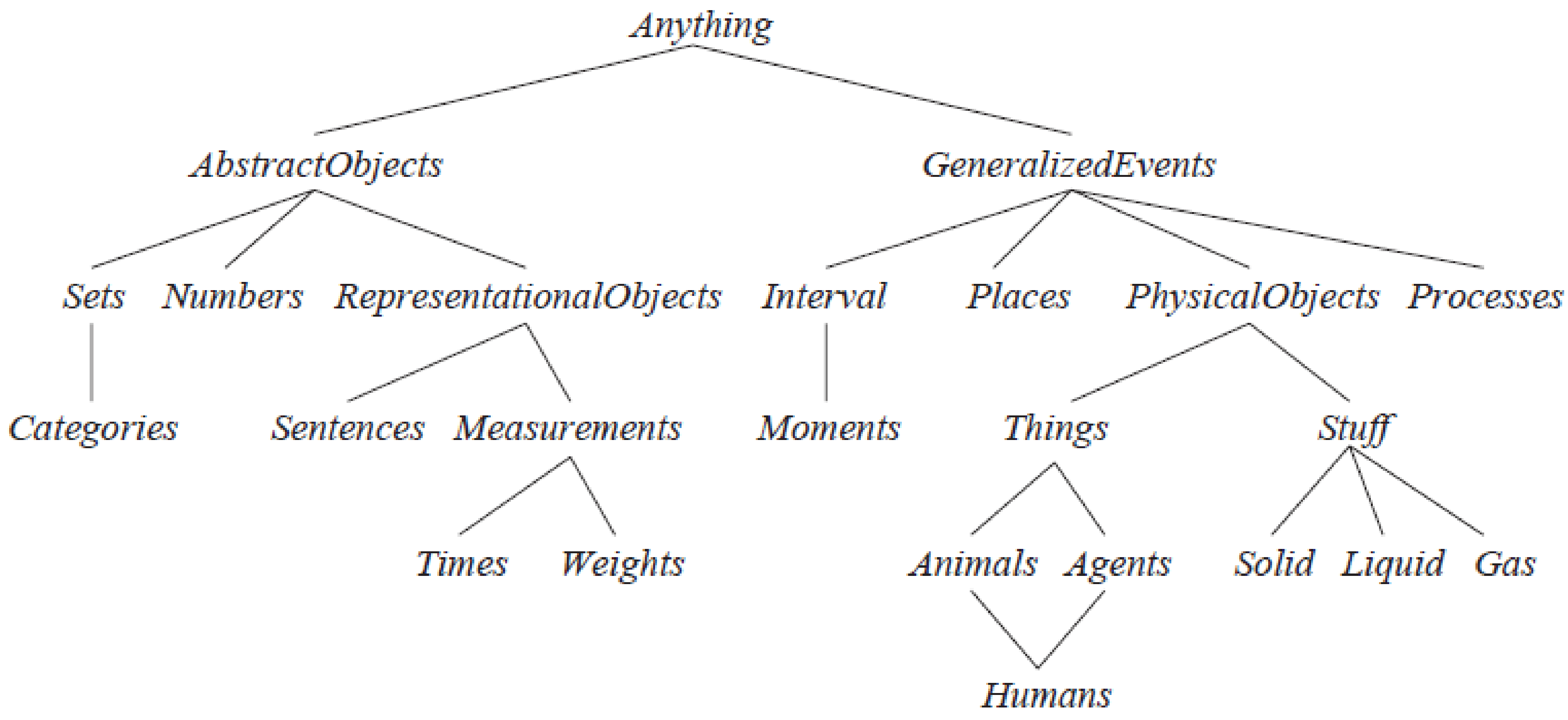
Ontology Engineering at UOS



Upper Ontology

- We will try to sketch a general-purpose ontology
- Focus on general concepts that occur in several domains
  - Events
  - Time
  - Physical Objects
  - Beliefs
- We cannot represent *everything* in the world
  - E.g., we will define what it means to be a physical object
  - Details of different object types (television, book, etc.) can be filled in later
- We call such a general framework *upper ontology*
  - Graphical representation
    - General concepts at the top
    - Specific concepts below

Upper Ontology



Representing Time

Relational Properties instead of Exact Measurements

Knowledge about Space and Time

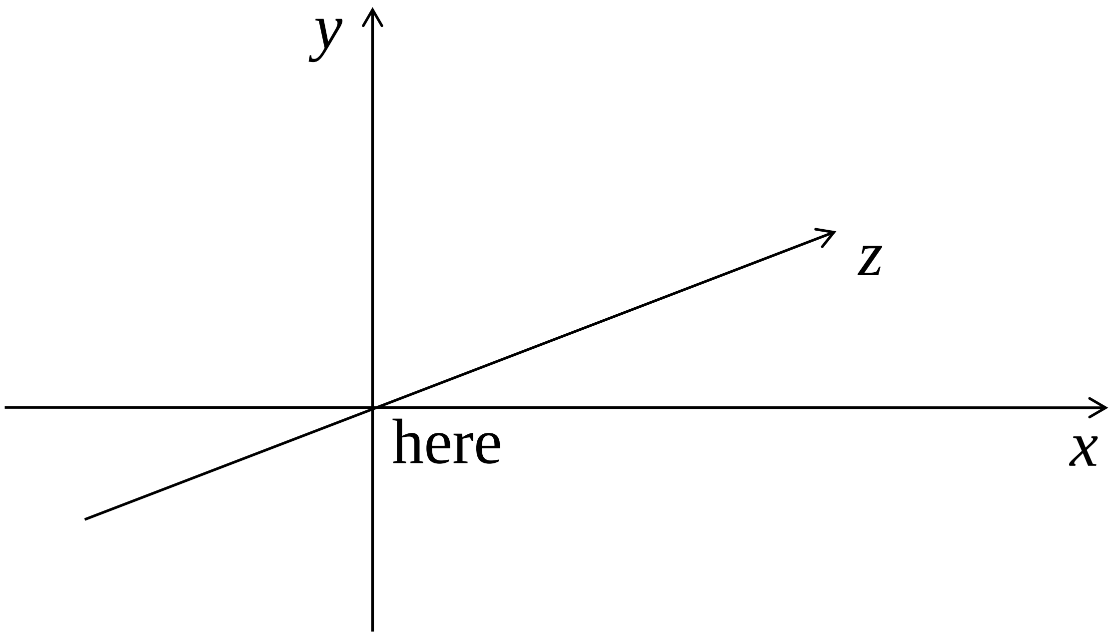
- Crucial questions are:
  How is our conceptualization of space and time related to planning and acting?
  How is knowledge about space and time represented?
  Which inference mechanisms exist for this domain?

- A secondary question is what the objective nature of space and time is
  - This is a question for physicists

- Here are some differences between space and time

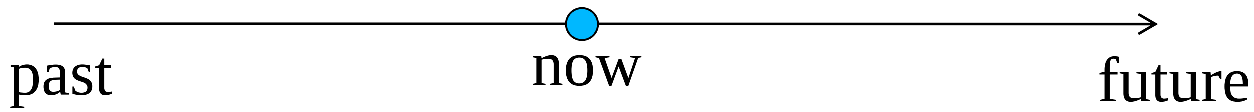
Space	**Time**
3-dimensional	1-dimensional
In all directions the same properties	There is a designated direction
Localization of objects	Localization of events
Direct perception possible	No direct perception possible

Knowledge about Space and Time



space

time

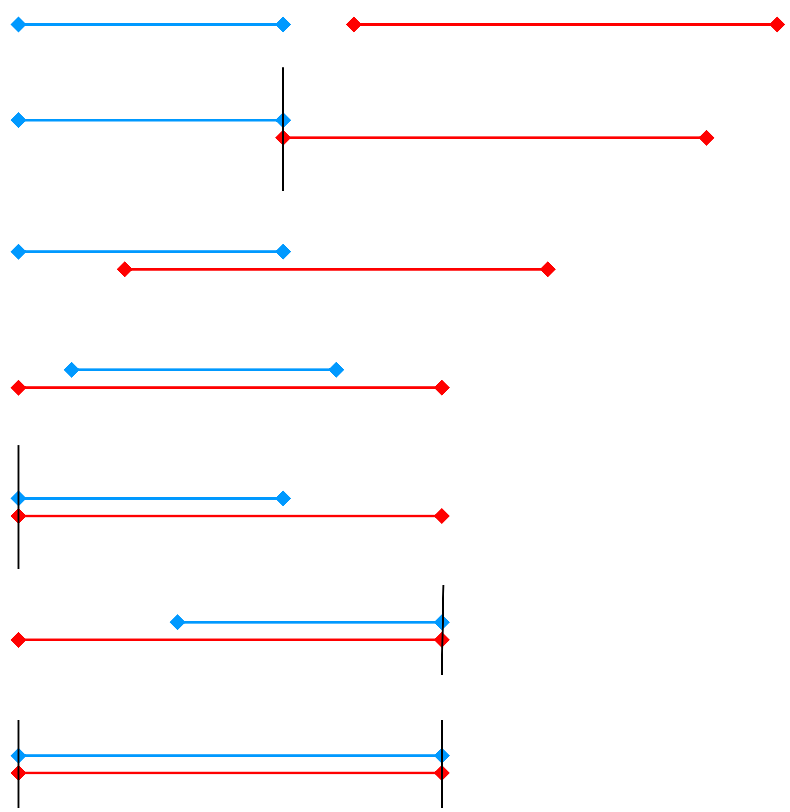


Knowledge about Space and Time

- What frameworks exist to represent space and time?
- Time: Allen’s tense logic (time intervals), time point logics (e.g. Since and Until Tense Logic), time series of feature-value pairs etc.
  - What about recurrent connections in NNs?
- Space: RCC-calculi, cross and double-cross calculus, cardinal direction calculus etc.
  - What about Euclidean geometry?
- Many of these approaches are based on (relational) algebra and/or logic.

Allen's Tense Logic

- 7 basic logical relations of time intervals
  - Before(t_1,t_2)
  - Meets(t_1,t_2)
  - Overlaps(t_1,t_2)
  - During(t_1,t_2)
  - Starts(t_1,t_2)
  - Finishes(t_1,t_2)
  - Equals(t_1,t_2)



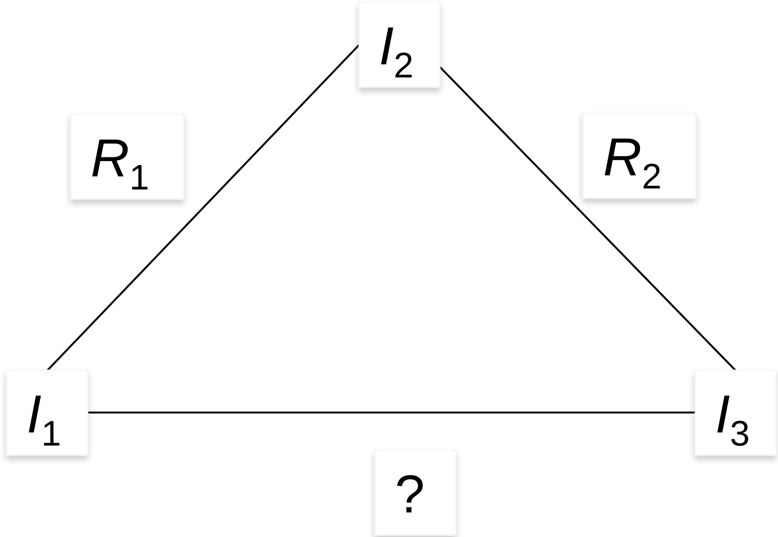
Allen's Tense Logic

- STARTS (t1,t2)
  - t1 has the same beginning as t2, but ends before the end of t2
- FINISHES (t1,t2)
  - t1 has the same end as t2, but starts after the start of t2
- DURING (t1,t2)
  - t1 is completely contained in t2
- BEFORE (t1,t2)
  - t1 is before t2 and t1 and t2 do not overlap
- OVERLAP (t1,t2)
  - t1 starts before t2 and ends after the beginning of t2, but before the end of t2
- MEETS (t1,t2)
  - t1 is before t2 and there is no interval between t1 and t2, i.e. t1 ends when t2 starts
- EQUAL (t1,t2)
  - t1 and t2 are the same interval

- IS-STARTED-BY (t2,t1)
- IS-ENDED-BY (t2,t1)
- CONTAINS (t2,t1)
- AFTER (t2,t1)
- IS-OVERLAPED-BY (t2,t1)
- IS-MET-BY (t2,t1)

Allen's Tense Logic

- How is it possible to reason in this interval structure?
- We need the composition table of relations:
- Given intervals I1, I2, I3
- and relations R1(I1,I2), R2(I2,I3)
- what are the possible relations between I1 and I3?

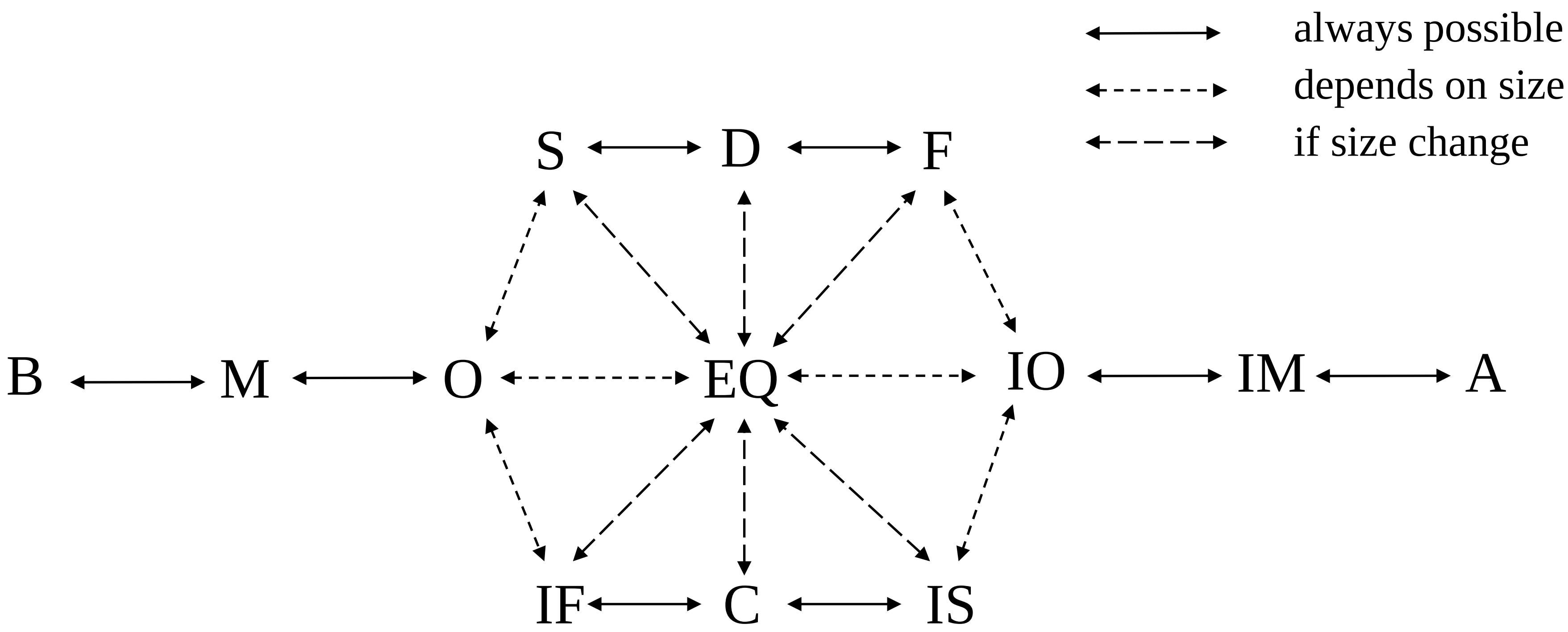


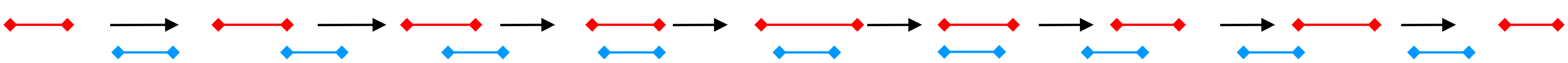
Allen's Tense Logic: Composition Table

	B	M	O	S	D	F	EQ	IF	C	IS	IO	IM	A
B	B	B	B	B	B,M,O, S,D	B,M,O, S,D	B	B	B	B	B,M,O ,S,D	B,M,O ,S,D	*
M	B	B	B	M	O,S,D	O,S,D	M	B	B	M	O,S,D	F,E, IF	C,IS,I O.IM, A
O	B	B	B,M,O	O	O,S,D	O,S,D	O	B,M,O	B,M,O ,IF,C	O,IF,C			
S	B	B	B,M,O	S	D	D	S						A
D	B	B		D	D	D	D					A	A
F	B	M		D	D	F	F					A	A
EQ	B	M	O	S	D	F	EQ	IF	C	IS	IO	IM	A
IF	B	M	O	O			IF	IF	C	C			
C	B,M,O, C						C	C	C	C			
IS	B,M,O, C					IO	IS	C	C	IS	IO	IM	A
IO	B,M,O, C					IO	IO				A,IM, IO	A	A
IM	B,M,O, C					IM	IM	IM	A	A	A	A	A
A	*					A	A	A	A	A	A	A	A

Allen’s Tense Logic

- Possible transitions between basic relations





Some Further Remarks

- Temporal Reasoning is studied in a variety of disciplines
  - Philosophy
    - Interest: Syntax and semantics of tense logic
    - Important example: *S* and *U* tense logic (Kamp, Burgess) (with operators *since* and *until*)
    - Aspects of branching time
  - Artificial Intelligence
    - Planning
    - Interest: practical applications, synchronizing processes
  - Linguistics
    - Interest: Models for time in natural language expressions
      - Notice that temporal relations coded in natural language can be quite complex
    - Usually connected with intentional logic

The Rochester Door (Allen 1991)

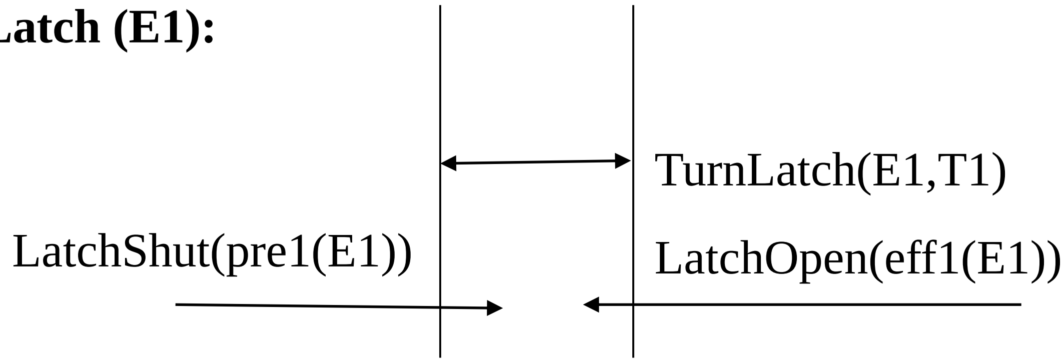
- The Rochester Door is taken from Allen 1991:
  - The door to the computer science building in Rochester is designed that it requires both hands to open it
  - A spring lock must be held open (with one hand), while the door is pulled open (with the other)
  - Notice: the effect of doing both actions (at the same time) is different from the sum of their individual effects

- How can we model this situation using time intervals?

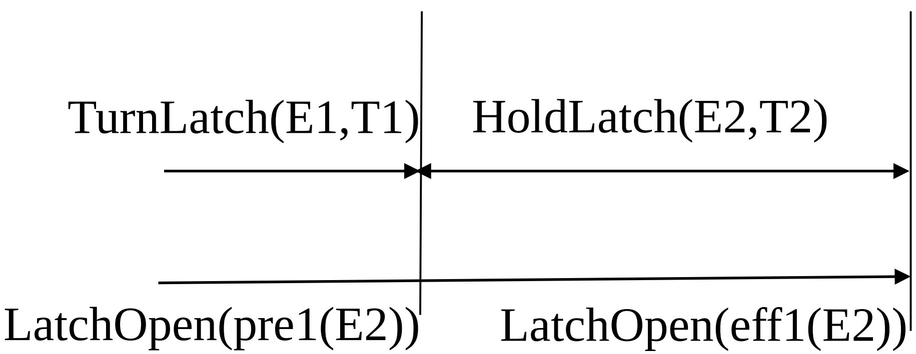
- The following two slides give a solution to the problem formulated in Allen 1991

Actions, Preconditions, and Effects

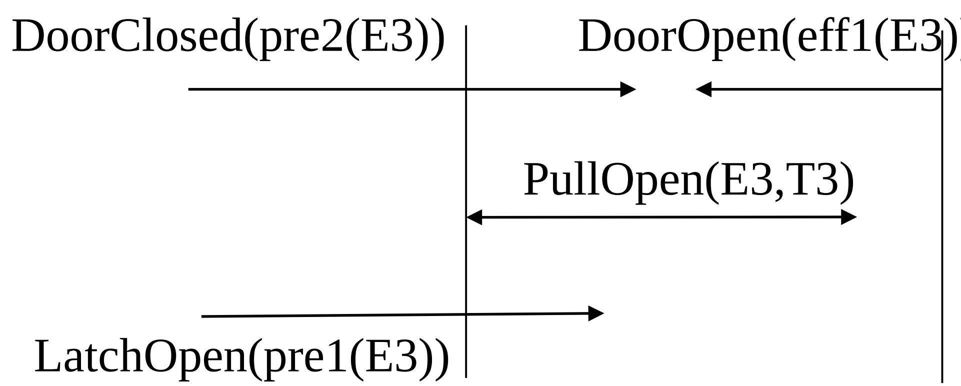
TurnLatch (E1):



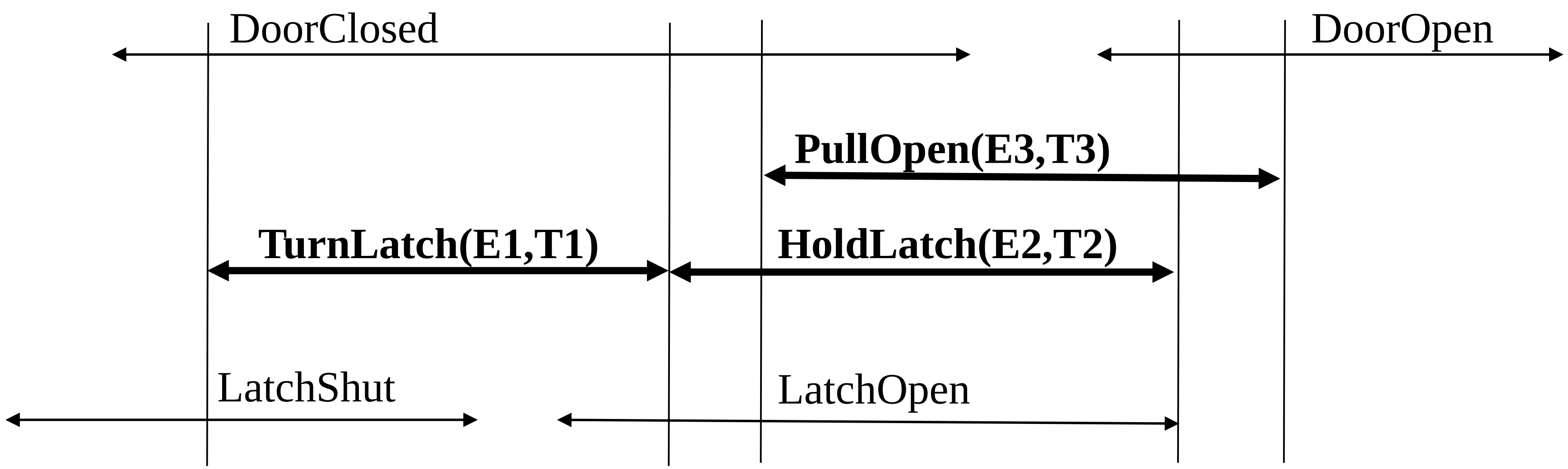
HoldLatch (E2):



PullOpen (E3):



The Rochester Door (Allen 1991)



Allen shows how a logical framework can be developed to make these intuitions precise, e.g. we get the following ‘simple’ qualitative temporal relations:

Meets[T1,T2]

Overlap[T2,T3]

QUESTIONS?