LUND UNIVERSITY

Knowledge Representation Some Remarks

but: Read Chapter 10 in AIMA!

Jacek Malec, AI@CS

26 February 2009

AI@CS

Department of Computing Science

Programming vs KR

Programming:

- construct an algorithm to solve a problem
- choose a programming language
- implement algorithm in the language
- run the program to solve the problem

Knowledge Representation and Engineering:

- identify knowledge necessary to solve the problem
- choose a representation language
- formulate knowledge in the language
- check if solution can be derived

AI@CS

Department of Computing Science

Knowledge

There is evidence that people memorize just **the important stuff.**

- objects
- facts, relations
- rules (generic knowledge)
- causality
- meta-knowledge
- procedural knowledge
- knowledge about processes
- ...

AI@CS

Department of Computing Science

Meta-knowledge

knowledge about other's (or our own)

- knowledge (Jacek knows that Earth revolves around Sun)
- beliefs (Jacek believes that his daughter is smart)
- needs (Jacek needs more time for checking assignments:-)
- goals (Jacek must write a paper by tomorrow)
- intentions (Jacek wants to be ready with assignment 3 in time)
- capabilities (Jacek can drive)
- ...

AI@CS

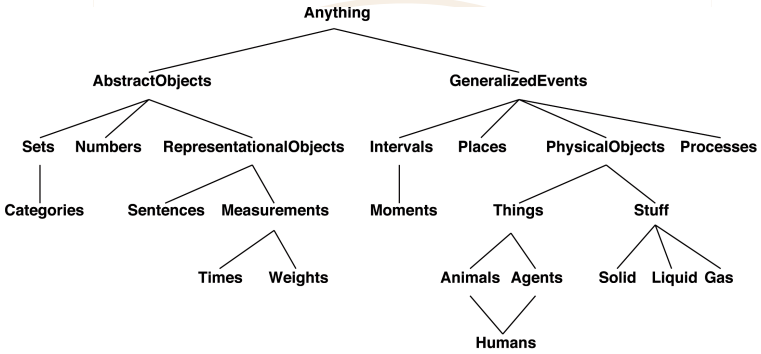
Department of Computing Science

Procedural knowledge

How to:

- change a diaper?
- serve in tennis?
- ride a bicycle?
- drive a car?
- ...

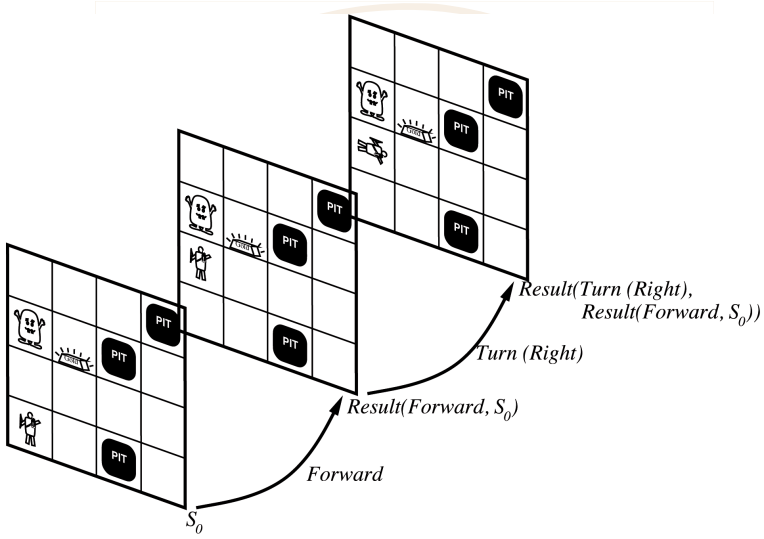
Ontology



Problems with predicate logic:

- static
- flat
- qualification/ramification/frame problem
- exceptions
- strength
- ...

Actions, situations



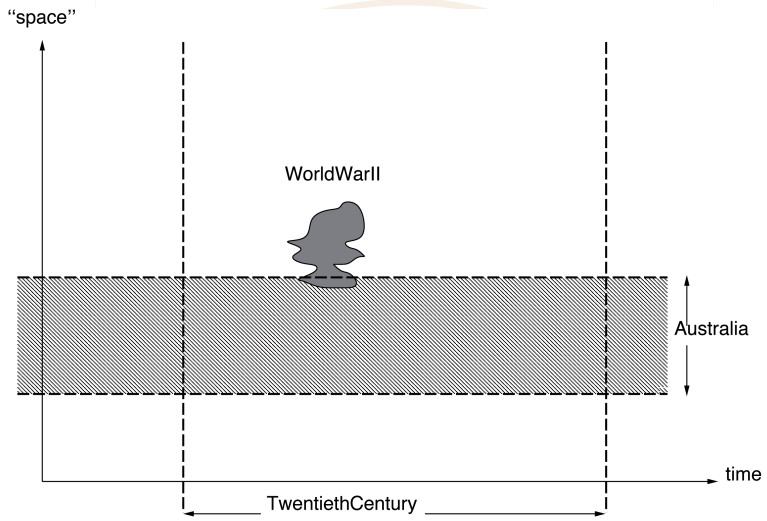
Situation calculus

Action descriptions:

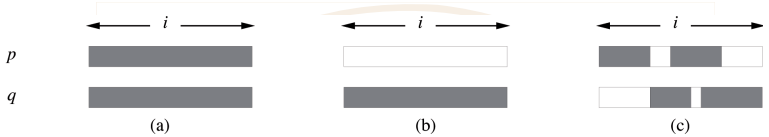
- possibility axioms (when is an action possible)
- effect axioms (what's its effect, what changes)
- frame axioms (what remains the same) Important issue!

Quite often we need richer ontology.

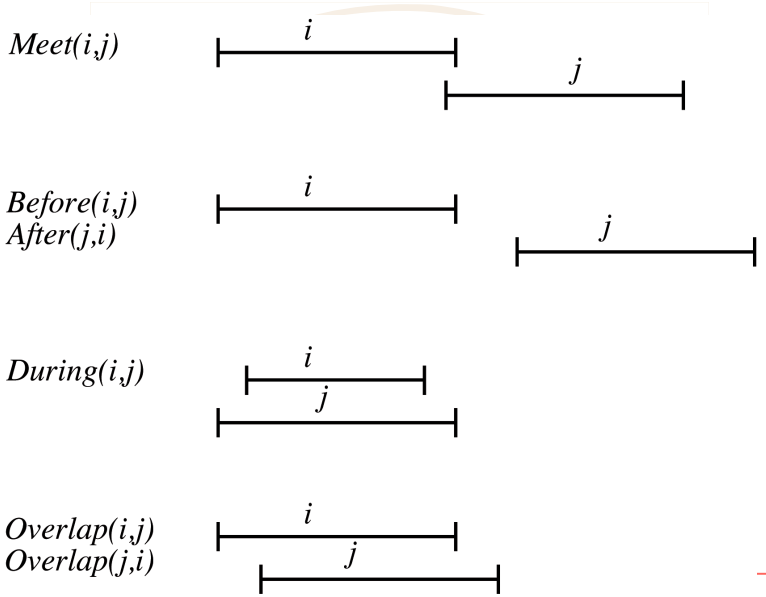
Events



Events



Interval calculus



Representation of exceptions:

$\forall x(Bird(x) \wedge \neg Pinguin(x) \wedge \neg Ostrich(x) \rightarrow Flies(x))$
But if we know $Bird(Tweety)$ we can't say whether it flies!

Qualification Problem:

Unfortunately, many other birds don't fly either: caged birds, dead birds, birds with broken wing, bird with feathers covered by oil, ...
$\forall x(Bird(x) \wedge \neg Pinguin(x) \wedge \neg Ostrich(x) \wedge \ldots \rightarrow Flies(x))$

Other related problems: frame problem, ramification problem

Solutions:

- non-monotonic logics
- fuzzy logic
- probabilistic logic
- possibilistic logic
- ...

Monotonic vs. non-monotonic

$Th(\Delta)$

monotonic logic:
if $\Delta \subseteq \Delta'$ then $Th(\Delta) \subseteq Th(\Delta')$

non-monotonic logic:
if $\Delta \subseteq \Delta'$ and $Th(\Delta) \supset Th(\Delta')$

Closed World Assumption:

Things that cannot be proven true are probably false

leads to:

Negation as failure

LUND UNIVERSITY

AI@CS

Department of Computing Science

Consider:

$$\Delta = \{A(Stockholm), A(Sturup), A(Copenhagen), A(Oslo), Conn(Oslo, Copenhagen), Conn(Stockholm, Oslo), \forall(x, y, z)(Conn(x, y) \wedge Conn(y, z) \rightarrow Conn(x, z))\}$$

Because it's not the case that $Conn(Sturup, Stockholm)$, CWA gives us immediately $\neg Conn(Sturup, Stockholm)$.
But let us add $Conn(Sturup, Copenhagen)$ to Δ.
$Conn(Sturup, Stockholm)$ can be proven now, so $\neg Conn(Sturup, Stockholm)$ does not appear as a consequence.
Non-monotonicity!

LUND UNIVERSITY

AI@CS

Department of Computing Science

CWA is syntax-dependent:

If $\Delta =$
$\{Single(John), Single(Mary), Clever(Kent)\}$
then CWA gives us:
$\{\neg Clever(John), \neg Clever(Mary), \neg Single(Kent)\}$.

But if $\Delta =$
$\{\neg Married(John), \neg Married(Mary), Clever(Kent)\}$
then CWA gives us:
$\{\neg Clever(John), \neg Clever(Mary), \neg Married(Kent)\}$

LUND UNIVERSITY

AI@CS

Department of Computing Science

Effective representation of knowledge:

- storing
- retrieval
- modification

What should be represented?

- use logic
- not necessarily FOL

How should it be represented?

- whatever method you find suitable

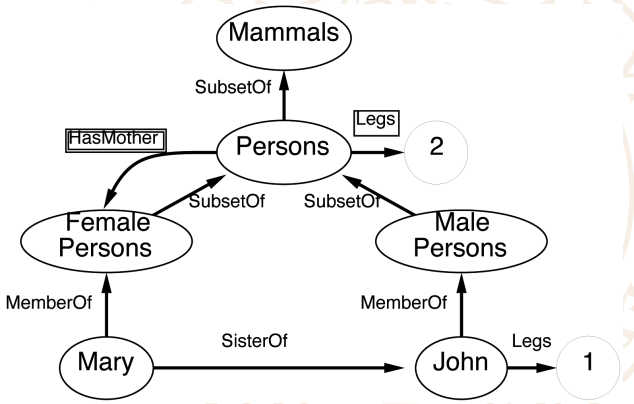
LUND UNIVERSITY

AI@CS

Department of Computing Science

Semantic networks:

Precursor of Description Logics and Semantic Web Languages (OWL, DAML, OIL, ...)



LUND UNIVERSITY

AI@CS

Department of Computing Science

Reasoning in SN:

- inheritance via Isa links
- intersection paths
- special meaning associated with some links (like Isa or $Is-part$)
- classification, consistency, subsumption

May be effective given some restrictions on the logic (semantic network).