Knowledge Representation

Chapter 10

Outline

- ♢ Ontological engineering
- ♢ Categories and objects
- ♢ Actions, situations and events
- ♢ Mental events and mental objects
- ♢ The Internet shopping world
- ♢ Reasoning systems for categories
- ♢ Reasoning with default information
- ♢ Truth maintenance systems

Ontological Engineering

How to create more general and flexible representations:

– Concepts like actions, time, physical object and beliefs

– Operates on a bigger scale than Knowledge Engineering

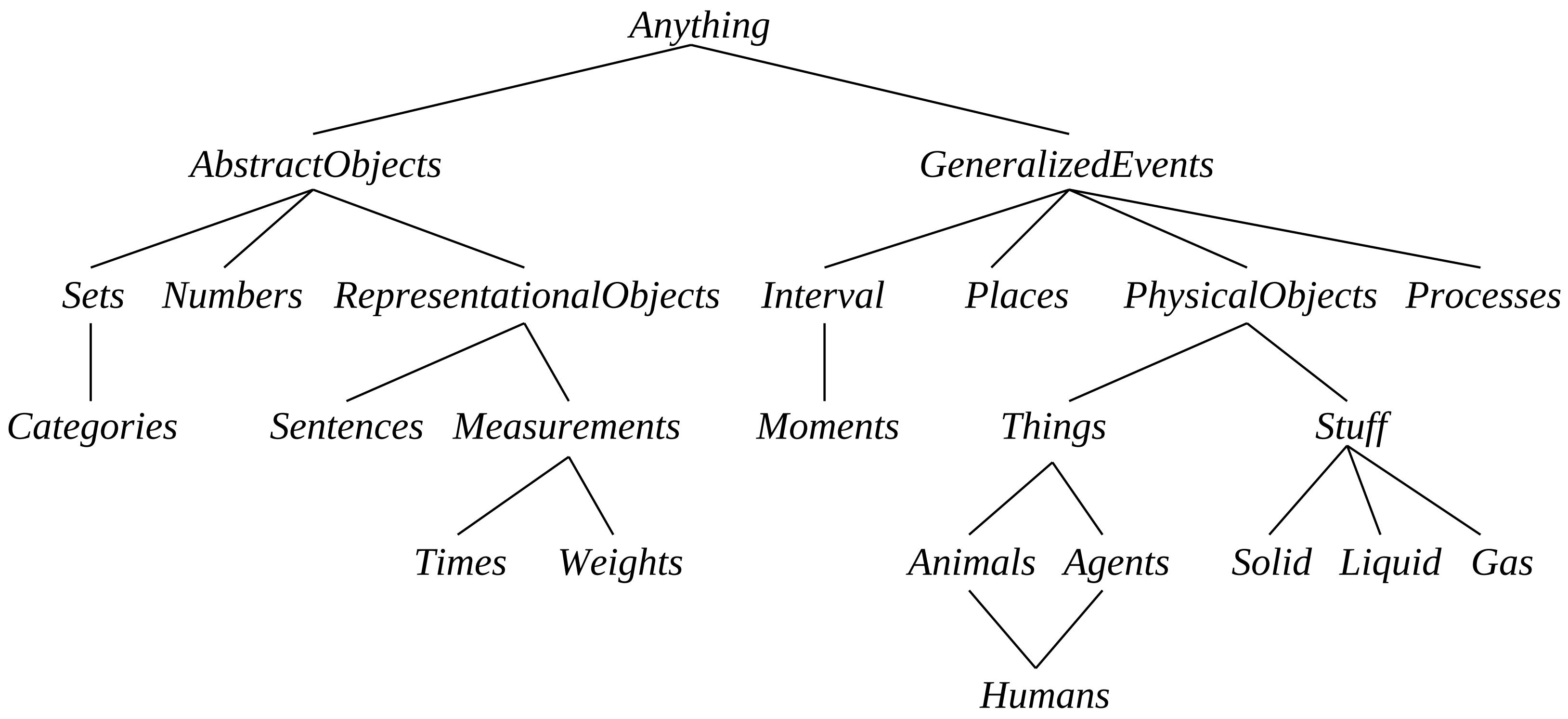
Define general framework of concepts

– Upper ontology

Limitations of logic representation

– Red, green and yellow tomatoes: exceptions and uncertainty

The upper ontology of the world



Difference with special-purpose ontologies

A general-purpose ontology should be applicable in more or less any special-purpose domain.

- Add domain-specific axioms

In any sufficiently demanding domain different areas of knowledge need to be unified.

- Reasoning and problem solving could involve several areas simultaneously

What do we need to express?

- Categories, Measures, Composite objects, Time, Space, Change, Events, Processes, Physical Objects, Substances, Mental Objects, Beliefs

Categories and Objects

KR requires the organization of objects into categories

- Interaction at the level of the object
- Reasoning at the level of categories

Categories play a role in predictions about objects

- Based on perceived properties

Categories can be represented in two ways by FOL

- Predicates: $Apple(x)$
- Reification of categories into objects: $Apples$

Category = set of its members

Category organization

◇ Relation = inheritance:

All instance of food are edible,
fruit is a subclass of food, and
apples is a subclass of fruit
then an apple is edible.

◇ Defines a taxonomy

FOL and categories

An object is a member of a category

$$MemberOf(BB_{12}, Basketballs)$$

A category is a subclass of another category

$$SubsetOf(Basketballs, Balls)$$

All members of a category have some properties

$$\forall x (MemberOf(x, Basketballs) \Rightarrow Round(x))$$

All members of a category can be recognized by some properties

$$\forall x (Orange(x) \land Round(x) \land Diameter(x) = 9.5\text{in} \land MemberOf(x, Balls) \Rightarrow MemberOf(x, BasketBalls))$$

A category as a whole has some properties

$$MemberOf(Dogs, DomesticatedSpecies)$$

Relations between categories

Two or more categories are disjoint if they have no members in common:

$Disjoint(s) \Leftrightarrow$
$(\forall c_1, c_2 \quad c_1 \in s \wedge c_2 \in s \wedge c1 \neq c2 \Rightarrow Intersection(c1, c2) = \{\})$

Example: $Disjoint(\{Animals, Vegetables\})$

A set of categories s constitutes an exhaustive decomposition of a category c if all members of the set c are covered by categories in s:

$ExhaustiveDecomposition(s, c) \Leftrightarrow (\forall i \quad i \in c \Leftrightarrow \exists c_2 \quad c_2 \in s \wedge i \in c_2)$

Example:
$ExhaustiveDecomposition(\{Americans, Canadian, Mexicans\}, NorthAmericans)$.

Relations between categories (contd.)

A partition is a disjoint exhaustive decomposition:

$$Partition(s, c) \Leftrightarrow Disjoint(s) \wedge ExhaustiveDecomposition(s, c)$$

Example: $Partition(\{Males, Females\}, Persons)$.

Example: Is $(\{Americans, Canadian, Mexicans\}, NorthAmericans)$ a partition? – No! There might be dual citizenships.

Categories can be defined by providing necessary and sufficient conditions for membership

$$\forall x \quad Bachelor(x) \Leftrightarrow Male(x) \wedge Adult(x) \wedge Unmarried(x)$$

Natural kinds

Many categories have no clear-cut definitions (e.g., chair, bush, book).

Example: Tomatoes: sometimes green, red, yellow, black. Mostly round.

One solution: subclass using category $Typical(Tomatoes)$.

$$\boxed{Typical(c) \subseteq c}$$

$$\forall x \quad x \in Typical(Tomatoes) \Rightarrow Red(x) \land Spherical(x).$$

We can write down useful facts about categories without providing exact definitions.

Wittgenstein (1953) gives an exhaustive summary about the problems involved when exact definitions for natural kinds are required in his book "Philosophische Untersuchungen".

What about "bachelor"? Quine (1953) challenged the utility of the notion of strict definition. We might question a statement such as "the Pope is a bachelor".

Physical composition

One object may be part of another:

- $PartOf(Bucharest, Romania)$
- $PartOf(Romania, EasternEurope)$
- $PartOf(EasternEurope, Europe)$

The $PartOf$ predicate is transitive (and reflexive), so we can infer that $PartOf(Bucharest, Europe)$

More generally:

$$\forall x \quad PartOf(x, x)$$

$$\forall x, y, z \quad PartOf(x, y) \land PartOf(y, z) \Rightarrow PartOf(x, z)$$

Physical composition (contd.)

Often characterized by structural relations among parts.

E.g. $Biped(a) \Rightarrow$

$$
\begin{aligned}
(\exists l_1, l_2, b)(Leg(l_1) \land Leg(l_2) \land Body(b) \land \\
PartOf(l_1, a) \land PartOf(l_2, a) \land PartOf(b, a) \land \\
Attached(l_1, b) \land Attached(l_2, b) \land \\
l_1 \neq l_2 \land (\forall l_3)(Leg(l_3) \Rightarrow (l_3 = l_1 \lor l_3 = l_2)))
\end{aligned}
$$

Physical composition (contd.)

$PartPartition$: a relation analogous to the $Partition$ relation for categories.

$BunchOf(X)$: a composite object consisting of all X's

$$BunchOf(\{Apple_1, Apple_2, Apple_3\})$$

$$\boxed{BunchOf(\{x\}) = x}$$

Definition of $BunchOf$ in terms of the $PartOf$ Relation:

$$\forall x \; x \in s \Rightarrow PartOf(x, BunchOf(s))$$

$BunchOf(s)$ is the smallest object satisfying this condition.

$$\forall y [\forall x \; x \in s \Rightarrow PartOf(x, y)] \Rightarrow PartOf(BunchOf(s), y)$$

logical minimization: defining an object as the smallest one satisfying certain condition.

Measurements

Objects have height, mass, cost, ...

Values that we assign to these are measures: measure objects

- Combine unit functions with a number:
  $Length(L_1) = Inches(1.5) = Centimeters(3.81)$.
- Conversion between units:
  $\forall i \quad Centimeters(2.54 \times i) = Inches(i)$.
- Some measures have no scale:
  Beauty, Difficulty, etc.
  - Most important aspect of measures:
    they are orderable.
  - Don't care about the actual numbers.
    (An apple can have deliciousness .9 or .1.)

Qualitative Physics

A sub-field of AI which investigates how to reason about physical systems without plunging into detailed equations and numerical simulations.

Substances and objects

The real world can be seen as consisting of
- ♢ primitive objects (particles), and
- ♢ composite objects built from them.

Stuff: a generic name for a significant portion of reality that seems to defy any obvious individuation (division into distinct objects).

- ♢ count nouns: aardvarks, holes, theorems
- ♢ mass nouns: butter, water, energy

$x \in Butter \land PartOf(y, x) \Rightarrow y \in Butter$

$x \in Butter \Rightarrow MeltingPoint(x, Centigrade(30))$

Substances and objects (contd.)

intrinsic properties:
belong to the very substance of object, rather than to the object as a whole.
(e.g. density, boiling point, color, . . .)

extrinsic properties:
not retained under subdivision.
(e.g. weight, length, shape, function, . . .)

- Substance (mass noun):
  a class of objects that includes in its definition only intrinsic properties.

- Count noun:
  a class that includes in its definition any extrinsic properties.

Stuff is the most general substance category, specifying no intrinsic properties.
Thing is the most general discrete object category, specifying no extrinsic properties.

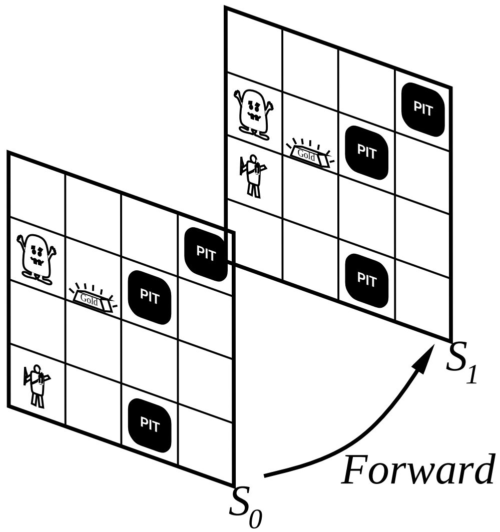
Actions, Situations, and Events

Reasoning about outcome of actions is central to KB-agent.

How can we keep track of location in FOL?
– Remember the multiple copies in PL.

Representing time by situations
(states resulting from the execution of actions).

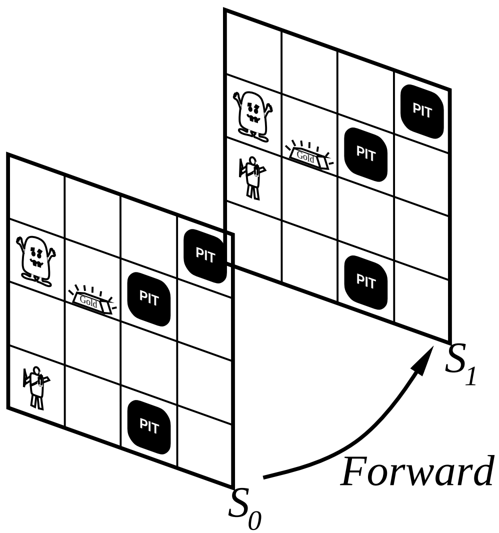
– Situation calculus



Actions, Situations, and Events (contd.)

Situation calculus:

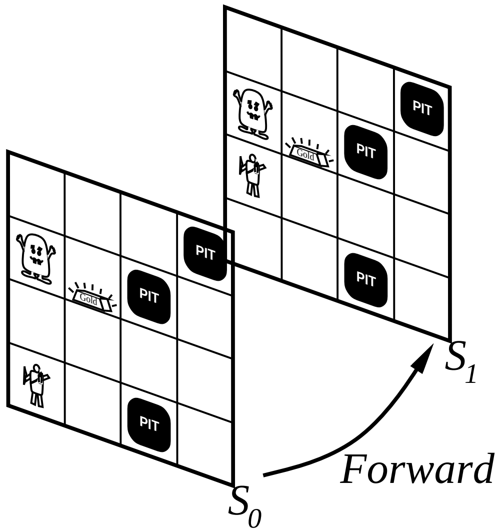
- Actions are logical terms
- Situations are logical terms consiting of
  - The initial situation I (S_0)
  - All situations resulting from the action on I ($= Result(a, s)$)
- Fluents are functions and predicates that vary from one situation to the next.
  E.g. $\neg Holding(G_1, S_0)$
- Atemporal (Eternal) predicates are also allowed
  E.g. $Gold(G_1), LeftLegOf(Wumpus)$



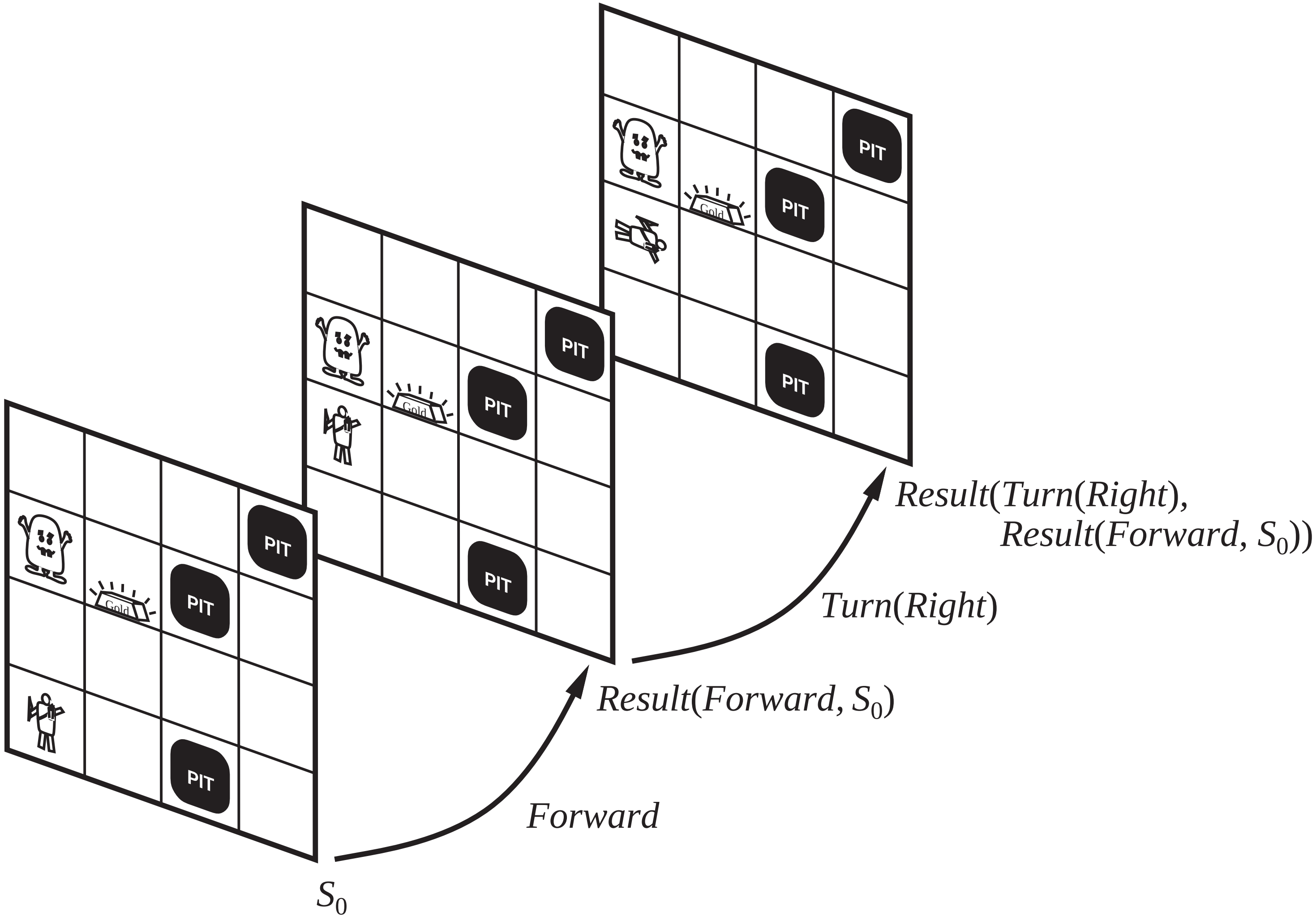
Actions, Situations, and Events (contd.)

Results of action sequences are determined by the individual actions.

- Projection task: an SC agent should be able to deduce the outcome of a sequence of actions.
- Planning task: find a sequence that achieves a desirable effect



Actions, Situations, and Events (contd.)



Describing change

Simples Situation calculus requires two axioms for each action to describe change:

- Possibility axiom: when is it possible to do the action

  $$At(Agent, x, s) \wedge Adjacent(x, y) \Rightarrow Poss(Go(x, y), s)$$

- Effect axiom: describe changes due to action

  $$Poss(Go(x, y), s) \Rightarrow At(Agent, y, Result(Go(x, y), s))$$

The problem is that the effect axioms say what changes, but don't say what stays the same.

- Frame problem: how to represent all things that stay the same?
- A solution, Frame axiom: describe non-changes due to actions

  $$At(o, x, s) \wedge (o \neq Agent) \wedge \neg Holding(o, s) \Rightarrow At(o, x, Result(Go(y, z), s))$$

Representational frame problem

If there are F fluents and A actions then we need $O(AF)$ frame axioms to describe other objects are stationary unless they are held.

If each action has at most E effects (E is typically much less than F), then we should be able to represent what happens with a much smaller KB of size $O(AE)$.

- We write down the effect of each actions

Solution; describe how each fluent changes over time: Successor-state axiom

Successor-state axiom

Action is possible $\Rightarrow$
$(Fluent\ is\ true\ in\ result\ state \Leftrightarrow Action\text{'}s\ effect\ made\ it\ true$
$\qquad \lor\ It\ was\ true\ before\ and\ action\ left\ it\ alone)$

- Note that next state is completely specified by current state.
- Each action effect is mentioned only once.

$$poss(a,s) \Rightarrow (At(Agent,y,Result(a,s)) \Leftrightarrow (a = Go(x,y))$$
$$\lor(At(Agent,y,s) \land a \neq Go(y,z)))$$

Successor-state axioms solve the representational frame problem because the total size of the axioms is $O(AE)$ literals: each of the E effects of the A actions is mentioned exactly one. The literals are spread over F different axioms, so the axioms have average size AE/F.

Other problems

- How to deal with secondary (implicit) effects?
  - If the agent is carrying the gold and the agent moves then the gold moves too.
  - Ramification problem
- How to decide EFFICIENTLY whether fluents hold in the future?
  - Inferential frame problem.
- Extensions:
  - Event calculus (when actions have a duration)
  - Process categories

The ramification problem and implicit effects

Successor-state axioms don't consider implicit effects (such as location changes of things the agent is holding), but can be generalized in order to deal with this case

$$
\begin{aligned}
& poss(a, s) \Rightarrow \\
& At(o, y, Result\ (a, s)) \Leftrightarrow (a = Go(x, y) \wedge (o = Agent \wedge Holding(o, s))) \\
& \qquad \vee (At(o, y, s) \wedge \neg(\exists z\ y \neq z \wedge a = Go(y, z) \wedge \\
& \qquad\qquad (o = Agent \vee Holding(o, s))))
\end{aligned}
$$

Unique names axiom, unique action assumption

Technicality: we need to deal with non-identities in the previous formulas. In general constants in FOL are not necessarily distinct for the previous axioms to work, we need to add:

For all distinct constants (unique names axiom, often simply assumed by provers):

$c_1 \neq c_2$

For all distinct Actions A, B (unique actions assumption):

$$A(x_1, \ldots, x_m) \neq B(y_1, \ldots, y_n)$$
$$A(x_1, \ldots, x_m) = A(y_1, \ldots, y_m) \Leftrightarrow x1 = y1 \wedge \ \ldots \ \wedge \ x_m = y_m$$

With these added, a theorem prover can prove that the proposed plan

$Go([1,1],[1,2],Grab(G1),Go([1,2],[1,1]))$

achieves the goal.

Inferential frame problem

To project the result of a t-step sequence of actions in time $O(Et)$, rather than time $O(Ft)$ or $O(AEt)$.

We already know exactly which action occurs at each step

Let's look at the successor-state-axioms:

$$poss(a, s) \Rightarrow$$

$$F_i(Result(a, s)) \Leftrightarrow \overbrace{(a = A_1 \lor \; a = A_2 \ldots)}^{actions\ making\ F_i\ true}$$

$$\lor F_i(s) \land \underbrace{(a \neq A_3) \land \; (a \neq A_4) \ldots}_{actions\ making\ F_i\ false}$$

Inferential frame problem (contd.)

$$\begin{aligned} poss(a,s) &\Rightarrow \\ F_i(Result(a,s)) &\Leftrightarrow PosEffect(a,F_i) \vee [F_i(s) \wedge \neg NegEffect(a,F_i)] \\ PosEffect(A_1,F_i) \\ PosEffect(A_2,F_i) \\ PosEffect(A_3,F_i) \\ PosEffect(A_4,F_i) \end{aligned}$$

An efficient algorithm now indexes $PosEffect$ and $NegEffect$ predicates per action and computes each successor state by a delta of the previous situation.

Such an algorithm runs in $O(Et)$ for the projection task.

Qualification problem

Ensuring that all necessary conditions for an action's success have been specified.

e.g., G_0 fails if the agent dies en route.

There is no complete solution for qualification problem.

Time and event calculus

Situation calculus is inappropriate for actions which have a duration where we want about time duration, to reason intervals of time, etc.

Rather reason about events which initiate, terminate fluents at certain points in time: event calculus

$Initiates(e, f, t)$:

the occurrence of event e at time t causes fluent f to become true.

$Terminates(e, f, t)$:

the occurrence of event e at time t causes fluent f ceases to be true.

$Happens(e, t)$:

event e happens at time t

$Clipped(f, t, t2)$:

f is terminated by some event sometime between t and t_2

$T(f, t)$

Event calculus axiom

$$T(f, t_2) \Leftrightarrow \exists\, e, t Happens(e, t) \land\ Initiates(e, f, t) \land\ (t < t_2) \\ \land \neg Clipped(f, t, t_2)$$

$$Clipped(f, t, t_2) \Leftrightarrow \exists\, e, t_1 Happens(e, t_1) \land\ Terminates(e, f, t_1) \\ \land (t < t_1) \land\ (t_1 < t_2)$$

Generalized events

Generalized event – a piece of the space-time universe

$SubEvent(BattleOfBritain, WorldWarII)$
$SubEvent(WorldWarII, TwentiethCentury)$

$Period(e)$ denotes the smallest interval enclosing an event e.
Intervals – chunks of space-time that include all space between two points

$Duration(e)$ denotes the length of time of an interval, e.

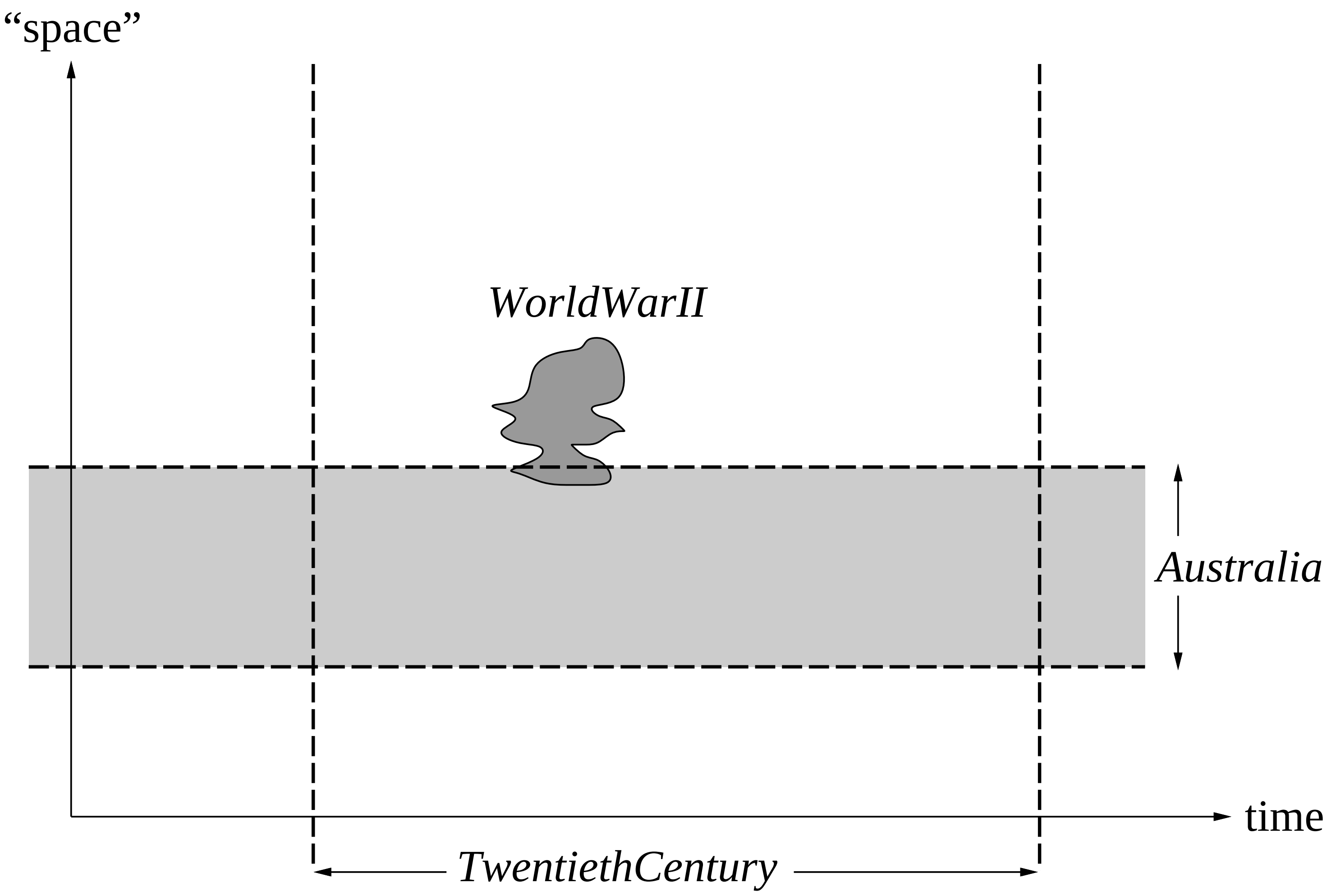
$Location(e)$ denotes the smallest place enclosing an event e.

$In(e1, e2)$ denotes $PartOf$ relationship of the spatial projection of an event.

$Duration(Period(WorldWarII)) > Years(5)$
$In(Sydney, Australia)$

$\exists\, w\; w \in CivilWars \land SubEvent(w, 1640s) \land In(Location(w), England)$

Generalized events (contd.)



Category of events

Actions like $Go([1,1],[1,2])$ denote a category of events and not single events; $Goto(y), GoFrom(x)$

– more general event categories.

Shortcuts for event categories:

$E(c,i) \Leftrightarrow \exists\ e\ e \in \wedge SubEvent(e,i)$
$E(Fly(Shankar, NewYork, NewDelhi), Yesterday)$

Processes

Discrete events vs. liquid events (processes) categories

Any subinterval of a process is a member of the same process category
$E(Flying(Shankar), Yesterday)$

One can say that a process is going on throughout an interval and not within an interval
$T(c,i) \Leftrightarrow E(c,i) \wedge$ "$the\ Event\ occurred\ throughout\ the\ whole\ interval\ i$"

$T(Working(Stuart), TodayLunchHour)$

Temporal substances (liquid) vs. Spatial substances (non-liquid)

Fluent calculus

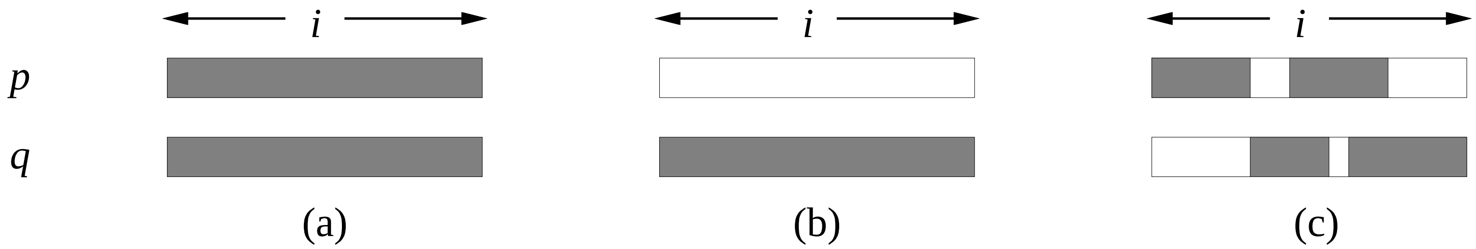
Fluent calculus reifies combinations of fluents, not just individual fluents.

$Both(e_1, e_2)$: the event of two things happening at once ($e_1 \circ e_2$)

e.g., "Someone walked and chewed gum at the same time":
$\exists p, i \ (p \in People) \wedge T(Walk(p) \circ ChewGum(p), i)$

"$\circ$" function is commutative and associative.

Fluent calculus: complex events



(a) $T(Both(p,q),i)$ or $T(p \circ q,i)$

(b) $T(OneOf(p,q),i)$

(c) $T(Either(p,q),i)$

Intervals

$Partition(\{Moments, ExtendedIntervals\}, Intervals)$

$i \in Moments \Leftrightarrow Duration(i) = Seconds(0)$

$Start(i)$, $End(i)$ denote the start and end moments of an interval i;

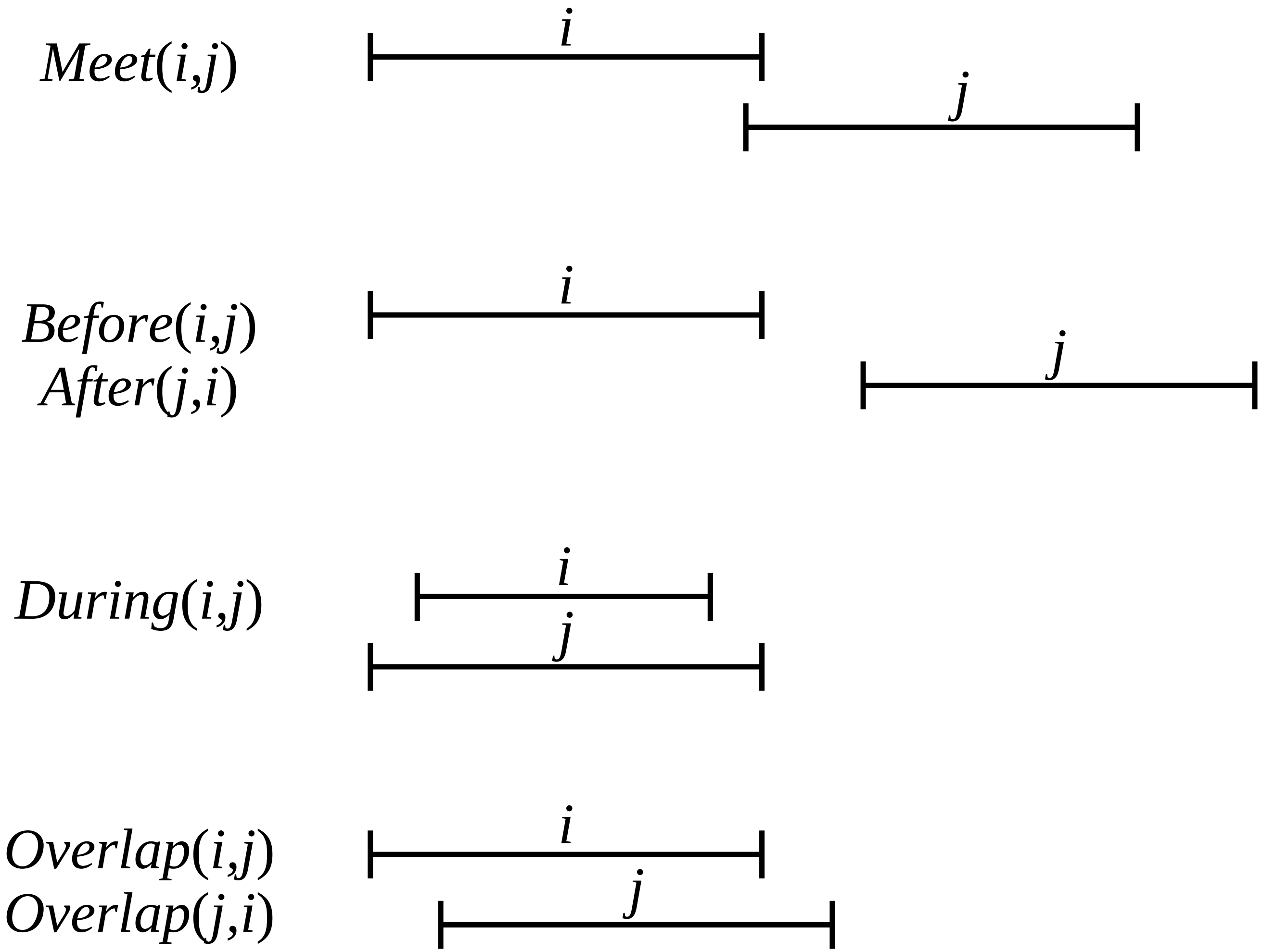
$Interval(i) \Rightarrow Duration(i) = (Time(End(i)) - Time(Start(i)))$

Time scale: such as $Seconds(s)$

$Time(Start(AD2001)) = Date(0, 0, 0, 1, Jan, 2001)$
$Date(0, 20, 21, 24, 1, 1995) = Seconds(3000000000)$

using these constructs one can define the functions $Meet(i, j)$, $Before(i, j)$, $Overlap(i, j)$, ...

Time intervals



Fluents and objects

Physical objects can be viewed as generalized event:
A physical object is a chunk of space-time.

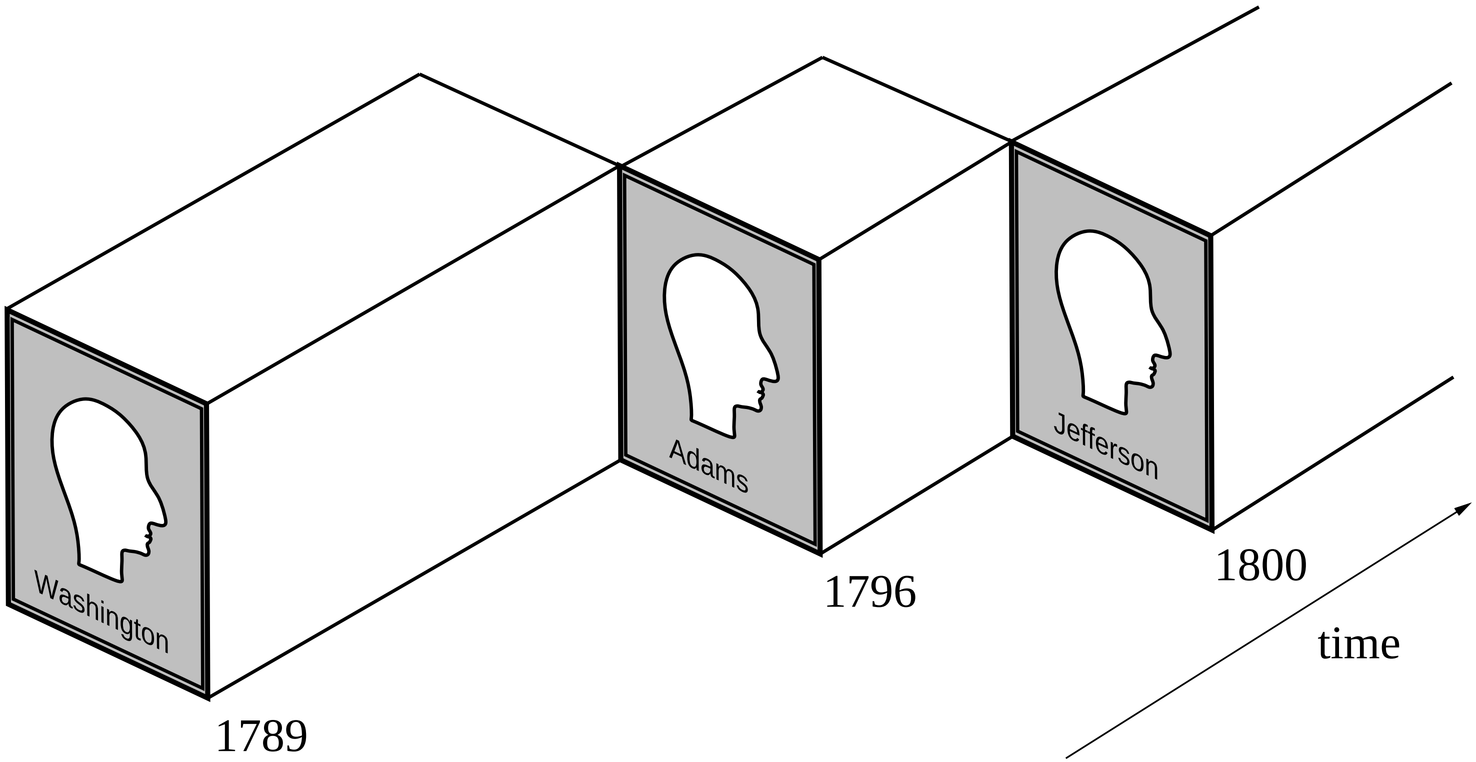
e.g., USA can be thought of as an event.

We can describe changing properties of USA using state fluents:
$E(Ppopulation(USA, 271000000), AD1999)$

$President(USA)$ denotes a single object that consists of different people at different times:
$T(President(USA) = GeorgeWashington, AD1790)$

$=$ is a function symbol,
logical $=$ is not something that can change over time.

Fluents and objects (contd.)



Mental Events and Mental Objects

So far, KB agents can have beliefs and deduce new beliefs

What about knowledge about beliefs?
What about knowledge about the inference process?

In multi-agent domains, it becomes important for an agent to reason about mental states of other agents.

Requires a model of the

- ♢ mental objects in someones head (KB) and
- ♢ the processes that manipulate these objects.

Relationships between agents and mental objects (propositional attitudes): believes, knows, wants, . . .

$Believes(Lois, Flies(Superman))$ with $Flies(Superman)$ being a function . . .
a candidate for a mental object (reification).

Agent can now reason about the beliefs of agents.

A formal theory of beliefs

Relationships between agents and mental objects (propositional attitudes): believes, knows, wants, ...

$Believes(Lois, Flies(Superman))$

Reification: What is $Flies(Superman)$ here? A term or a proposition? We need to be able to turn propositions/sentences into objects and vice versa!

$(Superman = Clark) \models$
$(Believes(Lois, Flies(Superman)) \Leftrightarrow Believes(Lois, Flies(Clark)))$

Referential transparency – being able to substitute a term freely for an equal term (not desired for reasoning about believes)
– $Believes$ relations with referentially **opaque** arguments for beliefs (one cannot substitute an equal term for the second argument without changing the meaning.)

Circumventing referential transparency

Two alternatives

- ♢ Special logics such as modal logics
- ♢ Syntactic theory of mental objects as strings

Unique string axioms, special function symbols

$``Superman" \neq ``Clark"$

Syntax, semantics, and proof theory for the string representation language:

- ♢ Den maps strings to the objects they denote
- ♢ $Name$ maps objects to a name string
- ♢ $Concat$ concatenates strings

$Den(``Clark") = ManOfSteel \wedge Den(``Superman") = ManOfSteel$
$Name(ManOfSteel) = ``X11"$

Modeling reification

We reify sentences as strings and emulate inference rules, e.g. Modus ponens

$LogicalAgent(a) \wedge Believes(a,p) \wedge Believes(a, Concat(p, \text{``} \Rightarrow \text{``}, q)) \Rightarrow Believes(a,q)$

Write short:
$LogicalAgent(a) \wedge Believes(a,p) \wedge Believes(a, \text{``}\underline{p} \Rightarrow \underline{q}\text{``}) \Rightarrow Believes(a,q)$

Logical omniscience:
$LogicalAgent(a) \wedge Believes(a,p) \Rightarrow Believes(a, \text{``}Believes(\underline{Name(a)}, \underline{p})\text{``})$

Knowledge and belief

Knowledge is justified true belief

$Knows(a, p)$: agent a knows that proposition p is true.

KnowsWhether:

$KnowsWhether(a, p) \Leftrightarrow Knows(a, p) \lor Knows(a, \text{``}\neg p\text{``})$

$Knows(a, s) \Rightarrow Believes(a, s)$

KnowsWhat:

$KnowsWhat(a, \text{``}PhoneNumber(b)\text{``}) \Leftrightarrow$
$\quad \exists x\ Knows(a, \text{``}x = PhoneNumber(b)\text{``}) \land x \in DigitStrings$

Knowledge, time, and action

Beliefs of an agents (or other agents) change over time.

Modeling knowledge effects
– Combine reasoning about events, action, time and knowledge:

$T(Believes(Lois, \text{``}Flies(Superman)\text{``}), Today)$
$T(Believes(Lois, \text{``}T(Flies(Superman), Yesterday)\text{``}, Today))$

Making plan involving beliefs

Using the machinery of event calculus

Actions can have:

- ♢ Knowledge preconditions
- ♢ Knowledge effects

$$Initiates(Lookup(a, \text{``}PhoneNumber(\underline{b})\text{``}),\ KnowsWhat(a, \text{``}PhoneNumber(\underline{b})\text{``}, DigitStrings), t)$$

Plans to gather and use information are often represented using a shorthand notation called runtime variable ($\underline{n}$)

$$[Lookup(Agent, \text{``}PhoneNumber(Bob)\text{``}, \underline{n}), Dial(\underline{n})]$$

The Internet shopping world

A Knowledge Engineering example

An agent that helps a buyer to find product offers on the internet.

IN = product description ($precise$ or $\neg precise$)

OUT = list of webpages that offer the product for sale.

Environment = WWW

Percepts = web pages (character strings)

Extracting useful information required.

The Internet shopping world (contd.)

Find relevant product offers

$$RelevantOffer(page, url, query) \Leftrightarrow$$
$$Relevant(page, url, query) \wedge Offer(page)$$

- Write axioms to define $Offer(x)$
- Find relevant pages: $Relevant(x, y, z)$?
  - Start from an initial set of stores.
  - What is a relevant category?
  - What are relevant connected pages?
- Require rich category vocabulary.
  - Synonymy and ambiguity
- How to retrieve pages: $GetPage(url)$?
  - Procedural attachment

Compare offers (information extraction).

Reasoning Systems for Categories

How to organize and reason with categories?

◇ Semantic networks
– Visualize knowledge-base
– Efficient algorithms for category membership inference

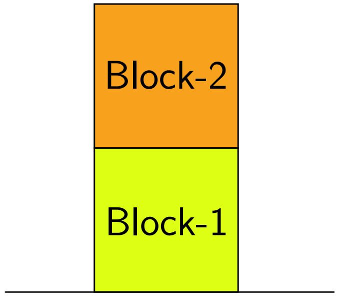
◇ Description logics
– Formal language for constructing and combining category definitions
– Efficient algorithms to decide subset and superset relationships between categories.

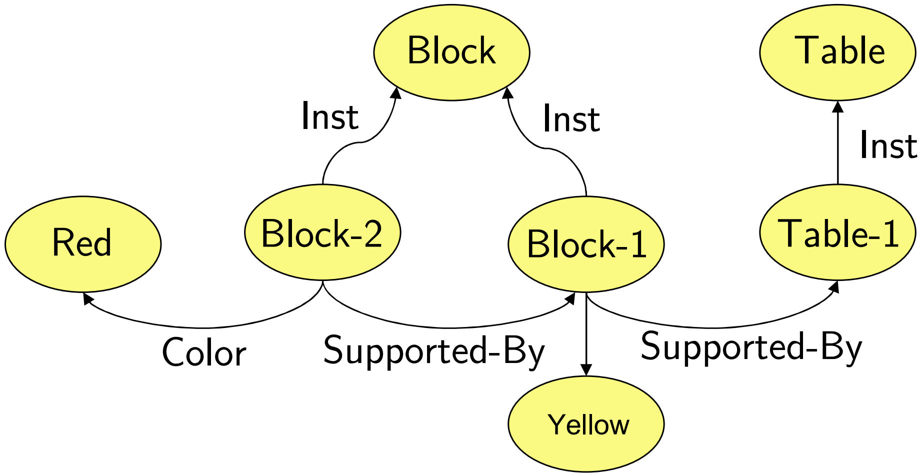
Representation of a Scene

◇ as set of logic expressions

```
(inst block-2 block)
(color block-2 red)
(supported-by block-2 block-1)
(inst block-1 block)
(color block-1 yellow)
(supported-by block-1 table-1)
(inst table-1 table)
```

◇ as Semantic Net





Semantic Networks

- Logic vs. semantic networks
- Many variations
- All represent individual objects, categories of objects and relationships among objects.
- Allows for inheritance reasoning
  - Female persons inherit all properties from person.
  - Cfr. Object-Orientd programming.
- Inference of inverse links
  - $SisterOf$ vs. $HasSister$

Alternative Notations

Semantic Nets (a.k.a. associative nets) and FOL sentences represent same information in different formats:

- Nodes correspond to terms
  marked out directed edges correspond to predicates
- they are alternative notations for the same content,
  not in principle different representations!

What differs?

Missing existential quantifier Functions (extensions exist) Semantic nets additionally provide pointers (and sometimes back pointers) which allow easy and high performance information access (e.g., to instances): INDEXING

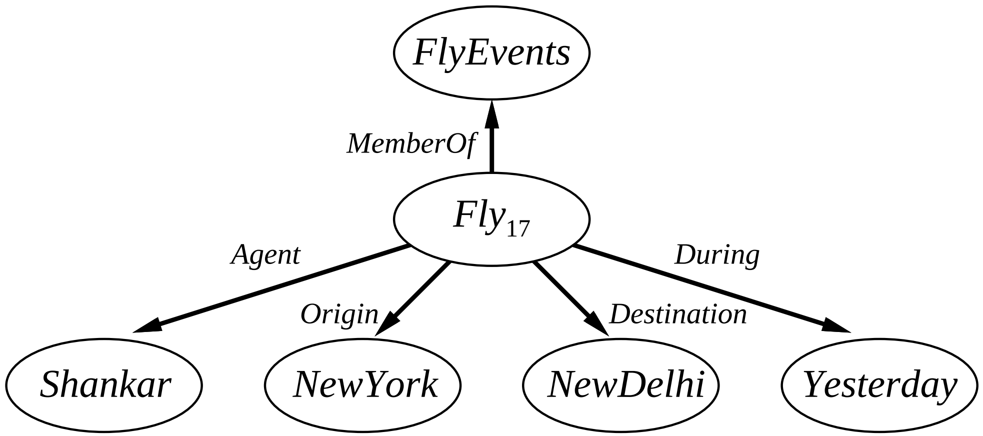
ISA-Hierarchy and Inheritance

- Key concept in the tradition of semantic nets
- Instances inherit properties which we attribute to sets of individuals (classes).
- This can be propagates along the complete isa hierarchy
  - Inheritance of properties
  - Reason: Knowledge representation economy
- Search along isa- and inst-links to access information not directly associated (using inheritance)
  - inst $\in$ member of
  - isa $\subseteq$ subset of

Semantic networks (contd.)

Drawbacks

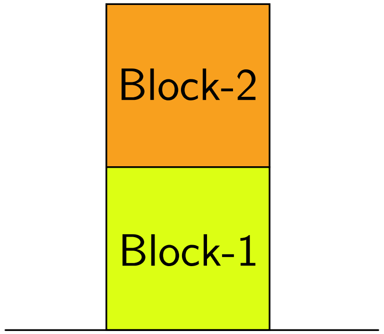
- Links can only assert binary relations
- Can be resolved by reification of the proposition as an event



Representation of default values
– Enforced by the inheritance mechanism.

Representation of a Scene (contd.)

◇ as Frames
(slot-and-filler-Notation)



"Alternative Notations"

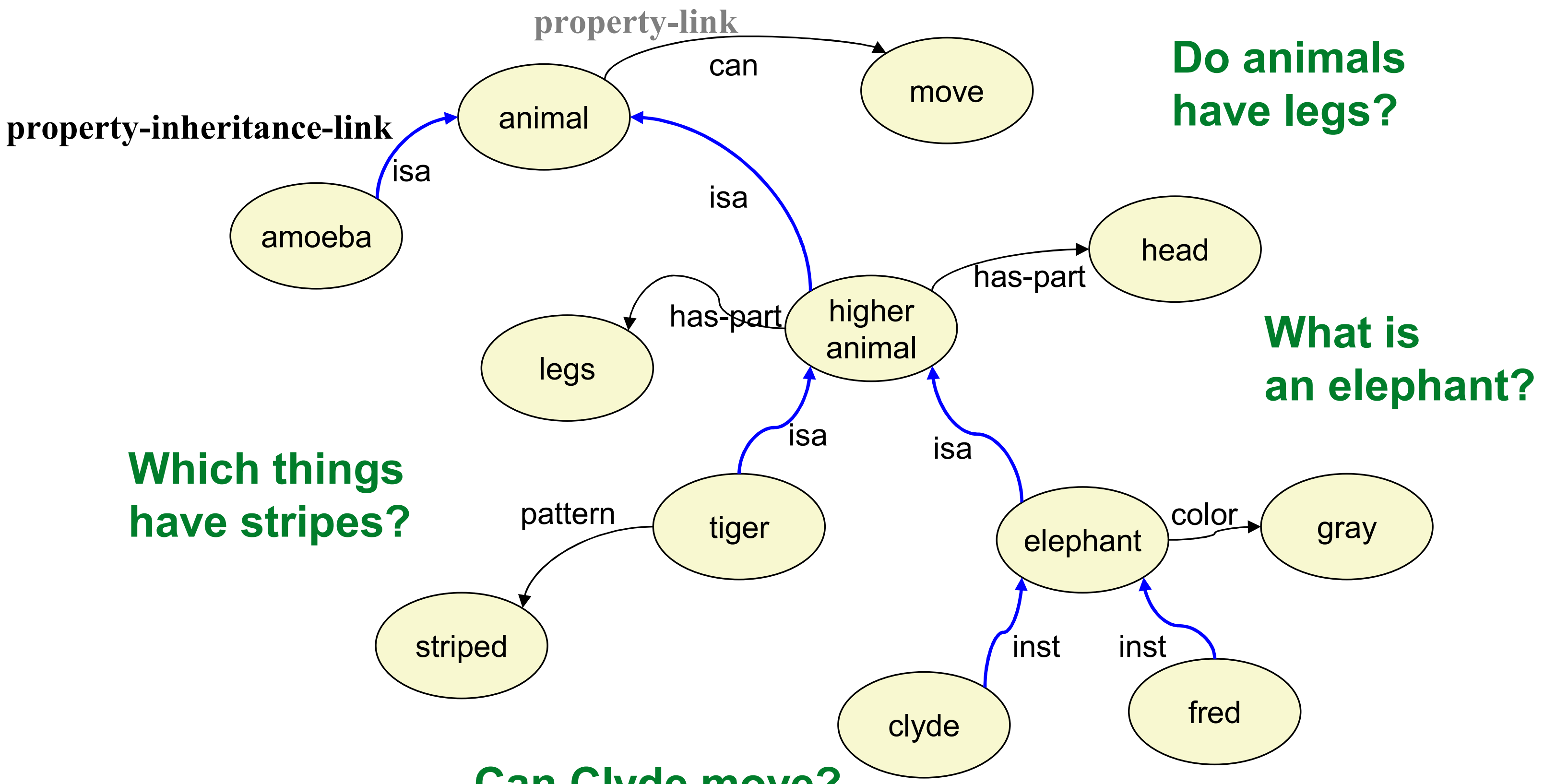
```
(inst block-2 block)
(color block-2 red)
(supported-by block-2 block-1)
(inst block-1 block)
(color block-1 yellow)
(supported-by block-1 table-1)
(inst table-1 table)
```

Frame	Attribute (slots)	Werte (fillers)
block-2 :	inst :	block
	color :	red
	supported-by :	block-1
	...	...

Frame	Attribute (slots)	Werte (fillers)
block-1 :	inst :	block
	color :	yellow
	supported-by :	table-1
	...	...

Frame	Attribute (slots)	Werte (fillers)
table-1 :	inst :	table
	color :	
	supported-by :	
	...	...

Example of an ISA-Hierarchy



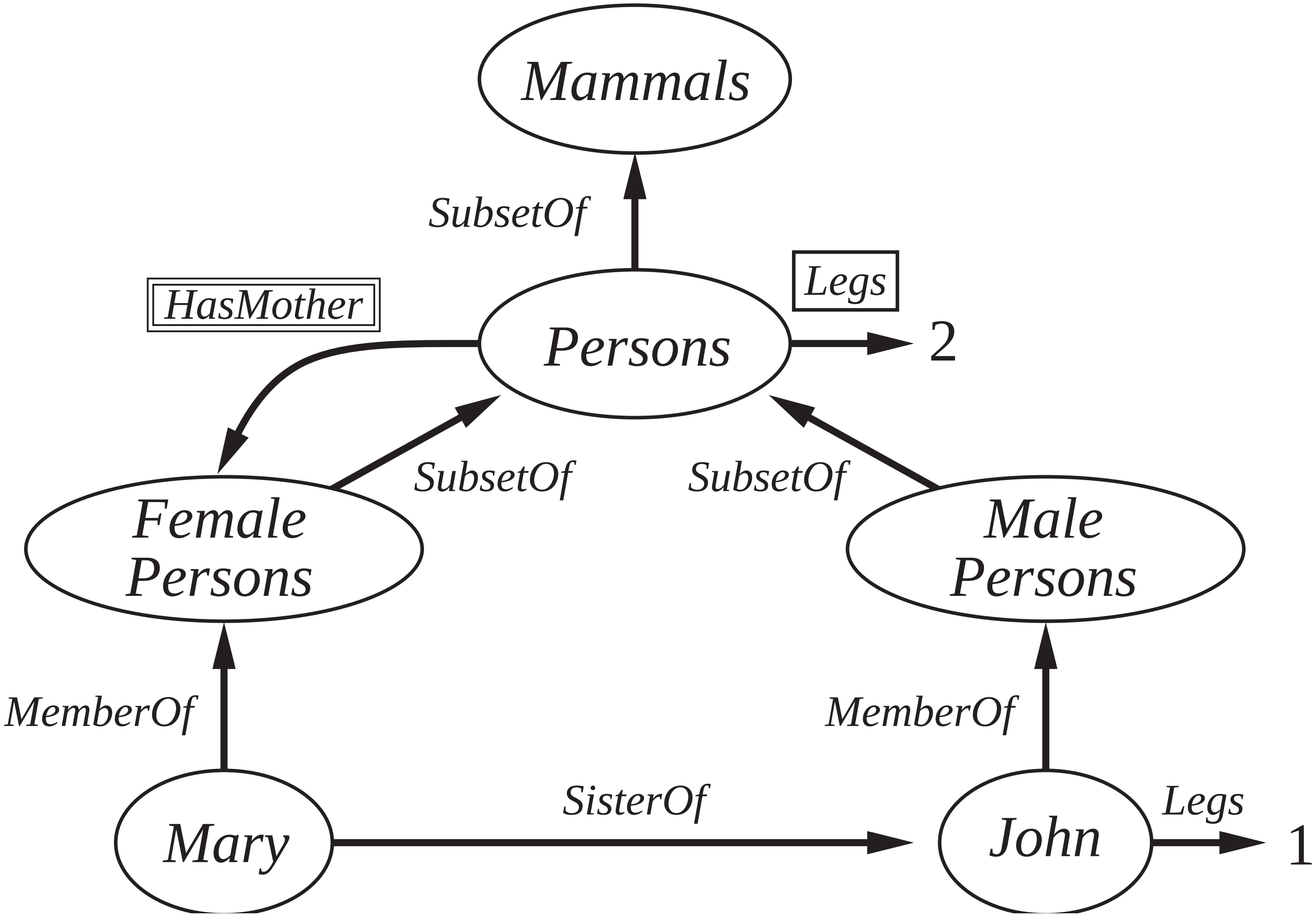
Do animals have legs?

What is an elephant?

Which things have stripes?

Can Clyde move?

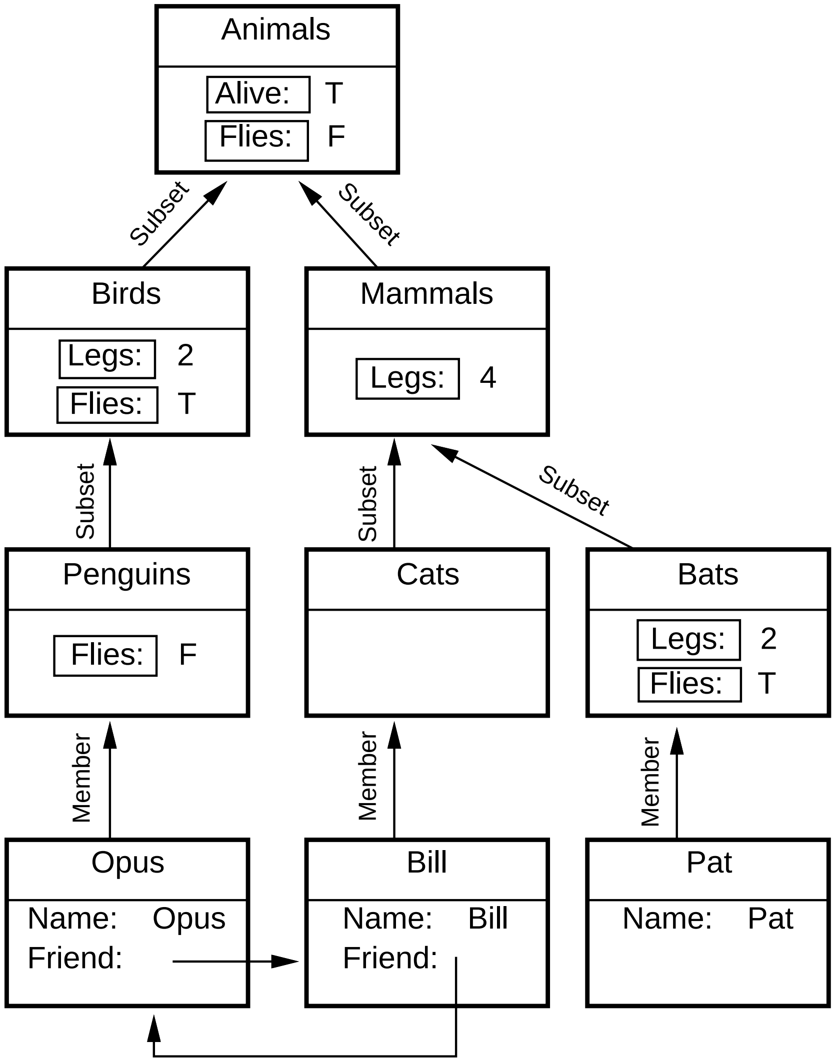
Semantic network example



Semantic network notation

Link Type	Semantics	Example
$A \xrightarrow{Subset} B$	$A \subset B$	$Cats \subset Mammals$
$A \xrightarrow{Member} B$	$A \in B$	$Bill \in Cats$
$A \xrightarrow{R} B$	$R(A,B)$	$Bill \xrightarrow{Age} 12$
$A \xrightarrow{\boxed{R}} B$	$\forall x \;\; x \in A \;\Rightarrow\; R(x,B)$	$Birds \xrightarrow{\boxed{Legs}} 2$
$A \xrightarrow{\boxed{\boxed{R}}} B$	$\forall x \;\; \exists y \;\; x \in A \;\Rightarrow\; y \in B \land R(x,y)$	$Birds \xrightarrow{\boxed{\boxed{Parent}}} Birds$

Frame-based KB vs. FOL



(a) A frame−based knowledge base

Rel(Alive,Animals,T)
Rel(Flies,Animals,F)

Birds $\subset$ Animals
Mammals $\subset$ Animals

Rel(Flies,Birds,T)
Rel(Legs,Birds,2)
Rel(Legs,Mammals,4)

Penguins $\subset$ Birds
Cats $\subset$ Mammals
Bats $\subset$ Mammals

Rel(Flies,Penguins,F)
Rel(Legs,Bats,2)
Rel(Flies,Bats,T)

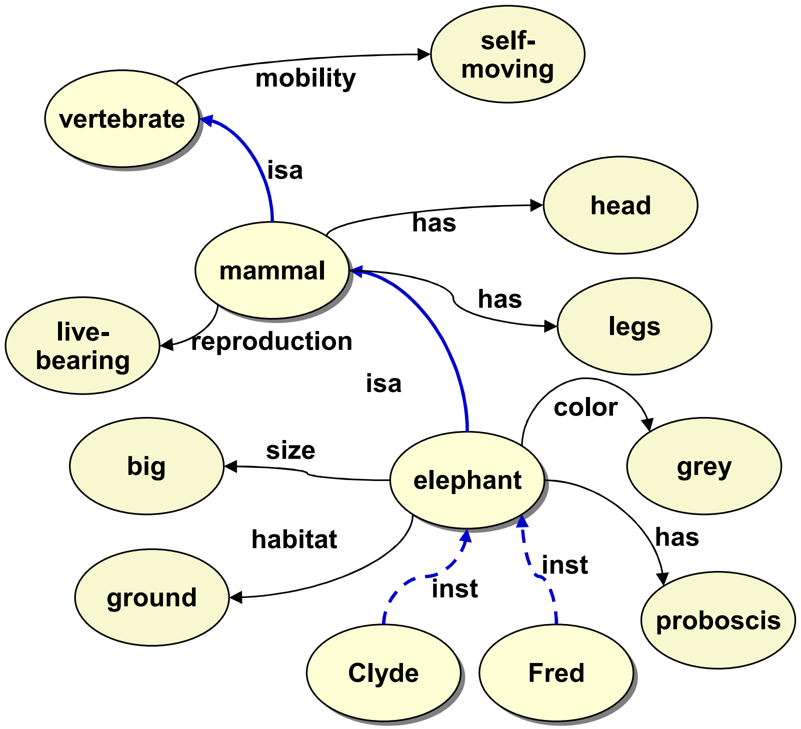
Opus $\in$ Penguins
Bill $\in$ Cats
Pat $\in$ Bats

Name(Opus,"Opus")
Name(Bill,"Bill")
Friend(Opus,Bill)
Friend(Bill,Opus)
Name(Pat,"Pat")

(b) Translation into first−order logic

Inheritance in Semantic Nets and Frames

(slightly different modelling than before)



object	property	value
mammal :	isa :	vertebrate
	reproduction :	livebearing
	has :	head, legs
	...	...

object	property	value
elephant :	isa :	mammal
	color :	grey
	has :	proboscis
	size :	big
	habitat :	Boden

object	property	value
Clyde :	inst :	elephant
	color :	grey
	has :	proboscis
	size :	big
	habitat :	ground

Origin of Frames

Cognitive theory about:

- Recognition of stereotype objects (e.g., living room)
- Action for stereotype events (e.g., children's birthday party)
- Replying to questions about stereotype or specific objects.

Marvin Minsky (1975):
A framework for representing knowledge.
In P.H. Winston (ed.): The Psychology of Computer Vision. New York: McGraw-Hill.

Description Logic

- Are designed to describe definitions and properties about categories
  - A formalization of semantic networks
- Principal inference task is
  - Subsumption: checking if one category is the subset of another by comparing their definitions
  - Classification: checking whether an object belongs to a category.
  - Consistency: whether the category membership criteria are logically satisfiable.

e.g., CLASSIC language.

Reasoning with Default Information

"The following courses are offered: CS101, CS102, CS106, EE101"

How many courses are offered?

- Four (db)
  - Assume that this information is complete (not asserted ground atomic sentences are false)
    = CLOSED WORLD ASSUMPTION (CWA)
  - Assume that distinct names refer to distinct objects
    = UNIQUE NAMES ASSUMPTION (UNA)
- Between one and infinity (logic) = OPEN WORLD ASSUMPTION (OWA)
  - Does not make these assumptions (CWA, UNA)
  - Requires completion.

Truth Maintenance Systems (TMS)

Many of the inferences have default status rather than being absolutely certain

– Inferred facts can be wrong and need to be retracted
  = BELIEF REVISION.

– Assume KB contains sentence P and we want to execute $\textsc{Tell}(KB, \neg P)$
  - To avoid contradiction: $\textsc{Retract}(KB, P)$
  - But what about sentences inferred from P?

Truth maintenance systems are designed to handle these complications.

Summary