

Why Knowledge Representation?

An Example

An Argument for Logic

Why Knowledge Representation? An Example

- Consider the task of understanding a simple story.
- How do we test understanding?
- Not easy, but understanding at least entails some ability to answer simple questions about the story.

Example.

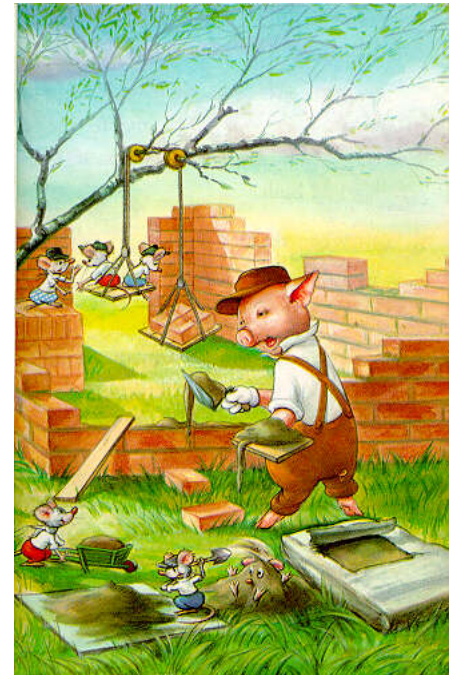
- Three little pigs: Mother sends them to “seek their fortune”



■ 1st pig builds a house of straw



■ 2nd pig builds a house of sticks



■ 3rd pig builds a house of bricks

Example.

- Three little pigs



■ Wolf blows down the straw house, and eats the pig!



■ Wolf blows down the sticks house, and eats the pig!



■ Wolf cannot huff and poof the brick house!

Example

- Why couldn't the wolf blow down the house made of bricks?
- What background knowledge are we applying to come to that conclusion?
 - Brick structures are stronger than straw and stick structures.
 - Objects, like the wolf, have physical limitations. The wolf can only blow so hard.

Why Knowledge Representation?

- Large amounts of knowledge are used to understand the world around us, and to communicate with others.
- We also have to be able to reason with that knowledge
 - Our knowledge won't be about the blowing ability of wolfs in particular, it is about physical limits of objects in general.
 - We have to employ reasoning to make conclusions about the wolf.
 - More generally, reasoning provides an exponential or more compression in the knowledge we need to store. I.e., without reasoning we would have to store a infeasible amount of information: e.g., Elephants can't fit into teacups.

Logical Representations

- AI typically employs logical representations of knowledge.
- Logical representations useful for a number of reasons:

Logical Representations

- They are mathematically precise, thus we can analyze their limitations, their properties, the complexity of inference etc.
- They are formal languages, thus computer programs can manipulate sentences in the language.
- They come with both a formal **syntax** and a formal **semantics**.
- Typically, have well developed **proof theories**: formal procedures for reasoning (achieved by manipulating sentences).

The Knowledge Base

- The Knowledge Base is a set of sentences.
 - Syntactically well-formed
 - Semantically meaningful
- A user can perform two actions to the KB:
 - ▶ Tell the KB a new fact
 - ▶ Ask the KB a question

Syntax of Sentences

English acceptable an one is sentence This.

vs.

This English sentence is an acceptable one.

$\vee P \neg \wedge Q R$

vs.

$P \vee \neg Q \wedge R$

Semantics of Sentences

This hungry classroom is a jobless moon.

- Why is this syntactically correct sentence not meaningful?

$$P \vee (\neg Q \wedge R)$$

- Represents a world where either P is true, or Q is not true and R is true.

Entailments

$$\alpha \models \beta$$

- read as “ α entails β ”, or “ β follows logically from α ”
- meaning that in any world in which α is true, β is true as well.

For example

$$(P \wedge Q) \models (P \vee R)$$

Syntactical Derivation

$$\alpha \vdash \beta$$

- read as “ α derives β ”
- meaning “from sentence α , following the syntactical derivation rules, we can obtain sentence β ”

For example

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

3.

Expressing Knowledge

Knowledge engineering

KR is first and foremost about knowledge

meaning and entailment

find individuals and properties, then encode facts sufficient for entailments

Before implementing, need to understand clearly

- what is to be computed?
- why and where inference is necessary?

Example domain: soap-opera world

people, places, companies, marriages, divorces, hanky-panky, deaths, kidnappings, crimes, ...

Task: KB with appropriate entailments

- what vocabulary?
- what facts to represent?

Vocabulary

Domain-dependent predicates and functions

main question: what are the individuals?

here: people, places, companies, ...

named individuals

john, sleezyTown, faultyInsuranceCorp, fic, johnQsmith, ...

basic types

Person, Place, Man, Woman, ...

attributes

Rich, Beautiful, Unscrupulous, ...

relationships

LivesAt, MarriedTo, DaughterOf, HadAnAffairWith, Blackmails, ...

functions

fatherOf, ceoOf, bestFriendOf, ...

Basic facts

Usually atomic sentences and negations

type facts

Man(john),

Woman(jane),

Company(faultyInsuranceCorp)

property facts

Rich(john),

\neg HappilyMarried(jim),

WorksFor(jim,fic)

equality facts

john = ceoOf(fic),

fic = faultyInsuranceCorp,

bestFriendOf(jim) = john

Like a simple database (can store in a table)

Complex facts

Universal abbreviations

$\forall y[\text{Woman}(y) \wedge y \neq \text{jane} \supset \text{Loves}(y, \text{john})]$

$\forall y[\text{Rich}(y) \wedge \text{Man}(y) \supset \text{Loves}(y, \text{jane})]$

$\forall x \forall y[\text{Loves}(x, y) \supset \neg \text{Blackmails}(x, y)]$

possible to express
without quantifiers

Incomplete knowledge

$\text{Loves}(\text{jane}, \text{john}) \vee \text{Loves}(\text{jane}, \text{jim})$

which?

$\exists x[\text{Adult}(x) \wedge \text{Blackmails}(x, \text{john})]$

who?

cannot write down
a more complete
version

Closure axioms

$\forall x[\text{Person}(x) \supset x = \text{jane} \vee x = \text{john} \vee x = \text{jim} \dots]$

$\forall x \forall y[\text{MarriedTo}(x, y) \supset \dots]$

$\forall x[x = \text{fic} \vee x = \text{jane} \vee x = \text{john} \vee x = \text{jim} \dots]$

limit the domain
of discourse

also useful to have $\text{jane} \neq \text{john} \dots$

Terminological facts

General relationships among predicates. For example:

disjoint $\forall x[\text{Man}(x) \supset \neg \text{Woman}(x)]$

subtype $\forall x[\text{Senator}(x) \supset \text{Legislator}(x)]$

exhaustive $\forall x[\text{Adult}(x) \supset \text{Man}(x) \vee \text{Woman}(x)]$

symmetry $\forall x \forall y [\text{MarriedTo}(x,y) \supset \text{MarriedTo}(y,x)]$

inverse $\forall x \forall y [\text{ChildOf}(x,y) \supset \text{ParentOf}(y,x)]$

type restriction $\forall x \forall y [\text{MarriedTo}(x,y) \supset$
 $\text{Person}(x) \wedge \text{Person}(y) \wedge \text{OppSex}(x,y)]$

sometimes

Usually universally quantified conditionals or biconditionals

Entailments: 1

Is there a company whose CEO loves Jane?

$\exists x [\text{Company}(x) \wedge \text{Loves}(\text{ceoOf}(x), \text{jane})]$??

Suppose $\mathcal{S} \models \text{KB}$.

Then $\mathcal{S} \models \text{Rich}(\text{john}), \text{Man}(\text{john})$,

and $\mathcal{S} \models \forall y [\text{Rich}(y) \wedge \text{Man}(y) \supset \text{Loves}(y, \text{jane})]$

so $\mathcal{S} \models \text{Loves}(\text{john}, \text{jane})$.

Also $\mathcal{S} \models \text{john} = \text{ceoOf}(\text{fic})$,

so $\mathcal{S} \models \text{Loves}(\text{ceoOf}(\text{fic}), \text{jane})$.

Finally $\mathcal{S} \models \text{Company}(\text{faultyInsuranceCorp})$,

and $\mathcal{S} \models \text{fic} = \text{faultyInsuranceCorp}$,

so $\mathcal{S} \models \text{Company}(\text{fic})$.

Thus, $\mathcal{S} \models \text{Company}(\text{fic}) \wedge \text{Loves}(\text{ceoOf}(\text{fic}), \text{jane})$,

and so

$\mathcal{S} \models \exists x [\text{Company}(x) \wedge \text{Loves}(\text{ceoOf}(x), \text{jane})]$.

Can extract identity of company from this proof

Entailments: 2

If no man is blackmailing John, then is he being blackmailed by somebody he loves?

$$\forall x[\text{Man}(x) \supset \neg \text{Blackmails}(x, \text{john})] \supset \\ \exists y[\text{Loves}(\text{john}, y) \wedge \text{Blackmails}(y, \text{john})] \quad ??$$

Note: $\text{KB} \models (\alpha \supset \beta)$ iff $\text{KB} \cup \{\alpha\} \models \beta$

Let: $\mathcal{S} \models \text{KB} \cup \{\forall x[\text{Man}(x) \supset \neg \text{Blackmails}(x, \text{john})]\}$

Show: $\mathcal{S} \models \exists y[\text{Loves}(\text{john}, y) \wedge \text{Blackmails}(y, \text{john})]$

Have: $\exists x[\text{Adult}(x) \wedge \text{Blackmails}(x, \text{john})]$ and $\forall x[\text{Adult}(x) \supset \text{Man}(x) \vee \text{Woman}(x)]$
so $\exists x[\text{Woman}(x) \wedge \text{Blackmails}(x, \text{john})]$.

Then: $\forall y[\text{Rich}(y) \wedge \text{Man}(y) \supset \text{Loves}(y, \text{jane})]$ and $\text{Rich}(\text{john}) \wedge \text{Man}(\text{john})$
so $\text{Loves}(\text{john}, \text{jane})!$

But: $\forall y[\text{Woman}(y) \wedge y \neq \text{jane} \supset \text{Loves}(y, \text{john})]$
and $\forall x \forall y[\text{Loves}(x, y) \supset \neg \text{Blackmails}(x, y)]$
so $\forall y[\text{Woman}(y) \wedge y \neq \text{jane} \supset \neg \text{Blackmails}(y, \text{john})]$ and $\text{Blackmails}(\text{jane}, \text{john})!!$

Finally: $\text{Loves}(\text{john}, \text{jane}) \wedge \text{Blackmails}(\text{jane}, \text{john})$
so: $\exists y[\text{Loves}(\text{john}, y) \wedge \text{Blackmails}(y, \text{john})]$

What individuals?

Sometimes useful to reduce n-ary predicates to 1-place predicates and 1-place functions

- involves reifying properties: new individuals
- typical of description logics / frame languages (later)

Flexibility in terms of arity:

Purchases(john,sears,bike) or
Purchases(john,sears,bike,feb14) or
Purchases(john,sears,bike,feb14,\$100)

Instead: introduce purchase objects

$\text{Purchase}(p) \wedge \text{agent}(p)=\text{john} \wedge \text{obj}(p)=\text{bike} \wedge \text{source}(p)=\text{sears} \wedge \dots$
allows purchase to be described at various levels of detail

Complex relationships: MarriedTo(x,y) VS. ReMarriedTo(x,y) VS. ...

Instead define marital status in terms of existence of marriage and divorce events.

$\text{Marriage}(m) \wedge \text{husband}(m)=x \wedge \text{wife}(m)=y \wedge \text{date}(m)=\dots \wedge \dots$

Abstract individuals

Also need individuals for numbers, dates, times, addresses, etc.

objects about which we ask wh-questions

Quantities as individuals

$\text{age}(\text{suzy}) = 14$

$\text{age-in-years}(\text{suzy}) = 14$

$\text{age-in-months}(\text{suzy}) = 168$

perhaps better to have an object for “the age of Suzy”, whose value in years is 14

$\text{years}(\text{age}(\text{suzy})) = 14$

$\text{months}(x) = 12 * \text{years}(x)$

$\text{centimeters}(x) = 100 * \text{meters}(x)$

Similarly with locations and times

instead of

$\text{time}(m) = \text{"Jan 5 2006 4:47:03EST"}$

can use

$\text{time}(m) = t \wedge \text{year}(t) = 2006 \wedge \dots$

Other sorts of facts

Statistical / probabilistic facts

- Half of the companies are located on the East Side.
- Most of the employees are restless.
- Almost none of the employees are completely trustworthy,

Default / prototypical facts

- Company presidents typically have secretaries intercepting their phone calls.
- Cars have four wheels.
- Companies generally do not allow employees that work together to be married.

Intentional facts

- John believes that Henry is trying to blackmail him.
- Jane does not want Jim to think that she loves John.

Others ...