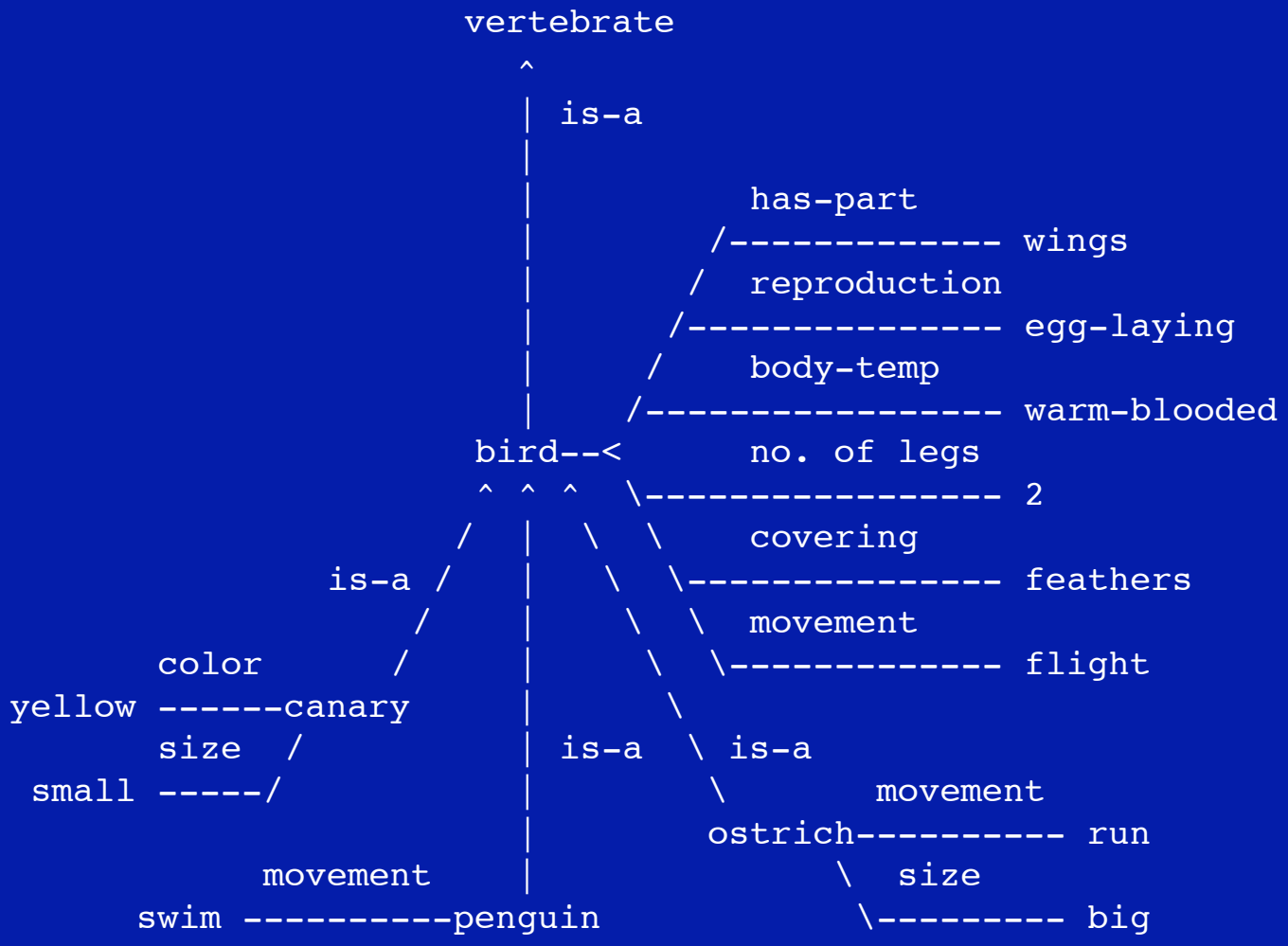
CPSC 322
Introduction to Artificial Intelligence

November 5, 2004

Primitive relations vs. derived relations

Does a bird fly? Does a canary fly? Does an ostrich fly?



Primitive relations vs. derived relations

It's called inheritance (or property inheritance)

AI people were using inheritance long before the object-oriented programming world appropriated it

Cognitive psychology was explaining memory organization with inheritance long before AI stole it

The *is_a* attributes/arcs are special -- they're what tells the reasoning system that the class at the start of the arc can inherit from the superclass at the end of the arc

Relational networks work great for nouns...

...but for representing individual verbs, the relational network as just described hasn't been as helpful

Many symbolic approaches to language understanding have adopted an approach of representing verbs as the composition of some number of primitive actions such as

physical transfer of an object from point A to point B
abstract transfer of possession of something form entity A to entity B
mental transfer of information from entity A to entity B

(these are just a few examples...there are others)

Slot-filler representation

John ordered a Big Mac.

is represented by the following

John mtrans <something> to the cashier

Slot-filler representation

John ordered a Big Mac.

is represented by the following

John mtrans to the cashier

cashier ptrans bigmac to John

+

cashier atrans bigmac to John

+

John ptrans money to cashier

+

John atrans money to cashier

Slot-filler representation

Each of these primitive actions can be represented as a collection of labeled slots and fillers (not unlike attributes and values)

John mtrans to the cashier

cashier ptrans bigmac to John

+

cashier atrans bigmac to John

+

John ptrans money to cashier

+

John atrans money to cashier

Slot-filler representation

Each of these primitive actions can be represented as a collection of labeled slots and fillers (not unlike attributes and values)

[action = ptrans, actor = cashier, object = bigmac, from = cashier, to = John]
+
[action = atrans, actor = cashier, object = bigmac, from = cashier, to = John]
+
[action = ptrans, actor = John, object = money, from = John, to = cashier]
+
[action = atrans, actor = John, object = money, from = John, to = cashier]

Slot-filler representation

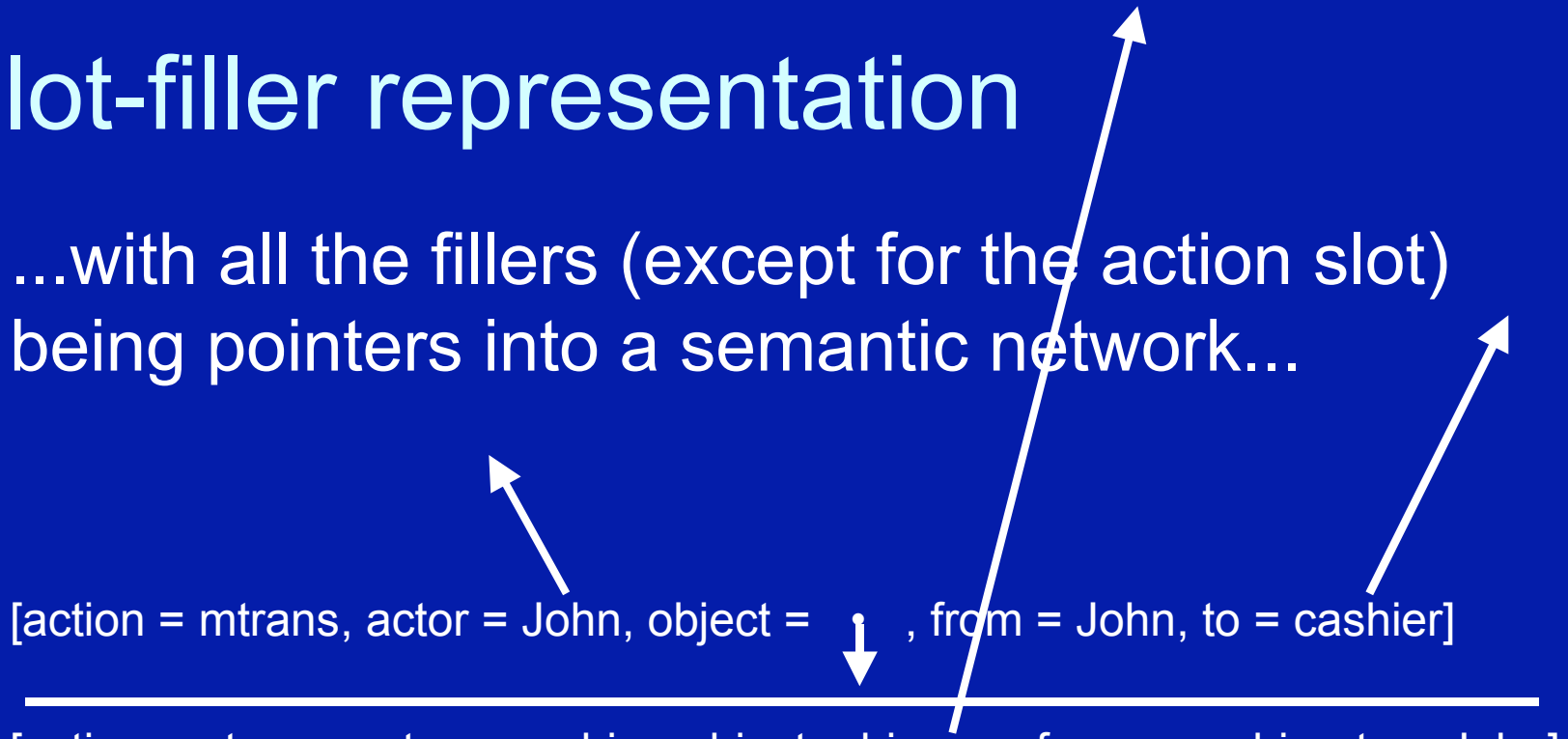
The representation for the sentence itself also becomes a slot-filler representation...

[action = mtrans, actor = John, object =  , from = John, to = cashier]

[action = ptrans, actor = cashier, object = bigmac, from = cashier, to = John]
+
[action = atrans, actor = cashier, object = bigmac, from = cashier, to = John]
+
[action = ptrans, actor = John, object = money, from = John, to = cashier]
+
[action = atrans, actor = John, object = money, from = John, to = cashier]

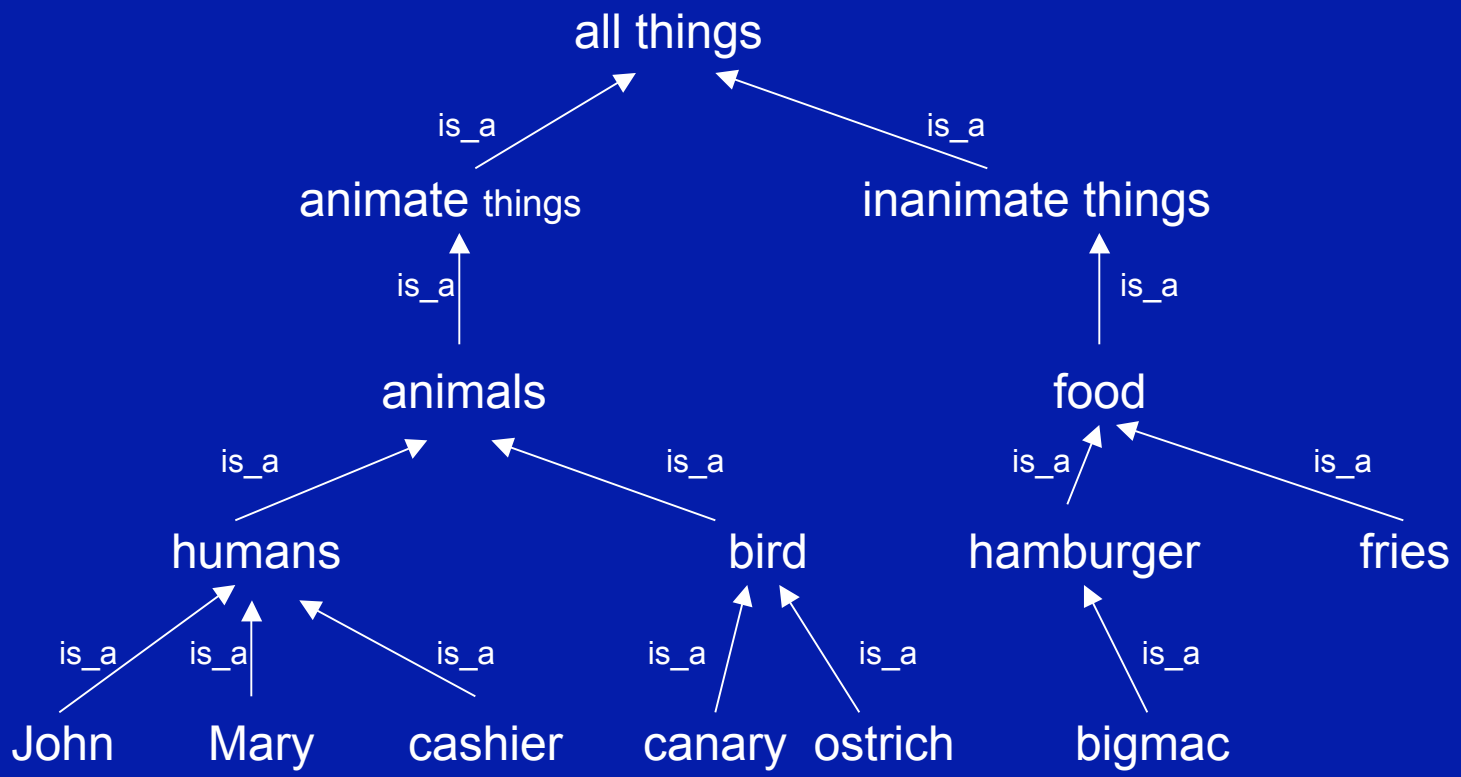
Slot-filler representation

...with all the fillers (except for the action slot) being pointers into a semantic network...



Slot-filler representation

...that could look like this, but this is far too simple



Representing Knowledge

What is a knowledge representation scheme?

- a set of conventions about how to describe a class of things
- a description makes use of the conventions of a representation to describe some particular thing within that class of things
- a given representation needs a set of symbols (vocabulary) with some understood mapping between the symbols and primitives in the world being represented (objects, attributes, relationships)
- the representation also needs some rules or conventions for how to order or combine symbols into more complex expressions which then become descriptions (these rules are a grammar for the representation language)
- a set of operators or procedures which permit the creation and manipulation of descriptions
- this should sound vaguely familiar - it's a discussion of the second R in RRS (reasoning and representation system)

Good News and Bad News

The good news is that once a problem is described using the appropriate representation, the problem is almost solved...the needed processing will be apparent

The bad news is that describing the knowledge correctly is really hard -- why?
Like we said before:

- it's voluminous
- it's hard to characterize accurately
- it's constantly changing
- it's organized in different ways depending on how it's used

Desirable attributes of a knowledge representation approach

- capture generalities in the world being modeled
- easily modifiable to reflect changes so that new knowledge can be derived from old knowledge
- transparent - understandable by people who provide the knowledge as well as those who look at it later
- usable even if not entirely accurate or complete
- explicitly represent important objects and relationships
- natural constraints on how one object or relation influences another should be obvious
- irrelevant detail should be suppressed (abstracted away)
- complete -- everything that needs to be represented can be represented
- concise -- what needs to be said can be said efficiently
- fast -- you can store and retrieve information quickly
- computable -- enables reasoning to proceed easily with known procedures (doesn't rely on bizarre coding tricks)

Or more succinctly, we want...

representational adequacy - the ability to represent all the kinds of knowledge that are needed in the domain

inferential adequacy - the ability to manipulate the structures in such a way as to derive new structures corresponding to new knowledge inferred from old

inferential efficiency - the ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mechanisms in the most promising directions

acquisitional efficiency - the ability to acquire new information easily

(from “Artificial Intelligence” by Elaine Rich and Kevin Knight)

What your book wants you to know

AI is a software engineering enterprise...there are lots of questions that you need answers to before you start hacking

Many of these questions have to do with how to represent the knowledge in your system

Some representations are qualitatively better than others depending on the problem you're trying to solve

Semantic (relational) networks and slot-filler representations are useful and flexible approaches to knowledge representation

You should read chapter 5, where you'll find different questions to be answered, including...