CPSC 322
Introduction to Artificial Intelligence

November 3, 2004

Representing Knowledge

Here are some general software engineering questions you may want to ask before building AI systems:

- What exactly is the activity that you want from this system you're going to create to solve some complex problem?
- What does your system need to know in order to perform that activity?
- How are you going to encode or represent that knowledge inside your system? (e.g., What will the language of the symbols be? What will the symbols map to?)
- How will your system know which piece(s) of knowledge to use at a given time, and how will it get at that knowledge without looking at all the knowledge?
- Once the system finds the appropriate knowledge, how will it use the knowledge?

Most of the questions have something to do with knowledge

As an example...

Let's say you're asked to build a system to understand simple stories

John and Mary went to McDonald's.
John ordered a Big Mac and fries.
Mary had a Quarter Pounder.
John put the trash in the wastebasket.
They went home.

- What exactly is the activity that you want from this system you're going to create to solve some complex problem?

As an example...

Answer questions

Paraphrase the story

Construct some internal representation that shows not only what was explicitly stated but also what can reasonably be inferred

As an example...

Let's say you're asked to build a system to understand simple stories

John and Mary went to McDonald's.
John ordered a Big Mac and fries.
Mary had a Quarter Pounder.
John put the trash in the wastebasket.
They went home.

- What does your system need to know in order to perform that activity?

As an example...

Language syntax

The meanings of words

The meanings of sentences (clauses)

Commonly-held background knowledge (e.g., what objects and events can you reasonably expect in the McDonald's domain? the more general fast-food restaurant domain?)

As an example...

Let's say you're asked to build a system to understand simple stories

John and Mary went to McDonald's.
John ordered a Big Mac and fries.
Mary had a Quarter Pounder.
John put the trash in the wastebasket.
They went home.

- How are you going to encode or represent that knowledge inside your system? (e.g., What will the language of the symbols be? What will the symbols map to?)

As an example...

Syntax as grammar rules:

```
sentence(T0,T2,s(NP,VP)) <- nounphrase(T0,T1,NP) & verbphrase(T1,T2,VP).


nounphrase(T0,T3,np(DET,NOUN,PP)) <- determiner(T0,T1,DET) &
                                     noun(T1,T2,NOUN) &
                                     prepositionalphrase(T2,T3,PP).

nounphrase(T0,T1,np(NAME)) <- name(T0,T1,NAME).

prepositionalphrase(T,T,[]).
prepositionalphrase(T0,T2,pp(PREP,NP)) <- preposition(T0,T1,PREP) &
                                          nounphrase(T1,T2,NP).

verbphrase(T0,T3,vp(VERB,NP,PP)) <- verb(T0,T1,VERB) &
                                    nounphrase(T1,T2,NP) &
                                    prepositionalphrase(T2,T3,PP).
```

As an example...

Syntax as grammar rules:

```
determiner(T,T,[]).
determiner([a|T],T,a).
determiner([the|T],T,the).
name([john|T],T,john).
name([mary|T],T,mary).
noun([mcdonalds|T],T,mcdonalds).
noun([bigmac|T],T,bigmac).
noun([fries|T],T,fries).
noun([quarterpounder|T],T,quarterpounder).
noun([trash|T],T,trash).
noun([wastebasket|T],T,wastebasket).
verb([went|T],T,went).
verb([ordered|T],T,ordered).
verb([had|T],T,had).
verb([put|T],T,put).
preposition([to|T],T,to).
preposition([in|T],T,in).
```

As an example...

Syntax as grammar rules:

```
cilog: ask sentence([john,ordered,a,bigmac],X,Y).
Answer: sentence([john, ordered, a, bigmac], [],
        s(np(john), vp(ordered, np(a, bigmac, []), [])))).
 Runtime since last report: 0 secs.
  [ok,more,how,help]: |: ok.
```

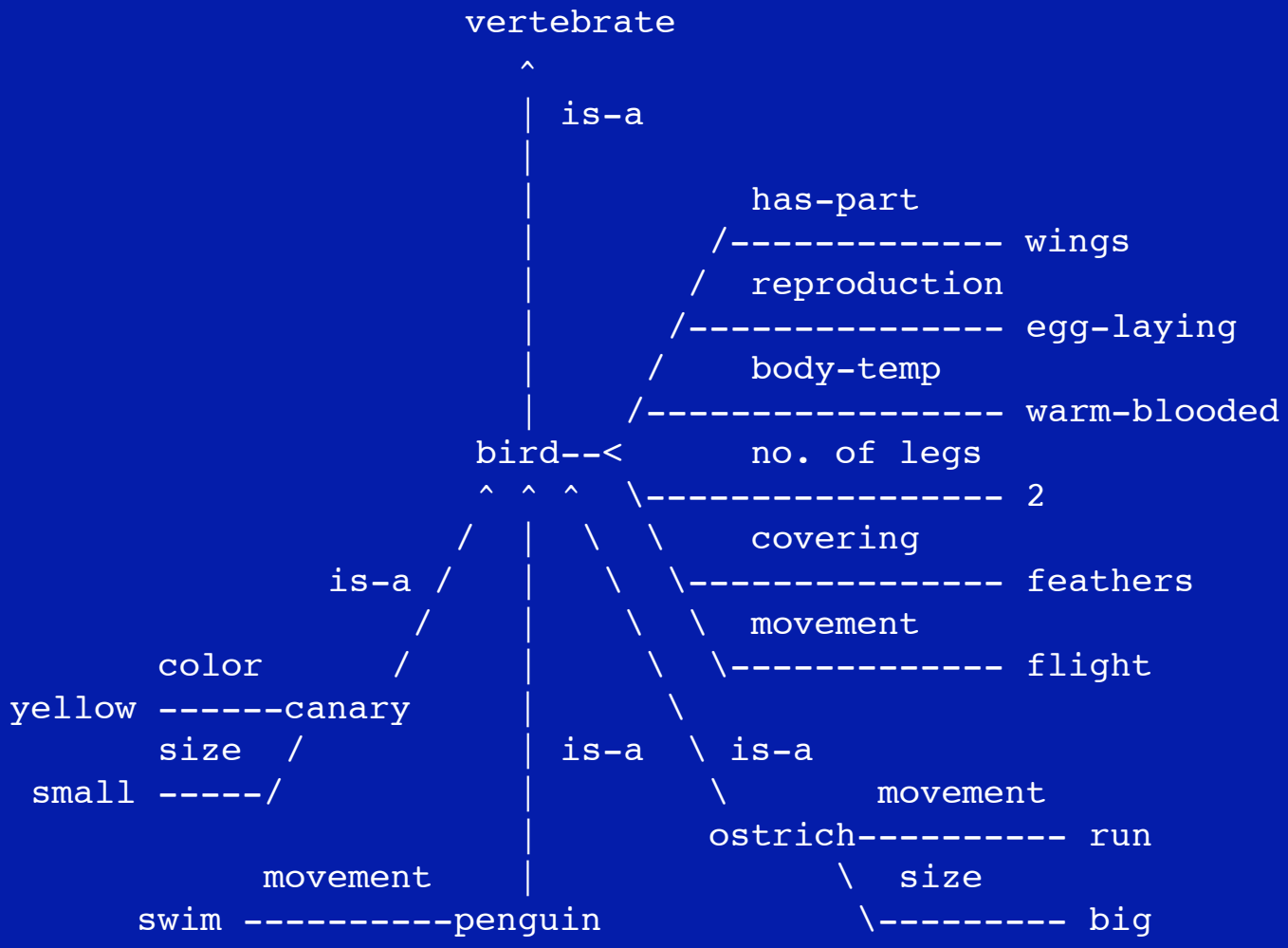
As an example...

Word meanings as collections of attributes and relations to other words -- consider the dictionary:

```
dog:        any of a large and varied group of domesticated
            animals related to the fox, wolf, and jackal

chihuahua:  any of an ancient Mexican breed of very small dog
            with large, pointed ears

bird:       any of a class of warm-blooded, two-legged,
            egg-laying vertebrates with feathers and wings

penguin:    any of an order of flight-less birds found in the
            Southern Hemisphere, having webbed feet and
            paddle-like flippers for swimming and diving

ostrich:    a large, swift-running bird of Africa and the
            Near East; the largest and most powerful of
            living birds; it has a long neck, long legs, two
            toes on each foot, and small useless wings

canary:     a small yellow songbird of the finch family,
            native to the Canary Islands
```

As an example...

Graphically, it looks like this (the bird part, at least):



As an example...

In CILOG, it might look like this (though there are other formats you could use):

```
prop(bird, is_a, vertebrate).
prop(bird, has_part, wings).
prop(bird, reproduction, egg_laying).
prop(bird, body_temp, warm_blooded).
prop(bird, no_of_legs, 2).
prop(bird, covering, feathers).
prop(bird, movement, flight).
prop(ostrich, is_a, bird).
prop(ostrich, movement, run).
prop(ostrich, size, big).
prop(penguin, is_a, bird).
prop(penguin, movement, swim).
prop(canary, is_a, bird).
prop(canary, size, small).
prop(canary, color, yellow).
```

As an example...

In CILOG, it might look like this (though there are other formats you could use):

```
prop(bigmac, is_a, hamburger).
prop(bigmac, has_part, special_sauce).
                :
                :
```

As an example...

In CILOG, it might look like this (though there are other formats you could use):

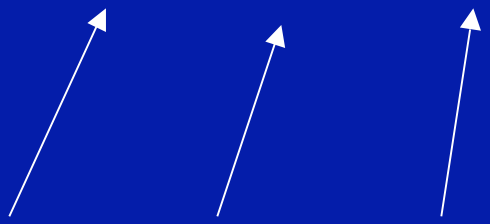
```
prop(bigmac, is_a, hamburger).
```

`prop` is just a predicate invented in the book. Short for property(?) -- it's not magic

As an example...

In CILOG, it might look like this (though there are other formats you could use):

```
prop(bigmac, is_a, hamburger).
```



object-attribute-value representation

- object is a node name
- attribute is an arc label
- value is a node name
- the graph is called a semantic network or a relational network
- the real “meaning” is as much in the arc labels as in the nodes, if not more so

As an example...

Don’t trust the dictionary to be endlessly helpful:

```
rock:       a large mass of stone

stone:      the hard, solid, nonmetallic mineral matter that
            rock is composed of
```

Other places to find relational networks

- Relational databases
- Cognitive psychology
- The Internet (Mapquest, Six Degrees of Kevin Bacon, the Web itself)
- Popular media