CPE/CSC 481: Knowledge-Based Systems

Franz J. Kurfess

Computer Science Department
California Polytechnic State University
San Luis Obispo, CA, U.S.A.

CAL POLY

Usage of the Slides

- these slides are intended for the students of my CPE/CSC 481 “Knowledge-Based Systems” class at Cal Poly SLO
  - if you want to use them outside of my class, please let me know (fkurfess@calpoly.edu)
- I usually put together a subset for each quarter as a “Custom Show”
  - to view these, go to “Slide Show => Custom Shows”, select the respective quarter, and click on “Show”
  - in Apple Keynote (.key files), I use the “Skip” feature to achieve similar results
- To print them, I suggest to use the “Handout” option
  - 4, 6, or 9 per page works fine
  - Black & White should be fine; there are few diagrams where color is important

Overview
Logic and Reasoning

- Motivation
- Objectives
- Knowledge and Reasoning
  - logic as prototypical reasoning system
  - syntax and semantics
  - validity and satisfiability
  - logic languages
- Reasoning Methods
  - propositional/predicate logic
  - inference methods
- Reasoning in Knowledge-Based Systems
  - shallow and deep reasoning
  - forward and backward chaining
  - rule-based systems
  - alternative inference methods
  - meta-knowledge
- Important Concepts and Terms
- Chapter Summary

Motivation

- without reasoning, knowledge-based systems would be practically worthless
  - derivation of new knowledge
  - examination of the consistency or validity of existing knowledge
- reasoning in KBS can perform certain tasks better than humans
  - reliability, availability, speed
  - also some limitations
    - common-sense reasoning
    - complex inferences

Objectives

- be familiar with the essential concepts of logic and reasoning
  - sentence, operators, syntax, semantics, inference methods
- appreciate the importance of reasoning for knowledge-based systems
  - generating new knowledge
  - explanations
- understand the main methods of reasoning used in KBS
  - shallow and deep reasoning
  - forward and backward chaining
- evaluate reasoning methods for specific tasks and scenarios
- apply reasoning methods to simple problems

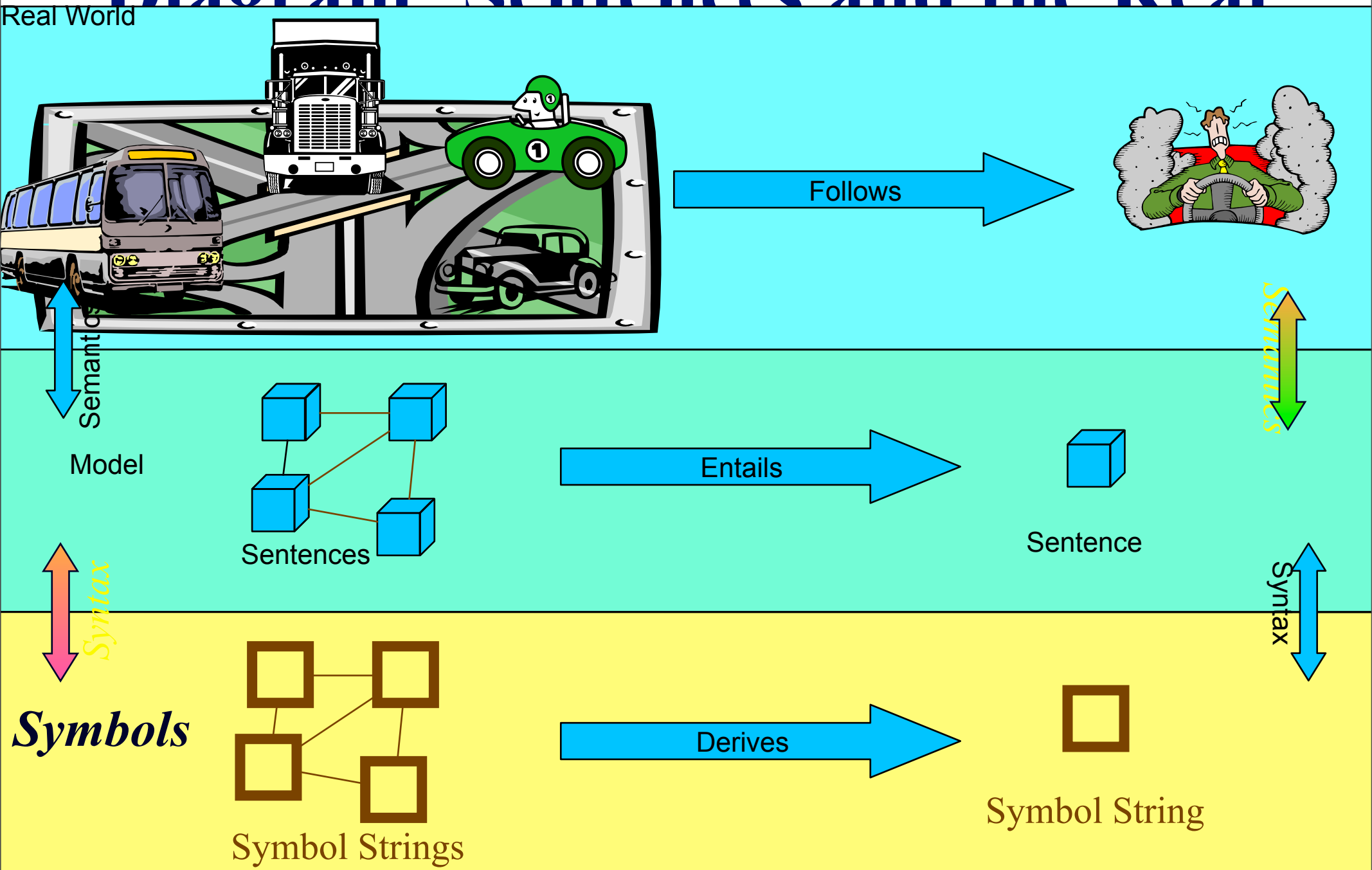
Knowledge Representation Languages

- syntax
  - sentences of the language that are built according to the syntactic rules
  - some sentences may be nonsensical, but syntactically correct
- semantics
  - refers to the facts about the world for a specific sentence
  - interprets the sentence in the context of the world
  - provides meaning for sentences
- languages with precisely defined syntax and semantics can be called logics

Sentences and the Real World

- syntax
  - describes the principles for constructing and combining sentences
    - e.g. BNF grammar for admissible sentences
    - inference rules to derive new sentences from existing ones
- semantics
  - establishes the relationship between a sentence and the aspects of the real world it describes
  - can be checked directly by comparing sentences with the corresponding objects in the real world
    - not always feasible or practical
  - compositional semantics
    - complex sentences can be checked by examining their individual parts

Diagram: Sentences and the Real



Logic

background
propositional logic
first order predicate logic

Introduction to Logic

- expresses knowledge in a particular mathematical notation

```
All birds have wings --> ¥x. Bird(x) -> HasWings(x)
```

- rules of inference
  - guarantee that, given true facts or premises, the new facts or premises derived by applying the rules are also true

```
All robins are birds --> ¥x Robin(x) -> Bird(x)
```

- given these two facts, application of an inference rule gives:

```
¥x Robin(x) -> HasWings(x)
```

Logic and Knowledge

- rules of inference act on the superficial structure or syntax of the first two sentences
  - doesn't say anything about the meaning of birds and robins
  - could have substituted mammals and elephants etc.
- major advantages of this approach
  - deductions are guaranteed to be correct to an extent that other representation schemes have not yet reached
  - easy to automate derivation of new facts
- problems
  - computational efficiency
  - uncertain, incomplete, imprecise knowledge

Summary of Logic Languages

- propositional logic
  - facts
  - true/false/unknown
- first-order logic
  - facts, objects, relations
  - true/false/unknown
- temporal logic
  - facts, objects, relations, times
  - true/false/unknown
- probability theory
  - facts
  - degree of belief [0..1]
- fuzzy logic
  - degree of truth
  - degree of belief [0..1]

Modus Ponens

- eliminates =>

$$\frac{(X \Rightarrow Y), \quad X}{Y}$$

  - If it rains, then the streets will be wet.
  - It is raining.
  - Infer the conclusion: The streets will be wet.
    - (affirms the antecedent)

Modus tollens

$$\frac{(X \Rightarrow Y),\ \sim Y}{\neg X}$$

- If it rains, then the streets will be wet.
- The streets are not wet.
- Infer the conclusion: It is not raining.

- NOTE: Avoid the fallacy of affirming the consequent:
  - If it rains, then the streets will be wet.
  - The streets are wet.
  - cannot conclude that it is raining.

  - If Bacon wrote Hamlet, then Bacon was a great writer.
  - Bacon was a great writer.
  - cannot conclude that Bacon wrote Hamlet.

Syllogism

- chain implications to deduce a conclusion

$$\frac{(X \Rightarrow Y),\ (Y \Rightarrow Z)}{(X \Rightarrow Z)}$$

More Inference Rules

- and-elimination
- and-introduction
- or-introduction
- double-negation elimination
- unit resolution

Resolution

$$\frac{(X \vee Y),\ (\neg Y \vee Z)}{(X \vee Z)}$$

- basis for the inference mechanism in the Prolog language and some theorem provers

Complexity issues

- truth table enumerates 2n rows of the table for any proof involving n symbol
  - it is complete
  - computation time is exponential in n
- checking a set of sentences for satisfiability is NP-complete
  - but there are some circumstances where the proof only involves a small subset of the KB, so can do some of the work in polynomial time
  - if a KB is monotonic (i.e., even if we add new sentences to a KB, all the sentences entailed by the original KB are still entailed by the new larger KB), then you can apply an inference rule locally (i.e., don't have to go checking the entire KB)

Reasoning in Knowledge-Based Systems

shallow and deep reasoning
forward and backward chaining
alternative inference methods
meta-knowledge

Shallow and Deep Reasoning

- shallow reasoning
  - also called experiential reasoning
  - aims at describing aspects of the world heuristically
  - short inference chains
  - possibly complex rules
- deep reasoning
  - also called causal reasoning
  - aims at building a model of the world that behaves like the "real thing"
  - long inference chains
  - often simple rules that describe cause and effect relationships

Examples Shallow and Deep Reasoning

- shallow reasoning

```
IF a car has
   a good battery
   good spark plugs
   gas
   good tires
  THEN the car can move
```

- deep reasoning

```
IF the battery is good
THEN there is electricity

IF there is electricity AND
   good spark plugs
THEN the spark plugs will fire

IF the spark plugs fire AND
   there is gas
THEN the engine will run

IF the engine runs AND
   there are good tires
THEN the car can move
```

Forward Chaining

- given a set of basic facts, we try to derive a conclusion from these facts
- example: What can we conjecture about Clyde?

```
IF elephant(x) THEN mammal(x)

IF mammal(x) THEN animal(x)

elephant (Clyde)
```

modus ponens:

```
IF p THEN q
p
---------------
q
```

unification:

find compatible values for variables

Forward Chaining Example

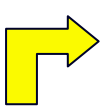
```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
```

```
q
```



```
IF elephant(   x   ) THEN mammal(   x   )
```

```
elephant (Clyde)
```

Forward Chaining Example

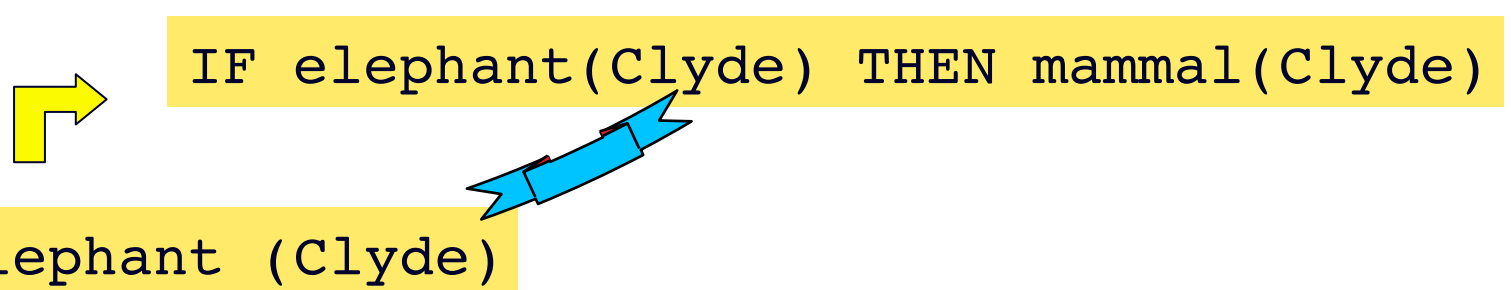
```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
```

```
q
```



Forward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---------
q
```

```
IF mammal(  x  ) THEN animal(  x  )
IF elephant(Clyde) THEN mammal(Clyde)
elephant (Clyde)
```

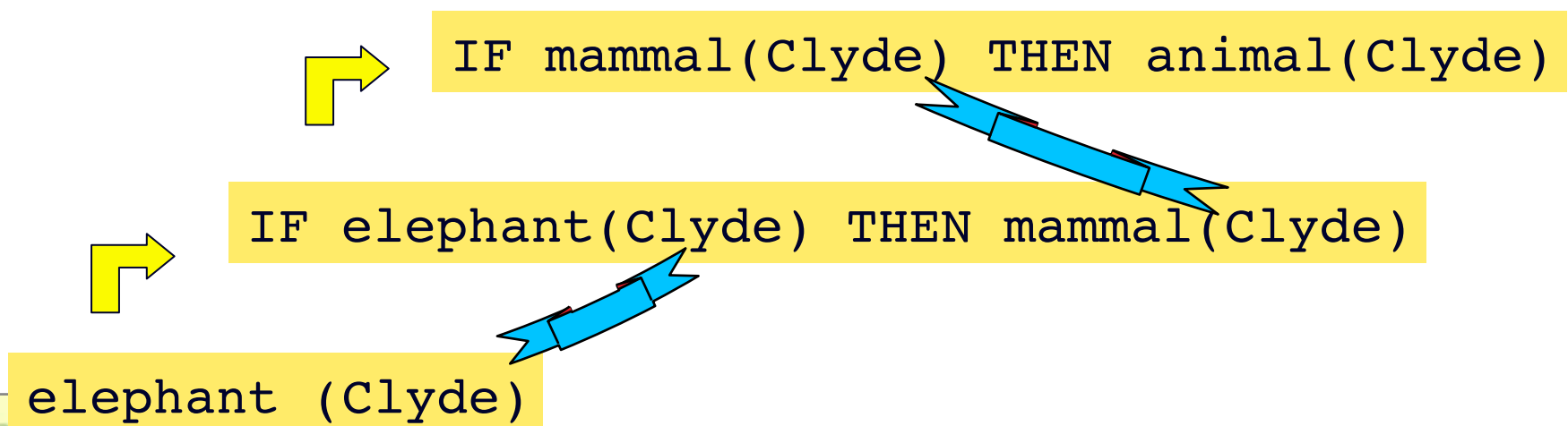
Forward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---------
q
```



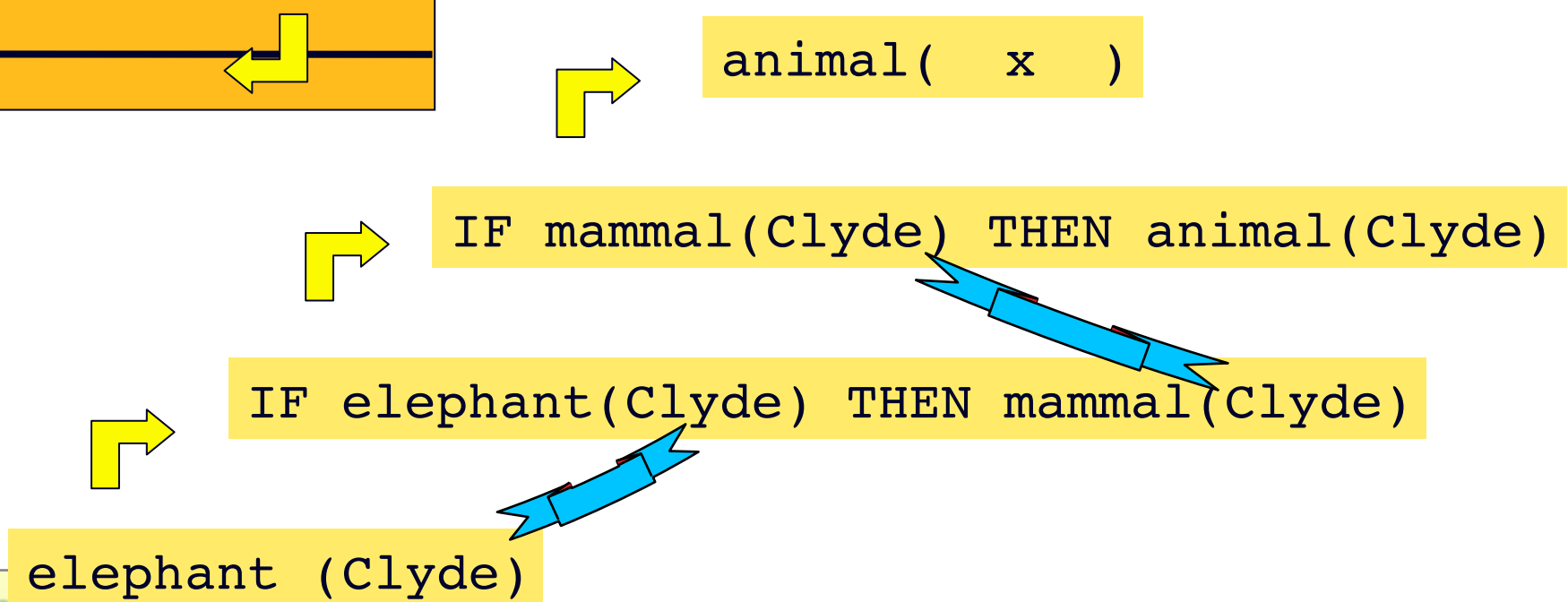
Forward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---------
q
```



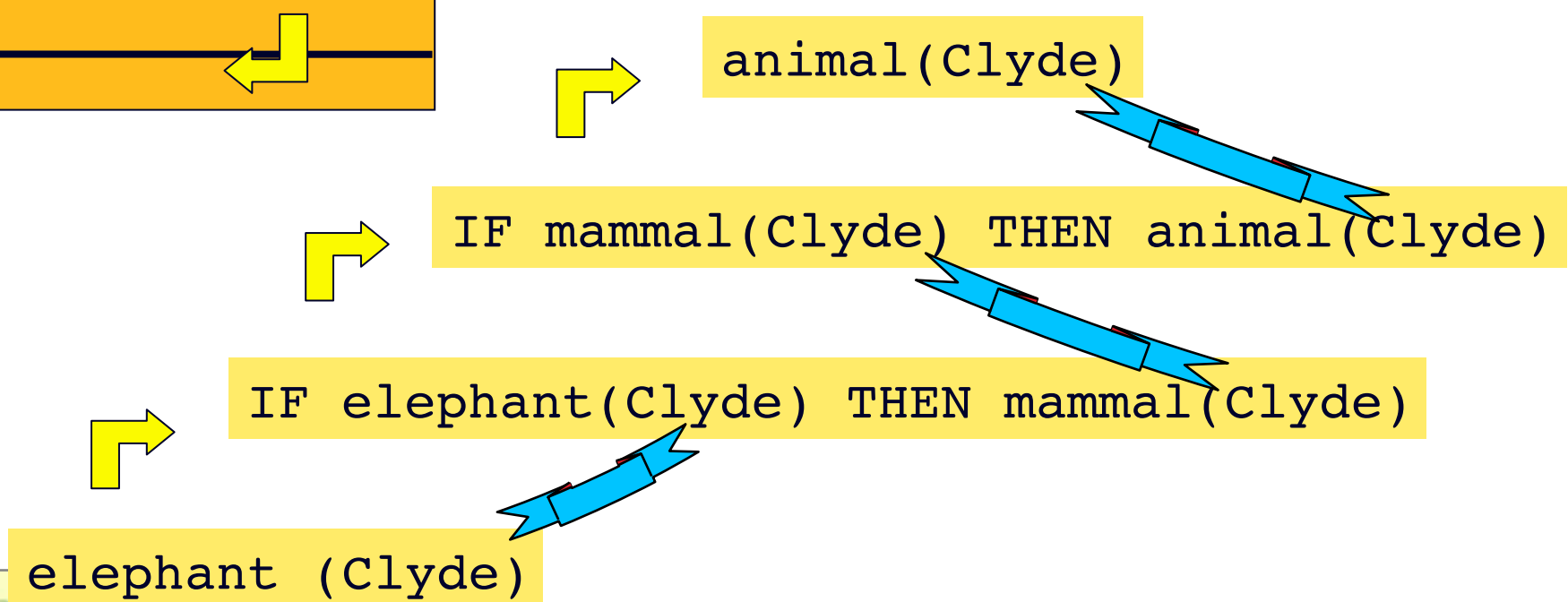
Forward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---
q
```



Backward Chaining

- try to find supportive evidence (i.e. facts) for a hypothesis
- example: Is there evidence that Clyde is an animal?

```
IF elephant(x) THEN mammal(x)

IF mammal(x) THEN animal(x)

elephant (Clyde)
```

modus ponens:

```
IF p THEN q
p
---------------
q
```

unification:

find compatible values for variables

Backward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---
q
```

```
animal(Clyde)
```

?

```
IF mammal(  x  ) THEN animal(  x  )
```

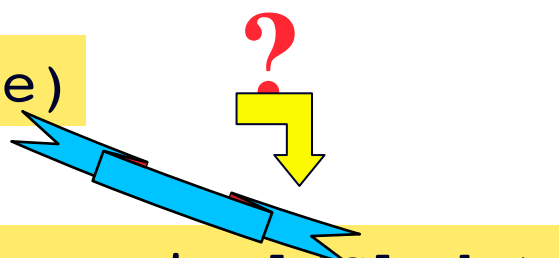
Backward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
q
```



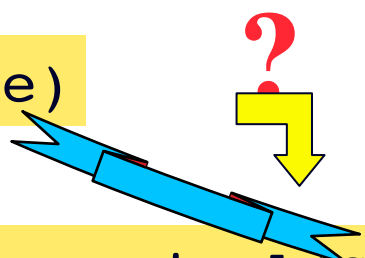
Backward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---------
q
```



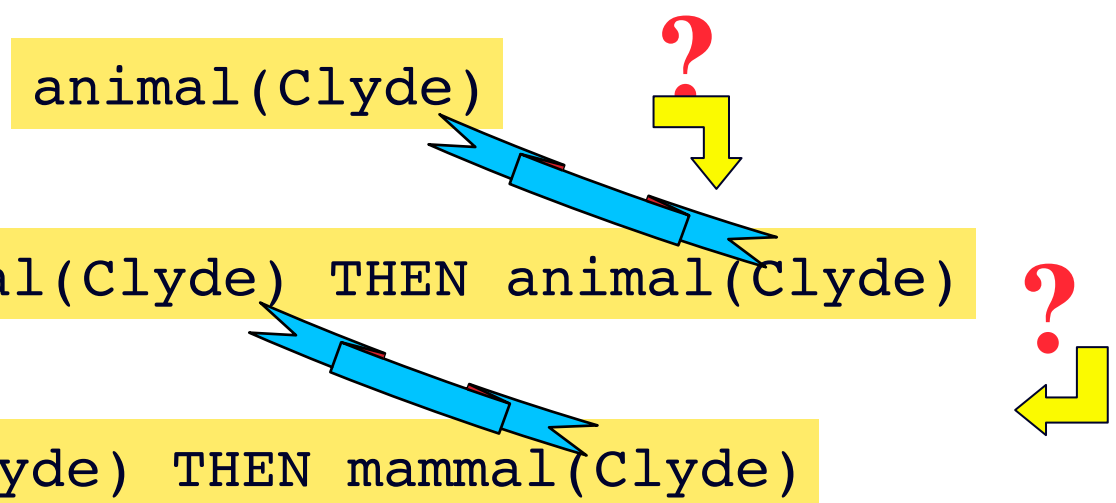
Backward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---------
q
```



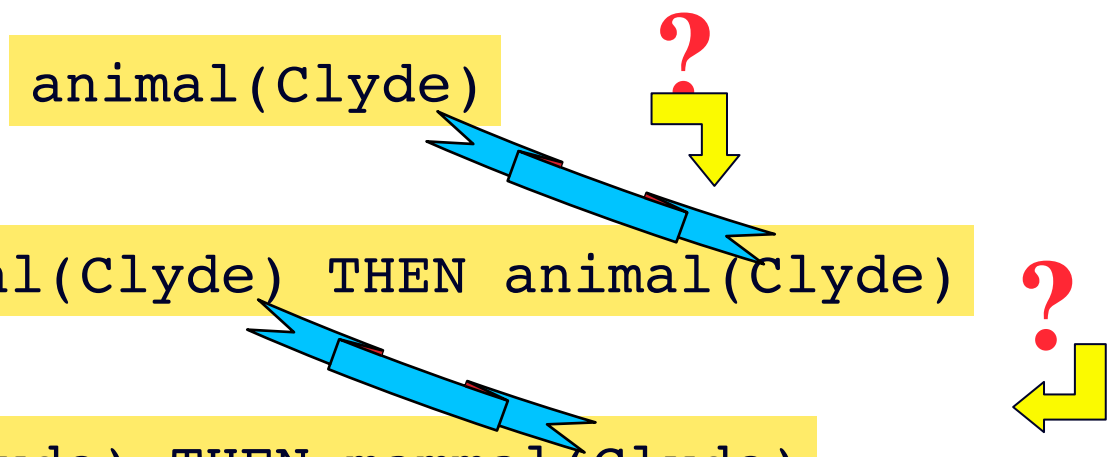
Backward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---------
q
```




```
elephant ( x )
```

?

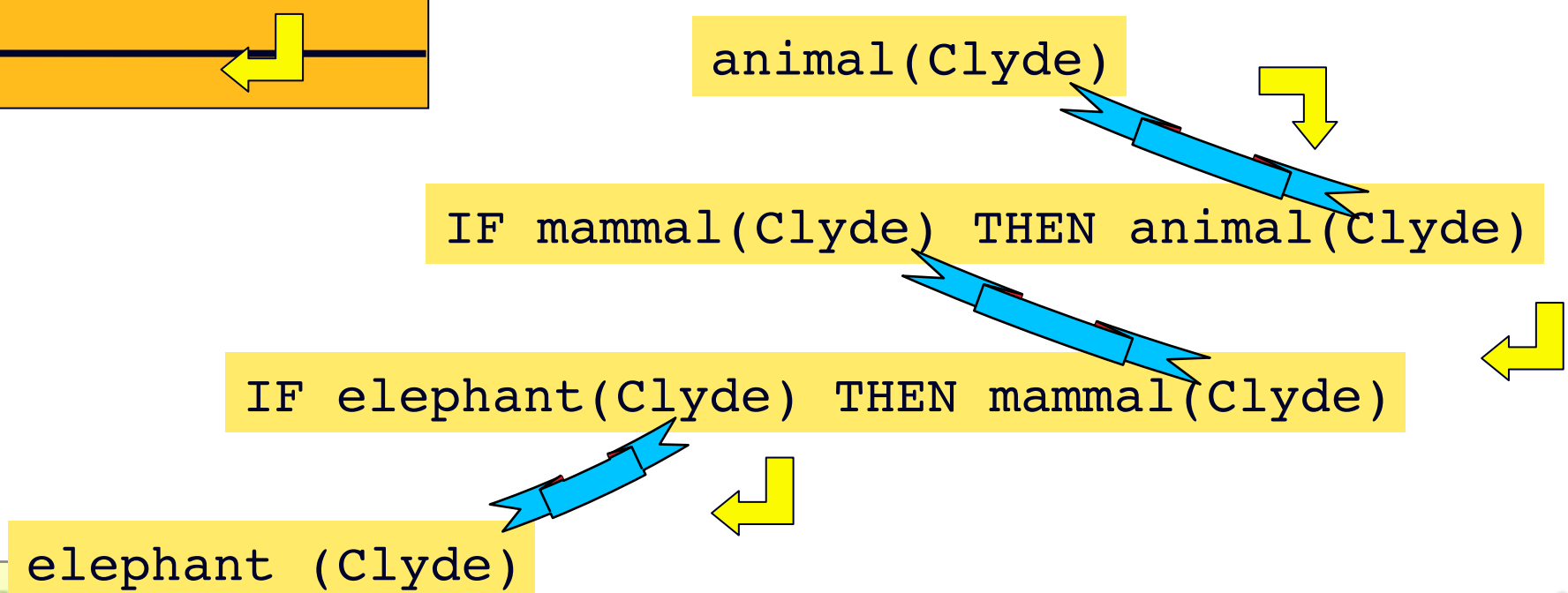
Backward Chaining Example

```
IF elephant(x) THEN mammal(x)
IF mammal(x) THEN animal(x)
elephant(Clyde)
```

unification:
find compatible values for variables

modus ponens:

```
IF p THEN q
p
---------
q
```



Forward vs. Backward Chaining

Forward Chaining	*Backward Chaining*
planning, control	diagnosis
data-driven	goal-driven (hypothesis)
bottom-up reasoning	top-down reasoning
find possible conclusions supported by given facts	find facts that support a given hypothesis
similar to breadth-first search	similar to depth-first search
antecedents (LHS) control evaluation	consequents (RHS) control evaluation

Alternative Inference Methods

theorem proving
probabilistic reasoning
fuzzy logic

Alternative Inference Methods

- theorem proving
  - emphasis on mathematical proofs, not so much on performance and ease of use
- probabilistic reasoning
  - integrates probabilities into the reasoning process
- fuzzy reasoning
  - enables the use of ill-defined predicates

Metaknowledge

- deals with “knowledge about knowledge”
  - e.g. reasoning about properties of knowledge representation schemes, or inference mechanisms
  - usually relies on higher order logic
    - in (first order) predicate logic, quantifiers are applied to variables
    - second-order predicate logic allows the use of quantifiers for function and predicate symbols
      - equality is an important second order axiom
        - two objects are equal if all their properties (predicates) are equal
    - may result in substantial performance problems

Important Concepts and Terms

- and operator
- atomic sentence
- backward chaining
- existential quantifier
- expert system shell
- forward chaining
- higher order logic
- Horn clause
- inference
- inference mechanism
- If-Then rules
- implication
- knowledge
- knowledge base
- knowledge-based system
- knowledge representation
- matching
- meta-knowledge
- not operator
- or operator
- predicate logic
- propositional logic
- production rules
- quantifier
- reasoning
- rule
- satisfiability
- semantics
- sentence
- symbol
- syntax
- term
- validity
- unification
- universal quantifier

Summary Reasoning

- reasoning relies on the ability to generate new knowledge from existing knowledge
  - implemented through inference rules
    - related terms: inference procedure, inference mechanism, inference engine
- computer-based reasoning relies on syntactic symbol manipulation (derivation)
  - inference rules prescribe which combination of sentences can be used to generate new sentences
  - ideally, the outcome should be consistent with the meaning of the respective sentences ("sound" inference rules)
- logic provides the formal foundations for many knowledge representation schemes
  - rules are frequently used in expert systems