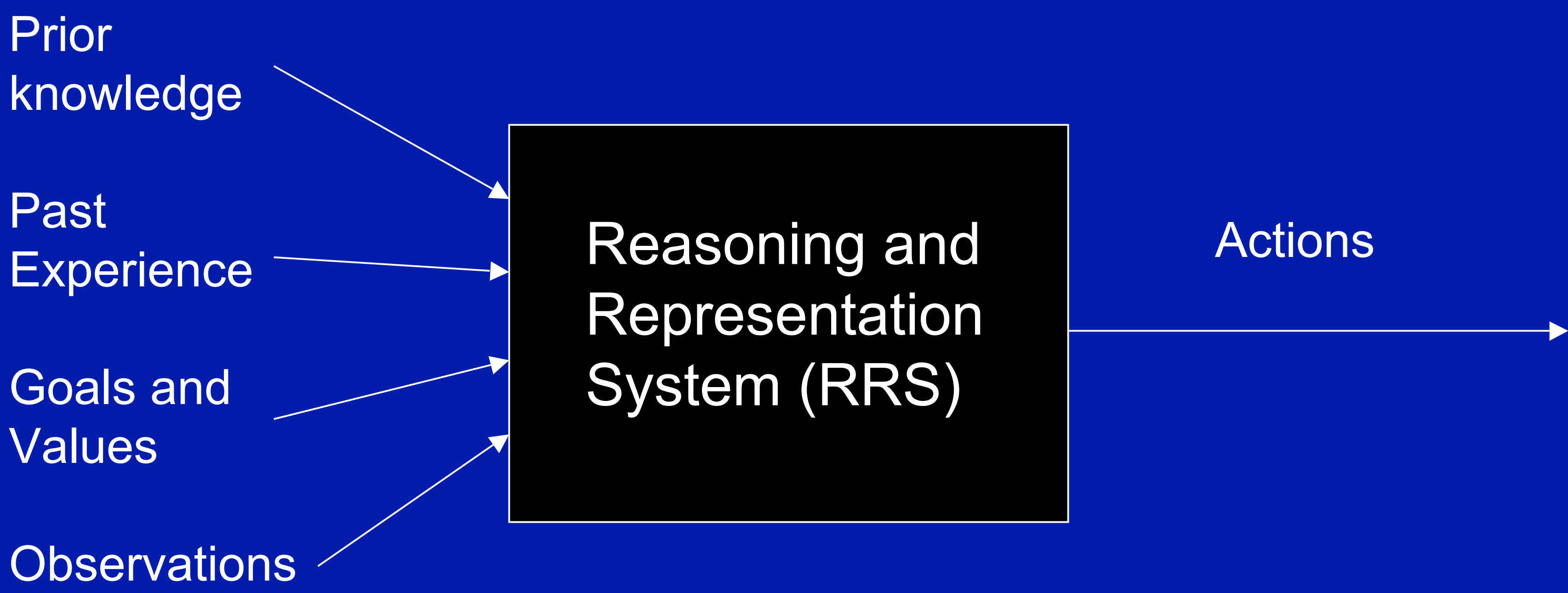
CPSC 322
Introduction to Artificial Intelligence

September 15, 2004

Highlights from last time

1. A peek at what’s inside the intelligent agent

The Intelligent Agent as Black Box



Reasoning and Representation System

A language for communication with the computer

A way to assign meaning to the language

Procedures to compute answers to problems given as input in the language

Highlights from last time

1. A peek at what’s inside the intelligent agent

2. Five steps to building an intelligent agent

Five Simple Steps to World Domination

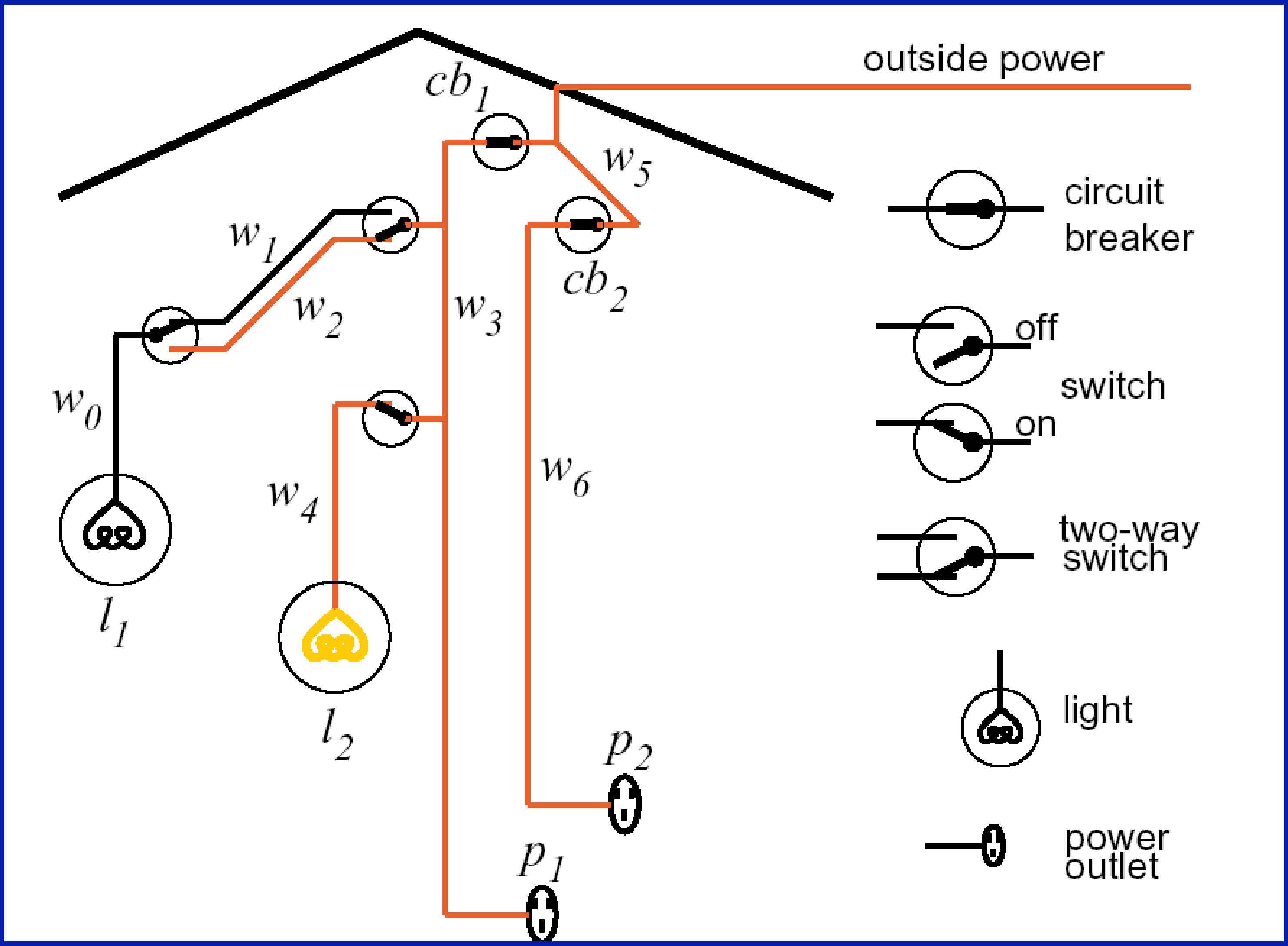
(or how to build the black box)

1. Begin with a task domain that you want to characterize
2. Distinguish the things you want to talk about in the domain (the ontology)
3. Use symbols in the computer to represent objects and relations in the domain. (Symbols *denote* objects...they're not really the objects.)
4. Tell the computer the knowledge about the domain.
5. Ask the RRS a question which prompts the RRS to reason with its knowledge to solve a problem, produce answers, or generate actions

Highlights from last time

1. A peek at what's inside the intelligent agent
2. Five steps to building an intelligent agent
3. We built an intelligent agent

The Diagnostic Assistant Domain



Our first intelligent agent*

```
cilog: tell on(l2) <- hot(w4).
cilog: tell hot(w4) <- closed(s3) & hot(w3).
cilog: tell hot(w3) <- closed(cb1) & hot(outside_power).
cilog: tell closed(cb1).
cilog: tell hot(outside_power).
cilog: tell closed(s3).
cilog: ask on(l2).
Answer: on(l2).
```

*OK, it has less intelligence than a plant, but everybody has to start somewhere.

Let's go back to our three-part Reasoning and Representation System

A language for communication with the computer
formal language

- Grammar defines legal symbols and how they can be put together in sentences
- Sentences express knowledge about domain
- The language is all the sentences that can be created from the grammar
- A knowledge base is a set of sentences from the language

Let's go back to our three-part Reasoning and Representation System

A way to assign meaning to the language

semantics

- Specifies the meaning of sentences in the language
- A commitment to how the symbols in the language relate to the task domain
- The semantic commitment is yours -- it's in your head, not in the computer

Let's go back to our three-part Reasoning and Representation System

Procedures to compute answers to problems given as input in the language

reasoning theory or proof procedure

- A (possibly nondeterministic) specification of how an answer can be derived from the knowledge base
- Often a set of inference rules for creating new knowledge from the knowledge base...
- ...or a (nondeterministic) specification of how an answer is computed

Two New Words

nondeterministic: exhibiting nondeterminism

nondeterminism: a property of a computation which may have more than one result.

How to implement nondeterminism:

- depth-first search with backtracking
- explore all possible solutions in parallel
- find an oracle

Two New Words

inference: (1) The act or process of deriving logical conclusions from premises known or assumed to be true. (2) The act of reasoning from facutal knowledge or evidence.

Three general classes of inference:

- deductive inference
- inductive inference
- abductive inference

An Implementation of a Reasoning and Representation System

Consists of:

A language parser that maps legal sentences of the formal language to some internal form stored as data structures

A reasoning procedure that combines a reasoning theory with a search strategy. The search strategy is a commitment to how to resolve the nondeterminism.

Note that this is all independent of semantics. This is just symbol manipulation by following a set of rules.

Simplifying Assumptions for our first Reasoning and Representation System

An agent's knowledge can be usefully described in terms of individuals and relations among individuals.

An agent's knowledge base consists of definite and positive statements. (i.e., nothing vague, no negation)

The environment is static. (i.e., nothing changes)

There are only a finite number of individuals of interest in the domain

Some of these assumptions will be relaxed as we go on.

Syntax for CILOG(Datalog)

A variable is a word that starts with an uppercase letter.

X, Y, Kurt, The_bald_guy, Q42

A constant is a word starting with a lowercase letter or it can be all digits (a numeral).

x, y, kurt, daughter, happy, q42, 493

A predicate symbol is a word starting with a lowercase letter.

x, y, kurt, daughter, happy, q42

A term is either a variable or a constant.

Syntax for CILOG(Datalog)

An atomic symbol (or atom) is of the form p or p(t_1 ,..., t_n) where p is a predicate symbol and each t_i is a term.

happy, teaches(kurt, cs322), between(s3, l2, cb1), mother(elizabeth, X)

A body is of the form a_1 & ... & a_m (or a_1 ^ ... ^ a_m) where each a_i is an atom.

A definite clause is either an atom, a, called a fact, or of the form a <- b, called a rule, where a, the head, is an atom and b is a body. The <- is read as “if”.

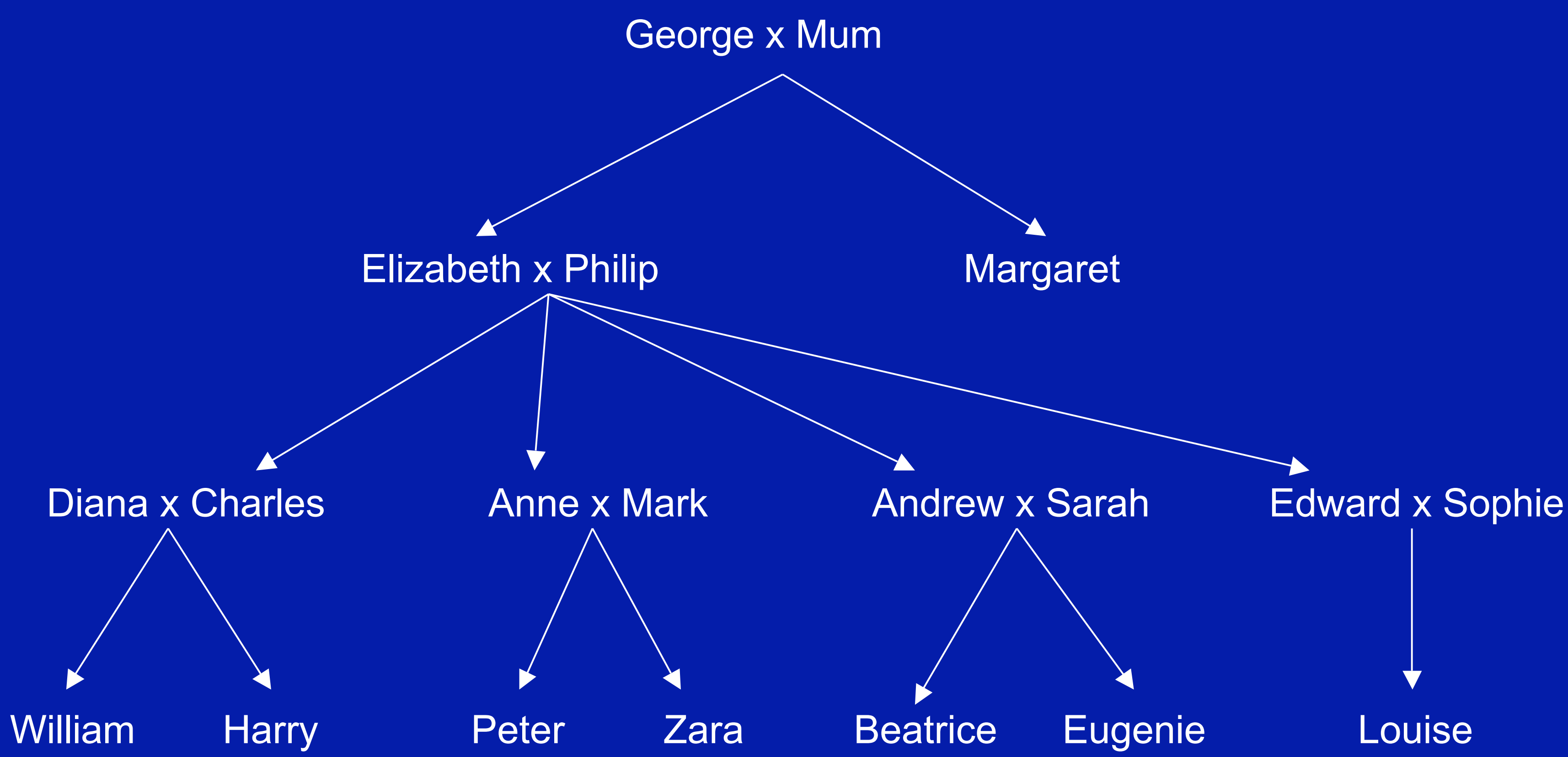
Syntax for CILOG(Datalog)

A query is of the form ask b (or ?b) where b is a body.

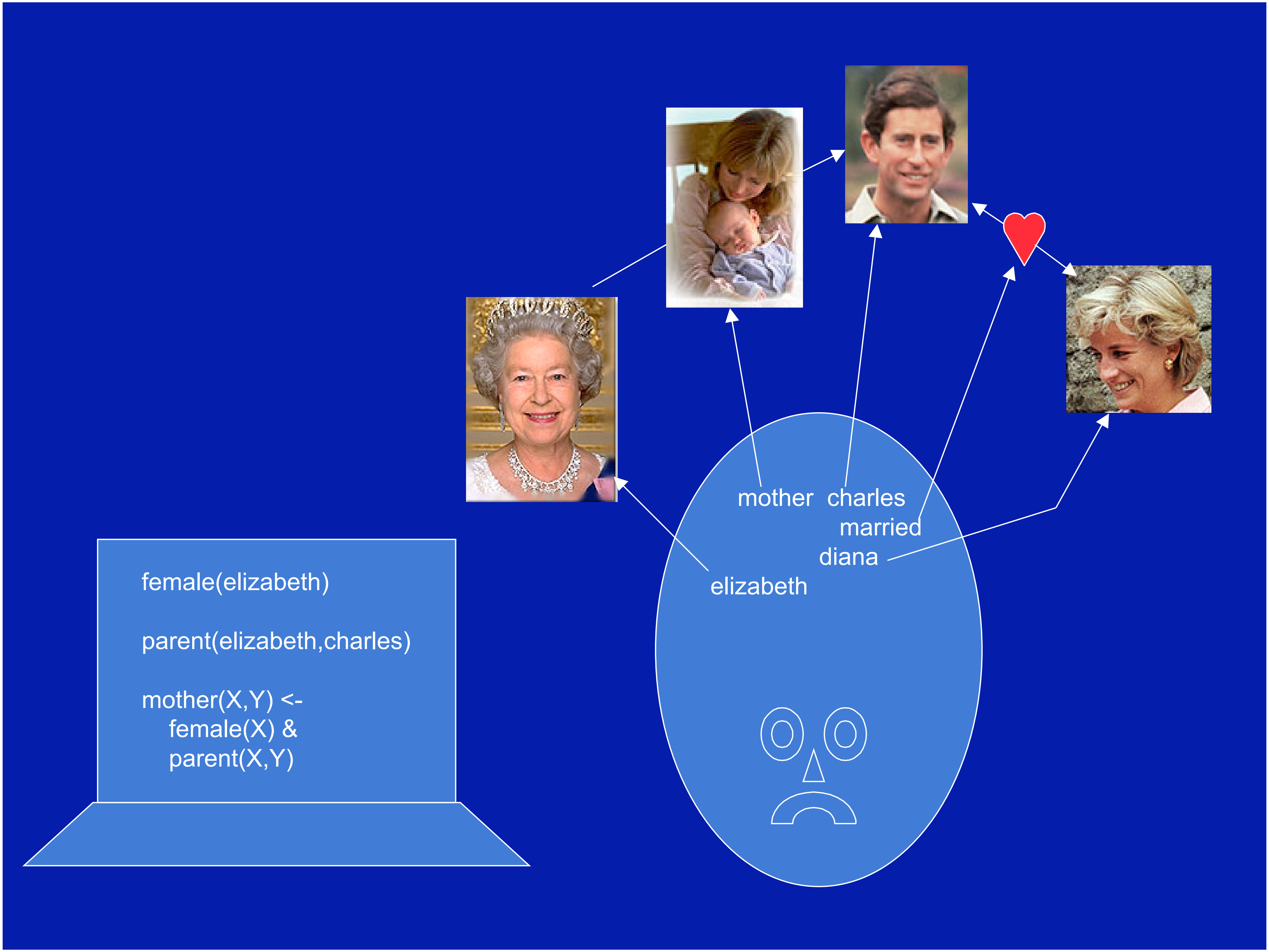
An expression is either a term, an atom, a definite clause, or a query.

A knowledge base is a set of definite clauses.

Your ordinary everyday family tree



Adapted from Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig
Updated from http://www.royal.gov.uk


female(elizabeth)
parent(elizabeth,charles)
mother(X,Y) <-
female(X) &
parent(X,Y)
mother
charles
married
diana
elizabeth