CPSC 322
Introduction to Artificial Intelligence

September 17, 2004

Highlights from last time
Reasoning and Representation System

A language for communication with the computer
formal language

A way to assign meaning to the language
semantics

Procedures to compute answers to problems given as input in the language
reasoning theory or proof procedure

Highlights from last time

An Implementation of an RRS 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.

Highlights from last time
Simplifying Assumptions for first RRS

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.

Highlights from last time
(One of) 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

Clarification

human inference may not be characterized as deductive inference, but...

- we don’t know how human inference works
- abduction and induction don’t maintain truth
- abduction and induction are poorly specified and harder to implement
- we don’t necessarily want our computers to exhibit human-like intelligence - maybe we just want the good parts and not the bad parts

Highlights from last time
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.

Highlights from last time
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”.

Highlights from last time
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.

Dissecting the definite clause

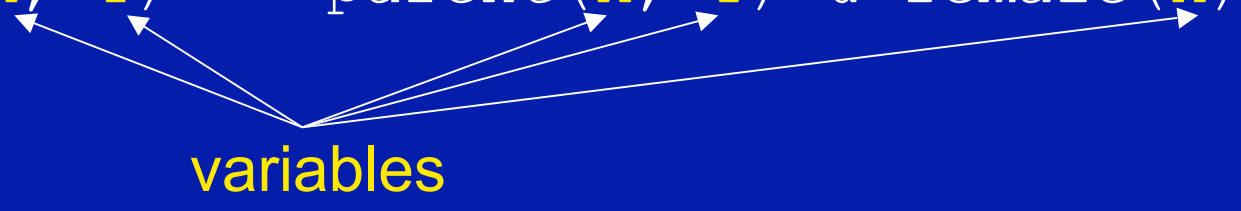
```
parent(elizabeth, charles)
```

```
mother(X, Y) <- parent(X, Y) & female(X)
```

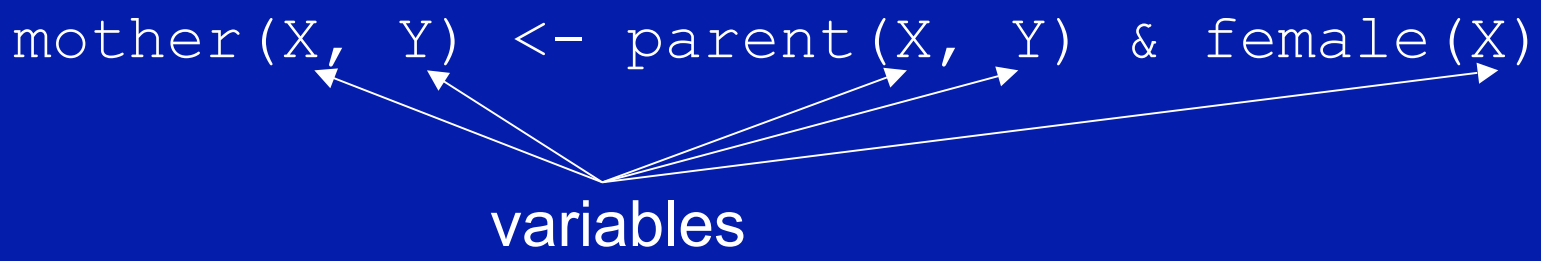
Dissecting the definite clause

```
parent(elizabeth, charles)
```

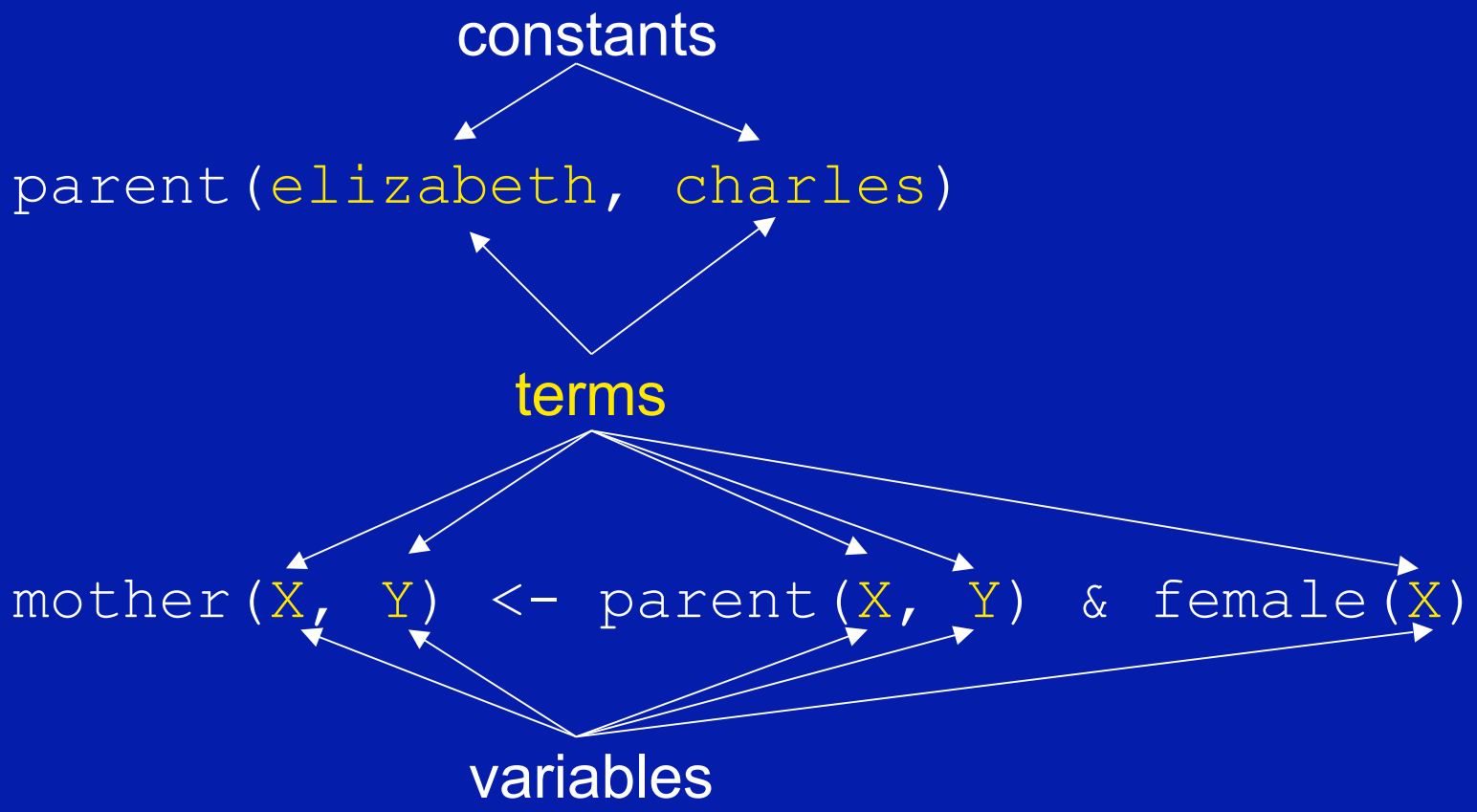
```
mother(X, Y) <- parent(X, Y) & female(X)
```



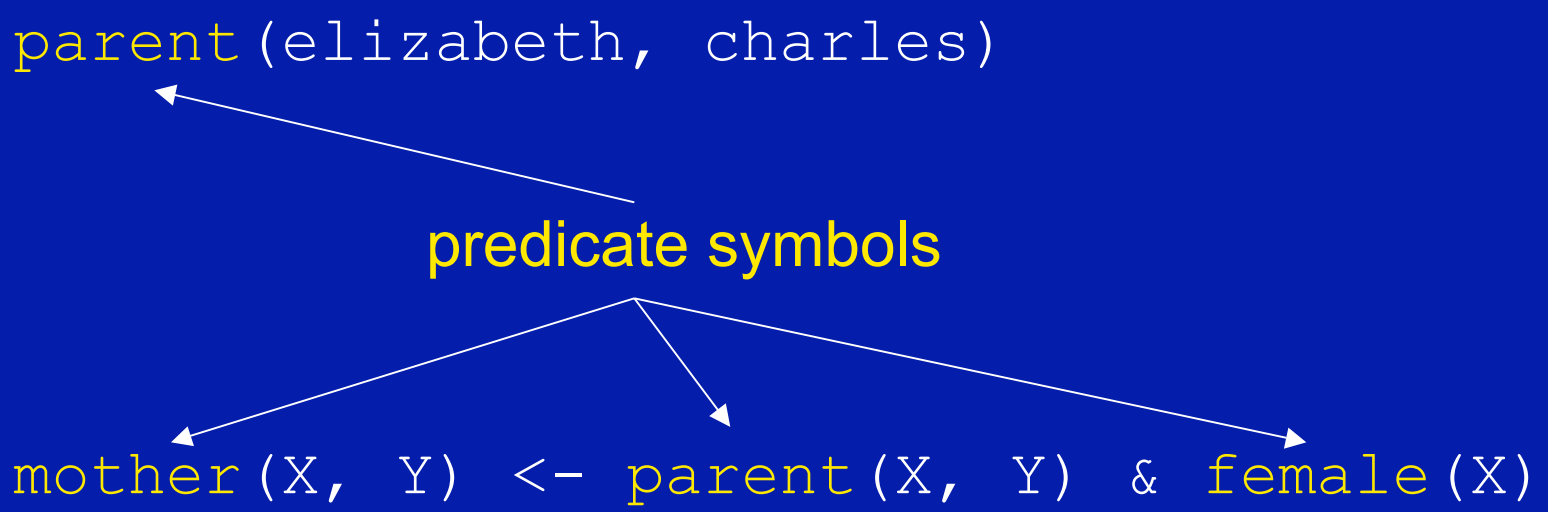
Dissecting the definite clause



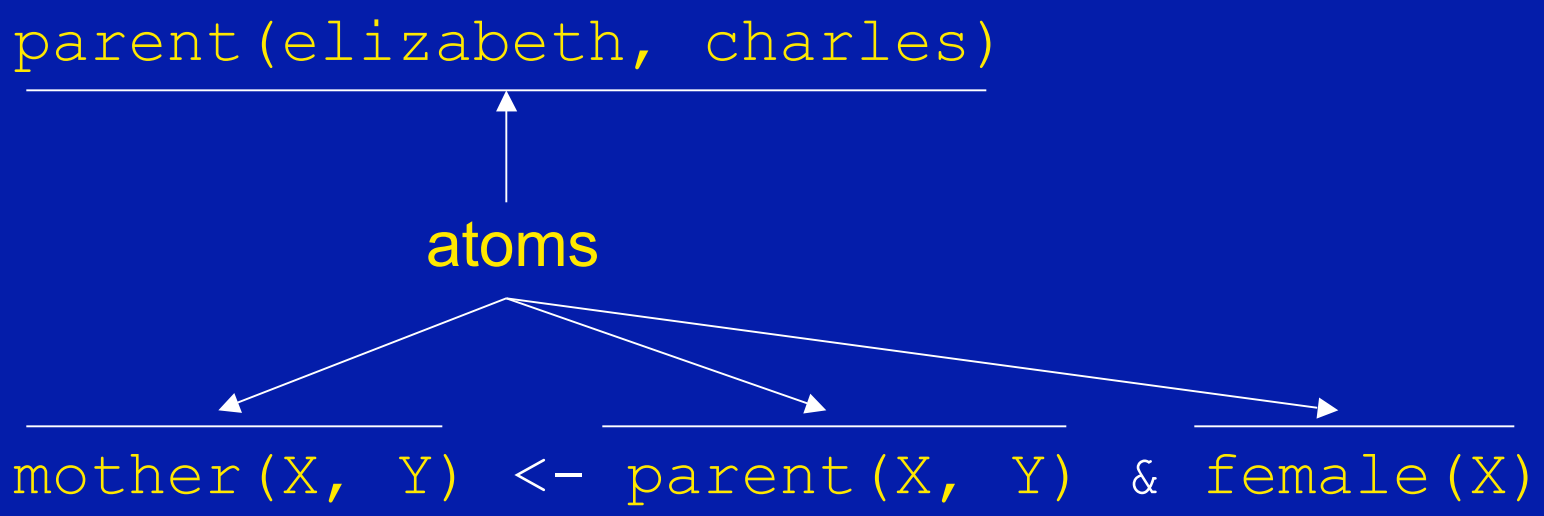
Dissecting the definite clause



Dissecting the definite clause

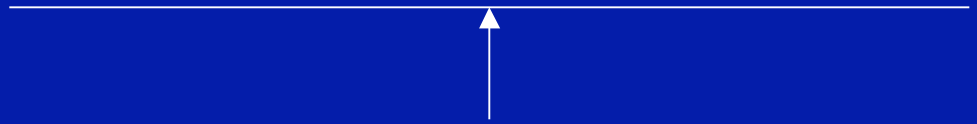


Dissecting the definite clause



Dissecting the definite clause

```
parent(elizabeth, charles)
```



fact (a single atom)

```
mother(X, Y) <- parent(X, Y) & female(X)
```

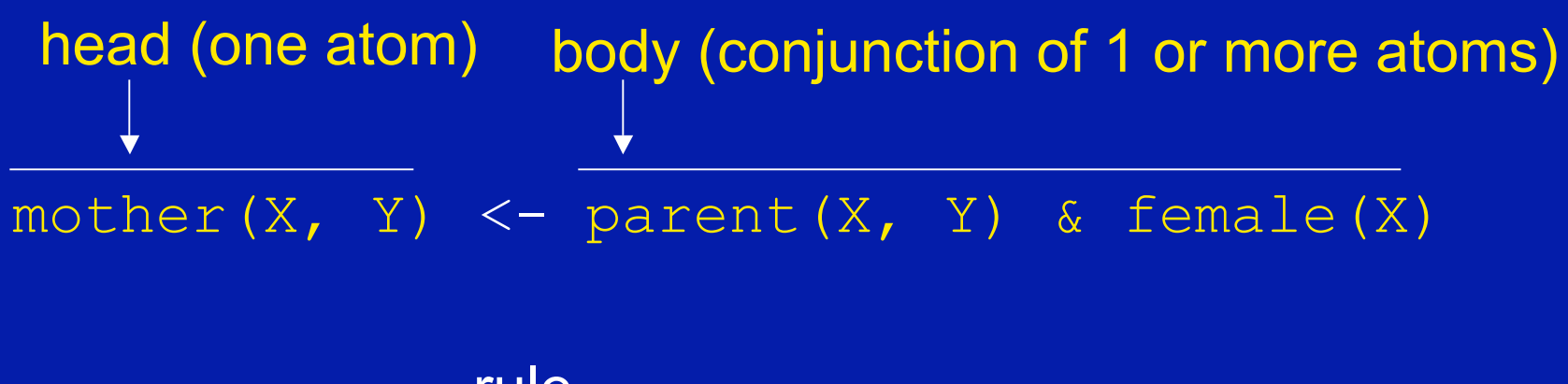
Dissecting the definite clause

```
parent(elizabeth, charles)
```

```
mother(X, Y) <- parent(X, Y) & female(X)
```

Dissecting the definite clause

```
parent(elizabeth, charles)
```



Semantics in the RRS

Informally:

A semantics tells YOU how symbols in the language correspond to things in the task domain

A semantics tells YOU how to use the correspondences to interpret sentences in the language

The correspondences between symbols in the language and objects in the domain define an interpretation for the language

Semantics in the RRS

Formally:

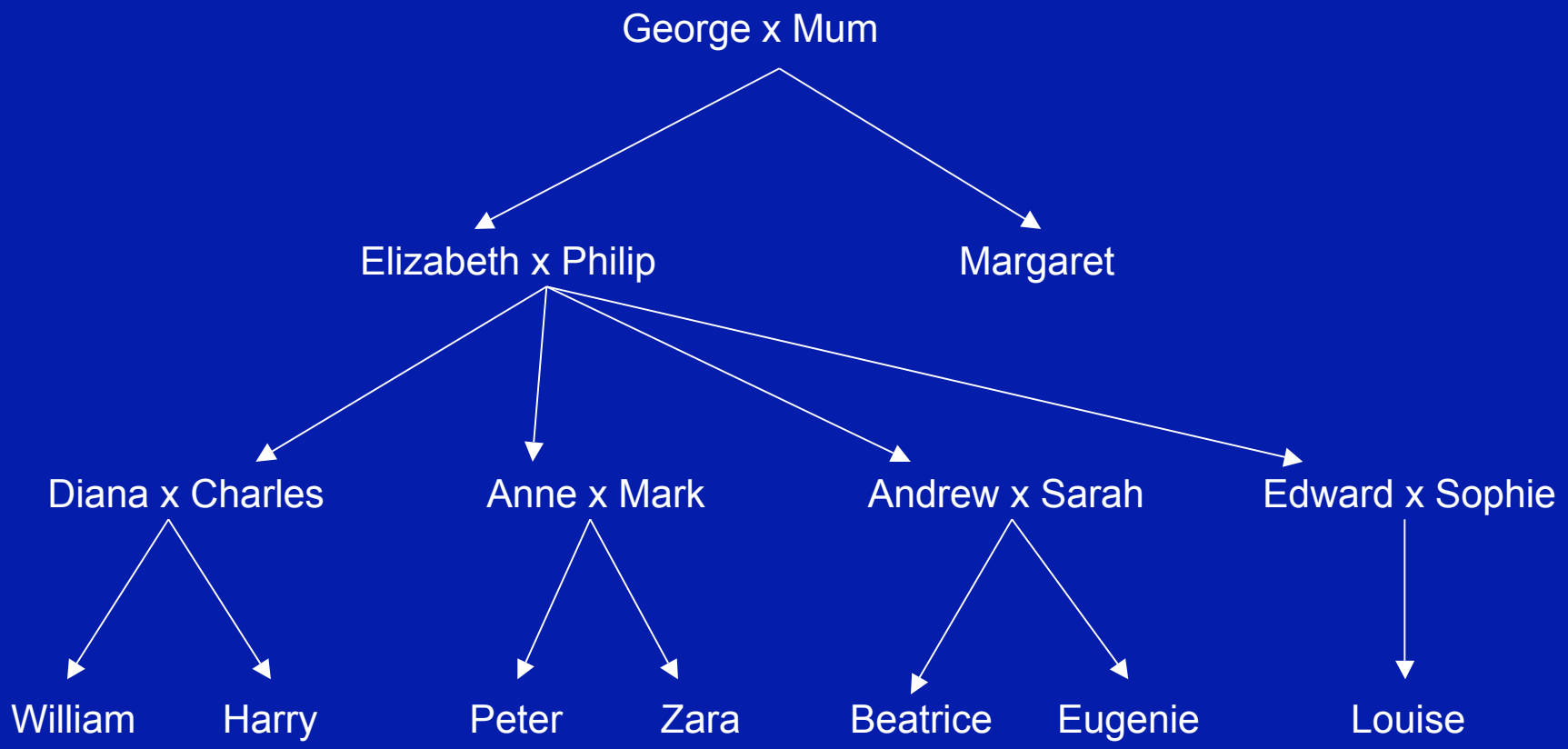
An interpretation is a triple $I = (D, \phi, \pi)$ where

D, the domain, is a nonempty set whose elements are individuals

ϕ is a mapping that assigns to each constant an element of D

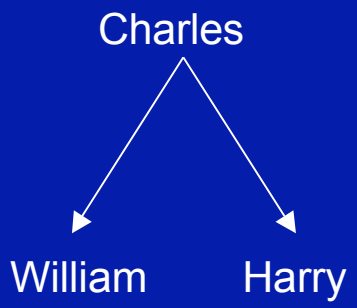
π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

Here's an example



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

Here's an example



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

A Sample Interpretation

D = {  ,  ,  }

constants = charles, william, harry

predicate symbols = male (unary)
female (unary)
parent (binary)

What about ϕ and π ?

A Sample Interpretation

ϕ is a mapping that assigns to each constant an element of D

ϕ(charles) = 

ϕ(william) = 

ϕ(harry) = 

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(male)() = *TRUE*

π(male)() = *TRUE*

π(male)() = *TRUE*

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(female)() = *FALSE*

π(female)() = *FALSE*

π(female)() = *FALSE*

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(parent)( , ) = *FALSE*

π(parent)( , ) = *TRUE*

π(parent)( , ) = *TRUE*

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(parent)( , ) = *FALSE*

π(parent)( , ) = *FALSE*

π(parent)( , ) = *FALSE*

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(parent)( , ) = *FALSE*

π(parent)( , ) = *FALSE*

π(parent)( , ) = *FALSE*

A Sample Interpretation

```
male(charles).
male(william).
male(harry).
parent(charles, william).
parent(charles, harry).
```

Wow. That was a lot of semantics to work out just to get to here. What if we want to add more individuals or more relationships?

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(brother)( , ) = *FALSE*

π(brother)( , ) = *FALSE*

π(brother)( , ) = *FALSE*

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(brother)( , ) = *FALSE*

π(brother)( , ) = *FALSE*

π(brother)( , ) = *TRUE*

A Sample Interpretation

π is a mapping that assigns to each n-ary predicate symbol a function from D^n into {*TRUE*, *FALSE*}

π(brother)( , ) = *FALSE*

π(brother)( , ) = *TRUE*

π(brother)( , ) = *FALSE*

A Sample Interpretation

```
male(charles).
male(william).
male(harry).
parent(charles, william).
parent(charles, harry).

brother(X, Y) <- male(X) &
                 parent(Z, X) &
                 parent(Z, Y).
```

Let's test it out and see how it works.

Let's talk about homework

The webpage at www.cs.ubc.ca/~eiselt/cs322 will have your homework assignment by 6:00pm today. Look at the entry for Friday, September 17.

Collaboration policy

Turning it in