Solving a Logic Puzzle

CS3100 Fall 2019

Review

Previously

- Logical Foundations
  - First-order logic
  - Definite clauses & Programs
  - SLD resolution

This lecture

- Solving the Zerbra puzzle

Zebra Puzzle

1. There are five houses.
2. The Englishman lives in the red house.
3. The Spaniard owns the dog.
4. Coffee is drunk in the green house.
5. The Ukrainian drinks tea.
6. The green house is immediately to the right of the ivory house.
7. The Old Gold smoker owns snails.
8. Kools are smoked in the yellow house.
9. Milk is drunk in the middle house.
10. The Norwegian lives in the first house.
11. The man who smokes Chesterfields lives in the house next to the man with the fox.
12. Kools are smoked in the house next to the house where the horse is kept.
13. The Lucky Strike smoker drinks orange juice.
14. The Japanese smokes Parliaments.
15. The Norwegian lives next to the blue house.

Now, who drinks water? Who owns the zebra?

Zebra Puzzle

1. There are **five** houses.
2. The **Englishman** lives in the **red** house.
3. The Spaniard owns the **dog**.
4. **Coffee** is drunk in the green house.
5. The Ukrainian drinks tea.
6. The green house is immediately to the **right of** the ivory house.
7. The **Old Gold** smoker owns snails.

8. Kools are smoked in the yellow house.
9. Milk is drunk in the **middle** house.
10. The Norwegian lives in the **first** house.
11. The man who smokes Chesterfields lives in the house next to the man with the fox.
12. Kools are smoked in the house **next to** the house where the horse is kept.
13. The Lucky Strike smoker drinks orange juice.
14. The Japanese smokes Parliaments.
15. The Norwegian lives **next to** the blue house.

Now, who drinks water? Who owns the zebra?

Representing a house

A house has 5 characteristics:

1. Nationality
2. Pet
3. Smoke
4. Drink
5. Colour

We can define a function `house(Nationality,Pet,Smoke,Drink,Colour)` to represent this.

We represent the row of houses as a 5-tuple `(H1,H2,H3,H4,H5)`.

Quiz

What sort of a term is `house(Nationality,Pet,Smoke,Drink,Colour)` ?

1. Number
2. Compound Term
3. Variable
4. Constant

Quiz

What sort of a term is `house(Nationality,Pet,Smoke,Drink,Colour)` ?

1. Number
2. Compound Term ✓
3. Variable
4. Constant

Quiz

What sort of a term is `(Nationality,Pet,Smoke,Drink,Colour)` ?

1. Number
2. Compound Term

3. Variable
4. Constant

Quiz

What sort of a term is `(Nationality,Pet,Smoke,Drink,Colour)` ?

1. Number
2. Compound Term ✓
3. Variable
4. Constant

A tuple is a compond term with no explicit function symbol.

Define the house-existence facts

- We want to convey this sort of house exists in this position.
  - So define a predicate `exists` that captures this fact.

In [1]:

```
exists(A,(A,_,_,_,_)).
exists(A,(_,A,_,_,_)).
exists(A,(_,_,A,_,_)).
exists(A,(_,_,_,A,_)).
exists(A,(_,_,_,_,A)).
```

```
Added 5 rule(s).
```

In [2]:

```
?- exists(h1,(h1,h2,h3,h4,h5)).
```

```
true.
```

In [3]:

```
?- exists(h2,(h1,h2,h3,h4,h5)).
```

```
true.
```

In [4]:

```
?- exists(h6,(h1,h2,h3,h4,h5)).
```

```
false.
```

Quiz

Which of these queries returns true?

1. exists(dog,(fly,spider,bird,cat,dog)).
2. exists(dog,(fly,spider,bird,cat)).

3. exists(dog).
4. exists(house(english,red), (house(spanish,green), house(french,orange), house(dutch,yellow), house(german,blue), house(english,_)).

Quiz

Which of these queries returns true?

1. exists(dog,(fly,spider,bird,cat,dog)). **yes**
2. exists(dog,(fly,spider,bird,cat)). **no**
3. exists(dog). **no**
4. exists(house(english,red), (house(spanish,green), house(french,orange), house(dutch,yellow), house(german,blue), house(english,_)). **yes**

rightOf

6. The green house is immediately to the right of the ivory house.

In [5]:

```
rightOf(A,B,(B,A,_,_,_)).
rightOf(A,B,(_,B,A,_,_)).
rightOf(A,B,(_,_,B,A,_)).
rightOf(A,B,(_,_,_,B,A)).
```

```
Added 4 rule(s).
```

middle and first house

9. Milk is drunk in the middle house.

In [6]:

```
middleHouse(A,(_,_,A,_,_)).
```

```
Added 1 rule(s).
```

10. The Norwegian lives in the first house.

In [7]:

```
firstHouse(A,(A,_,_,_,_)).
```

```
Added 1 rule(s).
```

nextTo

```
   12. Kools are smoked in the house next to the house where the horse is kep
   t.
   15. The Norwegian lives next to the blue house.
```

In [8]:

```
nextTo(A,B,H) :- rightOf(A,B,H).
nextTo(A,B,(A,B,_,_,_)).
nextTo(A,B,(_,A,B,_,_)).
nextTo(A,B,(_,_,A,B,_)).
nextTo(A,B,(_,_,_,A,B)).
```

```
Added 5 rule(s).
```

Express the puzzle as a Query

In [9]:

```
?- exists(house(british,_,_,_,red),Houses),
exists(house(spanish,dog,_,_,_),Houses),
exists(house(_,_,_,coffee,green),Houses),
exists(house(ukranian,_,_,tea,_),Houses),
rightOf(house(_,_,_,_,green),house(_,_,_,_,ivory),Houses),
exists(house(_,snail,oldgold,_,_),Houses),
exists(house(_,_,kools,_,yellow),Houses),
middleHouse(house(_,_,_,milk,_),Houses),
firstHouse(house(norwegian,_,_,_,_),Houses),
nextTo(house(_,_,chesterfields,_,_),house(_,fox,_,_,_),Houses),
nextTo(house(_,_,kools,_,_),house(_,horse,_,_,_),Houses),
exists(house(_,_,luckystrike,orangejuice,_),Houses),
exists(house(japanese,_,parliaments,_,_),Houses),
nextTo(house(norwegian,_,_,_,_),house(_,_,_,_,blue),Houses),
exists(house(WaterDrinker,_,_,water,_),Houses),
exists(house(ZebraOwner,zebra,_,_,_),Houses).
```

```
Houses = ,(house(norwegian, fox, kools, water, yellow), ,(house(ukrani
an, horse, chesterfields, tea, blue), ,(house(british, snail, oldgold,
milk, red), ,(house(spanish, dog, luckystrike, orangejuice, ivory), ho
use(japanese, zebra, parliaments, coffee, green))))), WaterDrinker = n
orwegian, ZebraOwner = japanese .
```

In [10]:

```
?- exists(house(british,_,_,_,red),Houses),
exists(house(spanish,dog,_,_,_),Houses) {40}.
```

```
Houses = ,(house(british, _G1062, _G1063, _G1064, red), ,(house(spanis
h, dog, _G1072, _G1073, _G1074), ,(_G1094, ,(_G1097, _G1098)))) ;
Houses = ,(house(british, _G1062, _G1063, _G1064, red), ,(_G1091, ,(ho
use(spanish, dog, _G1072, _G1073, _G1074), ,(_G1097, _G1098)))) ;
Houses = ,(house(british, _G1062, _G1063, _G1064, red), ,(_G1091, ,(_G
1094, ,(house(spanish, dog, _G1072, _G1073, _G1074), _G1098)))) ;
Houses = ,(house(british, _G1062, _G1063, _G1064, red), ,(_G1091, ,(_G
1094, ,(_G1097, house(spanish, dog, _G1072, _G1073, _G1074))))) ;
Houses = ,(house(spanish, dog, _G1072, _G1073, _G1074), ,(house(britis
h, _G1062, _G1063, _G1064, red), ,(_G1094, ,(_G1097, _G1098)))) ;
Houses = ,(_G1088, ,(house(british, _G1062, _G1063, _G1064, red), ,(ho
use(spanish, dog, _G1072, _G1073, _G1074), ,(_G1097, _G1098)))) ;
Houses = ,(_G1088, ,(house(british, _G1062, _G1063, _G1064, red), ,(_G
1094, ,(house(spanish, dog, _G1072, _G1073, _G1074), _G1098)))) ;
Houses = ,(_G1088, ,(house(british, _G1062, _G1063, _G1064, red), ,(_G
1094, ,(_G1097, house(spanish, dog, _G1072, _G1073, _G1074))))) ;
Houses = ,(house(spanish, dog, _G1072, _G1073, _G1074), ,(_G1091, ,(ho
use(british, _G1062, _G1063, _G1064, red), ,(_G1097, _G1098)))) ;
Houses = ,(_G1088, ,(house(spanish, dog, _G1072, _G1073, _G1074), ,(ho
use(british, _G1062, _G1063, _G1064, red), ,(_G1097, _G1098)))) ;
Houses = ,(_G1088, ,(_G1091, ,(house(british, _G1062, _G1063, _G1064,
red), ,(house(spanish, dog, _G1072, _G1073, _G1074), _G1098)))) ;
Houses = ,(_G1088, ,(_G1091, ,(house(british, _G1062, _G1063, _G1064,
red), ,(_G1097, house(spanish, dog, _G1072, _G1073, _G1074))))) ;
Houses = ,(house(spanish, dog, _G1072, _G1073, _G1074), ,(_G1091, ,(_G
1094, ,(house(british, _G1062, _G1063, _G1064, red), _G1098)))) ;
Houses = ,(_G1088, ,(house(spanish, dog, _G1072, _G1073, _G1074), ,(_G
1094, ,(house(british, _G1062, _G1063, _G1064, red), _G1098)))) ;
Houses = ,(_G1088, ,(_G1091, ,(house(spanish, dog, _G1072, _G1073, _G1
074), ,(house(british, _G1062, _G1063, _G1064, red), _G1098)))) ;
Houses = ,(_G1088, ,(_G1091, ,(_G1094, ,(house(british, _G1062, _G106
3, _G1064, red), house(spanish, dog, _G1072, _G1073, _G1074))))) ;
Houses = ,(house(spanish, dog, _G1072, _G1073, _G1074), ,(_G1091, ,(_G
1094, ,(_G1097, house(british, _G1062, _G1063, _G1064, red))))) ;
Houses = ,(_G1088, ,(house(spanish, dog, _G1072, _G1073, _G1074), ,(_G
1094, ,(_G1097, house(british, _G1062, _G1063, _G1064, red))))) ;
Houses = ,(_G1088, ,(_G1091, ,(house(spanish, dog, _G1072, _G1073, _G1
074), ,(_G1097, house(british, _G1062, _G1063, _G1064, red))))) ;
Houses = ,(_G1088, ,(_G1091, ,(_G1094, ,(house(spanish, dog, _G1072, _
G1073, _G1074), house(british, _G1062, _G1063, _G1064, red))))) .
```

Express the puzzle as a rule

In [11]:

```
puzzle(Houses) :- exists(house(british,_,_,_,red),Houses),
                  exists(house(spanish,dog,_,_,_),Houses),
                  exists(house(_,_,_,coffee,green),Houses),
                  exists(house(ukranian,_,_,tea,_),Houses),
                  rightOf(house(_,_,_,_,green),house(_,_,_,_,ivory),Houses),
                  exists(house(_,snail,oldgold,_,_),Houses),
                  exists(house(_,_,kools,_,yellow),Houses),
                  middleHouse(house(_,_,_,milk,_),Houses),
                  firstHouse(house(norwegian,_,_,_,_),Houses),
                  nextTo(house(_,_,chesterfields,_,_),house(_,fox,_,_,_),Houses),
                  nextTo(house(_,_,kools,_,_),house(_,horse,_,_,_),Houses),
                  exists(house(_,_,luckystrike,orangejuice,_),Houses),
                  exists(house(japanese,_,parliaments,_,_),Houses),
                  nextTo(house(norwegian,_,_,_,_),house(_,_,_,_,blue),Houses).
```

```
Added 1 rule(s).
```

Express the puzzle as a rule

Who is the zebra owner?

In [12]:

```
?- puzzle(Houses), exists(house(ZebraOwner,zebra,_,_,_),Houses).
```

```
Houses = ,(house(norwegian, fox, kools, _G1081, yellow), ,(house(ukran
ian, horse, chesterfields, tea, blue), ,(house(british, snail, oldgol
d, milk, red), ,(house(spanish, dog, luckystrike, orangejuice, ivory),
house(japanese, zebra, parliaments, coffee, green))))), ZebraOwner = j
apanese .
```

Who is the water drinker?

In [13]:

```
?- puzzle(Houses), exists(house(WaterDrinker,_,_,water,_),Houses).
```

```
Houses = ,(house(norwegian, fox, kools, water, yellow), ,(house(ukrani
an, horse, chesterfields, tea, blue), ,(house(british, snail, oldgold,
milk, red), ,(house(spanish, dog, luckystrike, orangejuice, ivory), ho
use(japanese, _G1055, parliaments, coffee, green))))), WaterDrinker =
norwegian .
```

Fin.