Parsing in Prolog

- **Rules for parsing English**

```
sentence(X) :- append(Y, Z, X), noun_phrase(Y), verb_phrase(Z).
noun_phrase(X) :- append(Y, Z, X), determiner(Y), noun(Z).
verb_phrase(X) :- append(Y, Z, X), verb(Y), noun_phrase(Z).
verb_phrase(X) :- verb(X).
```

- **Facts that define some words.**

```
sentence(X) :- append(Y, Z, X), noun_phrase(Y), verb_phrase(Z).
noun_phrase(X) :- append(Y, Z, X), determiner(Y), noun(Z).
verb_phrase(X) :- append(Y, Z, X), verb(Y), noun_phrase(Z).
verb_phrase(X) :- verb(X).
```

- **Parsing.**

```
?- sentence([the,man]).
No
?- sentence([the,man,eats]).
Yes
?- sentence([the,man,eats,the,apple]).
Yes
```

- **How to get the parse tree?**
  - Give the predicates a second argument that is the matching parse tree.

```
sentence(X, sentence(NP, VP)) :-
   append(Y, Z, X), noun_phrase(Y, NP), verb_phrase(Z, VP).
noun_phrase(X, noun_phrase(determiner(Y), noun(Z))) :-
   append(Y, Z, X), determiner(Y), noun(Z).
verb_phrase(X, verb_phrase(verb(Y), NP)) :-
   append(Y, Z, X), verb(Y), noun_phrase(Z, NP).
verb_phrase(X, verb_phrase(verb(X))) :- verb(X).
```

  - Compound term also called structure.
    - Used as argument.
  - We can now get the parse tree:

```
?- sentence([the,man], X).
No
?- sentence([the,man,eats], X).
X = sentence(noun_phrase(determiner([the]), noun([man])),
             verb_phrase(verb([eats])))
Yes
?- sentence([the,man,eats,the,apple], X).
X = sentence(noun_phrase(determiner([the]), noun([man])),
             verb_phrase(verb([eats]),
                         noun_phrase(determiner([the]),
                                     noun([apple]))))
Yes
```

Solving Einstein's riddle in Prolog

- **The original puzzle (Life International Magazine, 1962)**
  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?

In the interest of clarity, it must be added that each of the five houses is painted a different color, and their inhabitants are of different national extractions, own different pets, drink different beverages and smoke different brands of American cigarettes. One other thing: In statement 6, right means your right.

- **Smaller version:**
  - Three mice: mickey, minnie, mighty.
  - Three cheeses: gouda, emmental, brie.
  - Three TV shows: er, seinfeld, simpsons.
  - Three mouse holes: left, middle, right.
  - Rules:
    - Mickey Mouse loves Gouda.
    - Might Mouse's favorite TV show is ER.
    - The mouse that lives in the left hole never misses an episode of Seinfeld.
    - Mickey Mouse and Mighty Mouse are not next-door neighbors.
    - The Simpsons fan does not live on the left of the brie lover.
  - Solution
    - Conclusion: einstein(Mice) :-
    - Conditions:
      - Mice = [_, _, _], % There are three mice.
      - member([mickey, gouda, _, H1], Mice), % Rule 1.
      - member([mighty, _, er, H2], Mice), % Rule 2.
      - member([_, _, seinfeld, left], Mice), % Rule 3.
      - onebetween(H1, H2), % Rule 4.
      - notleft(H3, H4), % Rule 5.
      - member([_, _, simpsons, H3], Mice)
      - member([_, brie, _, H4], Mice),

        - member([minnie, _, _, _], Mice).
    - Missing facts:
        - onebetween(left, right).
        - onebetween(right, left).
        - notleft(X, X).
        - notleft(right, middle).
        - notleft(right, left).
        - notleft(middle, left).
- We can now query:

    ?- einstein(Mice).

    Mice = [[mickey, gouda, seinfeld, left],
        [minnie, brie, simpsons, middle],
        [mighty, emmental, er, right]]

    Yes