Contents

Prolog Theory

Unification

- Prolog uniques `woman(X)` with `woman(mia)`, thereby instantiating the variable `X` with the atom `mia`
- Two terms unify if they are the same term or if they contain variables that can be unfirmly instantiated such that the resulting terms are equal. For example, `mia` and `mia` unify, as do `42` and `42`, bu not `vincent` and `mia`
- After terms are unified, they are considered equal from there on

When do things unify?

Constants

Not that if `T~1~` and `T~2~` are constants, they unify if they are the same atom or number

Variables

If `T~1~` is a variable and `T~2~` as any atom, `T~1~` is instantiated to `T~2~` (and vice versa)

```
?- X=mia, X=vincent
no
```

Complex Terms

- Complex terms unify if
  - They have the same functor + arity
  - The arguments unify
  - The variable instantiation are compatible

```
?- k(s(g), Y) = k(X, t(k))
Y=t(k)
X=s(g)

?- loves(X, X) = loves(marsellus, mia)
no

?- father(X) = X
X=father(father(father(...)
yes
```

- The last example doesn't work in standard unification, as it will normally check whether the variableitself occurs within the term

```
?- unify_with_occurs_check(father(X), X)
no
```

Proof Search

- Search for whether a statement is true or false is done by creating a tree of the possible values of the variables in the statement and traversing it until we reach a state where everything unifies and all the varables are instantiated
- If there is no such state, the statement is false
- Otherwise, all the possible values of variables in the statement are returned