

Facts

First steps in Prolog

To read in a file:

```
?- consult('programfile.pl').
```

To override defitions:

```
?- reconsult('programfile.pl').
```

1

Definition:

- a.
- b.
- c.

Ask Prolog:

```
?- a.  
yes.
```

```
?- b.  
yes.
```

2

Predicates

Program:

```
is_town(london).  
is_town(berlin).  
is_town(paris).
```

Ask Prolog:

```
?- is_town(london).  
yes.
```

```
?- is_town(_).  
yes.
```

Reconsult without is_town:

```
ERROR: (user://2:43):  
    Undefined procedure: is_town/1
```

Bottom Line: `?- is_town(_).`

asks

```
is  $\exists x$  : town(x) true?
```

3

Remarks

Closed World Assumption: makes the *closed world assumption*. The prolog clauses constitute the "world". Can `is_town(_)` be derived from clauses in prolog program?

Note: `"_"` constitutes anonymous variable — matched by any object, london, berlin etc.

4

Named Variables

Variable naming: through uppercase words

```
?- consult('towns.pl').
```

```
?- is_town(X).
```

```
X = london
```

Note: <Return> ends the question. Use ";" to get more replies

```
X = berlin;
```

```
X = paris;
```

```
no.
```

Remark: so, `is_town(X)` is similar to `is_town(_)`, but `X` is a *named* variable, it allows to access the *instantiation* of `is_town`.

Intermediate Summary

We have learnt

1. loading prolog programs with `(re)consult`
2. facts
3. questions
4. anonymous variables
5. named variables
6. closed world assumption