Prolog 5
Some Additional Features

Fariba Sadri

Commenting

Comments have no effect on the execution of a program, but they are very useful for making programs more comprehensible.

Two forms of comments are allowed:

1. The character ‘%’ followed by any sequence of characters up to the end of the line.
2. The symbol ‘/*’ followed by any sequence of characters (including newlines) up to the symbol ‘*/’.

- listing. Lists all user definitions currently loaded.
- listing(p). Lists the definition of a given predicate p.

In Linux to be able to use listing you must use consult(<program>), rather than compile(<program>).

I/O

- write(T) writes a term T
- read(X) unifies with X the next term read (the term must be followed by “.” and carriage return if written from keyboad)
- nl writes a new line character
- tab(N) produces N tab spaces

tab may not work in some Prologs. But we can program it easily.

```
tab(0).
tab(N) :- N>0, write(' '), M is N-1, tab(M).
```

Example

A Prolg prgram that writes the content of a list, one element per line:

```
writelist([]).
writelist([X|L]):- write(X), nl, writelist(L).
| ?- writelist([this,is,great]).
this
is
great
yes
| ?-
```

Another Example

```
check_squares:-
        nl,
        writeMessage(['do you want to play', '? ']), read(X),
        (X=no -> write(goodbye);
        writeMessage(['give me an integer']),
        read(Y),
        askabout(Y)).
```

Example cntd.

```
askabout(X):-
         writeMessage(['what is the square of',
        X, '?']),
        read(Y),
        (Y is X*X -> write(correct), nl ;
        write(wrong),nl),
        check_squares.
```

Example cntd.

```
writeMessage([]).

writeMessage([X|L]):-
        write(X),
        write(' '),
        writeMessage(L),
        nl.
```

Inserting clauses at the terminal

```
| ?- [user].
% consulting user …
| p(X):-q(X).
| p(X):-r(X).
| q(1).
| r(2).
| end_of_file.
% consulted user in module user, 0 msec 560 bytes
yes
| ?-
```

Instead of

```
| end_of_file.
```

You can type

```
| <Ctrl> z
```

How do you save the terminal entered program to a file?

If you wish to save the program you have entered at the terminal as a text file, you can use the following steps:

- After the “[user].” part you have to declare all your predicates *dynamic*.
- Then enter the program, ending with end_of_file or <Ctrl>z.
- Then use *tell*, *listing*, *told*.

Example

```
| ?- [user].
| dynamic(plus_one/2).
| plus_one( X, Y ) :- Y is X + 1.
| end_of_file. (or '<Ctrl>-Z').
| ?- tell(yourFilename).
| ?- listing.
| ?- told.
yes
```

In windows use the file -> Working directory menu to choose directory where the file will go.

In Linux give the full file name.

Reading and writing from files

Not needed in this course.

But Prolog has a variety of predicates for these tasks, such as:

- see (F) opens file F as the input file
- seen closes F as input file
- tell(F) opens file F as the output file
- told closes F as output file