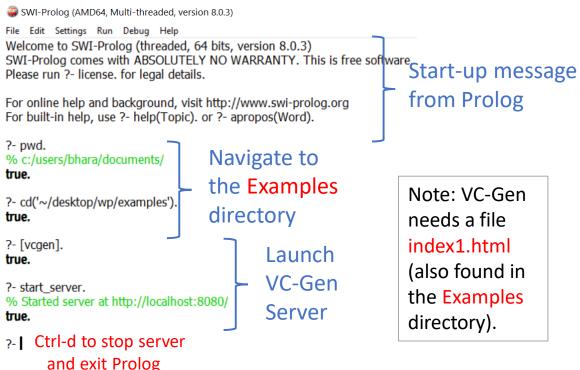# Installing and Running VC-Gen Server

1. Get SWI-Prolog
   a. Windows: download from https://www.swi-prolog.org/
   b. Linux: sudo apt-get install swi-prolog
   c. Mac: brew install … *sorry unfamiliar with details* …
2. Visit http://www.cse.buffalo.edu/LRG/WP
   a. Download Examples directory
   b. It also contains vcgen.pl, the code for VC-Gen.

```
bharat@bharat-VB:~$ pwd
/home/bharat
bharat@bharat-VB:~$ cd WP/Examples
bharat@bharat-VB:~/WP/Examples$ prolog
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [vcgen].
true.

?- start_server.
% Started server at http://localhost:8080/
true.
```

3. Linux:
   ➢ prolog
   … *startup msg* …
   ?- [vcgen].
   ?- start_server

3. Windows: Launch Prolog and then launch VC-Gen:

SWI-Prolog (AMD64, Multi-threaded, version 8.0.3)

File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
```
→ Start-up message from Prolog

```
?- pwd.
% c:/users/bhara/documents/
true.

?- cd('~/desktop/wp/examples').
true.
```
→ Navigate to the Examples directory

```
?- [vcgen].
true.
```
→ Launch VC-Gen Server

```
?- start_server.
% Started server at http://localhost:8080/
true.
```

?- | Ctrl-d to stop server and exit Prolog

Note: VC-Gen needs a file index1.html (also found in the Examples directory).

4. Go to localhost:8080 on your browser.
   a. Upload a program from Examples directory.
   b. Generate VCs and copy over to Alt-Ergo window.
   c. Ask Alt-Ergo to validate.

localhost:8080/verify

VC Generator

Choose File   No file chosen   Upload Local File

Program

```
@requires n >= 1
@ensures 2*s = n*(n+1)
@program {
  @var i, n, s : int;
  s = 1;
  i = 1;
  @invariant 2*s = i*(i+1) && i=<n
  while (i =< n-1) {
    i = i+1;
    s = s+i;
  }
}
```

Verification Conditions

```
(* Verification Conditions for Alt-Ergo *)

logic j, n, s: int

goal intra_loop:
(i<=(n-1) and
(2*s)=(i*(i+1)) and
i<=n ->
  (2*(s+(i+1)))=((i+1)*((i+1)+1)) and
  (i+1)<=n)

goal exit:
((j>=n) and
```

Generate VCs   Download VCs