Hi Ruben,

Some of the core functionality you ask about is achieved via function calls within the Perl CGI scripts (e.g. wff checking), some involves calls to external standalone programs that are gated by function calls within the CGI (e.g. proof checking). Validity checking is also a hybrid case where the program forks, and the parent continues if nothing is returned by the child within a certain time.

Here's more detail, and I hope it's what you need:

Proof checking is gated via the &check_the_proof function as defined in the file logic_qm_common_subrs.pl, but the actual proof checking is handled by the compiled C program 'logic' which can also be run as a standalone. The path to the standalone is stored as $daemonC within the CGI scripts. Communication is via a Perl open2 command which opens up two-way communication to the logic program, like this:

```
open2('RD','WR',$daemonC);
```

Pre-processed content from the web form collected in the browser is sent via a sequence of print statements:

```
print WR …
```

and the response lines are collected in a read loop:

```
 while (<RD>){ … }
```

The browser input and output are pre- and post-processed by the main procedures in logic.pl. The web interface can be interacted with directly at http://logic.tamu.edu/cgi-.

Wff checking is handled within the Perl scripts by the &iswff function defined within lpwff-subrs.pl. It can be interacted with directly at http://logic.tamu.edu/cgi-.

Equivalence checking is handled by code that is within the Perl scripts, but using the fork method mentioned above. The function to call is &is_equivalent, which is defined in valid_subrs.pl. The code here forks because equivalency checking runs by testing for validity and timing out if it can't be ascertained. It can be interacted with directly at http://logic.tamu.edu/cgi-.

Countermodel checking is by &is_countermodel in qmmodel.pl. The web interface can be directly interacted with at http://logic.tamu.edu/cgi-.

Quizmaster is basically a front end to much of the same functionality, but it also has some additional routines to handle multiple choice and T/F questions. It's alive at http://logic.tamu.edu/cgi-.

One last caveat / mea culpa: Much of this code was written within the first year or two of the WWW. As such, nobody had even thought of CSS, let alone anything like a model-controller-viewer separation. This means that presentational elements (i.e. html) are all mixed up with core logical tasks. That the code still runs 20 years after it was written is something of a miracle, but it would not be written the same way these days.

Let me know if you have any other questions.