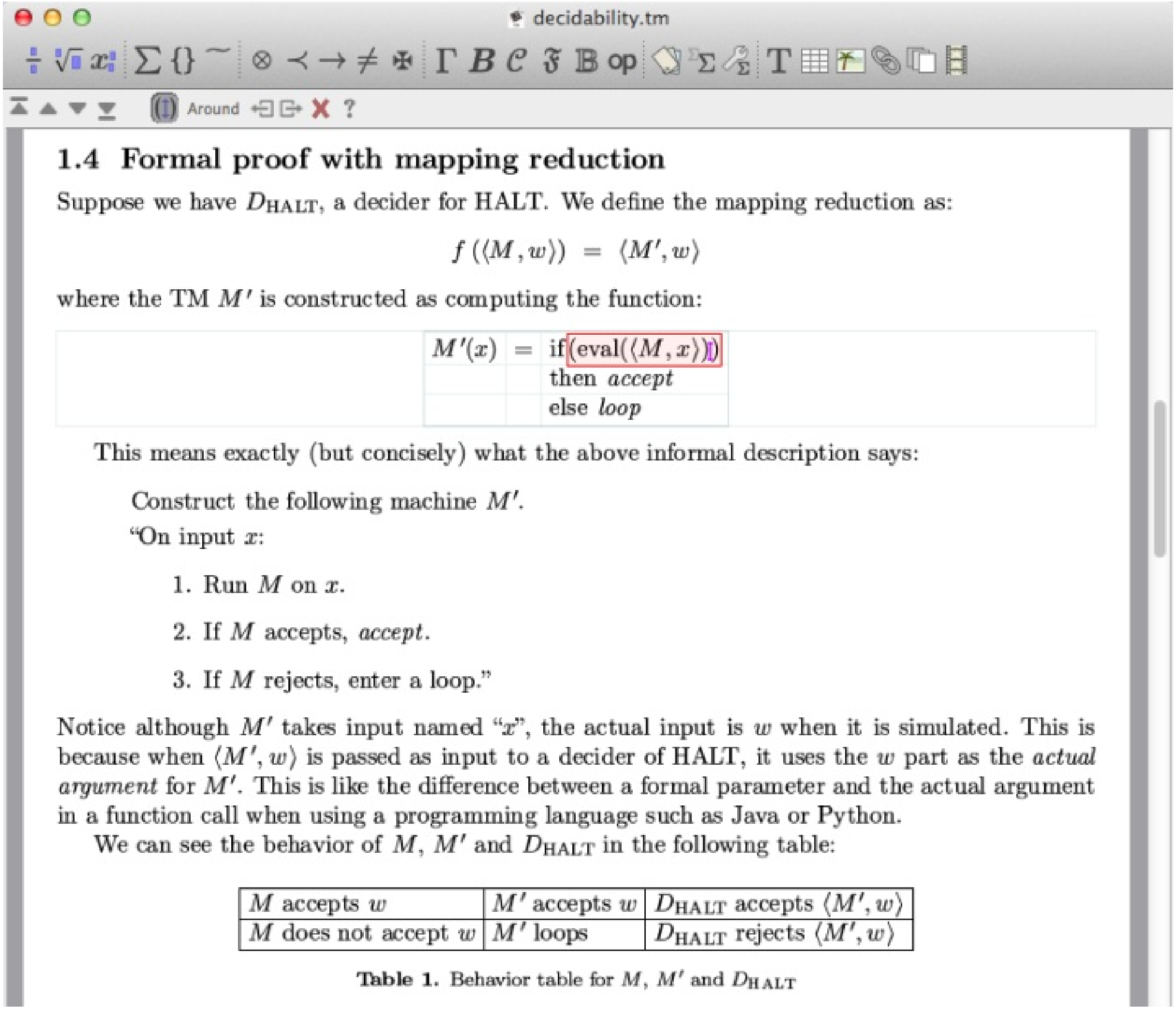
TeXmacs“”



1.4 Formal proof with mapping reduction

Suppose we have D_{HALT}, a decider for HALT. We define the mapping reduction as:

$$f(\langle M, w\rangle) = \langle M', w\rangle$$

where the TM M' is constructed as computing the function:

$$M'(x) = \text{if}(\mathrm{eval}(\langle M, x\rangle)) \text{ then } accept \text{ else } loop$$

This means exactly (but concisely) what the above informal description says:

Construct the following machine M'.
“On input x:

1. Run M on x.
2. If M accepts, *accept*.
3. If M rejects, enter a loop.”

Notice although M' takes input named “x”, the actual input is w when it is simulated. This is because when $\langle M', w\rangle$ is passed as input to a decider of HALT, it uses the w part as the *actual argument* for M'. This is like the difference between a formal parameter and the actual argument in a function call when using a programming language such as Java or Python.

We can see the behavior of M, M' and D_{HALT} in the following table:

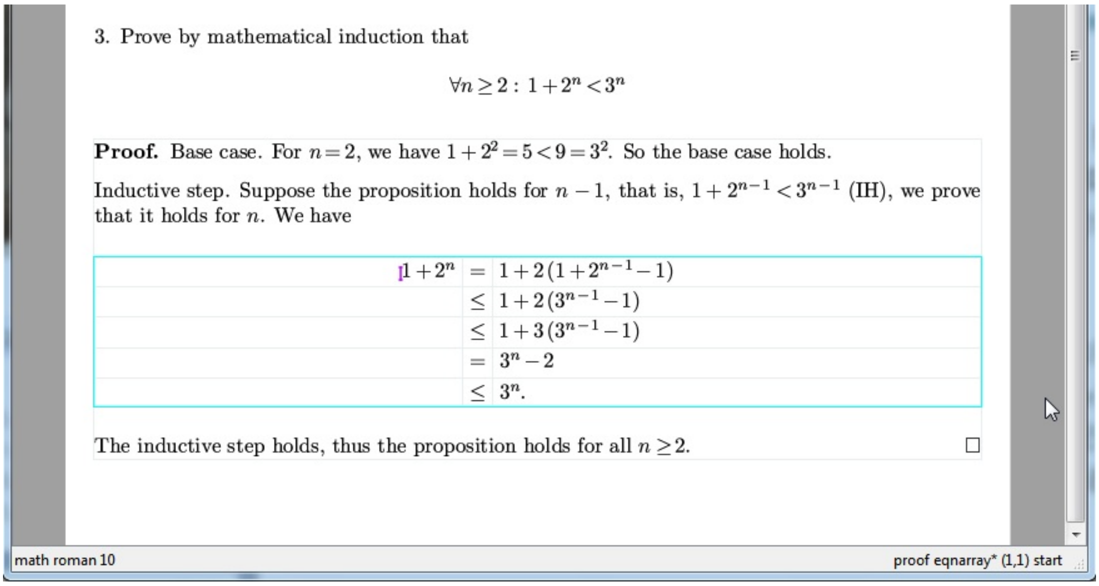
M accepts w	M' accepts w	D_{HALT} accepts $\langle M', w\rangle$
M does not accept w	M' loops	D_{HALT} rejects $\langle M', w\rangle$

Table 1. Behavior table for M, M' and D_{HALT}

TeXmacs TeXmacs

TeXmacs

- Lyx TeX “” TeX TeXmacs TeX TeX C++ TeX
- Word “” (WYSIWYG)Word “” Word TeX Lyx, Scientific Workspace
- TeX TeXmacs TeX
- TAB “” @ TAB > = TAB
- “”“proof eqnarry* (1,1) start” proof eqnarry (1,1) Ctrl-Backspace“” Ctrl-Backspace



- Ctrl-PgUp, Ctrl-PgDn “”
- MAXIMAOctave…… TeXmacs “TeX ” Scheme TeX

TeX TeXmacs “”

TeXmacs [TeXmacs]