Following an apparently highly arbitrary formal language, though later we will show how it represents an implmenetation of a generic theorem prover.

Let $\Sigma = \{0,1\}^*$ be a set bit vectors of varying length, called symbols.

Define a word $w \in \mathcal{W} \subset \Sigma^*$ to be a set[1] of symbols such that every two symbols in the word do not share a common prefix. For example, a word may contain the symbols 0010 and 1010 as they do not share a common prefix, but a string containing the symbols 0010 and 0011 is not considered a word on our language.

Define the set of sentences $\mathcal{S} = \mathcal{W}^*$ to be the set of all sequences of words. We are given a set of rules $\mathcal{R} \subset \mathcal{S} \times \mathcal{S}$ denoted by $s_1 \implies s_2$, with semantics such that if all words on s_1 can be proved, then all words on s_2 are proved. We are also given a set $\mathcal{F} \subset \mathcal{W}$ of facts. We construct the set of ground words $\mathcal{W}$ by:

$$\frac{w \in \mathcal{F}}{w \in \mathcal{G}} \tag{1}$$

means that every fact word is a ground word. The next rule

$$\frac{\begin{array}{c} w_1 \in \mathcal{G} \\ w_2 \in \mathcal{W} \\ w_1 \cup w_2 \in \mathcal{W} \end{array}}{w \in \mathcal{G}} \tag{2}$$

states that if a word is ground, then if its union (as a word is a set) with another word (valid word but not necessarily ground) yields another valid word, then this latter word is ground too. The last one is modus ponens:

$$\frac{\begin{array}{c} \{s_1, s_2\} \subset \mathcal{S} \\ s_1 \implies s_2 \\ \forall \omega \in s_1, \omega \in \mathcal{G} \\ w \in s_2 \end{array}}{w \in \mathcal{G}} \tag{3}$$

reads that if s_1, s_2 are sentences and we're given that $s_1 \implies s_2$, and all the words ω in s_1 are ground, then all words on s_2 are ground.

[1] Set as unordered indeed, unlike words in common languages where order of letters matters.