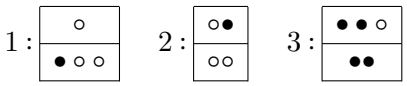
Instructions are labeled tales, such as:



Programs are label sequences, such as:

3 2 3 1

Figure 4.5: Post correspondence problem as a programming language

4.2 Post Correspondence

Turing machines allow us to formally state and prove that certain problems are *undecidable*, that is, that they cannot be solved by computers, no matter what programming language is being used. The canonical undecidable problem in the context of Turing machines is the *halting problem*: given a turing machine $\mathcal{M}$ and an input i, does $\mathcal{M}$ halt on i? Because of its reflective nature, the halting problem is somehow complicated to use in practice. In this section we discuss another canonical undecidable problem, the *Post correspondence problem*, showing how it can be reduced to a step-for-step equivalent rewrite theory. The reason for which we discuss this problem in this chapter dedicated to idealized programming languages, is because we can indeed regard it as an overly simplified programming language, whose basic instructions are two-dimentional tales and whose programs are sequences of such tales, as shown in Figure 4.5.

...