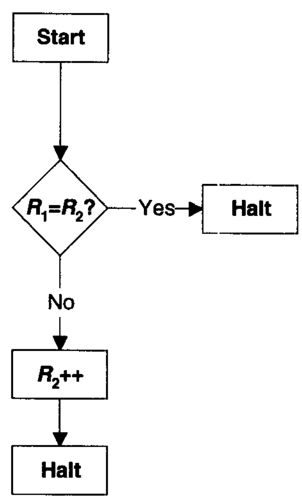
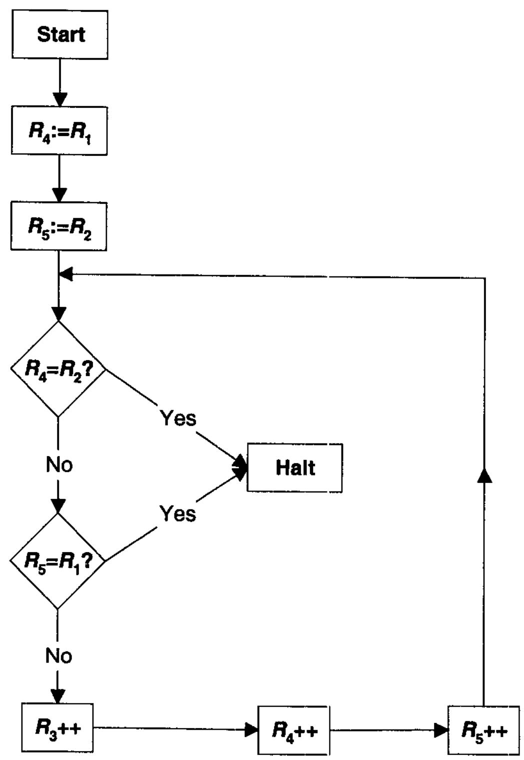
Solutions to Exercises for Chapter 5 of R. Gregory Taylor, *Models of Computation and Formal Languages* (Oxford University Press: New York, 1998)

Solutions to Exercises for § 5.1

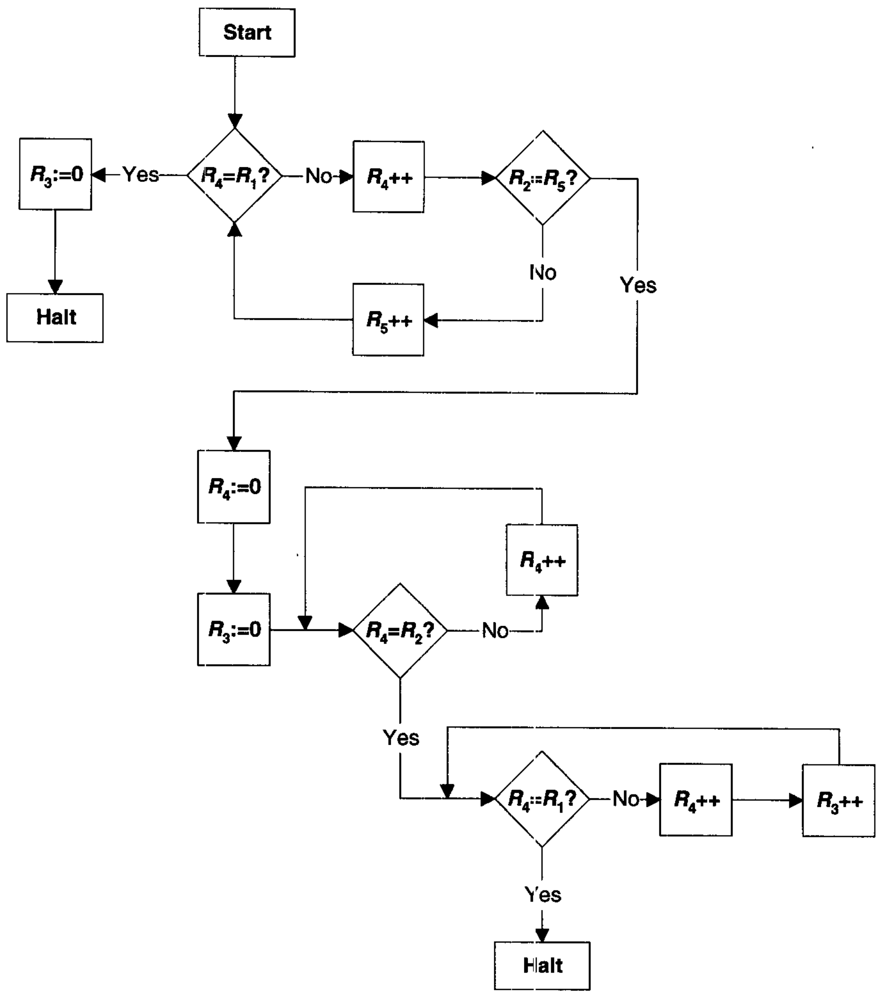
5.1.3. We make essential use of our assumption, in Definition 5.2, that all registers except argument registers contain 0 initially.



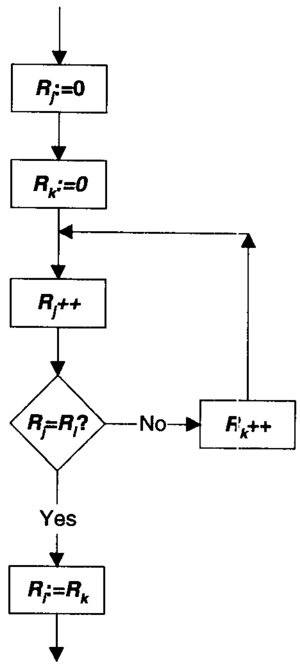
5.1.4 (a) This register machine computes the binary function $f(n,m) = |n-m|$.



(b) This register machine computes the binary monus function. Click twice on icon **Monus Machine** to observe its operation.



5.1.8.



Solutions to Exercises for § 5.4

5.4.2. Registers may be introduced by way of Read instructions, incrementation instructions, assignment statements, or goto instructions. The latter might introduce two registers simultaneously. For example, the goto instruction

$$R_9=R_{10}?\text{ goto } \mathcal{L}$$

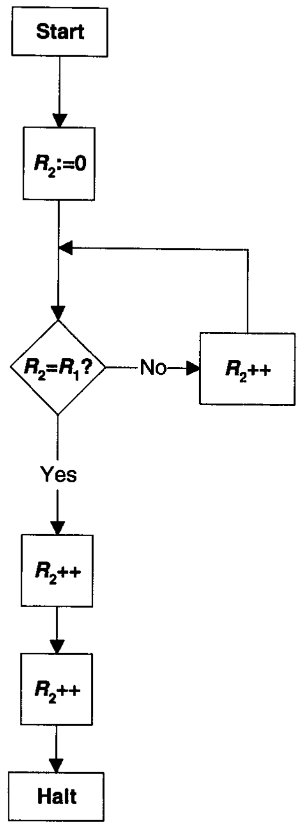
might be the first to mention either register R_9 or register R_{10}. Similarly, each of the other sorts of instructions introduces at most one or two new registers. It follows that the total number of registers used is bounded above by $2 \cdot f(n)$ and hence is $O(f(n))$.

5.4.3. The contents of a register may be the result of reading a natural number from *M*'s input tape. Any such number in no way depends upon n and hence is $O(1)$. Otherwise, a register's contents may increase as the result of repeated incrementation. But any such incrementation requires one step, from which we may conclude that the contents of any register used by M is $O(f(n))$.

5.4.4. The registers used by M in the case of input word w fall into two categories. First, there are those registers that are mentioned explicitly in *M*'s program. Their indices do not depend upon $n=|w|$ in any way and hence are $O(1)$. Second, there are those registers that are not mentioned explicitly but, rather, are introduced as the result of indirect addressing. Such a register R_i is introduced because some instruction takes operand $*R_j$, say, where the current contents of register R_j happens to be i. But, by Exercise 5.4.3, i is $O(f(n))$, from which it follows that any register index is $O(f(n))$ as well.

Solutions to Exercises for § 5.5

5.5.1.



5.5.10. A full description of a register machine M at a point during execution would consist of detailing the current contents of each of M's registers as well as the ("address" of the) instruction to be executed next. If M has input and output tapes, then their contents as well as read- and write-head locations would be part of this description.

5.5.12. In the case of a machine with multiple Halt instructions, replace all but one of them with unconditional branches to the one survivor.