The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Detailed Solution
Chapter 4, Problem 4.20, Part 4

1. Some experience writing production systems helps here. What we want to do is produce another n factors for 2 for each factor of 3, removing that factor when the n factors of 2 have been produced. However, we have to keep in mind that the rules are always scanned for applicability in the same order, so that, for instance, a rule that removes one factor of 3 without precaution, as in $3x \longrightarrow x$, will in fact get used over and over again, until all factors of 3 are gone. Removing only one factor takes some special care; we need to guard each rule by requiring that other factors be present, factors that we add in as needed and remove later. In particular, we want a single factor added in as a marker that we will remove when we remove the factor of 3; in spirit, we add the marking factor at the beginning of a loop and remove it at the end, enabling us to decrement the "loop counter" (the number of factors of 3) in a controlled fashion.
   In our solution we use 13 as the marking factor, 5 and 7 as stand-ins for 2, and 11 as a temporary replacement factor for 3. We accumulate all extra factors of 2 as factors of 5, so that we keep a count of n factors of 2 through all the loops—the 5s never get handled at all. Once we fall out of the last loop, we change all 5s back into 2s. The shift from 3s to 7s and back again into 3s and the shift of the n factors of 2 into 11s and back again into 2s are both used to sequence rules—we want to avoid going back to the first rules until the end of the transformation caused by one loop, so we change the factors and create additional rules that operate on the new factors. When we are ready to return to the beginning of the loop for a new iteration, we restore the original factors.
   A trick is used to introduce the single copy of the marking factor 13; obviously, we cannot just have a rule $x \longrightarrow 13x$, since that rule would be applicable forever. Instead, we set up 12 rules, each of the form $13x + i \longrightarrow 169x + 13i$, where I is between 1 and 12; such rules can

only apply if our number is *not* a multiple of 13 and only one such rule can apply. (Our actual rule is a bit more complex, because it also requires that we have present at least one factor each of 2 and of two of 3, so all numbers are in fact multiplied by 18—that is, we have 12 rules of the form $234x + 18i \longrightarrow 3042x + 234i$.) The various guards account for the large numbers in the rules, but one can see what the rule actually does by dividing throughout by the guards—as indicated in the comments.

```
A.  234x     ->  4095x        guarded by 13, which is conserved;
                              also guarded by 3*3, also conserved;
                              replaces each 2 by a 5 and a 7 (2x -> 35 x)
                              the 7 is just a temporary stand-in for 2;
                              the 5s form one more copy of the 2s
B. 1365x     ->  5005x        guarded by 13, which is conserved;
                              also guarded by 5*7, also conserved;
                              replaces each 3 by an 11 (3x -> 11x)
C.   91x     ->    26x        guarded by 13, which is conserved;
                              replaces each 7 by a 2 (7x -> 2x)
D.  143x     ->      x        guarded by 13, which is removed;
                              removes one factor of 3 (in its 11 guise)
E.   11x     ->     3x        unguarded; converts all factors of 11
                              back into factors of 3

F.                            the block below adds a single factor of 13
                              (but only to a number that does not have one)
                              it is guarded by 2*3*3, which is conserved
    234x+18  ->  3042x+234    (13x+1 -> 169x+13)
    234x+36  ->  3042x+468    (13x+2 -> 169x+26)
    234x+54  ->  3042x+702    (13x+3 -> 169x+39)
    234x+72  ->  3042x+936    (13x+4 -> 169x+52)
    234x+90  ->  3042x+1170   (13x+5 -> 169x+65)
    234x+108 ->  3042x+1404   (13x+6 -> 169x+78)
    234x+126 ->  3042x+1638   (13x+7 -> 169x+91)
    234x+144 ->  3042x+1872   (13x+8 -> 169x+104)
    234x+162 ->  3042x+2106   (13x+9 -> 169x+117)
    234x+180 ->  3042x+2340   (13x+10-> 169x+130)
    234x+198 ->  3042x+2574   (13x+11-> 169x+143)
    234x+216 ->  3042x+2808   (13x+12-> 169x+156)

G.                            one gets here only when all but one fac-
tor of 3
                              have been handled and removed;
                              just replaces all factors of 5 by fac-
tors of 2
                              and removes the last 3, if any
      5x     ->      x
      3x     ->      x
```

The system works as follows—keep in mind that it always tries to apply rule A first, then rule B if rule A is not applicable, then rule C, etc. When started with a number of the form $2^n \cdot 3^m$, if we have $n \geq 1$ and $m \geq 2$, then rules A through D cannot initially apply, since they require a factor of 13; rule E does not apply (no factor of 11), but one of the rules in block F does apply, yielding the new number $2^n \cdot 3^m \cdot 13$. Now rule A applies, over and over, replacing every factor of 2 by a factor of $5 \cdot 7$, so that we obtain $3^m \cdot 5^n \cdot 7^n \cdot 13$, at which point rule A no longer applies. At this point, though, rule B applies, since we have a factor of $5 \cdot 7 \cdot 13$ (the guard) in our number; this rule applies over and over again, replacing every factor of 3 by a factor of 11, so that we obtain $5^n \cdot 7^n \cdot 11^m \cdot 13$. Now neither rule A nor rule B applies, but rule C applies and replaces a factor of 7 by a factor of 2; notice that this change alone does not make rule A applicable (it would need a factor of 9 as well), so that rule C applies over and over again and replaces every factor of 7 by a factor of 2; we thus now have $2^n \cdot 5^n \cdot 11^m \cdot 13$. (A full copy of the 2s has been made, represented by the 5s.) Now none of rules A, B, and C applies, but rule D applies; it removes the guard of 13 and also one factor of 11, yielding $2^n \cdot 5^n \cdot 11^{m-1}$. Because the guard of 13 has been removed, rules A through D now will not apply until the guard is re-established; so rule E applies over and over again and replaces every factor of 11 by a factor of 3, yielding $2^n \cdot 3^{m-1} \cdot 5^n$. We have completed the loop: the loop counter (the power of 3) has been decreased by 1 and a copy of the 2s has been made. Now rules A through E fail to apply, so we fall through to block F, which adds again a factor of 13, thereby enabling rule A, etc. Eventually, we get to an application of rule E yielding the number $2^n \cdot 3 \cdot 5^{n(m-1)}$, at which point block F also fails to apply, since it requires two factors of 3. We then fall through to rule G, which applies over and over and replaces every 5 by a 2, yielding $2^{mn} \cdot 3$ and then removes the last 3, finally yielding 2^{mn} and stopping, as desired! Also note that special cases, such as the numbers 1, 2, 3, and 6, are all properly handled: nothing at all applies to either 1 or 2, which are output unchanged (as should be) and only the very last rule applies to 3 and 6, which are changed into 1 and 2 (again as should be).