ERRATA FOR ROSEN, DISCRETE MATHEMATICS AND ITS APPLICATIONS

BJORN POONEN

The following is a list of all errors known to me in Rosen, *Discrete Mathematics and Its Applications*, fourth edition. These were found by students, teaching assistants, and me, in the fall of 2001. I intend to email this list to the author at the end of the course. If you notice any errors not already in this list, please email them to me!

- p.77, bottom of page: In some copies of the fourth edition, the two cases are interchanged. (But in others, it is printed correctly, so presumably this was fixed recently.)
- p.78, line 4: should be "$d_4 = 4$ since $d_{44} \neq 4$".
- p.79, section 1.7, exercise 22 refers to exercise 13b — should that be 21b?
- p.80, section 1.7, exercise 40: should assume $a \neq 0$, or at least that a, b, c are not all zero.
- p.92, section 1.8, exercise 62: this seems to be contained in example 5 in the section. Perhaps the O was supposed to be a Θ in exercise 62?
- p.121, section 2.3, Gauss biography: "Fundamental Theorem of Arithmetic" should be "Fundamental Theorem of Algebra".
- p.147, section 2.5: In the RSA DECRYPTION section, the first displayed sequence of equalities should be a congruence $\pmod n$. This is because C is not equal to M^e; C is only congruent to M^e modulo n.
- p.210, section 3.3, exercise 30: "palindrome" is not defined in the book until p.243.
- p.223, section 3.5: The sentences

  > "An assertion that remains true each time S is executed must be chosen. Such an assertion is called a **loop invariant.**"

  suggest that a loop invariant must actually be true at the start of the **while** loop, and must be true after each execution of the S within the **while** loop. This is not in agreement with the definition "p is a loop invariant if $(p \wedge \textit{condition})\{S\}p$ is true." (This second definition says only that *if* p is true at the beginning of one execution of **while** loop, then p is true at the end of that execution.) I assume it is the latter definition that is correct.
- p.223, section 3.5: The proof in Example 4 is confused. As in the previous comment, whether p is true at the beginning of the **while** loop is irrelevant to the question of whether p is a loop invariant. Also, while induction could be used to prove that p is true after each iteration of the loop, induction is not relevant to the proof that p is a loop invariant. (One other change suggested is to call p an assertion rather than a proposition, since the truth value of p varies during the course of the program: it can be false just after an $i := i + 1$ step.) I suggest the following replacement.

Let p be the assertion "*factorial* $= i!$ and $i \leq n$." We first prove that p is a loop invariant. Suppose that, at the beginning of one execution of the **while** loop, p is true and the condition of the **while** loop holds; in other words, assume that *factorial* $= i!$ and that $i < n$. The values i_{new} and $\textit{factorial}_{\text{new}}$ of i and *factorial* at the end of the loop are given in terms of the original i and *factorial* by $i_{\text{new}} = i+1$ and $\textit{factorial}_{\text{new}} = \textit{factorial} \cdot (i+1)$. By assumption *factorial* $= i!$, so $\textit{factorial}_{\text{new}} = i! \cdot (i+1) = (i+1)! = i_{\text{new}}!$. Since $i < n$, we also have $i_{\text{new}} = i+1 \leq n$. Thus p is true at the end of the execution of the loop. This shows that p is a loop invariant.

Now we consider the whole program. Just before entering the loop, $i = 1 \leq n$ and *factorial* $= 1 = 1! = i!$ both hold, so p is true. Since p is a loop invariant, the rule of inference just introduced implies that if the **while** loop terminates, it terminates with p being true and with $i < n$ being false. In this case, at the end, *factorial* $= i!$ and $i \leq n$ are true, but $i < n$ is false; in other words, $i = n$ and *factorial* $= i! = n!$, as desired.

Finally, we need to check that the **while** loop actually terminates. At the beginning of the program i is assigned the value 1, so after $n-1$ traversals of the loop, the new value of i will be n, and the loop terminates then.

- p.227, supplementary exercise 9: the x_i should be distinct.
- p.228, supplementary exercise 28: the equation should be

$$\sum_{j=1}^{n} \cos jx = \cos\left[(n+1)x/2\right] \sin(nx/2) \,/ \sin(x/2) \ .$$

  (The / was missing.)
- p.229, supplementary exercise 50: $(a/2, b)$ should be $\gcd(a/2, b)$. More seriously, the facts about gcd given are insufficient to yield a terminating algorithm. See what happens on input $(a, b) = (3, 17)$, for instance.

  One way to fix this problem is to do all of the following:
  (1) Add the rule "$\gcd(a, b) = \gcd(b, a)$ if $a > b$".
  (2) Replace "$\gcd(a, b) = a$ if $a = b$" by "$\gcd(0, b) = 0$". (This is needed to handle the case where one of a and b is zero.)
  (3) Replace "$\gcd(a, b) = \gcd(b - a, b)$" by "$\gcd(a, b) = \gcd(a, b - a)$".
- p.252, section 4.3, Theorem 2: It would be better to allow n to be any *nonnegative integer* in the theorem's statement, since otherwise Corollary 1 does not follow in the indicated way. One could also allow $n = 0$ in Theorems 4 and 6 (but not in Theorem 7!)
- p.271, section 4.5, Example 5: The problem statement refers to the event that the bit string of length four has an even number of 0s, but the solution counts bit strings of length four having an even number of 1s. It would be good to fix the inconsistency, even though it turns out that the two conditions are equivalent.

  Also, as currently written, it almost sounds as if *two* bit strings are generated randomly, one for E, and one for F. With this interpretation, E and F would automatically be independent. Therefore I think it would

be better to define F as "the event that this bit string contains an even number of..." (where "this" refers to the bit string used to define E).

- p.282, section 4.5, proof of Theorem 6: When the sum for $V(X)$ is split in two, there is a $p(s)$ missing in the second sum.
- p.291, section 4.6, example 7: The solution refers several times to a "set of integers," which is confusing, since it is really a sequence: multiplicities matter.
- p.292, section 4.6, proof of Theorem 3: There are some extraneous z's in the next to last line!
- p.294, section 4.6, exercise 16(b): Is $x_5 > 5$ supposed to be $x_5 \geq 5$, perhaps?
- p.331, section 5.2, exercise 37: $a_n - 1$ should be a_{n-1}.
- p.349, section 5.4, line 6: The final expression for $G(x)$ is missing an x^n.
- p.360, section 5.5, exercise 24: If the coin is supposed to be fair, the problem should say so.
- p.368, section 5.6, exercise 22: "integers" should be "positive integers"