Tutorial 11: Part III Solutions

CS 135 Fall 2007

November 23, 2007

1. Will not terminate when $m = n > 0$; for example, `(bar 1 1)`. If we require $m \neq n$, then $(m-n)^2 > 0$, so either $mn = 0$ or $q > 0$. Therefore the values of `m` and `n` are always strictly decreasing. Furthermore, since they decrease by the same amount, they will still not be equal on the recursive call.

2. Always terminates. Note that the second condition is true whenever `n` is even. So if `n` is even, then the value on the recursive call is strictly decreasing. Now consider the binary representation of `n`. The number of 1's in the binary representation stays the same when `n` is even and decreases by exactly 1 when `n` is odd. Therefore `n` can only be odd a finite number of times, and therefore the algorithm terminates.

3. Will not terminate when i is not divisible by the gcd of m and n. If we require that m and n are relatively prime, then the nontermination condition will never hold, and so the algorithm will always stop eventually.

4. Always terminates. Note that the base (i.e. terminating) case here is when $m + n \geq 1000$. We can see that $(m - n) + (n + 1)^2 > m + n$, so whenever the second case holds, the value of $m+n$ is strictly increasing. And the first case can never hold twice in a row. So the function always terminates.