# Notes on Lecture 2
# 9 Sep., 2005 1
# Recursion Theory

Institute for Logic, Language and Computation
Universiteit van Amsterdam

---

[CT] is *Computability Theory* by Barry Cooper.

1. Let $sg(n)$ and $\overline{sg}(n)$ be defined as follows (Exercise 2.1.9, pg. 15):

$$sg(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n \neq 0 \end{cases}$$

$$\overline{sg}(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases}$$

   It is straightforward to show that both $sg$ and $\overline{sg}$ are primitive recursive (prove this).

2. The primitive recursive scheme for $rm(m, n)$ given on pg. 16 is:

$$\begin{aligned} rm(m, 0) &= 0 \\ rm(m, n+1) &= rm(m, n)' \times sg(|n - rm(m, n)'|) \end{aligned}$$

   In fact this is incorrect:

   **Counterexample:** $r(2, 3) = 1$; however

$$rm(2,1) = r(2,0)' \times sg(|0 - rm(2,0)'|) = 0' \times sg(|0 - 0'|) = 1 \times sg(1) = 1$$
$$rm(2,2) = r(2,1)' \times sg(|1 - rm(2,1)'|) = 1' \times sg(|1 - 2|) = 2 \times sg(1) = 2$$
$$rm(2,3) = r(2,2)' \times sg(|2 - rm(2,2)'|) = 2' \times sg(|2 - 3|) = 3 \times sg(1) = 3$$

   Corrected version:

$$\begin{aligned} rm(m, 0) &= 0 \\ rm(m, n+1) &= rm(m, n)' \times sg(|m - rm(m, n)'|) \end{aligned}$$

**Claim:** $rm(m, n)$ gives the remainder upon dividing $n$ by $m$.

**Proof:** Let $R(m, n)$ be the remainder upond dividing $n$ by $m$. That is $R(m, n) = r$ means that there is a (positive) $q$ such that $n = qm + r$. We will show by induction on $n$ that $rm(m, n) = R(m, n)$, i.e. there is a $q$ such that $n = qm + rm(m, n)$. The base case is straightforward: $R(m, 0) = 0 = rm(m, 0)$. Suppose that $rm(m, n) = R(m, n)$. Then there is a $q$ such that $n = qm + rm(m, n)$. We will show that $rm(m, n + 1) = R(m, n + 1)$, i.e., there is a $q'$ such that $n + 1 = q'm + rm(m, n + 1)$. We have two cases:

(a) $sg(|m - rm(m, n)'|) = 0$. In this case, $rm(m, n + 1) = 0$ so we must show there is a $q'$ such that $n + 1 = q'm$. By the definition of $sg$, we have $m = rm(m, n)'$, i.e., $m = rm(m, n)+1$. Since $n = qm+rm(m, n)$, $n+1 = qm+rm(m, n)+1 = qm+m$. Then we are done (let $q' = q+1$).

(b) $sg(|m - rm(m, n)'|) = 1$. In this case $rm(m, n + 1) = rm(m, n)' = rm(m, n)+1$. Since $n = qm+rm(m, n)$, $n+1 = (qm+rm(m, n))+1 = qm + (rm(m, n) + 1) = qm + rm(m, n + 1)$, as desired.

3. Define the function $eq(m, n)$ to be 1 if $m = n$ and 0 otherwise. Then as discussed in class, it is easy to see that $eq$ is a primitive recursive function.

4. Let $Q(m, n)$ be the quotient upon dividing $n$ by $m$, i.e., $Q(m, n) = q$ means that there is an $r$ such that $n = qm + r$. Consider the following primitive recursive function:

$$\begin{aligned} q(m, 0) &= 0 \\ q(m, n + 1) &= q(m, n) + eq(m \times q(m, n) + m, n + 1) \end{aligned}$$

**Claim:** $q(m, n) = Q(m, n)$.

**Proof:** The proof is by induction on $n$. The base case is obvious. Suppose that $q(m, n) = Q(m, n)$, i.e., there is an $r$ such that $n = q(m, n)m + r$. We must show there is an $r'$ such that $n + 1 = q(m, n + 1)m + r'$. We have two cases:

(a) $eq(m \times q(m, n) + m, n + 1) = 1$. Then by the definition of $eq$, $m \times q(m, n)+m = n+1$. By the definition of $q$, $mq(m, n+1) = m(q(m, n)+1) = mq(m, n) + m = n + 1$. Then we are done (let $r' = 0$).

(b) $eq(m\times q(m, n)+m, n+1) = 0$. In this case $q(m, n+1) = q(m, n)$. Then if $r' = r + 1$, we have $n + 1 = mq(m, n) + (r + 1) = mq(m, n + 1) + r'$.