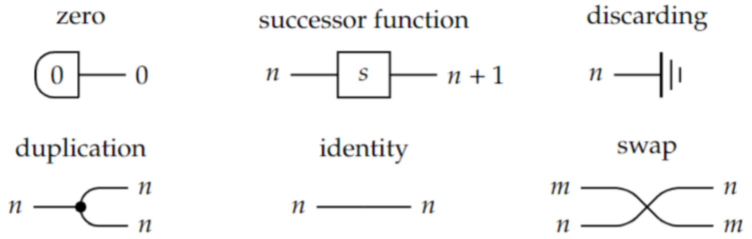
(1) In our definition of primitive (or partial) recursive funtions we use the box-and-wire notation and take the following six functions as basic (see the *Recursive Functions* slide deck, p. 2):



Nevertheless, a conventional textbook uses the following three kinds of functions as basic:

- The zero function $z^k : \mathbb{N}^k \to \mathbb{N} :: (n_1, \ldots, n_k) \mapsto 0$ for each k.
- The successor function $s : \mathbb{N} \to \mathbb{N} :: n \mapsto n + 1$ (the same function as the box above).
- The projection $\Pi^k_i : \mathbb{N}^k \to \mathbb{N} :: (n_1, \ldots, n_k) \mapsto n_i$ for each k and every i with $1 \leqslant i \leqslant k$.

(See the *Turing Machines* slide deck, p. 8, for the notation "$f : \mathbb{N}^k \to \mathbb{N} :: (n_1, \ldots, n_k) \mapsto \cdots$".) Using the box-and-wire notation and our definition, show that the projection Π^5_3 (with 5 and 3 in particular in place of k and i in the Π^k_i above) is primitive recursive.

(2) Prove that the following "conditional function"

$$\mathrm{cond} : \mathbb{N}^3 \to \mathbb{N} :: (x, y, n) \mapsto \begin{cases} x & \text{if } n > 0, \\ y & \text{if } n = 0 \end{cases}$$

is primitive recursive, by the following two steps:

(a) Define cond by primitive recursion. This is to spell out the two functions f and g used in the base clause (the 0th step, using f) and the inductive clause (the ith step within the `for` loop, using g) in primitive recursion defining h = cond.

```
cond(x, y, 0) := f(x, y)
for i in (0, ..., n-1):
    cond(x, y, i+1) := g(x, y, i, cond(x, y, i))
```

(b) Show that these two functions f and g are both primitive recursive.

[Note (not necessarily a hint): cond *defines an* `if` ... `else` ... *branching*.]

(3) In the (untyped) lambda calculus, let us define

$$\overline{\mathrm{True}} := (\lambda x. (\lambda y. x)), \qquad \overline{\mathrm{False}} := (\lambda x. (\lambda y. y)).$$

Then show that

$$((\overline{\mathrm{True}}\, M)\, N) \xrightarrow{\beta} \cdots \xrightarrow{\beta} M, \qquad ((\overline{\mathrm{False}}\, M)\, N) \xrightarrow{\beta} \cdots \xrightarrow{\beta} N.$$

Just in case you wonder "But what if x or y occurs in M or N?", we assume that neither x nor y occurs in M or N. [Note (again, not a hint): *These also give an* `if` ... `else` ... *branching!*]