1 Algorithms

A sketch of a formal definition of an algorithm:

A *computational method* is a quadruple (Q, I, Ω, f), where

- $I \subset Q$ is the *input*
- $\Omega \subset Q$ is the *output*
- $f : Q \to Q$ is the *computational rule*, which satisfies $f(\omega) = \omega\ \forall \omega \in \Omega$.

Each $x \in I$ defines a *computational sequence* $x_0, x_1, \ldots,$, where $x_0 = x$ and $x_{k+1} = f(x_k)$. The sequence *terminates* in k steps if k is the smallest integer such that $x_k \in \Omega$. An *algorithm* is a computational method that terminates in some finite number of steps for all x in I.

As an example, we present Euclid's algorithm in this formalization: let Q be the set of all singletons $\{n\}$, all ordered pairs (m, n), and all ordered quadruples $(m, n, r, 1), (m, n, r, 2), (m, n, p, 3)$ where m, n, p are positive integers and r is a nonnegative integer. Let I be the ordered pairs (m, n) and Ω the set of singletons $\{n\}$. Define f by

$$f((m,n)) = (m,n,0,1);\ f((n)) = (n); \tag{1}$$
$$f((m,n,r,1)) = (m,n,m \% n,2); \tag{2}$$
$$f((m,n,r,2)) = (n) \text{ if } r = 0,\ (m,n,r,3) \text{ otherwise}; \tag{3}$$
$$f((m,n,p,3)) = (n,p,p,1) \tag{4}$$

2 Algorithms - solutions to exercises

1.1 $t \leftarrow a$, $a \leftarrow b$, $b \leftarrow c$, $c \leftarrow d$, $d \leftarrow t$.

1.2 We have $m \leftarrow n$ and $n \leftarrow r$. Since $r < n$, after assignment $n < m$.

1.3 Algorithm F. Given two positive integers m and n, find the greatest common divisor.

F1 Divide m by n.
F2 Set m equal to the remainder.
F3 If $m = 0$ then the answer is n.
F4 Otherwise divide n by m.
F5 Set n equal to the remainder.
F6 If $n = 0$ then the answer is m.
F7 Go to **F1**.
1.4 $6099\%2166 = 1767 \Rightarrow 2166\%1767 = 399$
$\Rightarrow 1767\%399 = 171 \Rightarrow 399\$171 = 57 \Rightarrow 171\%57 = 0$.
So the GCD is 57.
1.5 Not finite, not definite, not effective.

1.6 $n = 1$: $1\%5 = 1 \Rightarrow 5\%1 = 0$, 2 steps
$n = 2$: $2\%5 = 2 \Rightarrow 5\%2 = 1 \Rightarrow 2\%1 = 0$, 3 steps
$n = 3$: $3\%5 = 3 \Rightarrow 5\%3 = 2 \Rightarrow 3\%2 = 1 \Rightarrow 2\%1 = 0$, 4 steps
$n = 4$: $4\%5 = 4 \Rightarrow 5\%4 = 4 \Rightarrow 4\%4 = 0$, 3 steps
$n = 5$: $5\%5 = 0$, 1 step
So $T_5 = 2.6$.

1.7 U_m is well-defined: if $n > m$, the first step of the Euclidean algorithm simply swaps n and m (since $m \ \% \ n = m$) and $U_m = T_m + 1$. If $n < m$ then there are only finitely many cases.

3 Mathematical Preliminaries - Induction

Algorithmic proof procedure:

Algorithm I - Construct a proof Given a positive integer n and proposition $P(n)$, this algorithm will output a proof that $P(n)$ is true (if it succeeds).

I1 [Prove $P(1)$] Set $k \leftarrow 1$ and use another algorithm to output a proof of $P(1)$.

I2 [$k = n$?] If $k = n$, terminate - the required proof was found in the previous step.

I3 [$k < n$] Otherwise $k < n$. Use another algorithm to output a proof of the following statement: "If $P(1)$, $P(2)$,...,$P(k)$ is true, then $P(k+1)$ is true." Then output the statement "We have already proved $P(1), \ldots, P(k)$, hence $P(k+1)$ is true." Combine these statements.

I4 Set $k \leftarrow k + 1$. Go to step **I2**.

Here is an inductive proof of a fact about the Fibonacci sequence. Let $F_0 = 0, F_1 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$. Define $\phi = (1 + \sqrt{5})/2$. Then $F_n \leq \phi^{n-1}$ for all positive n.

We proceed according to the algorithm above. This is clearly true for $n = 0$ and $n = 1$,l so we have obtained a proof of $P(1)$. For $P(2)$, $F_2 = 1$ and $\phi > 1.6$, so we have a (computational) proof of $P(2)$. Now assume our target is $k + 1$ with $k > 1$ and we have k proofs $P(1), \ldots, P(k)$. Since $F_{k+1} = F_k + F_{k-1}$, and by hypothesis $F_k \leq \phi^{k-1}$ and $F_{k-1} \leq \phi^{k-2}$,

$$F_{k+1} \leq \phi^{k-1} + \phi^{k-2} = \phi^{k-2}(1 + \phi). \tag{5}$$

ϕ is actually the positive solution to $1 + \phi = \phi^2$. So plugging this in gives $F_{k+1} \leq \phi^k$, as desired.

Note that our proof would have failed if we didn't have direct proofs of $P(1)$ and $P(2)$: $P(1)$ would fail at the inductive step since the theorem is not true for $n = 0$, and for $P(2)$ we couldn't have applied the method at the $F_{k-1} \leq \phi^{k-2}$ step (since $k = 1$).