Solution for sheet 4

12 a) proof by contradiction: let r be the order of x mod n and assume

$$m = kr + s, \quad 0 < s < r$$

$\rightarrow \quad 1 = x^{kr+s} = (x^r)^k x^s = x^s \mod n$,

which contradicts r being the least integer s.t. $x^r = 1 \mod n$.

b) by Euler's theorem $x^{\varphi(n)} = 1 \mod n$ and therefore according to a) $\varphi(n) = kr$

c) (note: x co-prime $n \Rightarrow x^e$ co-prime n)

- r order of x^e mod n means

$$1 = (x^e)^r = x^{er} \mod n,$$

$$\xrightarrow{a)} \qquad er = k\tilde{r} \qquad (*)$$

- according to b) both r and $\tilde{r}$

are factors of $\varphi(n)$

$\rightarrow$ r and $\tilde{r}$ are co-prime e,

as e is co-prime $\varphi(n)$

this means that h in $(*)$ must

be divisible by e!

$\rightarrow$ $\underline{r = \ell \tilde{r}}$ with $\underline{\ell \geq 1}$;

- by $(x^e)^{\tilde{r}} = (x^{\tilde{r}})^e = 1 \mod n$

and as: $\underline{\tilde{r} = \ell' r}$ with $\underline{\ell' \geq 1}$

$\rightarrow$ $r = \tilde{r}$.

15 a) M moves to the right-hand end of the input string x (first 2 rules), adds "11" (3rd and 4th rule) and moves back to the left-hand end of x (last 2 rules);

$\Rightarrow$ M computes $f(x) = 4x + 3$

Example: (blanks "$\square$" only written where needed)

$s_0\,101 \vdash 1\,s_0\,01 \vdash 10\,s_0\,1 \vdash 101\,s_0\,\square$

$\vdash 1011\,s_1\,\square \vdash 101\,s_2\,11 \vdash 10\,s_2\,111$

$\vdash 1\,s_2\,0111 \vdash s_2\,10111 \vdash s_2\,\square 10111$

$\vdash s_e\,10111$

i.e. $\underset{\substack{\shortparallel \\ 5}}{s_0\,101} \vdash^* \underset{\substack{\shortparallel \\ 23 \,=\, 4\cdot 5+3}}{s_e\,10111}$

d) $x \bmod 2$ means "erase all bits but the last one"

→ states s_0, s_1, s_2, s_3, s_e with

rules δ: $s_0, a \to s_0, a, R$ $\quad a = 0, 1$

$s_0, \square \to s_1, \square, L$

$s_1, a \to s_2, a, L$

$s_2, a \to s_2, \square, L$

$s_2, \square \to s_3, \square, R$

$s_3, \square \to s_3, \square, R$

$s_3, a \to s_e, a, N$

e) • final state s_e of M becomes a non-final state of M'

• add new final state s_e' and new state s_{10} →

new rules: $s_e, 0 \to s_e', 0, N$

$s_e, 1 \to s_\infty, 1, L$

$s_\infty, \square \to s_\infty, \square, L$

with this states and rules M' works as follows:

if $h(w) = 0$:

$M': s_0 w \overset{M}{\vdash^*} s_0 0 \vdash s_e' 0$ (halts)

if $h(w) = 1$:

$M': s_0 w \overset{M}{\vdash^*} s_0 1 \vdash s_\infty \square 1$

$\vdash s_\infty \square\square 1 \vdash s_\infty \square\square\square 1 \vdash \dots$

(ad infinitum ...)

14) for $y \in \mathbb{R}$ let

$$A_y = \{ i \mid x_i < y \}$$

$$t_y = |A_y|$$

and $G(y)$ the Grover search that outputs an $x \in A_y$ within

$O(\sqrt{N/t_y})$ queries

$\rightarrow$ algorithm:

$y_0 = x_1$

repeat

$y_{i+1} = G(y_i)$

until $t_{y_{i+1}} = 0$

$\rightarrow x_{min} = y_{i+1}$

worst case: $O(N \cdot \sqrt{N})$ queries!

expected number of queries:

$$n_\ell = \sum_{\ell=0}^{n} \sqrt{\frac{N}{N/2^\ell}} = \sum_{\ell=0}^{n} 2^{\ell/2} = \frac{1-2^{\frac{n+1}{2}}}{1-\sqrt{2}}$$

$n = \log_2 N$

i.e. $$n_\ell = O(2^{n/2}) = O(\sqrt{N})$$