I2CN exercise sheet 3

Dimitri Tabatadze ~ 2025-03-25

1 Leader election in rings

Assume anonymous processes, indirect addressing, randomization and konwn ring size N. Give pseudo code of a probabilistic algorithm for leader election.

> **Hint:** you have to deal with duplicates in the random choices, possibly repeatedly.

Solution

```
leader_elected = False
am_leader = False
participating = True
am_unique = True

while not leader_elected:
  if participating:
    ov = 0 if not participating else 1 + rand()
    v = ov
    for _ in range(N):
      send(v)

      r = recv()
      if r == "LEADER_ELECTED":
        v = "LEADER_ELECTED"
        leader_elected = True
      elif r > v:
        v = r
        participating = False
      elif r == v:
        am_unique = False
    am_leader = v == ov and am_unique
    if am_leader:
      send("LEADER_ELECTED")
      leader_elected = True
```

2 Reconsider exercise 1

When the algorithm in exercise 1 termiates it always has solved leader election. Show that this cannot work with unknown ring size for any algorithm.

Hint: I explained the solution in the classroom.

Solution

Assume the ring has size N and the processes, without knowing the size, elected a leader p_l. If we were to have made the ring twice as large with the new processes being identical copies of the original processes as follows:

$$p_i = p_{i+N} \forall i \in [0 : N-1] \tag{1}$$

the processes would not know the difference between p_l and p_{l+N} and they would elect two leaders.

3 One's complement numbers

Let $a, b \in \mathbb{B}^n$ and $s \in \mathbb{B}^{n+1}$ and $[\![a]\!] + [\![b]\!] = [\![s]\!]$. Prove or disprove

1. sign extension: $[\![a_{n-1}a]\!] = [\![a]\!]$
2. detecting overflow: $[\![s]\!] \notin OC_n \leftrightarrow s_n \neq s_{n-1}$

Hint: look at the proof of similar statements for two's complement numbers.

Solution

1. - $\boldsymbol{\neg a_{n-1}}$: $[\![a]\!] \geq 0 \rightarrow [\![a]\!] = [\![0a]\!]$
   - $\boldsymbol{a_{n-1}}$: $[\![a]\!] \leq 0 \rightarrow [\![a]\!] = -[\![\bar{a}]\!] = -[\![0\bar{a}]\!] = [\![1a]\!]$
2. First, by definition.

$$[\![s]\!] \notin OC_n \rightarrow [\![s]\!] \in [-2^n + 1 : 2^n - 1] \setminus [-2^{n-1} + 1 : 2^{n-1} - 1] \tag{2.1}$$
$$\rightarrow [\![s]\!] \in [-2^n + 1 : -2^{n-1}] \vee [\![s]\!] \in [2^{n-1} : 2^n - 1] \tag{2.2}$$

Now consider the cases:

- $[\![s]\!] \in [\boldsymbol{2^{n-1} : 2^n - 1}]$:

$$\rightarrow s_{n-1} \wedge \neg s_n \tag{3}$$

- $[\![s]\!] \in [\boldsymbol{-2^n + 1 : -2^{n-1}}]$:

$$\rightarrow s_n \wedge \neg s_{n-1} \tag{4}$$

so $[\![s]\!] \notin OC_n \rightarrow s_n \neq s_{n-1}$. Now consider $s_n \neq s_{n-1}$.

- $\boldsymbol{\neg s_n \wedge s_{n-1}}$:

$$\neg s_n \rightarrow [\![s]\!] \geq 0, \qquad s_{n-1} \rightarrow [\![s]\!] \geq 2^{n-1} \rightarrow [\![s]\!] \notin OC_n \tag{5}$$

- $\boldsymbol{s_n \wedge \neg s_{n-1}}$:

$$s_n \to [\![s]\!] \leq 0, \qquad \neg s_{n-1} \to [\![s]\!] \leq -2^{n-1} \to [\![s]\!] \notin OC_n \tag{6}$$

4 Correctness of Polynomial Division

show the following properties of the division algorithm for polynomials

$$f_{i+1} = f - q_i g \forall i \tag{7.1}$$

$$a_{i,n-1} \neq 0 \to \deg(f_{i+1}) < \deg(f_i) \tag{7.2}$$

What are the outputs (quotient and reminder) of the division algorithm?

Solution

- $$f_{i+1} = f_i - t_i g = (f_{i-1} - t_{i-1} g) - t_i g = \cdots = f - (t_0 + \cdots + t_i) g = f - q_i g \tag{8}$$

  .

- $$\begin{aligned} & a_{i,n-i} \neq 0 \wedge b_m \neq 0 \to c_i \neq 0 && (9.1)\\ \to f_{i+1} &= f_i - g t_i && (9.2)\\ &= \sum_{j=0}^{n-i} a_{i,j} x^j - \sum_{j=0}^{m} b_j x^j x^{n-m-i} \left(\frac{a_{i,n-i}}{b_m}\right) && (9.3)\\ &= \cancel{a_{i,n-i} x^{n-i}} + \sum_{j=0}^{n-i-1} a_{i,j} x^j - \cancel{b_m x^{n-i} \left(\frac{a_{i,n-i}}{b_m}\right)} - \sum_{j=0}^{m-1} b_j x^j x^{n-m-i} \left(\frac{a_{i,n-i}}{b_m}\right) && (9.4)\\ &= \sum_{j=0}^{n-i-1} a_{i,j} x^j - \sum_{j=0}^{m-1} b_j x^j x^{n-m-i} \left(\frac{a_{i,n-i}}{b_m}\right) && (9.5) \end{aligned}$$

  and therefore the degree of f_i is greater than f_{i+1} given that $a_{i,n-i} \neq 0$.

5 CRC Computation

Consider message $u = 1010001101$ and generator polynomial $x^5 + x^4 + x^2 + 1$ as the divisor.

1. What polynomial division is performed?
2. Compute the result.
3. What message is sent in binary?

Solution

1. $x^5 \cdot (x^9 + x^7 + x^3 + x^2 + 1)$ should be divided by the generator $x^5 + x^4 + x^2 + 1$

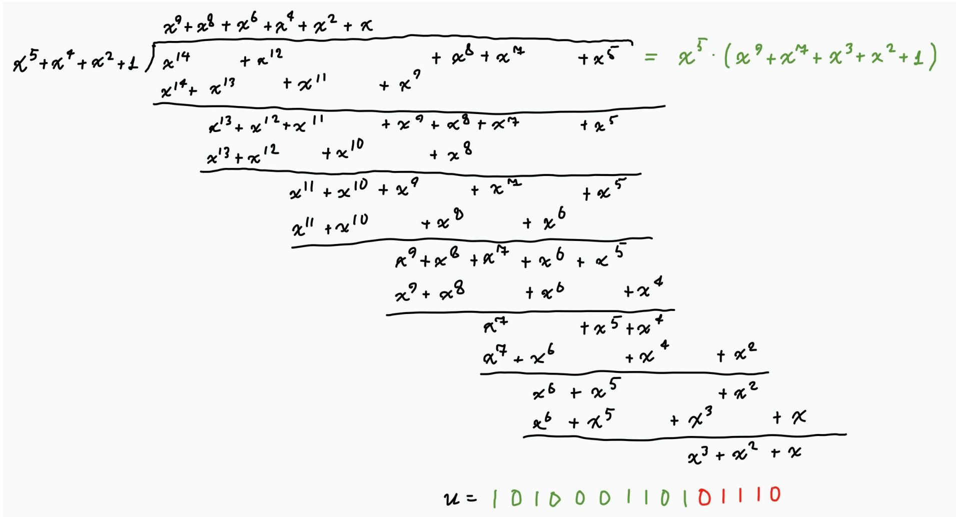
2. 

Figure 1: the division

3. What should be sent is 101000110101110.