TP - Architecture et Système - 03

04 October 2019

1 De Bruijn sequences

In this part of the exam, we will develop an efficient method to count the number of trailing zero bits in a given (unsigned) integer value x such that $x > 0$. Equivalently, we can compute the position of the least significant bit whose value is 1. Incidentally, one concrete application – when x has 64 bits – is to encode the positions of pieces on a chess board and iterate over these. Here however, we will simplify matters by assuming that x has only 8 bits.

An *index* in a bit string is identified from right to left starting at zero. E.g., for $x = (10110100)_2$, the bits of x at index 0 and 1 are 0, and the bit with index 2 is 1.

Given $x \in \mathbb{N}$ such that $0 < x < 2^8$, we will be interested in implementing a function $\ell : \{1, \ldots, 2^8 - 1\} \to \{0, \ldots, 7\}$ such that $\ell(x)$ is equal to smallest index that is set to 1 in the binary representation of x. In the example above, we have $\ell(x) = 2$.

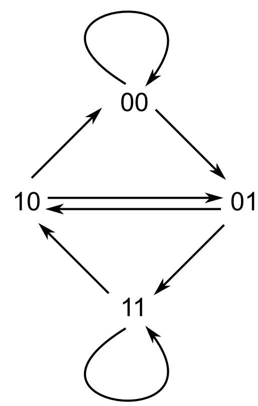
In principle, we could solve the problem using the C function below, which shifts x to the right until the least significant bit is 1.

```
unsigned int l (unsigned int x) {   // we assume 0 < x < 256
  int result = 0;
  while (x & 1 == 0) {
    result++;
    x = x >> 1;
  }
  return result;
}
```

However, the running time of this function depends on the number of bits in x. We will develop another algorithm that has *constant* running time, i.e. independent of the actual number of zeros. To this end, we will study *de Bruijn sequences*. A de Bruijn sequence $s(n)$ of order n is a cyclic bit string such that every binary string of length n occurs exactly once in s. Cyclic means that once you reach the end of $s(n)$ you may continue at the beginning of $s(n)$. For example, for $n = 2$ we can set $s(n) = 0011$ since 00, 01, 10 and 11 can all be found in $s(n)$; in particular 10 starts at index 0 of $s(n)$ and then continues at index 3 of $s(n)$.

An obvious lower bound for the minimal length of a de Bruijn sequence $s(n)$ is 2^n. We will see that a sequence of this length can always be found, then use it to achieve our initial goal.

(a) (3 points) De Bruijn sequences can be obtained from paths in *de Bruijn graphs*. The vertices of a de Bruijn graph of order n are all bit strings of length n. There is a directed edge between two vertices $b_1b_2\cdots b_n$ and $c_1c_2\cdots c_n$ if and only if $b_2 = c_1, b_3 = c_2, \ldots, b_n = c_{n-1}$. The figure below depicts the de Bruijn graph of order 2.



Draw the de Bruijn graph of order 3.

(b) (2 points) A de Bruijn sequence can be obtained from a de Bruijn graph by following a *Hamiltonian cycle* that starts and ends in the vertex $0\cdots 0$. A Hamiltonian cycle is a cycle that visits each vertex exactly once before returning to the starting vertex. For instance, the only Hamiltonian cycle in the graph in the figure above is $00 \to 01 \to 11 \to 10 \to 00$. This cycle corresponds to the aforementioned de Bruijn sequence 0011. One can in fact prove that such a Hamiltonian cycle exists in every de Bruijn graph.

Read off two different de Bruijn sequences of order 3 by following two different Hamiltonian paths in your de Bruijn graph of order 3 starting in vertex 000.

(c) (2 points) *Choose a de Bruijn sequence* $s(3)$ *of order 3 from (b) and complete the following table:*

bit-string	7 - index in $s(3)$
000	0
001	
010	
011	
100	
101	
110	
111	

(d) (2 points) *Let* $s(3)$ *be the de Bruijn sequence from (c) and* $0 \leq j < 8$. *What is the* *value assigned by the table in (c) of the bit string*

$$((s(3) \ll j) \gg 5) \; \& \; 0x7$$

Here, $\ll$ and $\gg$ mean shift-left and shift-right, respectively, and & is binary AND.

(e) (2 points) *Given an unsigned integer $k > 0$, what is the value of $k \; \& \; (-k)$, where $-k$ is the two's complement of k?*

(f) (3 points) *Complete the following code skeleton such that it computes $\ell(x)$:*

```
const int index[8] = { 0, ... }; // the right-hand side of the table
                                 // in (c) here
const int s3 = 0b...;            // your de Bruijn sequence used in (c)

unsigned int l(unsigned int x) { // we assume 0 < x < 256
  return index[ ... ];           // complete code in the brackets
}
```

1.1 Extra:

For the following exercise denote by $G_n = (V_n, E_n)$ the De Bruijn graph of size n, and $b_n : V_n \to B_n$ (B_n- bit-string of length n). The following shows that De Bruijn graphs have a Hamiltonian cycle:

1. For all $n \geq 1$, the graph G_n has a Eulerian cycle (goes through every edge exactly once).
2. For all graphs $G = (V, E)$, denote by $A(G) = (V^a, E^a)$ their line graph. Where $V^a = E$ and $(uv, wx) \in E^a$ iff $v = w$. Show that $G_{n+1} \cong A(G_n)$ for all $n \geq 1$.
3. Show that any Eulerian cycle in a graph G corresponds to a Hamiltonian cycle of $A(G)$.