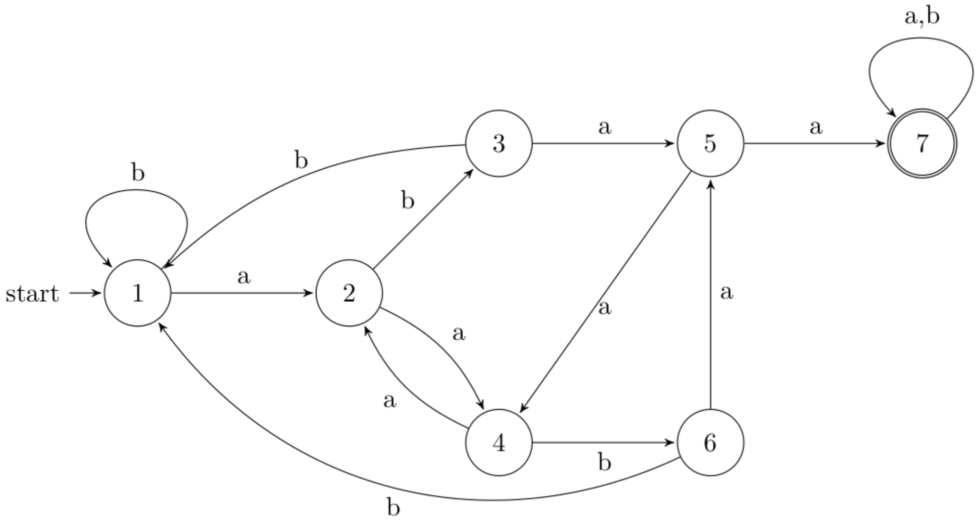
Compiler Design GP

March 2024

1.1 Hopcroft's Minimization Algorithm



(i) Fill out the table with Hopcroft's algorithm.

Iteration	Groups	Split on a	Split on b

(ii) Draw the minimized DFA.

2.1 Grammars

$$
\begin{aligned}
S &\to ABC \\
A &\to aA \mid \varepsilon \\
B &\to bB \mid \varepsilon \\
C &\to cA \mid \varepsilon
\end{aligned}
$$

Show a leftmost derivation for the string *abbc*. At each step underline which non terminal is derived.

3.1 CFG

```
1  n = ...;
2  i = 0;
3  while(n > 1) {
4      if (n % 2 == 0) {
5          n = n/2;
6      } else {
7          n = 3*n+1;
8      }
9      i = i + 1;
10 }
11 r = i;
```

(i) Construct the control flow graph.

(ii) Find the reaching definitions by filling in the table of IN and OUT.

Line	IN	OUT
1		
2		
3		
4		
5		
7		
8		
9		
11		

Register Allocation

```
1      load r1, @a
2      load r2, @b
3      mult r3, r1, r2
4      load r4, @c
5      load r5, @d
6      div r6, r4, r5
7      add r7, r3, r6
```

(i) Fill out the table of liveness ranges.

	1	2	3	4	5	6	7
r1							
r2							
r3							
r4							
r5							
r6							
r7							

(ii) Draw the register allocation graph.

(iii) Find a minimal coloring for the register allocation graph.

(iv) Without changing the order of instructions rewrite the code to use a minimal number of registers.

Optimizations

```
y = 100 * x;
```

Assume multiplication takes 10 cycles and addition and shifting each take only 1 cycle.

(i) Rewrite the code to use fewer cycles.

(ii) Now assuming multiplication takes x cycles, calculate the minimum x, such that your code is still faster.