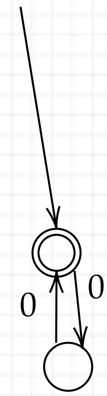
CSE491/596 Lecture Wed. 6 Sept.: Regular Expressions and FAs

Built up from characters and the empty string $(\epsilon \text{ or } \lambda)$
via the operations + (also written $\cup$ or |), $\cdot$, and *:
this + that: this or that
this $\cdot$ that: this followed by that
(this)* : zero or more occurrences of this. *Examples:*

$(a+b)\cdot(a+c) = aa + ac + ba + bc$ $\quad (0+01)\cdot(10+0) = 010 + 00 + 0110$

$(a+bc)^* = \{\epsilon, a, bc, aa, abc, bcbc, bca, aaa, \dots\}$ But not bac for instance.



$(00)^* = \{\epsilon, 00, 0000, \dots\} = \left\{0^n : n \text{ is even}\right\}$.

$(11)^*1 = \{1, 111, 11111, \dots = \left\{1^n : n \text{ is odd}\right\}$. $\quad 1(11)^*$ *is equivalent*.

Now how about strings over {0,1} containing an odd number of 1s?

Try even 1s first: $\left(0^*10^*10^*\right)^*$. Fixing the even case, use $\left(0^*10^*1\right)^*0^*$.

Then add a 1 to make it odd: $\left(0^*10^*10^*\right)^*1$. Is that good? ***Sound***? ***Comprehensive***?

(needs to allow ending in 0s) Note incidentaly that $0^*0^* = 0^*$.

Economical is: $\left(0^*10^*1\right)^*0^*10^*$. This was not comprehensive, did not match 0,

How about $\left\{x \in \{0,1\}^* : \text{every 5th char of x from the first is a 1}\right\}$? We can try

$\left(1(0+1)^4\right)^*$.

But this forces the string to have length a multiple of 5. To allow other ~~lengths, try:~~

$\left(1(0+1)^4\right)^*(\epsilon + 1(\epsilon+0+1)(\epsilon+0+1)(\epsilon+0+1))$ [Will pause for why it works.]

Now how do we apply these ideas to make a regular expression for Wed.'s language $L(M_{5,2}) \;=\; \{x:\; x\, has\, an\, odd\, \#\, of\; 1s\; in\; positions\;\; \equiv 2 \bmod 5\}$?

First, we need at least 3 chars, to get at least one 1 in such a position.
The first two such chars are arbitrary: $(0+1)^2$. Then we see the equation:

$$L(M_{5,2}) \;=\; (0+1)^2 \cdot L(M_{5,0})$$

Thus we can focus on "blocks" of the form $Z \;=\; 0(0+1)^4$ or $I \;=\; 1(0+1)^4$.
Take our previous "template" for an odd number of 1's and sub. 0 by Z, 1 by I:

$$L(M_{5,0}) \;=\; \left(Z^*IZ^*I\right)^* Z^*IZ^*$$

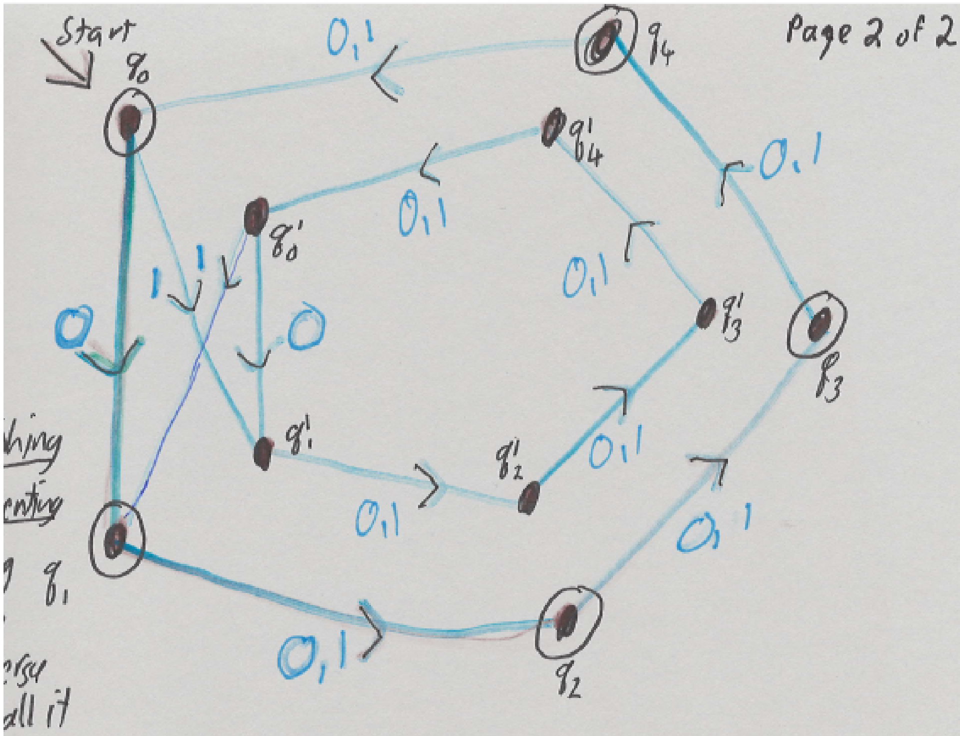
But this has another "overkill" problem: The last 1 in a multiple-of-5 position need not be followed by 4 chars. So instead define $Y \;=\; (0+1)^4 0$. Then:

$$L(M_{5,2}) \;=\; (0+1)^2 \cdot \left(Z^*IZ^*I\right)^* Z^*1Y^*(\epsilon+0+1)^4.$$

$$= \; (0+1)^2 \cdot \left(\left(0(0+1)^4\right)^* 1(0+1)^4 \left(0(0+1)^4\right)^* 1(0+1)^4\right)^* \left(0(0+1)^4\right)^* 1\left((0+1)^4 0\right)^* (\epsilon+0+1)^4.$$

Yuck---?---! We got it by top-down reasoning---but maybe there's a better way...

Here is the DFA $M_{5,0}$ that was referred to, from the first-day lecture in 2021 :



If we start this machine up in state q_3 then we get $M_{5,2}$: the machine either gets just zero or one char and accepts, or it gets two chars corresponding to the initial $(0+1)^2$ and then goes into the same machinations as $M_{5,0}$.