Now let us abbreviate $A \cdot A$ as A^2, $A \cdot A \cdot A = A^3$, and so on. This is OK even though concatenation isn't commutative on languages either---hey, neither is matrix multiplication, but A^3 is like raising a matrix to the third power, and that's fine. Just like with numbers and matrices, strings and languages obey the additive law of powers: $A^i \cdot A^j = A^{i+j}$.

We have $A^1 = A$, of course, but what is A^0? In particular, what is $\emptyset^0$? Well, suppose we wrote a program loop to fetch and test a mandated number n of chunks of sensor data?

```
for (int i = 0; i < n; i++) {
   string xi = getNewSensorData();
   if (! A(xi)) { throw Failure; }
}
```

If we invoke this loop with a given number n then it will perform n tests and allow processing to continue without exception only if all n of the tests pass.

- If $A = \emptyset$ and we enter the loop, then the body surely fails, *finito*.
- If $n = 0$ then what happens? The loop is a fall-through. Do we die? No: processing continues undisturbed.
- So what happens if $A = \emptyset$ and $n = 0$? *The same as the second case*: we never get put to the death test, and processing continues undisturbed.

In the third case, nor is any data taken. Thus this is exactly the same situation as when concatenating with $\{\epsilon\}$. It is a "free pass". So $A^0 = \{\epsilon\}$ for any language A, and in particular:

$$\emptyset^0 = \{\epsilon\}.$$

Well, this is just like the numerical convention $0^0 = 1$. In all cases, this is needed to make the additive power law $A^i \cdot A^j = A^{i+j}$ work even when $j = 0$.

Also note: $A^* = \bigcup_{i=0}^{\infty} A^i = A^0 + A^1 + A^2 + \cdots$ always includes the empty string ϵ.

[If even more time allows, tell the story at https://rjlipton.wordpress.com/2015/02/23/the-right-stuff-of-emptiness/ .]

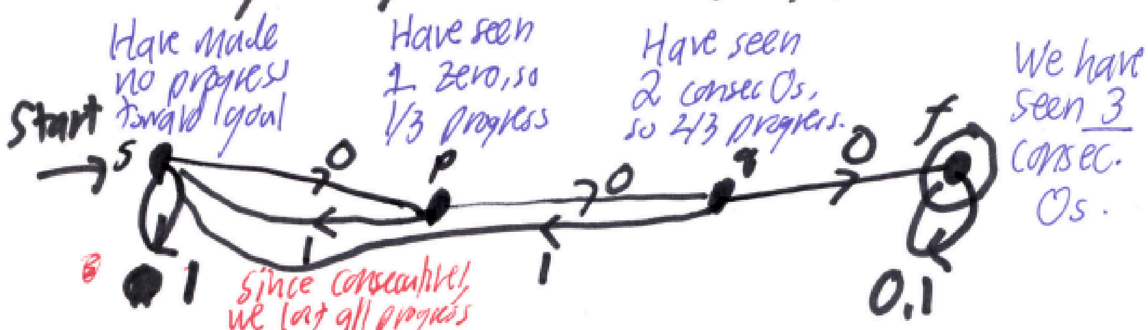
Another DFA Example

Let $A = \{x \in \{0,1\}^* : x$ does <u>not</u> have 3 consecutive 0s in it$\}$.

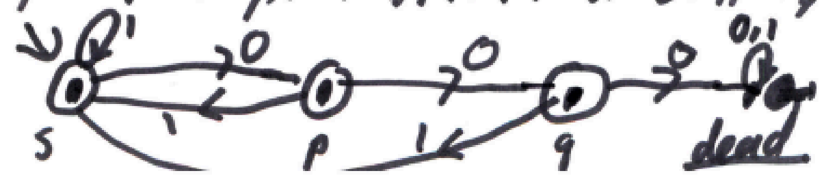
Build a DFA M such that $L(M) = A$.

The complement of A, which I denote by $\sim A$ or $\tilde{A}$ (text $\overline{A}$) is $\{x \in \{0,1\}^* : x$ <u>does</u> have 3 consecutive 0s in it$\}$.

First design a "goal-oriented" DFA M' s.t. $L(M') = \tilde{A}$.



To get the original DFA M for $L(M)=A$, complement the accepting and rejection states.



This also exemplifies the general theorem that if we have a DFA M such that $L(M) = A$, then we can build a DFA M' such that $L(M) =$ the complement of A.