\[ \ell = \{ \text{all } n \text{ is prime } \} \subseteq \mathbb{Z} \]

By P.L., \( \exists p \neq \ell \) ...

Let \( q \) be some prime \( \neq p \)

\[ \exists x, y \in \mathbb{Z} \]

\[ \ell q = xy \]

\[ \ell y \not\in 0 \]

\[ 1 \leq \ell y \leq p \]

\[ xy^i z \leq \ell \forall i \geq 0 \]

\[ 0 \leq 1 \ell \leq p \]

\[ 1 \leq \ell y \leq p \]

\[ 1 \ell = k \]

\[ 1 y \ell = d \]

\[ \ell y z = a x d \ell a^k \}

\[ \ell = 8 - k - j \]
Try 1: \( c = g + k \)

\[
xyz = a^k a^{j(g+k)} a^l = a^{k+jg+jk+l}
\]

Try 2

\[
z = 26k - 1
\]

Try 3

\[
i = k + 1
\]

Try 4

\[
i = g + 1
\]

\[
xyz = a^k a^{i(g+l)} a^l = a^{g+jg+l}
\]

Promising, but if \( k+l = 1 \), it's original expression (could be fixed by choosing \( g > p+2 \))

\[
xyz = a^{g+jg+l} = a^g a^j a^l
\]

\[
= 2^j 8^{j+1}
\]

\[
= 8 (j+1)
\]

Success \((14120)\), \( j+1 \neq 1 \) (8 \((j+1)\) is composite)
\[ \Sigma = \{ (, ) \} \]

\[ L = \Sigma \cup \{ \text{parameters balanced} \} \]

\[ \exists, (, ) , (\text{not } ) ) \]

\[ \exists \leq 3 \]

if \( L \) is regular, so is \( L' \)

\[ L' = L \cup \{ (* *) \} \]

\[ L' = \{ (n)^n \mid n \geq 0 \} \]

\[ \subseteq \{ a^m b^n \mid m \geq 3 \} \]
C - (the programming language)

C satisfies the pumping lemma

```
main () {} return (<<<<<0>>>>) ;
```

If regular \( \exists P \) and

\( \forall x \in \Sigma^* , y \in \Sigma^+ \) pumps nearly - get new fn. names

But C is not regular

```
main () {} return (<<<0>>>); \exists P
```

\( L = C \cap L^2 \cap L^3 \cap L^4 \cap \cdots \)

\( L \) is not regular \( \exists P \) ...

\( \delta = \) main () {} return (Po); \exists

Then if \( y \notin \Sigma^* \), \exists \( a \) given invalid prefix

\( y \in \Sigma^* \), \exists \( a \) given unbalanced prefix
Given a regular language $L$ and a string $x$, how hard is it to decide:

- $x \in L$?
- $L = \emptyset$?
- $L = \Sigma^*$?

A key issue: how is $L$ (in general an infinite thing) "given" as input to our program? Some options:

1. DFA
2. NFA
4. A Java program

Does it matter?