Example of a Deterministic Finite Automaton (DFA): Check whether parity of a binary string is an odd number of 1s.

```
00101001
00101001
```

Components are:
- States $Q = \{ \text{even, odd} \}$ means: Parity of the 1's read thus far.
- Start in even because zero is an even number.
- Goal: state odd $F = \{ \text{odd} \}$ desired.
- Characters $\Sigma = \{ 0, 1 \}$
- Rules $\delta$ shown in a picture.

$$M = \begin{align*}
\begin{array}{c}
\text{even} \\
\text{odd}
\end{array} \quad \begin{array}{l}
\text{o/0} \\
\text{1/1}
\end{array}
\end{align*}
$$

$L(M)$, the language of $M$, is the set of binary strings that cause $M$ to end in a goal state when started at $s$.

Formal Def in "Tuple Style" pioneered for math in the 1920s. Example: A Field is a 5-tuple $(F, +, \cdot, 0, 1)$ where [laws of + and \cdot, identity, div, etc. hold].
Def: A DFA is a 5-tuple \( M = (Q, \Sigma, \delta, s, F) \) where:
- \( Q \) is a finite set of states
- \( s \), a member of \( Q \), is the start state
- \( F \), a subset of \( Q \), is the set of final states
- \( \Sigma \) is a finite alphabet of characters, and
- \( \delta \) is a function from \( Q \times \Sigma \) to \( Q \).

Example had:
- \( \delta(\text{even}, 0) = \text{even} \)
- \( \delta(\text{odd}, 0) = \text{odd} \)
- \( \delta(\text{even}, 1) = \text{odd} \)
- \( \delta(\text{odd}, 1) = \text{even} \)

\[ \delta : Q \times \Sigma \rightarrow Q \]

Typical argument:
- \( \delta(p, c) = q \)
- \( p, q : \text{states} \)
- \( c : \text{char} \)

Graphically:
- \( p \rightarrow \bullet \rightarrow q \)
- \( q = p \) allowed
- Then it's a "loop"

View as an instruction \((p, c, q)\).

Example:
- \( \delta = \{ (\text{even}, 0, \text{even}), (\text{odd}, 0, \text{odd}), (\text{even}, 1, \text{odd}), (\text{odd}, 1, \text{even}) \} \)

(Justified on next slide)

Benefits: Same notation is also good for NFAs to come soon.
class DFA {
    Set<State> Q;
    State s; //start state
    Set<State> F;
    set<char> Sigma;
    State(delta)(State p, char c);
}

Better

Set<Tuple> delta: where

Tuple is a type for ((Q x Sigma) x Q)

If \( S(p, c) = q \), write the instruction as \( (p, c, q) \)

Every function \( f: A \rightarrow B \) is formally a relation \( R_f \subseteq A \times B \),

\( R_f = \{ (a, b) \mid b = f(a) \} \), such that \( (\forall a \in A)(\exists! b \in B) (a, b) \in R_f \)

Here, \( A = Q \times \Sigma \), \( B = Q \).

M will be a DFA if its delta set of instructions has this functional property.

Clearly, set(Tuple) delta is a class member

- can be different for different DFAs.

[The Friday 8/31 lecture was a demo of the "Turing Kit"]