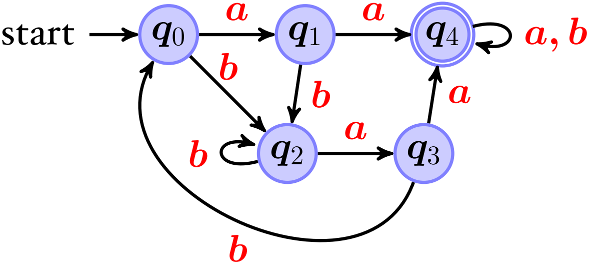
Automata and Formal Languages (5)

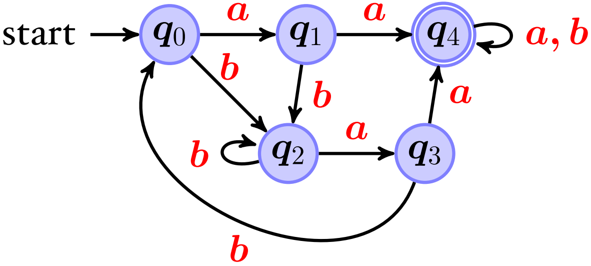
Email:	christian.urban at kcl.ac.uk
Office:	S1.27 (1st floor Strand Building)
Slides:	KEATS (also home work is there)

DFA Minimisation

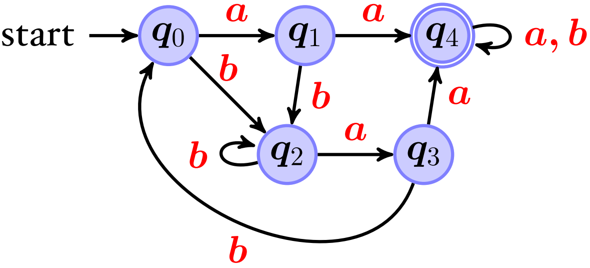
1. Take all pairs (q, p) with $q \neq p$
2. Mark all pairs that accepting and non-accepting states
3. For all unmarked pairs (q, p) and all characters c tests wether
   $$(\delta(q, c), \delta(p, c))$$
   are marked. If yes, then also mark (q, p).
4. Repeat last step until no chance.
5. All unmarked pairs can be merged.



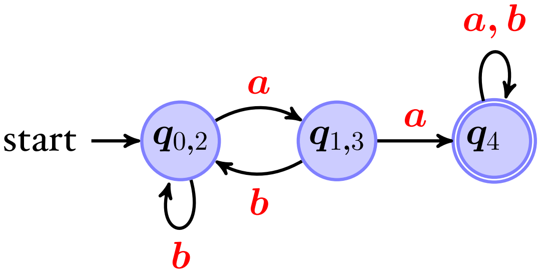
	q_0	q_1	q_2	q_3
q_1				
q_2				
q_3				
q_4	⋆	⋆	⋆	⋆



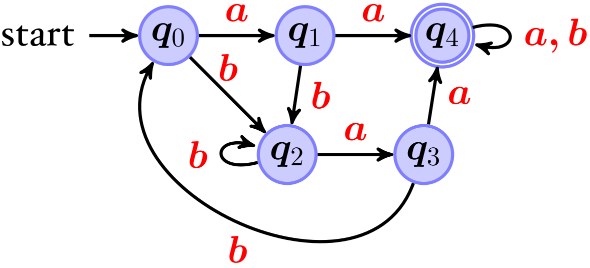
	q_0	q_1	q_2	q_3
q_1				
q_2				
q_3				
q_4	⋆	⋆	⋆	⋆



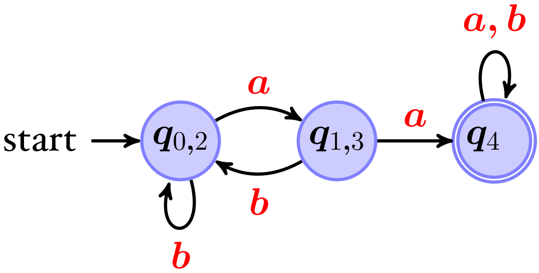
q_1	★			
q_2		★		
q_3	★		★	
q_4	★	★	★	★
	q_0	q_1	q_2	q_3



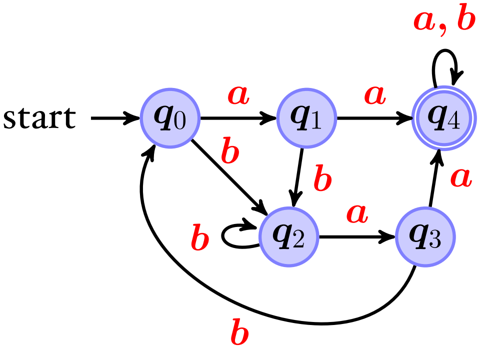
minimal automaton

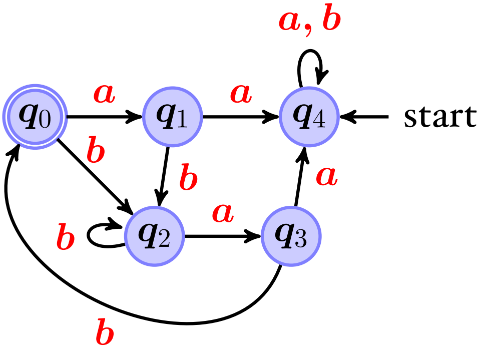


	q_0	q_1	q_2	q_3
q_1	★			
q_2		★		
q_3	★		★	
q_4	★	★	★	★

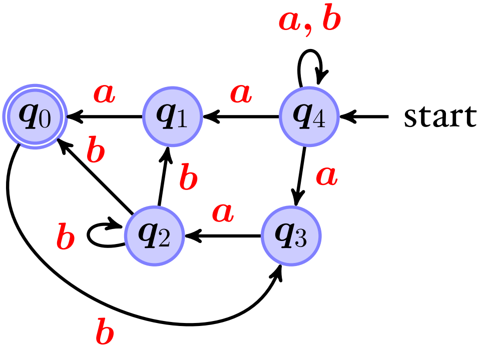


minimal automaton

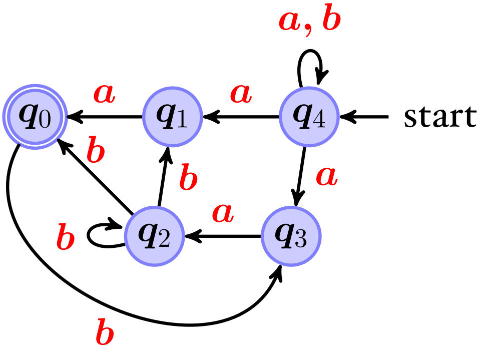

start
q_0
q_1
q_2
q_3
q_4
a
a
b
b
a
a
b
b
a, b



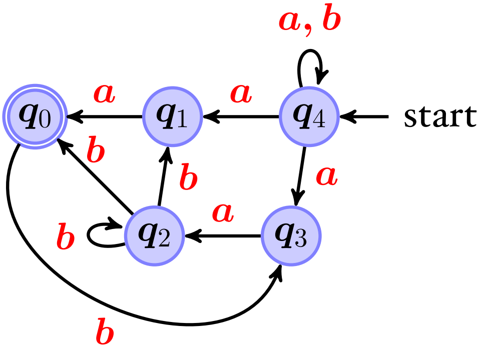
- exchange initial / accepting states



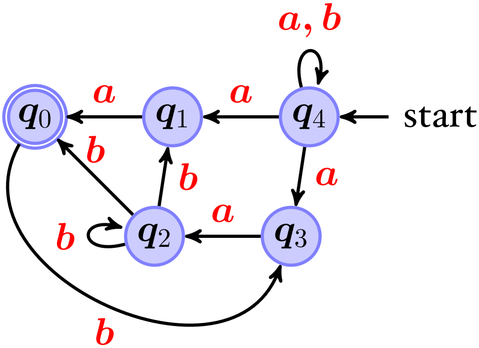
- exchange initial / accepting states
- reverse all edges



- exchange initial / accepting states
- reverse all edges
- subset construction ⇒ DFA



- exchange initial / accepting states
- reverse all edges
- subset construction ⇒ DFA
- repeat once more



- exchange initial / accepting states
- reverse all edges
- subset construction ⇒ DFA
- repeat once more ⇒ minimal DFA

```
write ”Input a number ”;
read n;
x := 0;
y := 1;
while n > 0 do {
  temp := y;
  y := x + y;
  x := temp;
  n := n - 1
};
write ”Result ”;
write y
write ”Input a number ”;
read n;
while n > 1 do {
  if n % 2 == 0
  then n := n/2
  else n := 3*n+1;
};
write ”Yes”;
```

Grammars

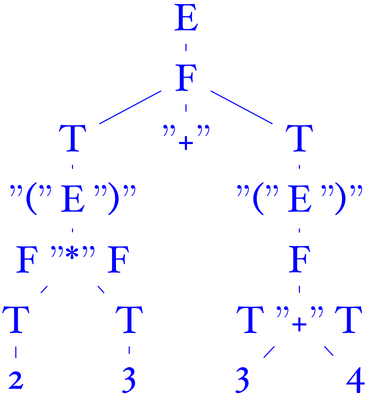
$$\begin{array}{lcl}
\boldsymbol{E} & \rightarrow & \boldsymbol{F} + (\boldsymbol{F} \cdot {''}*{''} \cdot \boldsymbol{F}) + (\boldsymbol{F} \cdot {''}\backslash{''} \cdot \boldsymbol{F}) \\
\boldsymbol{F} & \rightarrow & \boldsymbol{T} + (\boldsymbol{T} \cdot {''}{+}{''} \cdot \boldsymbol{T}) + (\boldsymbol{T} \cdot {''}\text{-}{''} \cdot \boldsymbol{T}) \\
\boldsymbol{T} & \rightarrow & \boldsymbol{num} + ({''}({''} \cdot \boldsymbol{E} \cdot {''}){''})
\end{array}$$

$\boldsymbol{E}$, $\boldsymbol{F}$ and $\boldsymbol{T}$ are non-terminals
$\boldsymbol{E}$ is start symbol
$\boldsymbol{num}$, (,), + ...are terminals

$(2*3)+(3+4)$

$$\begin{array}{lcl} \boldsymbol{E} & \rightarrow & \boldsymbol{F} + (\boldsymbol{F} \cdot {''}*{''} \cdot \boldsymbol{F}) + (\boldsymbol{F} \cdot {''}\backslash{''} \cdot \boldsymbol{F}) \\ \boldsymbol{F} & \rightarrow & \boldsymbol{T} + (\boldsymbol{T} \cdot {''}+{''} \cdot \boldsymbol{T}) + (\boldsymbol{T} \cdot {''}\text{-}{''} \cdot \boldsymbol{T}) \\ \boldsymbol{T} & \rightarrow & \boldsymbol{num} + ({''}({''} \cdot \boldsymbol{E} \cdot {''}){''}) \end{array}$$

(2*3)+(3+4)



E
F
T
”+”
T
”(” E ”)”
”(” E ”)”
F ”*” F
F
T
T
T ”+” T
2
3
3
4