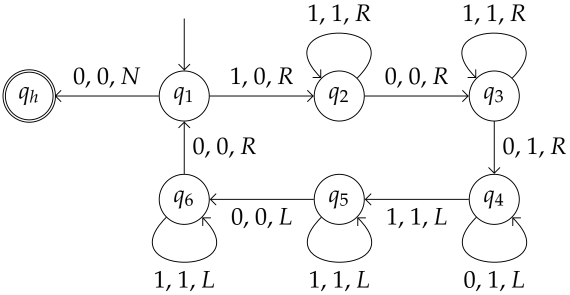
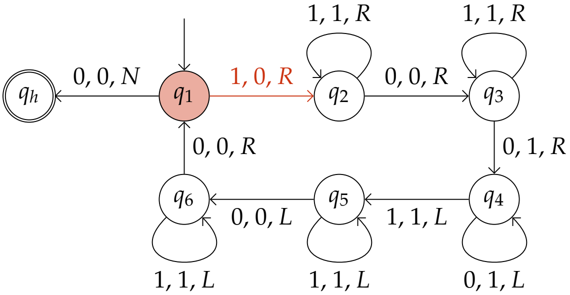
PHIL 222
Philosophical Foundations of Computer Science
Week 5, Tuesday

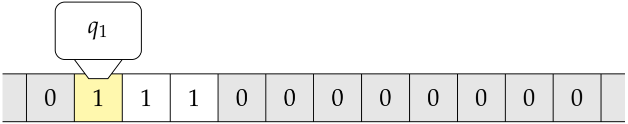
Sept. 24, 2024

Technical Exercise 1
Review

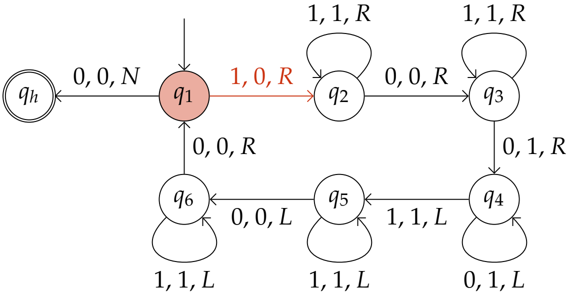

1, 1, R
1, 1, R
0, 0, N
1, 0, R
0, 0, R
q_h
q_1
q_2
q_3
0, 0, R
0, 1, R
q_6
q_5
q_4
0, 0, L
1, 1, L
1, 1, L
1, 1, L
0, 1, L

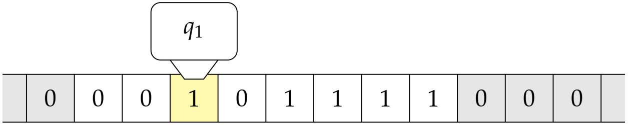
$q_1 111$


q_h
0, 0, N
q_1
1, 0, R
q_2
1, 1, R
0, 0, R
q_3
1, 1, R
0, 1, R
q_4
0, 1, L
1, 1, L
q_5
1, 1, L
0, 0, L
q_6
1, 1, L
0, 0, R


q_1
0 1 1 1 0 0 0 0 0 0 0 0

$q_1 111$

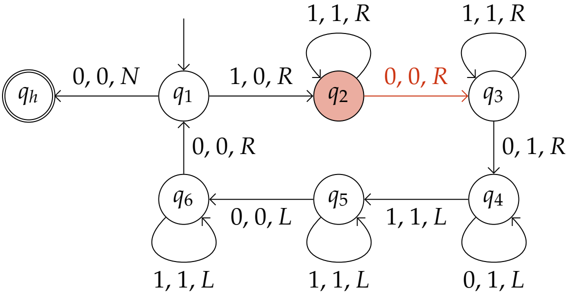


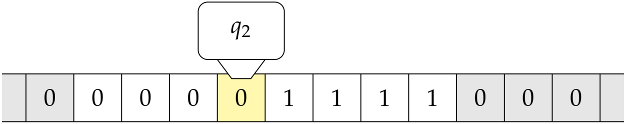


$q_1 111$

$\vdots$

$00 q_1 101111$

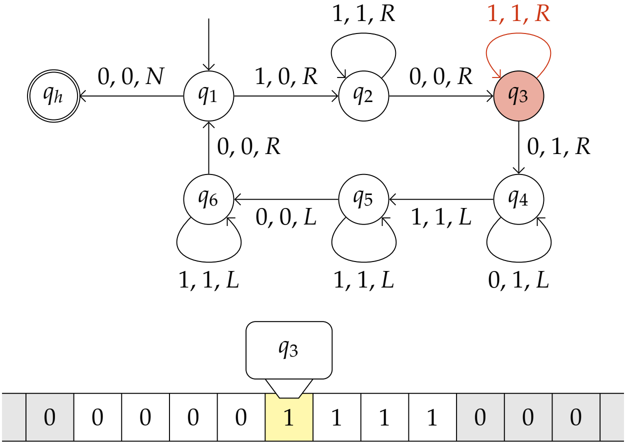




q_1111 $\quad$ $000q_201111$

$\vdots$

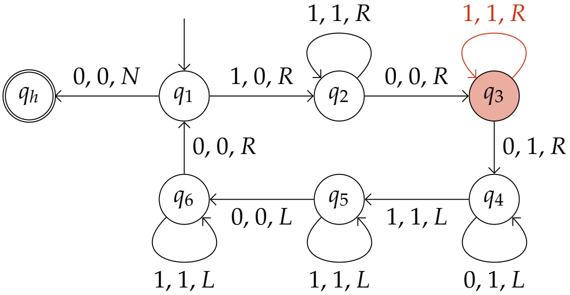
$00q_1101111$

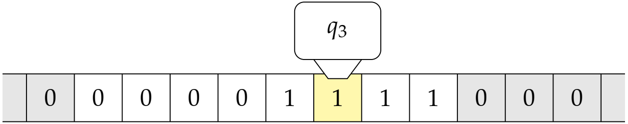


$q_1 111$ $\qquad$ $000q_2 01111$

$\vdots$ $\qquad$ $0000q_3 1111$

$00q_1 101111$





$q_1 111$

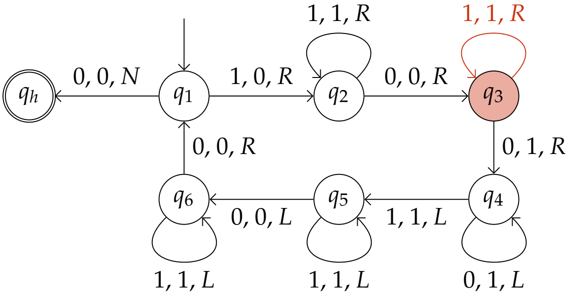
$\vdots$

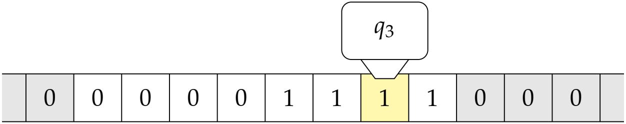
$00q_1 101111$

$000q_2 01111$

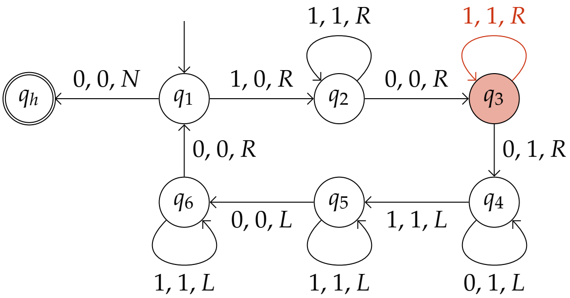
$0000q_3 1111$

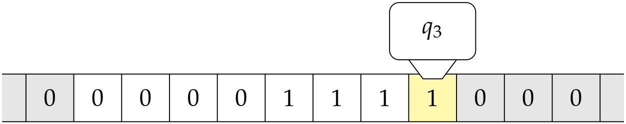
$00001q_3 111$



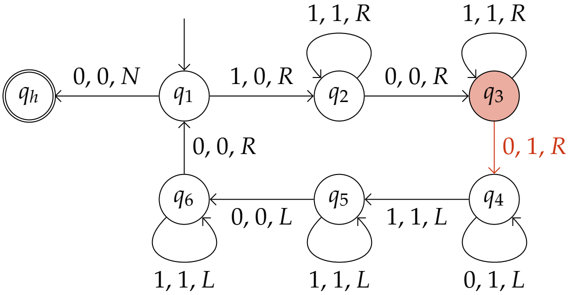


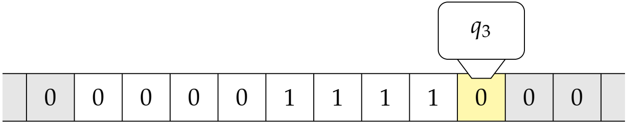
$q_1 111$	$000q_2 01111$
$\vdots$	$0000q_3 1111$
	$00001q_3 111$
$00q_1 101111$	$000011q_3 11$





$q_1 111$	$000q_2 01111$	$0000111q_3 1$
$\vdots$	$0000q_3 1111$	
	$00001q_3 111$	
$00q_1 101111$	$000011q_3 11$	





$q_1 111$

$\vdots$

$00q_1 101111$

$000q_2 01111$

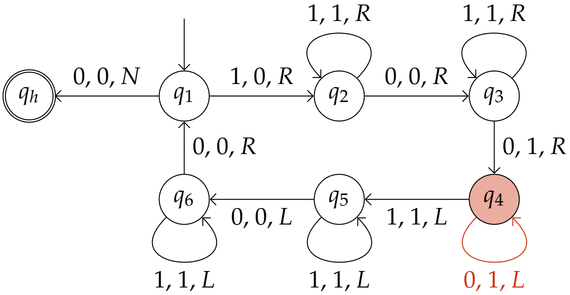
$0000q_3 1111$

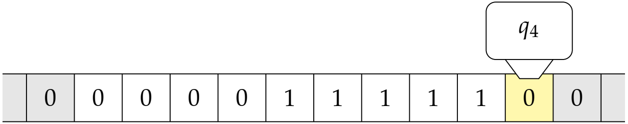
$00001q_3 111$

$000011q_3 11$

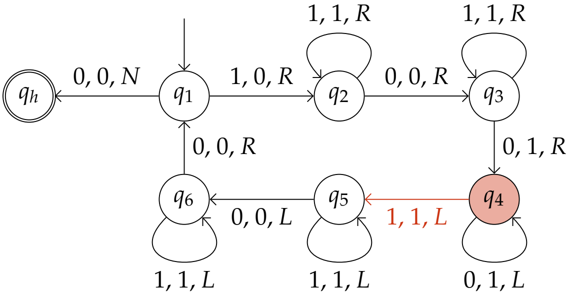
$0000111q_3 1$

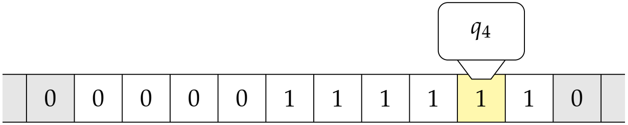
$00001111q_3 0$



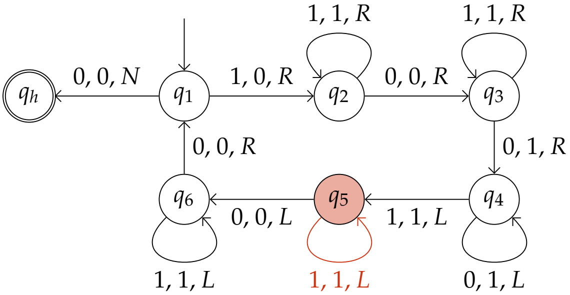


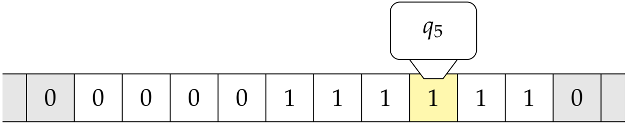
$q_1 111$	$000q_2 01111$	$0000111q_3 1$
$\vdots$	$0000q_3 1111$	$00001111q_3 0$
	$00001q_3 111$	$000011111q_4 0$
$00q_1 101111$	$000011q_3 11$	



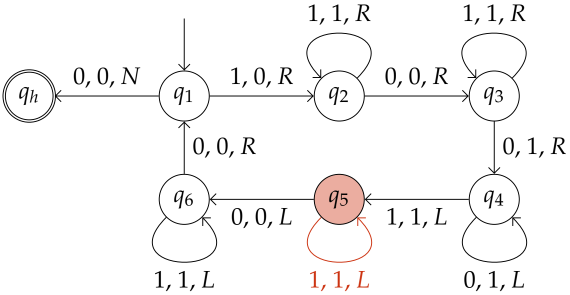


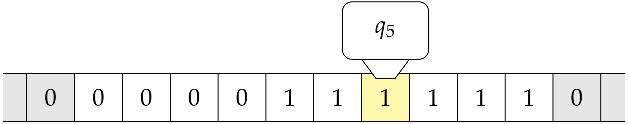
$q_1 111$	$000q_2 01111$	$0000111q_3 1$
$\vdots$	$0000q_3 1111$	$00001111q_3 0$
	$00001q_3 111$	$000011111q_4 0$
$00q_1 101111$	$000011q_3 11$	$00001111q_4 11$



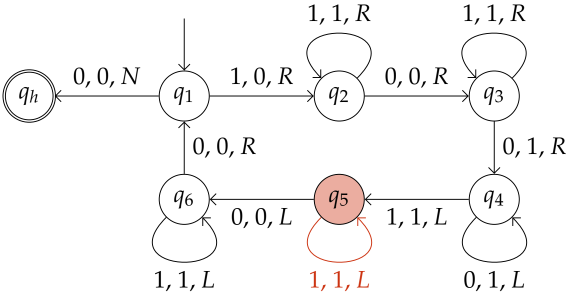


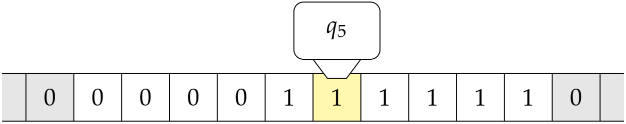
q_1111	$000q_201111$	$0000111q_31$	$0000111q_5111$
$\vdots$	$0000q_31111$	$00001111q_30$	
	$00001q_3111$	$000011111q_40$	
$00q_1101111$	$000011q_311$	$00001111q_411$	



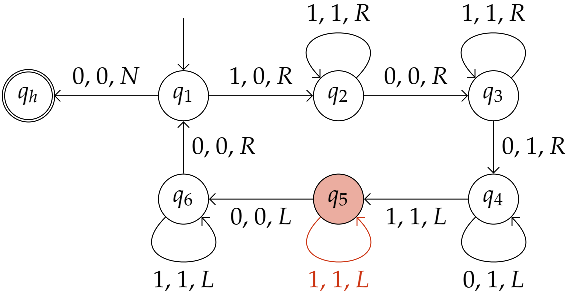


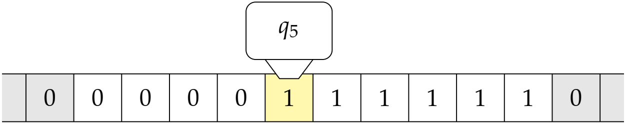
q_1111	$000q_201111$	$0000111q_31$	$0000111q_5111$
$\vdots$	$0000q_31111$	$00001111q_30$	$000011q_51111$
	$00001q_3111$	$000011111q_40$	
$00q_1101111$	$000011q_311$	$00001111q_411$	



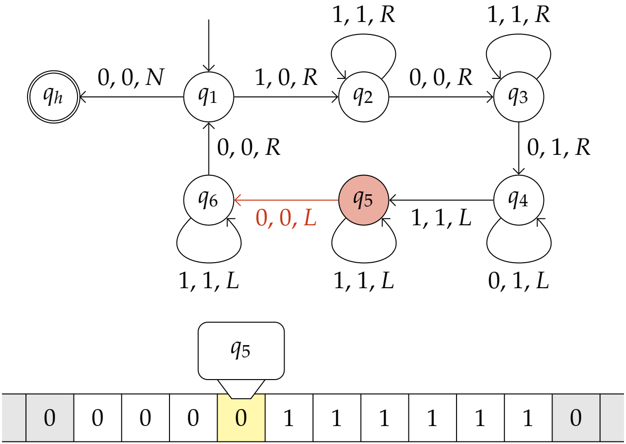


$q_1 111$	$000q_2 01111$	$0000111q_3 1$	$0000111q_5 111$
$\vdots$	$0000q_3 1111$	$00001111q_3 0$	$000011q_5 1111$
	$00001q_3 111$	$000011111q_4 0$	$00001q_5 11111$
$00q_1 101111$	$000011q_3 11$	$00001111q_4 11$	

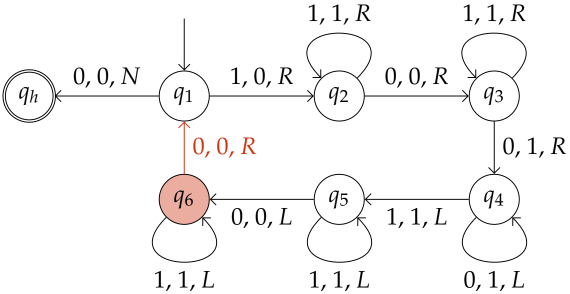


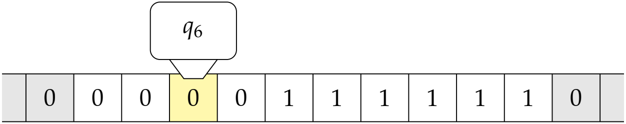


$q_1 111$	$000q_2 01111$	$0000111q_3 1$	$0000111q_5 111$
$\vdots$	$0000q_3 1111$	$00001111q_3 0$	$000011q_5 1111$
	$00001q_3 111$	$000011111q_4 0$	$00001q_5 11111$
$00q_1 101111$	$000011q_3 11$	$00001111q_4 11$	$0000q_5 111111$

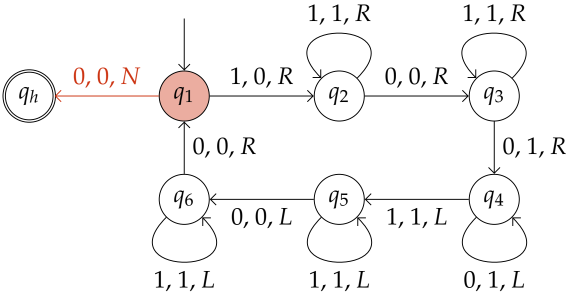


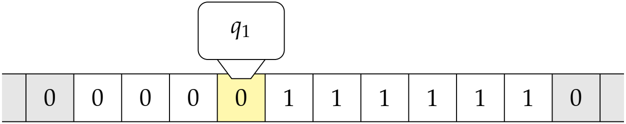
$q_1 111$	$000q_2 01111$	$0000111q_3 1$	$0000111q_5 111$	$000q_5 0111111$
$\vdots$	$0000q_3 1111$	$00001111q_3 0$	$000011q_5 1111$	
	$00001q_3 111$	$000011111q_4 0$	$00001q_5 11111$	
$00q_1 101111$	$000011q_3 11$	$00001111q_4 11$	$0000q_5 111111$	



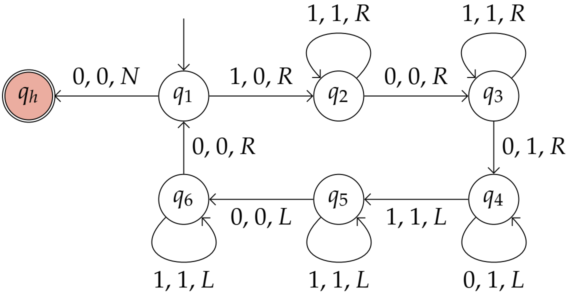


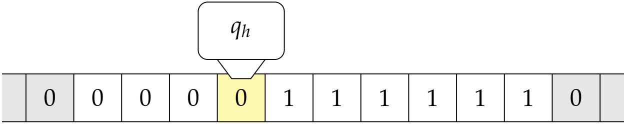
q_1111	$000q_201111$	$0000111q_31$	$0000111q_5111$	$000q_50111111$
$\vdots$	$0000q_31111$	$00001111q_30$	$000011q_51111$	$00q_600111111$
	$00001q_3111$	$000011111q_40$	$00001q_511111$	
$00q_1101111$	$000011q_311$	$00001111q_411$	$0000q_5111111$	



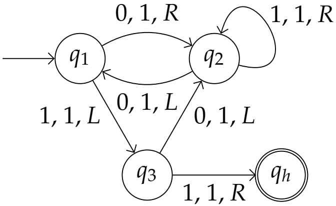


$q_1 111$	$000q_2 01111$	$0000111q_3 1$	$0000111q_5 111$	$000q_5 0111111$
$\vdots$	$0000q_3 1111$	$00001111q_3 0$	$000011q_5 1111$	$00q_6 00111111$
	$00001q_3 111$	$000011111q_4 0$	$00001q_5 11111$	$000q_1 0111111$
$00q_1 101111$	$000011q_3 11$	$00001111q_4 11$	$0000q_5 111111$	

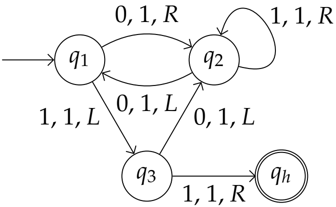




$q_1 111$	$000 q_2 01111$	$0000111 q_3 1$	$0000111 q_5 111$	$000 q_5 0111111$
$\vdots$	$0000 q_3 1111$	$00001111 q_3 0$	$000011 q_5 1111$	$00 q_6 00111111$
	$00001 q_3 111$	$000011111 q_4 0$	$00001 q_5 11111$	$000 q_1 0111111$
$00 q_1 101111$	$000011 q_3 11$	$00001111 q_4 11$	$0000 q_5 111111$	$000 q_h 0111111$

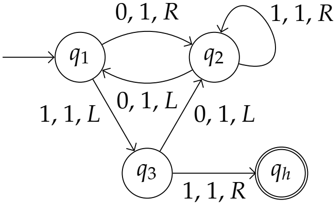


$q_1 0$



$q_1 0$

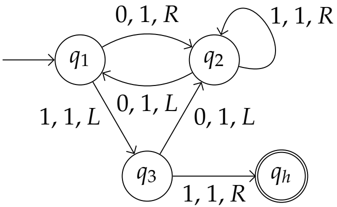
$1 q_2 0$



$q_1 0$

$1 q_2 0$

$q_1 11$

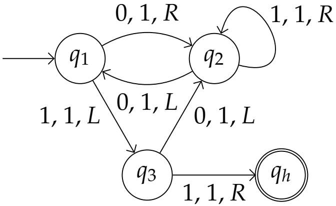


$q_1 0$

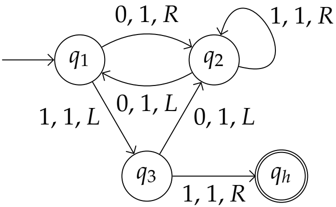
$1 q_2 0$

$q_1 11$

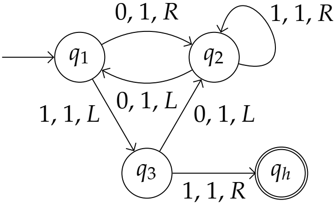
$q_3 011$



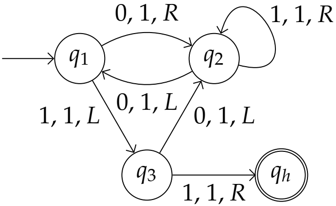
$$q_1 0$$
$$1 q_2 0$$
$$q_1 11$$
$$q_3 011$$
$$q_2 0111$$



$$q_1 0$$
$$1 q_2 0$$
$$q_1 11$$
$$q_3 011$$
$$q_2 0111$$
$$q_1 01111$$



$$
\begin{array}{r}
q_1 0 \\
1 q_2 0 \\
q_1 11 \\
q_3 011 \\
q_2 0111 \\
q_1 01111 \\
1 q_2 1111
\end{array}
$$



$q_1 0$

$1 q_2 0$

$q_1 11$

$q_3 011$

$q_2 0111$

$q_1 01111$

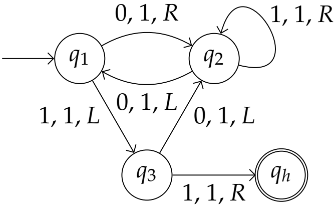
$1 q_2 1111$

$11 q_2 111$

$111 q_2 11$

$1111 q_2 1$

$11111 q_2 0$



$q_1 0$

$1 q_2 0$

$q_1 11$

$q_3 011$

$q_2 0111$

$q_1 01111$

$1 q_2 1111$

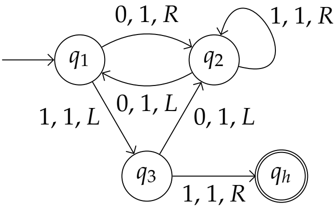
$11 q_2 111$

$111 q_2 11$

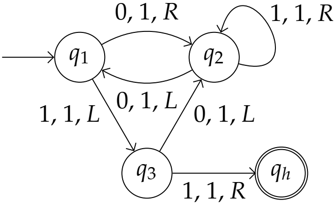
$1111 q_2 1$

$11111 q_2 0$

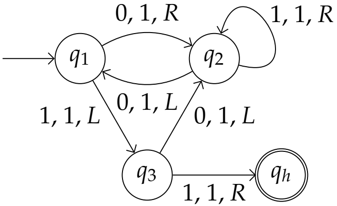
$1111 q_1 11$



$q_1 0$	$11 q_2 111$
$1 q_2 0$	$111 q_2 11$
$q_1 11$	$1111 q_2 1$
$q_3 011$	$11111 q_2 0$
$q_2 0111$	$1111 q_1 11$
$q_1 01111$	$111 q_3 111$
$1 q_2 1111$	



$q_1 0$	$11 q_2 111$
$1 q_2 0$	$111 q_2 11$
$q_1 11$	$1111 q_2 1$
$q_3 011$	$11111 q_2 0$
$q_2 0111$	$1111 q_1 11$
$q_1 01111$	$111 q_3 111$
$1 q_2 1111$	$1111 q_h 11$



$q_1 0$	$11q_2111$
$1q_20$	$111q_211$
q_111	$1111q_21$
q_3011	$11111q_20$
q_20111	$1111q_111$
q_101111	$111q_3111$
$1q_21111$	$1111q_h11$

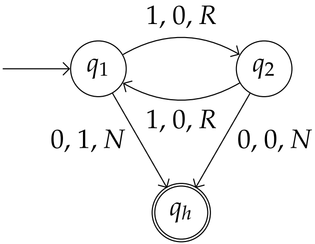
This Turing machine is called a “3-state busy beaver”.

$$f(n) = \begin{cases} 1 & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd.} \end{cases}$$

$$f(n) = \begin{cases} 1 & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd.} \end{cases}$$

E.g.,

We use the representation in which a string of n strokes refers to n. Take the following Turing machine, and feed it with a tape with a string of n strokes, placing the scanner on the leftmost stroke.



Then the machine halts with $f(n)$ stroke.

The Lambda Calculus
(cont'd)

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n\text{ times}}\,y))))))$$

f — $\boxed{\bar{n}}$ — f^n, i.e., applying f n times

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\, (\lambda y.\, (\underbrace{x(x(x(\cdots(x}_{n \text{ times}}\, y))))))$$

$$f \;\text{---}\; \boxed{\bar{n}} \;\text{---}\; f^n, \text{ i.e., applying } f \ n \text{ times}$$

$$(\bar{3} f)\, a = ((\lambda x.\, (\lambda y.\, (x(x(xy)))))\, f)\, a$$

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n\text{ times}}\,y))))))$$

$$f \;-\!\!\!-\!\!\!-\; \boxed{\bar{n}} \;-\!\!\!-\!\!\!-\; f^n, \text{ i.e., applying } f \; n \text{ times}$$

$$(\bar{3}f)\,a = ((\lambda x.\,(\lambda y.\,(x(x(xy)))))\,f)\,a$$

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n \text{ times}}\,y))))))$$

$$f \;\text{———}\; \boxed{\bar{n}} \;\text{———}\; f^{n}\text{, i.e., applying } f \ n \text{ times}$$

$$(\bar{3}f)\,a = ((\lambda x.\,(\lambda y.\,(x(x(xy)))))\,f)\,a$$

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n \text{ times}}\,y))))))$$

$$f \;\text{---}\; \boxed{\bar{n}} \;\text{---}\; f^n, \text{ i.e., applying } f \; n \text{ times}$$

$$\begin{aligned}(\bar{3}f)\,a &= ((\lambda x.\,(\lambda y.\,(x(x(xy)))))\,f)\,a \\ &\to (\lambda y.\,(f(f(fy))))\,a\end{aligned}$$

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n\text{ times}}\,y))))))$$

$$f \;—\; \boxed{\bar{n}} \;—\; f^n\text{, i.e., applying } f \; n \text{ times}$$

$$\begin{aligned}(\bar{3}f)\,a &= ((\lambda x.\,(\lambda y.\,(x(x(xy)))))\,f)\,a \\ &\to (\lambda y.\,(f(f(fy))))\,a\end{aligned}$$

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n \text{ times}}\,y))))))$$

$$f \;\text{——}\; \boxed{\bar{n}} \;\text{——}\; f^n, \text{ i.e., applying } f \; n \text{ times}$$

$$\begin{aligned} (\bar{3}f)\,a &= ((\lambda x.\,(\lambda y.\,(x(x(xy)))))\,f)\,a \\ &\to \quad (\lambda y.\,(f(f(fy))))\quad a \end{aligned}$$

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n\text{ times}}\,y))))))$$

$$f \;\text{——}\; \boxed{\bar{n}} \;\text{——}\; f^{n}\text{, i.e., applying } f \; n \text{ times}$$

$$\begin{aligned}
(\bar{3}f)\,a &= ((\lambda x.\,(\lambda y.\,(x(x(xy)))))\,f)\,a \\
&\to (\lambda y.\,(f(f(fy))))\quad a \\
&\to f(f(fa))
\end{aligned}$$

Church numerals. In this calculus everything is a function, so we need functions that stand for natural numbers.

$$\bar{n} := \lambda x.\,(\lambda y.\,(\underbrace{x(x(x(\cdots(x}_{n \text{ times}}\,y))))))$$

$$f \;\text{---}\; \boxed{\bar{n}} \;\text{---}\; f^n\text{, i.e., applying } f \; n \text{ times}$$

$$\begin{aligned}(\bar{3}f)\,a &= ((\lambda x.\,(\lambda y.\,(x(x(xy)))))\,f)\,a\\ &\to \quad (\lambda y.\,(f(f(fy))))\quad a\\ &\to \quad f(f(fa))\end{aligned}$$

Definition. We say that a term F of the lambda calculus "λ-defines" a k-argument partial function f of natural numbers if

- whenever $f(n_1, \ldots, n_k)$ is defined and equals n,

$$((((F\overline{n_1})\,\overline{n_2})\cdots)\,\overline{n_k}) \to \cdots \to \bar{n},$$

- but when $f(n_1, \ldots, n_k)$ is undefined, the β-reduction of $((((F\overline{n_1})\,\overline{n_2})\cdots)$ never terminates.

E.g., Add $:= \lambda u.\, (\lambda v.\, (\lambda x.\, (\lambda y.\, ((ux)((vx)\, y)))))$ defines $+$.

E.g., Add := $\lambda u. (\lambda v. (\lambda x. (\lambda y. ((ux)((vx)\, y)))))$ defines +.

Here is how the computation of 2 + 3 = 5 goes:

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$(\mathrm{Add}\,\bar{2})\,\bar{3} = ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3}$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$(\mathrm{Add}\,\bar{2})\,\bar{3} = ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))\,\bar{2})\,\bar{3}$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$(\mathrm{Add}\,\bar{2})\,\bar{3} = ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3}$$

E.g., Add $:= \lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines $+$.

Here is how the computation of $2 + 3 = 5$ goes:

$$
\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))\,\bar{2})\,\bar{3} \\
&\to \quad (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3}
\end{aligned}
$$

E.g., Add $:= \lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of 2 + 3 = 5 goes:

$$
\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
&\to \qquad (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y))))) \qquad \bar{3}
\end{aligned}
$$

E.g., Add := $\lambda u. (\lambda v. (\lambda x. (\lambda y. ((ux)((vx)\, y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$\begin{aligned}(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u. (\lambda v. (\lambda x. (\lambda y. ((ux)((vx)\, y))))))\,\bar{2})\,\bar{3} \\ &\to \quad (\lambda v. (\lambda x. (\lambda y. ((\bar{2}x)((vx)\, y)))))\quad \bar{3}\end{aligned}$$

E.g., Add $:= \lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
&\to \quad (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3} \\
&\to \quad \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y)))
\end{aligned}$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
&\to (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3} \\
&\to \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y)))
\end{aligned}$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$
\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
&\to (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3} \\
&\to \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y))) \\
\to \cdots &\to \lambda x.\,(\lambda y.\,((\bar{2}x)(x(x(xy)))))
\end{aligned}
$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$
\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3}\\
&\to (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3}\\
&\to \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y)))\\
\to \cdots &\to \lambda x.\,(\lambda y.\,((\bar{2}x)(x(x(xy)))))
\end{aligned}
$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
&\to \quad (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3} \\
&\to \quad \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y))) \\
\to \cdots &\to \quad \lambda x.\,(\lambda y.\,((\bar{2}x)(x(x(xy))))) \\
\to \cdots &\to \quad \lambda x.\,(\lambda y.\,(x(x(x(x(xy))))))
\end{aligned}$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of $2 + 3 = 5$ goes:

$$\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
\to\quad &\qquad (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\qquad \bar{3} \\
\to\quad &\qquad\qquad \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y))) \\
\to \cdots \to\quad &\qquad\qquad \lambda x.\,(\lambda y.\,((\bar{2}x)(x(x(xy))))) \\
\to \cdots \to\quad &\qquad\qquad \lambda x.\,(\lambda y.\,(x(x(x(x(xy))))))
\end{aligned}$$

E.g., Add := $\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines +.

Here is how the computation of 2 + 3 = 5 goes:

$$\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
&\to (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3} \\
&\to \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y))) \\
\to \cdots &\to \lambda x.\,(\lambda y.\,((\bar{2}x)(x(x(xy))))) \\
\to \cdots &\to \lambda x.\,(\lambda y.\,(x(x(x(x(xy)))))) \\
&= \bar{5}
\end{aligned}$$

E.g., Add $:= \lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y)))))$ defines $+$.

Here is how the computation of $2 + 3 = 5$ goes:

$$
\begin{aligned}
(\mathrm{Add}\,\bar{2})\,\bar{3} &= ((\lambda u.\,(\lambda v.\,(\lambda x.\,(\lambda y.\,((ux)((vx)\,y))))))\,\bar{2})\,\bar{3} \\
&\to (\lambda v.\,(\lambda x.\,(\lambda y.\,((\bar{2}x)((vx)\,y)))))\quad \bar{3} \\
&\to \lambda x.\,(\lambda y.\,((\bar{2}x)((\bar{3}x)\,y))) \\
\to \cdots &\to \lambda x.\,(\lambda y.\,((\bar{2}x)(x(x(xy))))) \\
\to \cdots &\to \lambda x.\,(\lambda y.\,(x(x(x(x(xy)))))) \\
&= \bar{5}
\end{aligned}
$$

Theorem. For any partial function f of natural numbers,

$$
\begin{array}{ccc}
 & f \text{ is Turing computable} & \\
\Updownarrow & & \Updownarrow \\
f \text{ is partial recursive} & \Longleftrightarrow & f \text{ is } \lambda\text{-definable.}
\end{array}
$$

The Church-Turing Thesis

So we have three models of computation agreeing with each other:

$$\begin{tikzcd}[column sep=small] f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \arrow[dl, Leftrightarrow] \\ & f \text{ is Turing computable} & \end{tikzcd}$$

So we have three models of computation agreeing with each other:

$$\begin{array}{ccc} f \text{ is partial recursive} & \Longleftrightarrow & f \text{ is } \lambda\text{-definable.} \\ & \nwarrow\!\!\searrow \quad \swarrow\!\!\nearrow & \\ & f \text{ is Turing computable} & \end{array}$$

f "can be computed"

So we have three models of computation agreeing with each other:

$$\begin{tikzcd} f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \arrow[dl, Leftrightarrow] \\ & f \text{ is Turing computable} \arrow[d, Rightarrow] & \\ & f \text{ ``can be computed''} & \end{tikzcd}$$

So we have three models of computation agreeing with each other:

$$\begin{tikzcd}
f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \arrow[dl, Leftrightarrow] \\
& f \text{ is Turing computable} \arrow[d, Rightarrow, shift right] & \\
& f \text{ ``can be computed''} \arrow[u, Rightarrow, shift right] &
\end{tikzcd}$$

So we have three models of computation agreeing with each other:

$$\begin{tikzcd}[column sep=small]
f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \arrow[dl, Leftrightarrow] \\
& f \text{ is Turing computable} \arrow[d, Rightarrow, shift right] & \\
& f \text{ “can be computed”} \arrow[u, Rightarrow, shift right, blue] &
\end{tikzcd}$$

The “$\Longrightarrow$” is roughly what is called the **Church-Turing thesis**, but ...

So we have three models of computation agreeing with each other:

$$\begin{tikzcd} f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \arrow[dl, Leftrightarrow] \\ & f \text{ is Turing computable} \arrow[d, Rightarrow, shift right] & \\ & f \text{ ``can be computed''} \arrow[u, Rightarrow, shift right] & \end{tikzcd}$$

The "$\Longrightarrow$" is roughly what is called the **Church-Turing thesis**, but . . .

1. What is the right notion to be placed in "can be computed"?

So we have three models of computation agreeing with each other:

$$\begin{array}{ccc} f \text{ is partial recursive} & \Longleftrightarrow & f \text{ is } \lambda\text{-definable.} \\ & \nwarrow\!\!\searrow \quad \swarrow\!\!\nearrow & \\ & f \text{ is Turing computable} & \\ & \Downarrow \;\; \Uparrow & \\ & f \text{ “can be computed”} & \end{array}$$

The “$\Longrightarrow$” is roughly what is called the **Church-Turing thesis**, but . . .

1. What is the right notion to be placed in “can be computed”? “Human beings can do it”? “Some physical machines can do it”? But in what sort of ways? “There is an algorithm for it”?

So we have three models of computation agreeing with each other:

$$\begin{tikzcd}
f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \\
& f \text{ is Turing computable} \arrow[ur, Leftrightarrow] \arrow[d, Rightarrow, shift right] & \\
& f \text{ ``can be computed''} \arrow[u, Rightarrow, shift right] &
\end{tikzcd}$$

The "$\Longrightarrow$" is roughly what is called the **Church-Turing thesis**, but . . .

1. What is the right notion to be placed in "can be computed"? "Human beings can do it"? "Some physical machines can do it"? But in what sort of ways? "There is an algorithm for it"?
2. What type of method is supposed to justify "$\Longrightarrow$"?

So we have three models of computation agreeing with each other:

$$
\begin{tikzcd}
f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \arrow[dl, Leftrightarrow] \\
& f \text{ is Turing computable} \arrow[d, Rightarrow, shift right] & \\
& f \text{ “can be computed”} \arrow[u, Rightarrow, shift right] &
\end{tikzcd}
$$

The "$\Longrightarrow$" is roughly what is called the **Church-Turing thesis**, but ...

1. What is the right notion to be placed in "can be computed"?
   "Human beings can do it"? "Some physical machines can do it"?
   But in what sort of ways? "There is an algorithm for it"?
2. What type of method is supposed to justify "$\Longrightarrow$"?
   Math? Philosophy? Physics? Cataloging a zoo of computers?

So we have three models of computation agreeing with each other:

$$\begin{tikzcd}[column sep=small]
f \text{ is partial recursive} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & f \text{ is } \lambda\text{-definable.} \arrow[dl, Leftrightarrow] \\
& f \text{ is Turing computable} \arrow[d, Rightarrow, shift right] & \\
& f \text{ ``can be computed''} \arrow[u, Rightarrow, shift right] &
\end{tikzcd}$$

The "$\Longrightarrow$" is roughly what is called the **Church-Turing thesis**, but . . .

1. What is the right notion to be placed in "can be computed"?
   "Human beings can do it"? "Some physical machines can do it"?
   But in what sort of ways? "There is an algorithm for it"?

2. What type of method is supposed to justify "$\Longrightarrow$"?
   Math? Philosophy? Physics? Cataloging a zoo of computers?

And for each of 1 and 2, we can ask "What did Turing think?" and "What is the right answer?" as well as "Is the thesis then true"?

The Church-Turing Thesis: Church and Turing's Version

Let’s first consider what Turing (and Church) thought.

Let's first consider what Turing (and Church) thought.

Turing (1948):

> It is found in practice that [Turing machines] can do anything that could be described as 'rule of thumb' or 'purely mechanical'. This is sufficiently well established that it is now agreed amongst logicians that 'calculable by means of [a Turing machine]' is the correct accurate rendering of such phrases.

Church (1936):

> We now define the notion, already discussed, of an *effectively calculable* function of positive integers by identifying it with the notion of a recursive function of positive integers (or of a λ-definable function of positive integers). This definition is thought to be justified by the considerations which follow, so far as positive justification can ever be obtained for the selection of a formal definition to correspond to an intuitive notion.

1. What do they mean by “can be done purely mechanically” or
“can be calculated / computed effectively”?

1. What do they mean by “can be done purely mechanically” or “can be calculated / computed effectively”?

They understood it as follows (quoting Copeland’s summary):

1. What do they mean by “can be done purely mechanically” or
“can be calculated / computed effectively”?

They understood it as follows (quoting Copeland’s summary):

> A method, or procedure, M, for achieving some desired result is called ‘effective’ (or ‘systematic’ or ‘mechanical’) just in case:
>
> a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
> b. M will, if carried out without error, produce the desired result in a finite number of steps;
> c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
> d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

This is why Copeland & Shagrir summarize the Church-Turing thesis (as Church and Turing conceived of it) as

> **"CTT-Original".** Every function that can be computed by the idealized human computer, which is to say, can be effectively computed, is Turing computable,

where "computed by the idealized human computer" is cashed out by a + b + c + d.

This is why Copeland & Shagrir summarize the Church-Turing thesis (as Church and Turing conceived of it) as

> **"CTT-Original".** Every function that can be computed by the idealized human computer, which is to say, can be effectively computed, is Turing computable,

where "computed by the idealized human computer" is cashed out by a + b + c + d.

(There are similar but different versions of the thesis introduced by others, either on purpose or by misunderstanding. We will discuss some of them later.)

2. What type of method do they think could justify the thesis?

2. What type of method do they think could justify the thesis?

Turing (1936):

> All arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically. [. . . .]
>
> The arguments which I shall use are of three kinds.
>
> (a) A direct appeal to intuition.
>
> (b) A proof of the equivalence of two definitions (in case the new definition has a greater intuitive appeal).
>
> (c) Giving examples of large classes of numbers which are computable.

2. What type of method do they think could justify the thesis?

Turing (1936):

> All arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically. [. . . .]
>
> The arguments which I shall use are of three kinds.
>
> (a) A direct appeal to intuition.
> (b) A proof of the equivalence of two definitions (in case the new definition has a greater intuitive appeal).
> (c) Giving examples of large classes of numbers which are computable.

Church (1936):

> This definition is thought to be justified by the considerations which follow, so far as positive justification can ever be obtained for the selection of a formal definition to correspond to an intuitive notion.

2. What type of method do they think could justify the thesis?

Turing (1936):

> All arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically. [. . . .]
>
> The arguments which I shall use are of three kinds.
>
> (a) A direct appeal to intuition.
>
> (b) A proof of the equivalence of two definitions (in case the new definition has a greater intuitive appeal).
>
> (c) Giving examples of large classes of numbers which are computable.

Church (1936):

> This definition is thought to be justified by the considerations which follow, so far as positive justification can ever be obtained for the selection of a formal definition to correspond to an intuitive notion.

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) *M* is called "effective" just in case

- a. *M* is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
- b. *M* will, if carried out without error, produce the desired result in a finite number of steps;
- c. *M* can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
- d. *M* demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Again, Turing and Church understood “purely mechanical” or “effective” as follows:

A method (or procedure) M is called “effective” just in case

- a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
- b. M will, if carried out without error, produce the desired result in a finite number of steps;
- c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
- d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) M is called "effective" just in case

a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
b. M will, if carried out without error, produce the desired result in a finite number of steps;
c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) M is called "effective" just in case

a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
b. M will, if carried out without error, produce the desired result in a finite number of steps;
c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) M is called "effective" just in case

a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
b. M will, if carried out without error, produce the desired result in a finite number of steps;
c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) M is called "effective" just in case

a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
b. M will, if carried out without error, produce the desired result in a finite number of steps;
c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood “purely mechanical” or “effective” as follows:

A method (or procedure) M is called “effective” just in case

a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
b. M will, if carried out without error, produce the desired result in a finite number of steps;
c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood “purely mechanical” or “effective” as follows:

A method (or procedure) M is called “effective” just in case

- (a) M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
- (b) M will, if carried out without error, produce the desired result in a finite number of steps;
- (c) M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
- (d) M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

> A method (or procedure) M is called "effective" just in case
>
> a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
> b. M will, if carried out without error, produce the desired result in a finite number of steps;
> c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
> d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) M is called "effective" just in case

- (a) M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
- (b) M will, if carried out without error, produce the desired result in a finite number of steps;
- (c) M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
- (d) M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) *M* is called "effective" just in case

a. *M* is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
b. *M* will, if carried out without error, produce the desired result in a finite number of steps;
c. *M* can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
d. *M* demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood "purely mechanical" or "effective" as follows:

A method (or procedure) M is called "effective" just in case

a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
b. M will, if carried out without error, produce the desired result in a finite number of steps;
c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

Again, Turing and Church understood “purely mechanical” or “effective” as follows:

A method (or procedure) M is called “effective” just in case

- a. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols);
- b. M will, if carried out without error, produce the desired result in a finite number of steps;
- c. M can (in practice or in principle) be carried out by a human being unaided by any machinery except paper and pencil;
- d. M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

Green: Do / can admit mathematical definition / treatment.

Red: Can they?

(d) M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

$$\text{(a)} + \text{(b)} + \text{(c)} + \text{(d)} \;\; \overset{\Longleftarrow}{\Longrightarrow} \;\; \text{Turing computable, etc.}$$

(d) M demands no insight, intuition, or ingenuity, on the part of the human being carrying out the method.

$$\text{(a)} + \text{(b)} + \text{(c)} + \text{(d)} \;\underset{\Longrightarrow}{\Longleftarrow}\; \text{Turing computable, etc.}$$

"an intuitive notion"

Turing (1936):

> All arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically.

Church (1936):

> This definition is thought to be justified by the considerations which follow, so far as positive justification can ever be obtained for the selection of a formal definition to correspond to an intuitive notion.