

Hasenjaeger's electromechanical small universal Turing machine is time efficient

Rainer Glaschick¹, Turlough Neary², Damien Woods³, **Niall Murphy**⁴

¹Paderborn, Germany

² **ETH** Institute for Neuroinformatics, University of Zürich and ETH Zürich, Switzerland

³ California Institute of Technology, USA

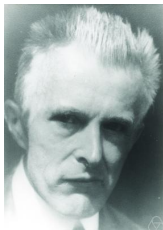
⁴ Universidad Politécnica de Madrid, CEI Campus Moncloa, UCM-UPM, Madrid, Spain

Turing in Context II, October 11, 2012

- Hasenjaeger, a (near) contemporary of Turing's
- Built electromechanical universal Turing machine that was remarkably small
- Simulated Wang's B-machine (non-erasing machine)
- We prove that Wang B-machines are a time efficient model of computation (an exponential improvement)
- As an immediate corollary we find that Hasenjaeger's small machine is efficiently universal

Some context

- 1936 “On Computable Numbers, with an Application to the *Entscheidungs* problem”
- Scholz in Münster and Braithwaite in Cambridge requested preprints
- Scholz was founding a mathematical logic group in Münster
- Undergraduate Hasenjaeger



Copyright MFO/CC BY-SA 2.0



- Scholz got him a place in the High Command cryptography group 1942
- Hasenjaeger looked for weaknesses in the Enigma machine
- After the war Hasenjaeger became Scholz's PhD student
- Became professor of Mathematical logic in Münster
- Didn't learn of Turing's involvement with Enigma until the 70's



Copyright: Reinhold Remmert, Oberwolfach Photo Collection

Hasenjaeger and the machine

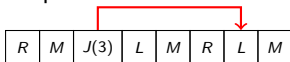
- Aspired to build a cheap electromechanical universal Turing machine

Hasenjaeger and the machine

- Aspired to build a cheap electromechanical universal Turing machine
- 1951, Moore gave the first small universal Turing machine
 - 2 regular infinite tapes, 1 circular tape
 - simulates standard TM rules, $(q_i, 1, 0, R, q_k)$
 - simple encoding system

Hasenjaeger and the machine

- Aspired to build a cheap electromechanical universal Turing machine
- 1951, Moore gave the first small universal Turing machine
 - 2 regular infinite tapes, 1 circular tape
 - simulates standard TM rules, $(q_i, 1, 0, R, q_k)$
 - simple encoding system
- 1957, Wang publishes his *B*-machine
 - non-erasing machine
 - simpler instruction set

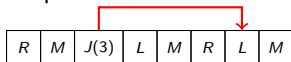


- complicated encoding (powers of primes)

Hasenjaeger and the machine

- Aspired to build a cheap electromechanical universal Turing machine
- 1951, Moore gave the first small universal Turing machine
 - 2 regular infinite tapes, 1 circular tape
 - simulates standard TM rules, $(q_i, 1, 0, R, q_k)$
 - simple encoding system

- 1957, Wang publishes his *B*-machine
 - non-erasing machine
 - simpler instruction set



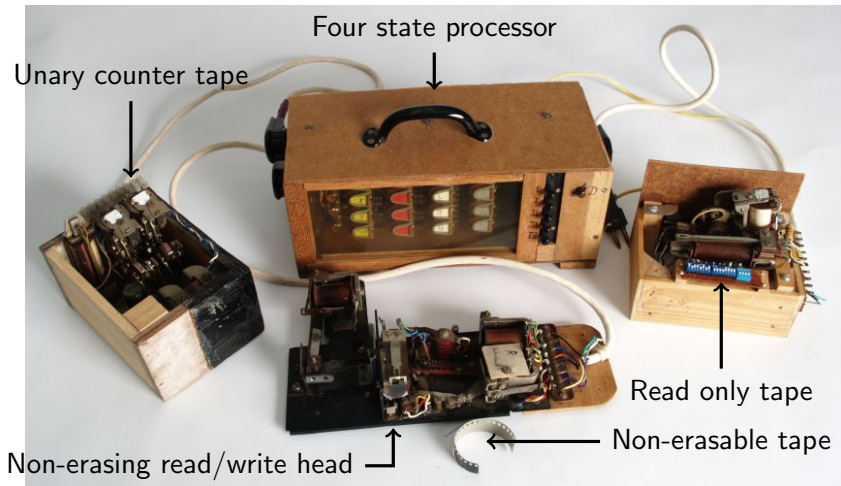
- complicated encoding (powers of primes)
- 1963, Hasenjaeger built a universal Turing machine using old telephone relays
 - used some ideas from Wang and Moore
 - 3-tapes, 4 states, 2 symbols

Anatomy of Hasenjaeger's machine



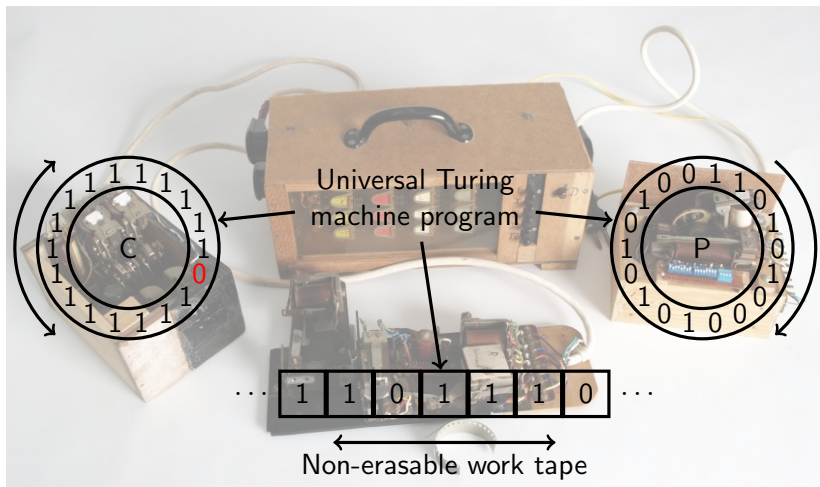
Donated by Hasenjaeger's family to the Heinz Nixdorf MuseumsForum, Paderborn

Anatomy of Hasenjaeger's machine



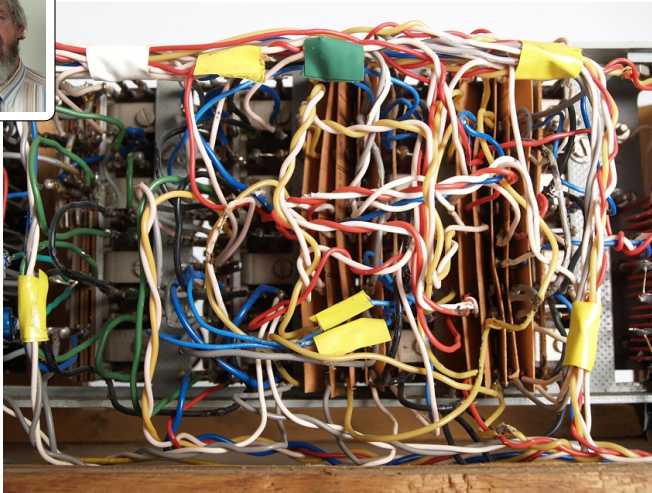
Donated by Hasenjaeger's family to the Heinz Nixdorf MuseumsForum, Paderborn

Anatomy of Hasenjaeger's machine

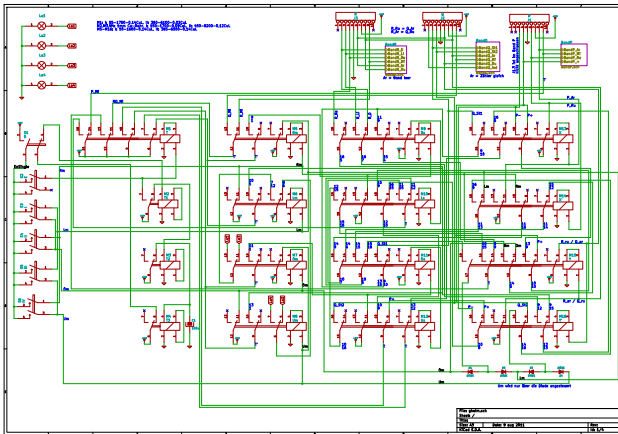


Donated by Hasenjaeger's family to the Heinz Nixdorf MuseumsForum, Paderborn

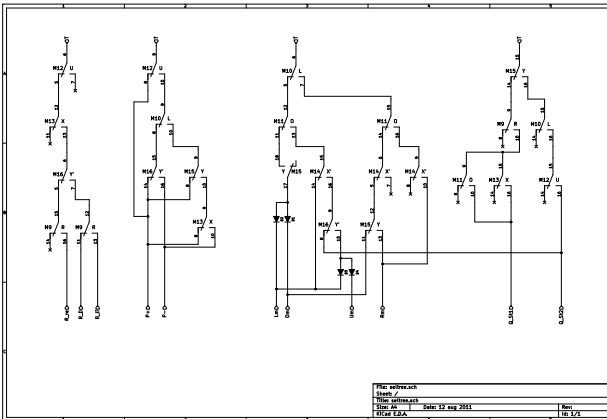
Rainer Glaschick figured out Hasenjaeger's Universal machine



Rainer Glaschick figured out Hasenjaeger's Universal machine



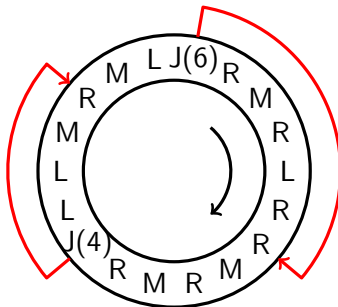
Rainer Glaschick figured out Hasenjaeger's Universal machine



Hasenjaeger's Universal machine: Rules

#	Q	P	C	R	P	C	R	Q'
1	q_1	1	-	0	\triangleright	\square	\boxtimes	q_1
2	q_1	1	-	1	\triangleright	\square	\square	q_1
3	q_1	0	-	-	\triangleright	\square	\square	q_2
4	q_2	1	0	-	\triangleright	\square	\triangleright	q_1
5	q_2	0	0	-	\triangleright	\triangleright	\square	q_2
6	q_2	1	1	-	\triangleright	\triangleleft	\triangleleft	q_1
7	q_2	0	1	-	\triangleright	\triangleleft	\square	q_3
8	q_3	0	-	0	\triangleright	\square	\square	q_3
9	q_3	1	-	0	\triangleright	\square	\square	q_1
10	q_3	0	-	1	\triangleright	\triangleright	\square	q_3
11	q_3	1	-	1	\triangleright	\triangleright	\square	q_4
12	q_4	0	1	-	\triangleright	\square	\square	q_4
13	q_4	1	1	-	\triangleright	\triangleleft	\square	q_4
14	q_4	-	0	-	\square	\square	\square	q_1

Circular Wang machines



What do the rules do?

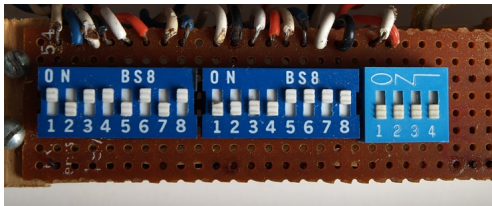
- Simulate Wang's B-machine
- Using an encoding scheme inspired by Moore

M : 1 : mark: write a 1 to the current tape cell,

R : 01 : move head right,

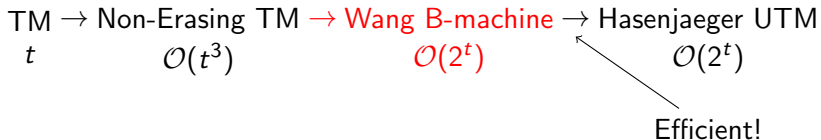
L : 001 : move head left,

$J(x)$: 00 $0^x 1$: if the current cell is 1 (marked) then skip x instructions, otherwise move to the next instruction.



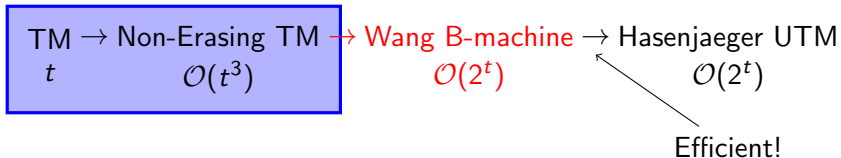
Hasenjaeger built a small
efficient Wang B-machine
simulator

Proving Hasenjaeger's machine is efficiently universal



$A \rightarrow B$, indicates A is simulated by B

Proving Hasenjaeger's machine is efficiently universal



$A \rightarrow B$, indicates A is simulated by B


Turing machine to Non-erasing Turing machine $\mathcal{O}(t^3)$

1: 

Turing machine to Non-erasing Turing machine $\mathcal{O}(t^3)$

1: 

2: 


 write

Turing machine to Non-erasing Turing machine $\mathcal{O}(t^3)$

1: 

2: 

3: 

 write

Turing machine to Non-erasing Turing machine $\mathcal{O}(t^3)$

1: 

2: 

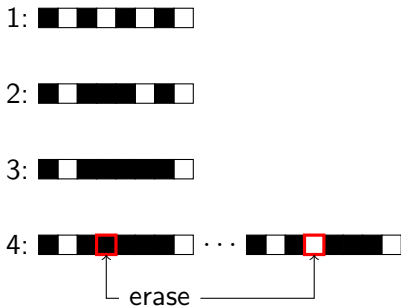
3: 

4: 

 erase

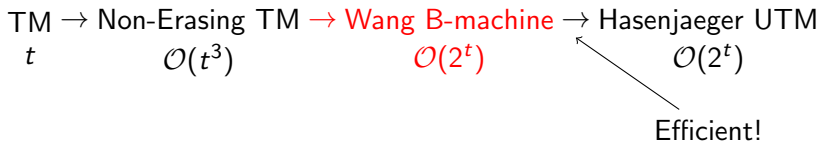
To erase a bit, make a new copy of the work tape.

Turing machine to Non-erasing Turing machine $\mathcal{O}(t^3)$



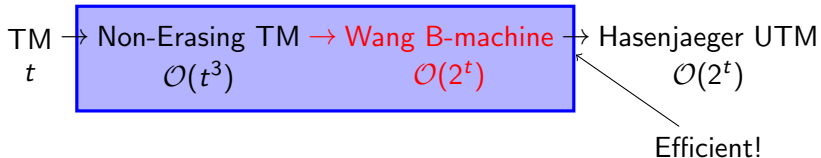
To erase a bit, make a new copy of the work tape.

Proving Hasenjaeger's machine is efficiently universal



$A \rightarrow B$, indicates A is simulated by B

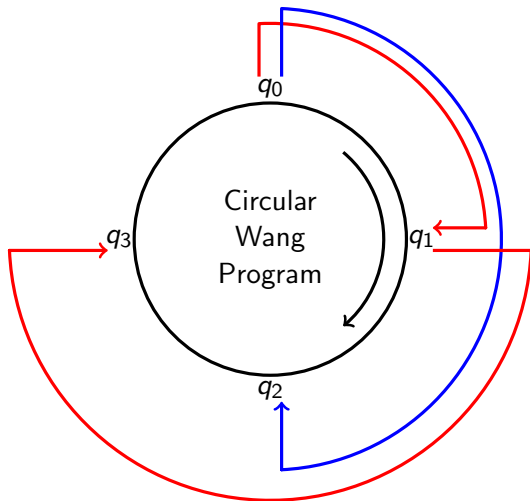
Proving Hasenjaeger's machine is efficiently universal

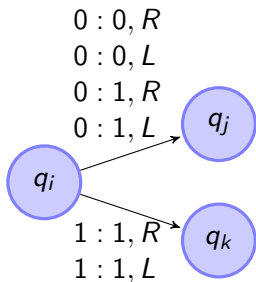


$A \rightarrow B$, indicates A is simulated by B

Transforming a non-erasing Turing machine to a Wang machine

$(q_0, 0 : 0, R, q_1)$
 $(q_0, 1 : 1, R, q_2)$
 $(q_1, 0 : 1, R, q_3)$
 $(q_1, 1 : 1, R, q_1)$
 $(q_1, 0 : 0, R, q_2)$
 $(q_2, 0 : 0, R, q_2)$
 $(q_2, 1 : 1, R, q_3)$
 $(q_3, 1 : 1, R, q_3)$
 $(q_3, 0 : 1, L, q_0)$

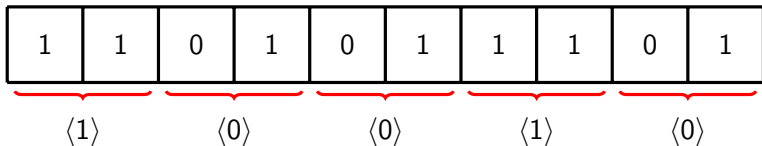




$q_i, 0 : 0, R, q_j$	
$q_i, 1 : 1, R, q_k$	
$q_i, 0 : 0, R, q_j$	
$q_i, 1 : 1, L, q_k$	
$q_i, 0 : 0, L, q_j$	
$q_i, 1 : 1, R, q_k$	
$q_i, 0 : 0, L, q_j$	
$q_i, 1 : 1, L, q_k$	
$q_i, 0 : 1, R, q_j$	
$q_i, 1 : 1, R, q_k$	
$q_i, 0 : 1, R, q_j$	
$q_i, 1 : 1, L, q_k$	
$q_i, 0 : 1, L, q_j$	
$q_i, 1 : 1, R, q_k$	
$q_i, 0 : 1, L, q_j$	
$q_i, 1 : 1, L, q_k$	

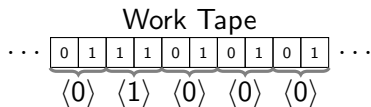
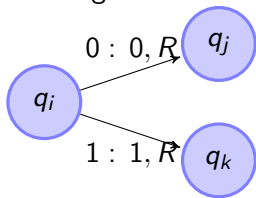
Idea: The encoding on the input tape

- Wang machines only jump on 1's
- $\langle 1 \rangle = 11$
- $\langle 0 \rangle = 01$

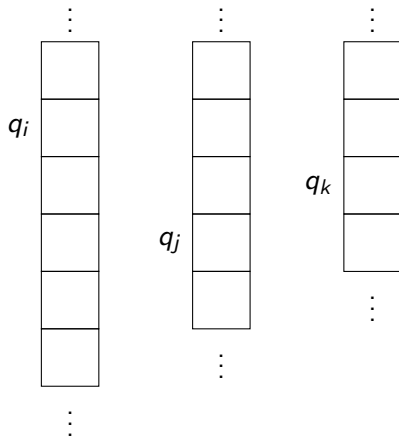


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

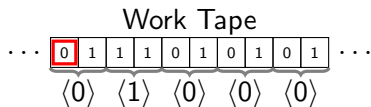
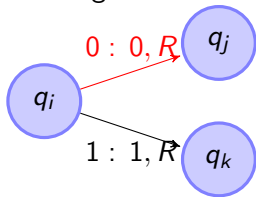


Circular Wang program sections

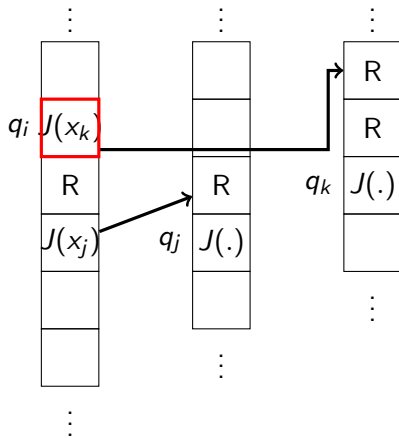


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

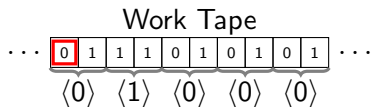
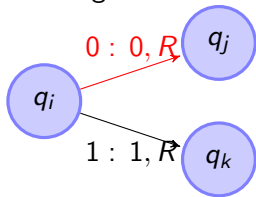


Circular Wang program sections

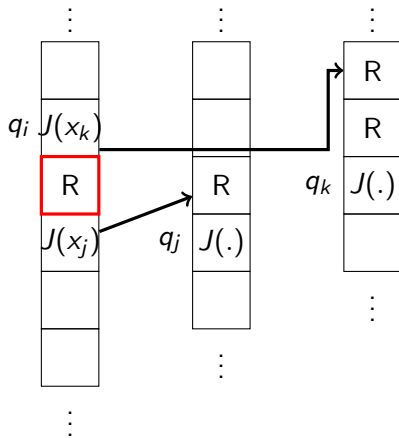


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

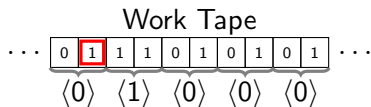
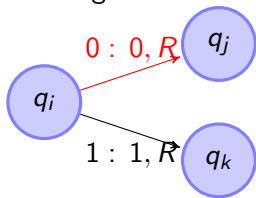


Circular Wang program sections

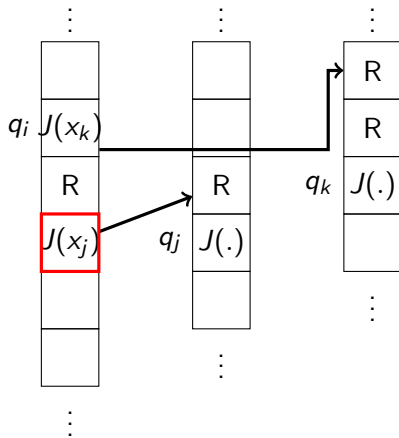


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

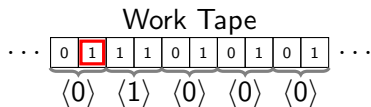
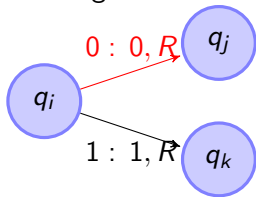


Circular Wang program sections

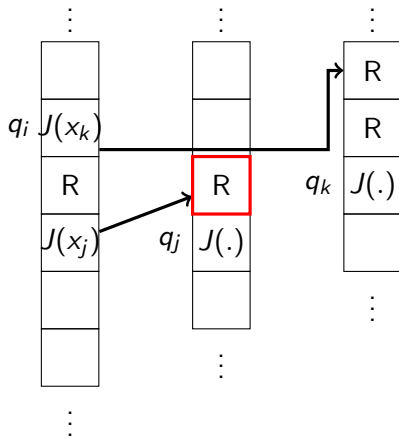


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

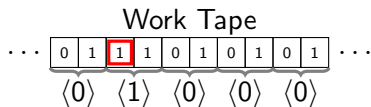
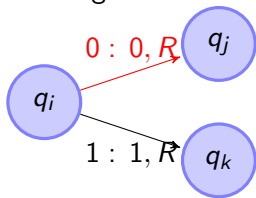


Circular Wang program sections

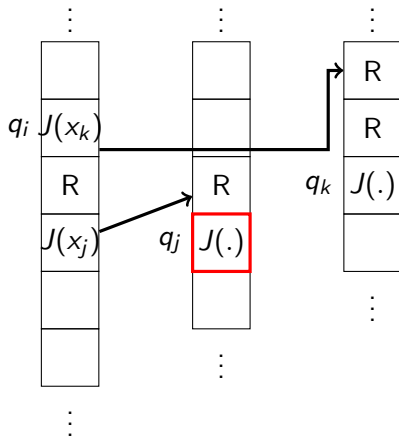


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

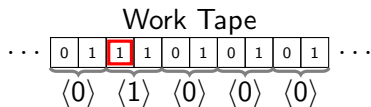
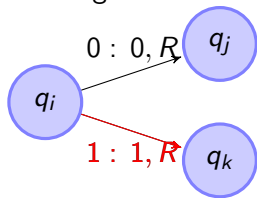


Circular Wang program sections

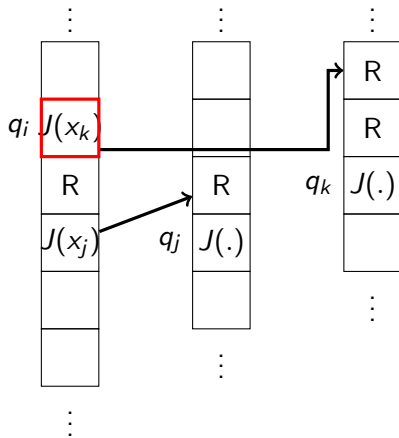


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

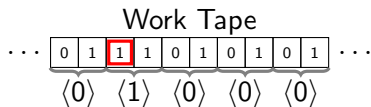
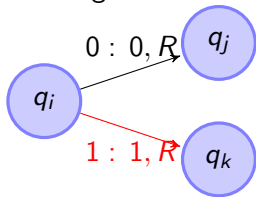


Circular Wang program sections

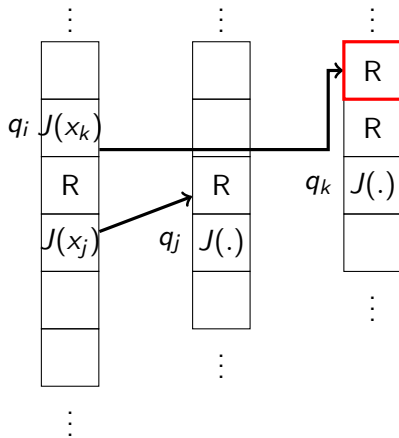


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

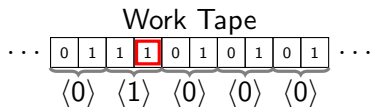
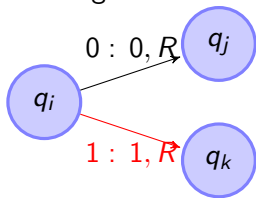


Circular Wang program sections

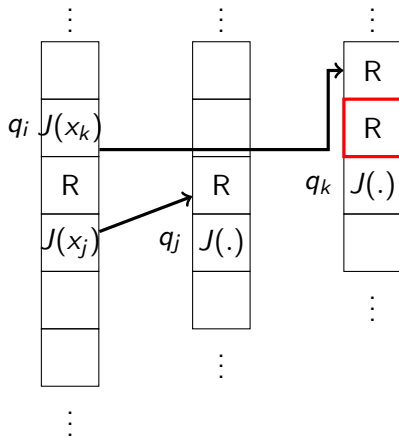


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules

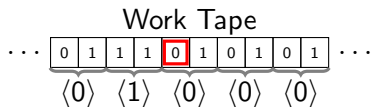
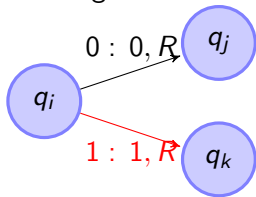


Circular Wang program sections

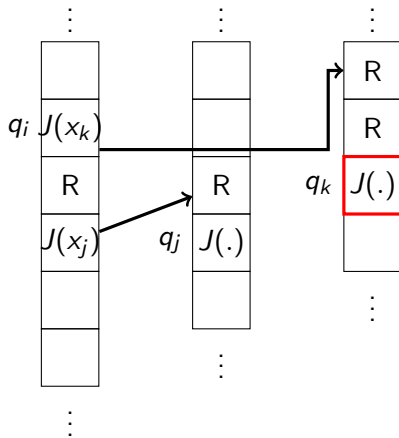


Key Ideas: Jumping on reading a 0 or 1

NE Turing Machine rules



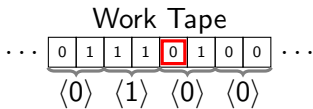
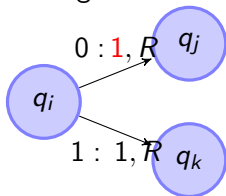
Circular Wang program sections



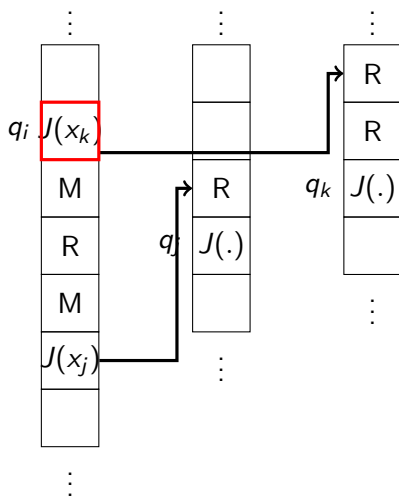
Read a 0/write a 1

Key Ideas: Read a 0/write a 1

NE Turing Machine rules

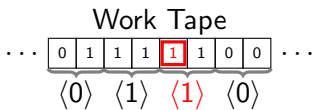
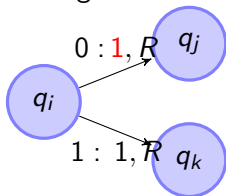


Circular Wang program sections

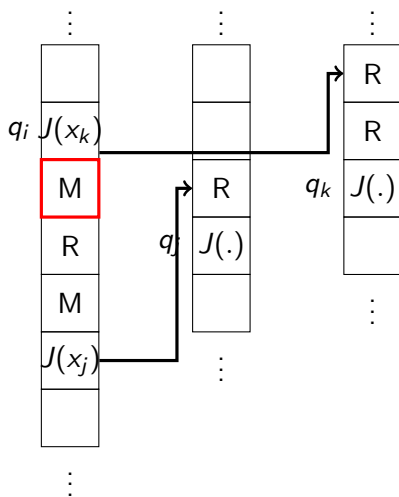


Key Ideas: Read a 0/write a 1

NE Turing Machine rules

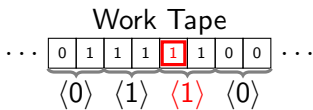
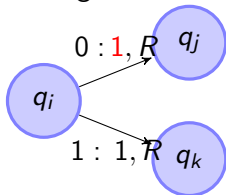


Circular Wang program sections

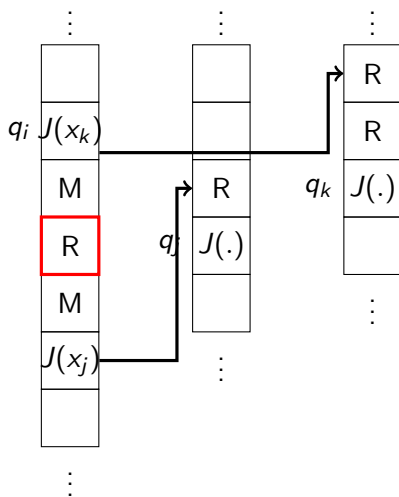


Key Ideas: Read a 0/write a 1

NE Turing Machine rules

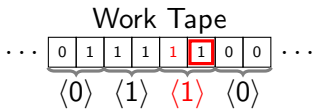
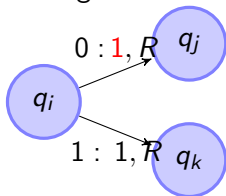


Circular Wang program sections

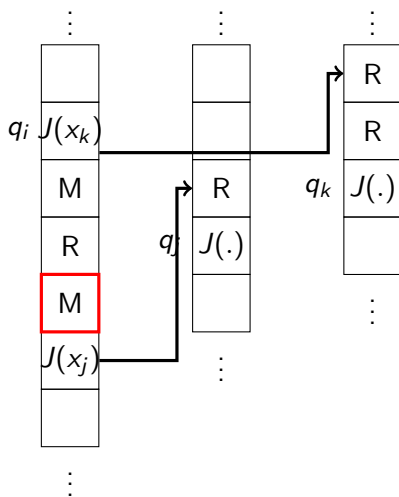


Key Ideas: Read a 0/write a 1

NE Turing Machine rules

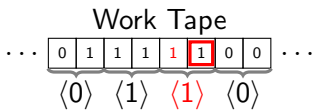
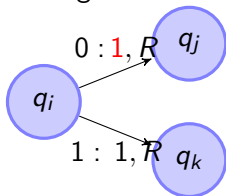


Circular Wang program sections

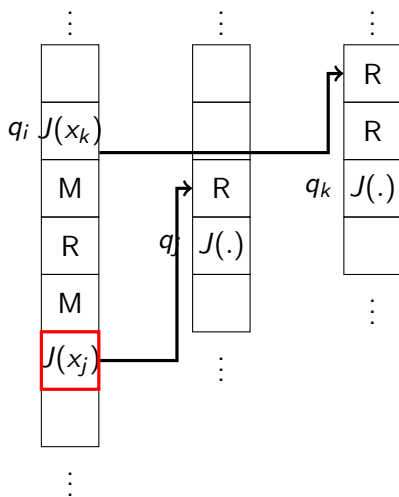


Key Ideas: Read a 0/write a 1

NE Turing Machine rules

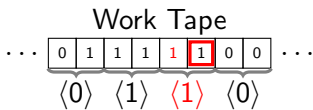
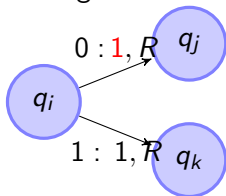


Circular Wang program sections

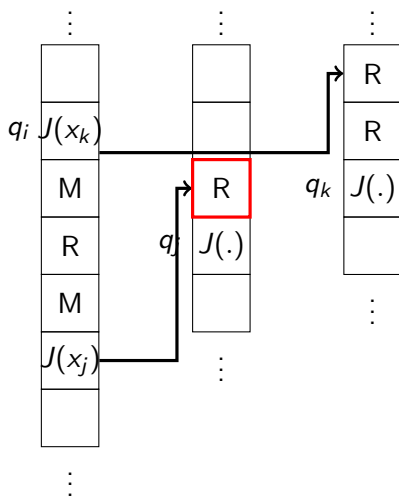


Key Ideas: Read a 0/write a 1

NE Turing Machine rules



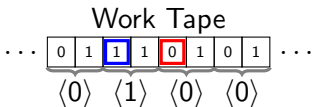
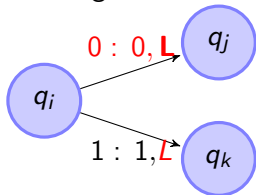
Circular Wang program sections



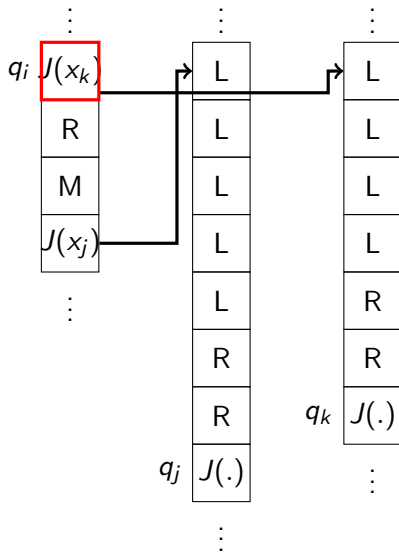
Moving left

Key Ideas: Moving Left

NE Turing Machine rules

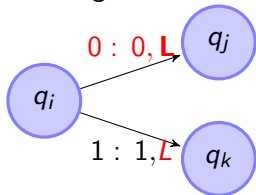


Circular Wang program sections

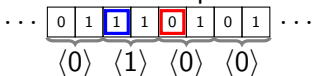


Key Ideas: Moving Left

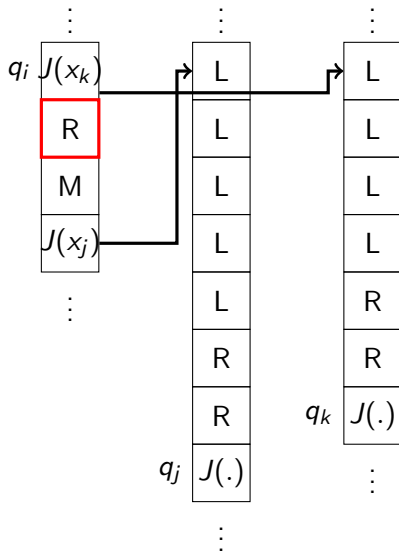
NE Turing Machine rules



Work Tape

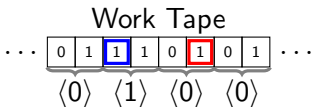
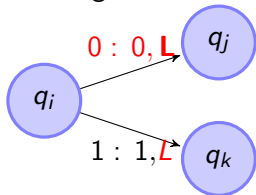


Circular Wang program sections

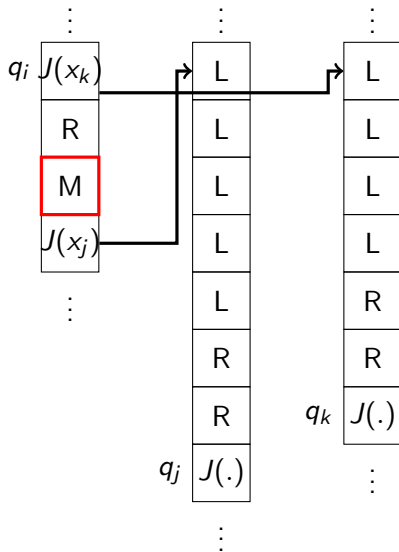


Key Ideas: Moving Left

NE Turing Machine rules

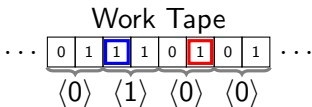
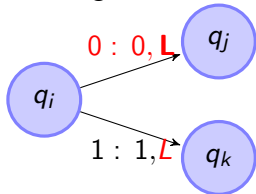


Circular Wang program sections

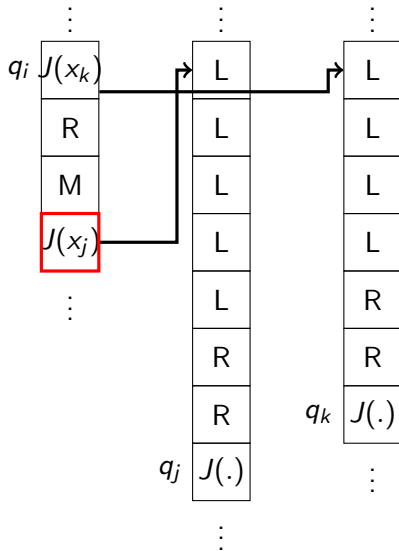


Key Ideas: Moving Left

NE Turing Machine rules

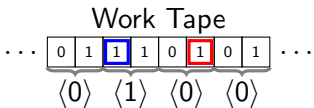
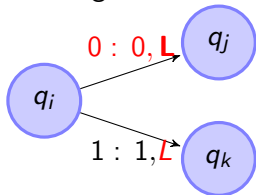


Circular Wang program sections

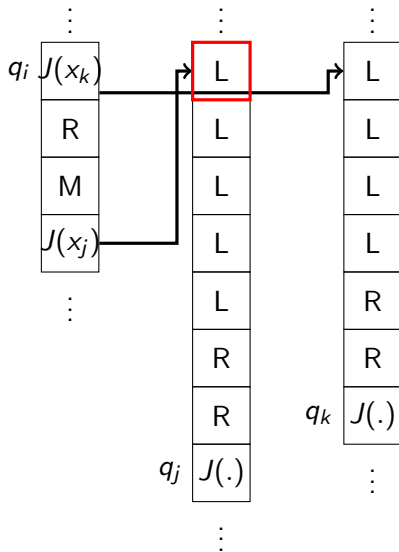


Key Ideas: Moving Left

NE Turing Machine rules

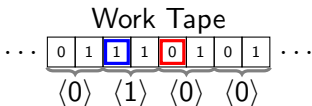
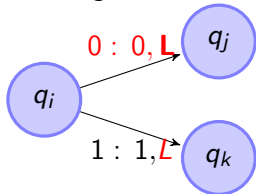


Circular Wang program sections

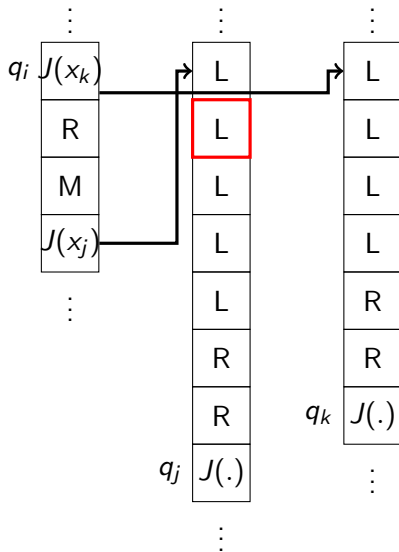


Key Ideas: Moving Left

NE Turing Machine rules

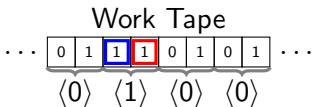
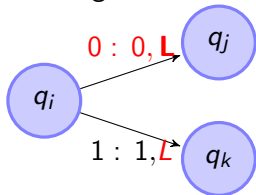


Circular Wang program sections

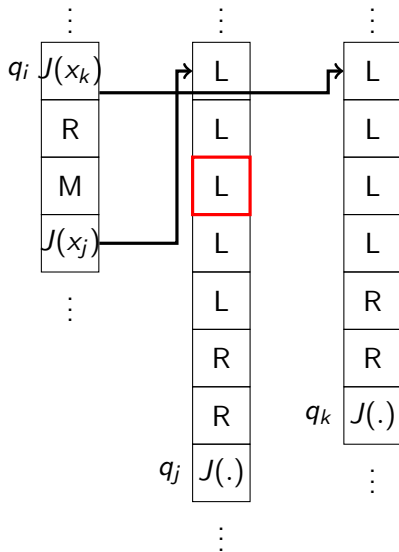


Key Ideas: Moving Left

NE Turing Machine rules

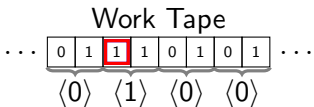
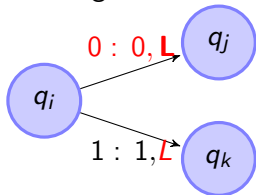


Circular Wang program sections

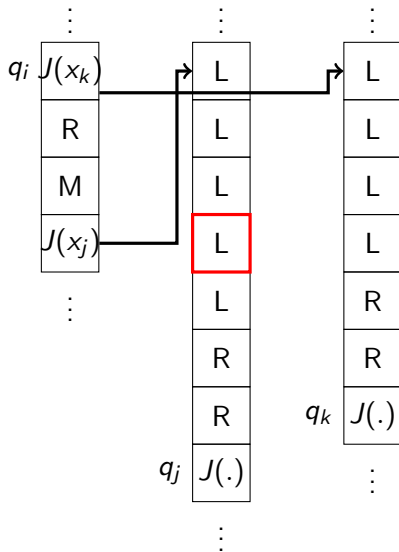


Key Ideas: Moving Left

NE Turing Machine rules

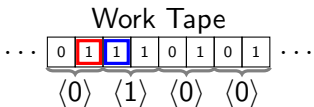
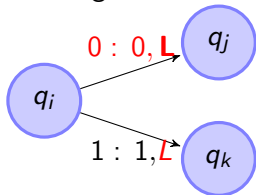


Circular Wang program sections

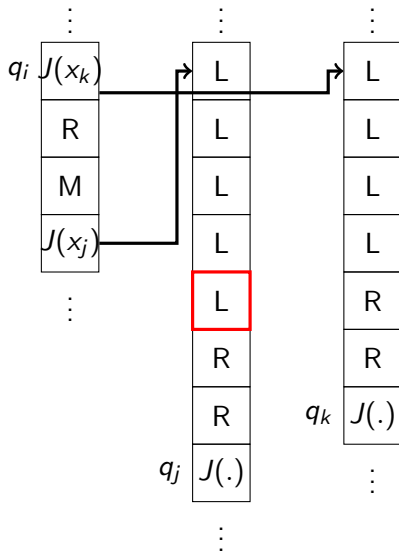


Key Ideas: Moving Left

NE Turing Machine rules

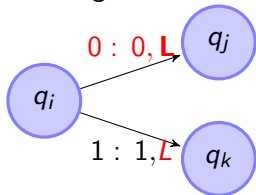


Circular Wang program sections

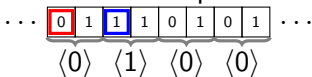


Key Ideas: Moving Left

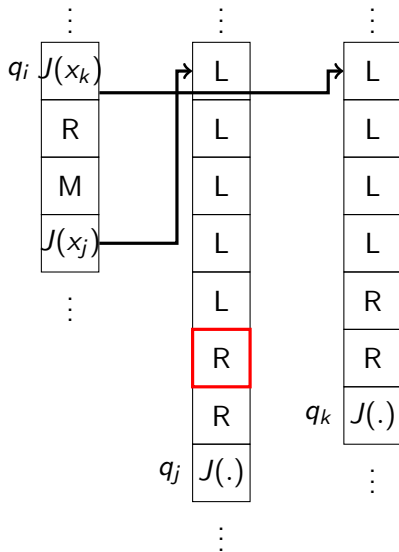
NE Turing Machine rules



Work Tape

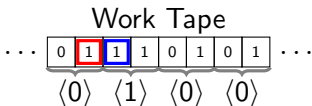
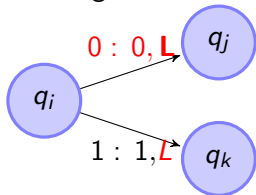


Circular Wang program sections

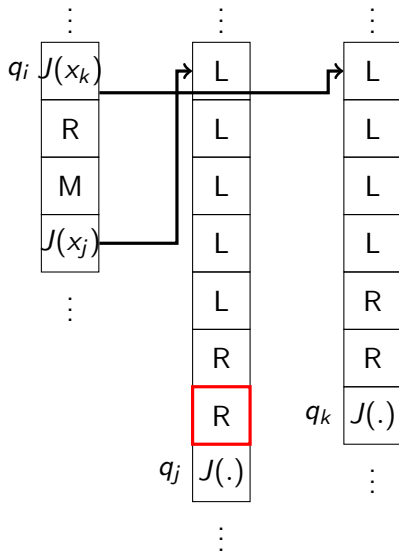


Key Ideas: Moving Left

NE Turing Machine rules

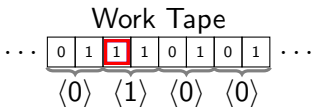
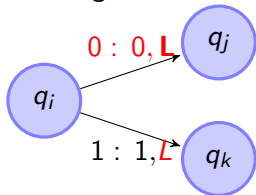


Circular Wang program sections

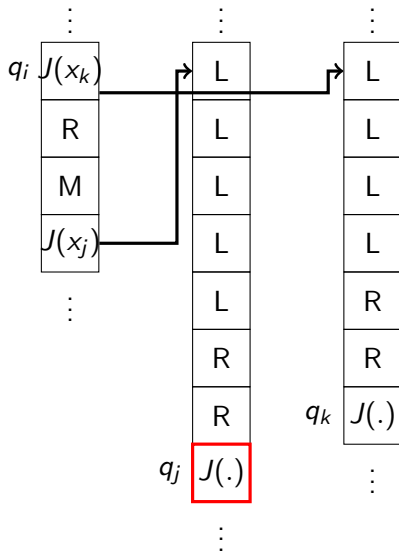


Key Ideas: Moving Left

NE Turing Machine rules



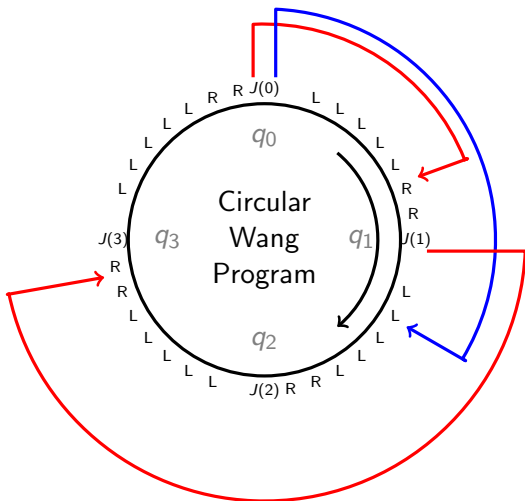
Circular Wang program sections



Mapping from the table to the rules

$q_i, 0, 0, R, q_j$ $q_i, 1, 1, R, q_k$	$L, L, L, L, L, R, R, J(q_k - 2), R, M, J(q_j - 1)$
$q_i, 0, 0, R, q_j$ $q_i, 1, 1, L, q_k$	$L, L, L, L, L, R, R, J(q_k - 6), R, M, J(q_j - 1)$
$q_i, 0, 0, L, q_j$ $q_i, 1, 1, R, q_k$	$L, L, L, L, L, R, R, J(q_k - 2), R, M, J(q_j - 7)$
$q_i, 0, 0, L, q_j$ $q_i, 1, 1, L, q_k$	$L, L, L, L, L, R, R, J(q_k - 6), R, M, J(q_j - 7)$
$q_i, 0, 1, R, q_j$ $q_i, 1, 1, R, q_k$	$L, L, L, L, L, R, R, J(q_k - 2), M, R, M, J(q_j - 1)$
$q_i, 0, 1, R, q_j$ $q_i, 1, 1, L, q_k$	$L, L, L, L, L, R, R, J(q_k - 6), M, R, M, J(q_j - 1)$
$q_i, 0, 1, L, q_j$ $q_i, 1, 1, R, q_k$	$L, L, L, L, L, R, R, J(q_k - 2), M, R, M, J(q_j - 7)$
$q_i, 0, 1, L, q_j$ $q_i, 1, 1, L, q_k$	$L, L, L, L, L, R, R, J(q_k - 6), M, R, M, J(q_j - 7)$

Final circular Wang program



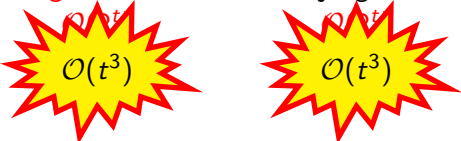
Conclusion

TM \rightarrow Non-Erasing TM \rightarrow Wang B-machine \rightarrow Hasenjaeger UTM
 t $\mathcal{O}(t^3)$ $\mathcal{O}(2^t)$ $\mathcal{O}(2^t)$

$A \rightarrow B$, indicates A is simulated by B

Conclusion

TM \rightarrow Non-Erasing TM \rightarrow Wang B-machine \rightarrow Hasenjaeger UTM
 t $\mathcal{O}(t^3)$ $\mathcal{O}(t^3)$ $\mathcal{O}(t^3)$



- 4 States, 2 symbols, 3 tapes is a remarkably small machine

$A \rightarrow B$, indicates A is simulated by B

- Benedikt Löwe and the Isaac Newton Institute, Cambridge, UK, “Semantics and Syntax: A Legacy of Alan Turing” 2012
- R. Glaschick cooperates with the Heinz Nixdorf MuseumsForum, Germany
- T. Neary is supported by Swiss National Science Foundation Grant 200021-141029
- D. Woods is supported by National Science Foundation Grant 0832824, the Molecular Programming Project, and NSF grants CCF-1219274 and CCF-1162589
- N. Murphy is supported by The Campus de Excelencia Internacional Moncloa, PICATA Program and Spanish National project C09101514.