Two-Dimensional Typewriter Automata

NCMA 2022

Taylor J. Smith

Department of Computer Science
St. Francis Xavier University
Antigonish, Nova Scotia, Canada

August 27, 2022

Table of Contents

Table of Contents

Two-Dimensional Automata

- A two-dimensional (2D) automaton is a generalization of a one-dimensional automaton.
- Two major differences:
  1. Different input word
  2. Different transition function

Two-Dimensional Automata

- A two-dimensional (2D) automaton is a generalization of a one-dimensional automaton.
- Two major differences:
  1. **Different input word**
  2. Different transition function

$$
\begin{array}{cccccc}
\# & \# & \# & \cdots & \# & \# \\
\# & a_{1,1} & a_{1,2} & \cdots & a_{1,n} & \# \\
\# & a_{2,1} & a_{2,2} & \cdots & a_{2,n} & \# \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\# & a_{m,1} & a_{m,2} & \cdots & a_{m,n} & \# \\
\# & \# & \# & \cdots & \# & \#
\end{array}
$$

Two-Dimensional Automata

- A two-dimensional (2D) automaton is a generalization of a one-dimensional automaton.
- Two major differences:
  1. Different input word
  2. **Different transition function**

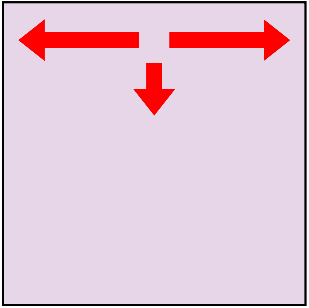
$$\delta : (Q \setminus q_{\text{accept}}) \times (\Sigma \cup \{\#\}) \rightarrow Q \times \{U, D, L, R\}$$

Deterministic four-way (2DFA-4W)

$$\delta : (Q \setminus q_{\text{accept}}) \times (\Sigma \cup \{\#\}) \rightarrow 2^{Q \times \{U,D,L,R\}}$$

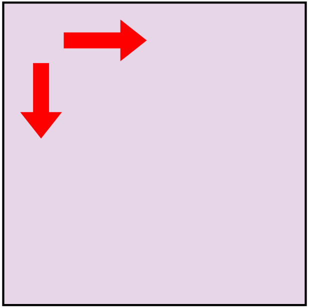
Nondeterministic four-way (2NFA-4W)

- Three-way 2D automata (2DFA-3W/2NFA-3W)
  - Can only move its input head downward, leftward, or rightward.
  - Can read the current row as much as it wants, but can never return to previous rows.
  - Known relationships:
    $\mathcal{L}_{\text{2DFA-3W}} \subset \mathcal{L}_{\text{2NFA-3W}}$
    $\mathcal{L}_{\text{2DFA-3W}} \subset \mathcal{L}_{\text{2DFA-4W}}$
    $\mathcal{L}_{\text{2NFA-3W}} \subset \mathcal{L}_{\text{2NFA-4W}}$

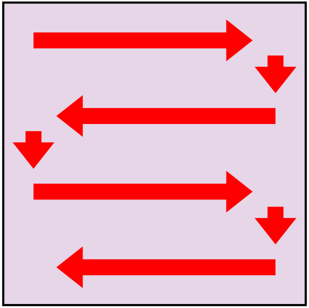


Variants of 2D Automata

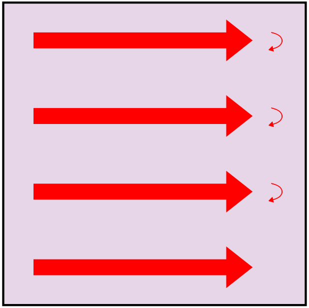
- Two-way 2D automata (2DFA-2W/2NFA-2W)
  - Can only move its input head downward or rightward.
  - Can read any given symbol of the input word at most once.
  - Cannot read every symbol of input words where $m, n \geq 2$.
  - Known relationships:
    $L_{\text{2DFA-2W}} \subset L_{\text{2NFA-2W}}$
    $L_{\text{2DFA-2W}} \subset L_{\text{2DFA-3W}}$
    $L_{\text{2NFA-2W}} \subset L_{\text{2NFA-3W}}$



- Boustrophedon automata (BDFA/BFA)
  - Introduced by Fernau et al. (2015, ’16, ’18)
  - Moves its input head “as the ox turns”.
  - More precisely, moves left-to-right and right-to-left in alternating rows.
  - Must read each symbol of the input word exactly once before accepting.
  - Known relationships:
    $L_{\text{BFA}} \subset L_{\text{2NFA-3W}}$
    $L_{\text{BFA}} = L_{\text{BDFA}}$

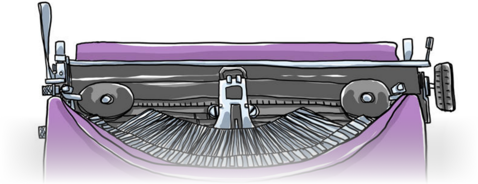


- Returning automata (RFA)
  - Also introduced by Fernau et al. (2015)
  - Moves its input head from the leftmost symbol to the rightmost symbol in each row.
  - Syntactically identical to boustrophedon automata.
  - Known relationships: $L_{\text{RFA}} = L_{\text{BFA}}$



Typewriter Automata

- A typewriter automaton (TFA) is a tuple $(Q, \Sigma, \delta, q_0, q_{\text{accept}})$.
  - Q, Σ, q_0, and q_{accept} are the same as in any other 2D automaton.
  - The transition function δ is restricted to making two moves: a rightward move R and a "reset" move $\curvearrowleft$.
- Like a typewriter's carriage return, the reset move $\curvearrowleft$ shifts the input head all the way to the left and one row downward.
- Symbols in a row aren't re-read during a reset move.



Typewriter Automata

- Typewriter automata only move in one direction through each row of the input word, and can reset any time. (+)
  - Boustrophedon automata move in alternating directions through adjacent rows, and can't choose when to move downward.
- Typewriter automata don't need to read entire rows of their input words. (+)
  - Returning automata must read an entire row before moving downward.
- Typewriter automata can only make downward/leftward moves as one "unit" during a reset move. (−)
  - Three-way 2D automata can move downward and leftward separately and at any time.

Motivating Questions

- What (recognition/closure/decidability/etc.) properties do TFAs have?
- How do TFAs relate to BFAs, RFAs, and other models?
- What can we do with TFAs?
- How can we apply TFAs to real-world problems?

Table of Contents

NCMA 2022

Recognition Properties

- All of the following relations were previously known.
- An arrow $L_A \to L_B$ indicates a subset relation $L_A \subset L_B$.

$$\begin{tikzcd}
 & L_{\text{2NFA-4W}} & & & \\
L_{\text{2DFA-4W}} \arrow[ur] & & & & \\
 & L_{\text{2NFA-3W}} \arrow[uu] & & & \\
L_{\text{2DFA-3W}} \arrow[uu] \arrow[ur] & & & & \\
 & L_{\text{2NFA-2W}} \arrow[uu] & L_{\text{BFA}} \arrow[uul] & = & L_{\text{BDFA}} \\
L_{\text{2DFA-2W}} \arrow[uu] \arrow[ur] & & & &
\end{tikzcd}$$

Theorem

$L_{TDFA} \subset L_{TFA}$.

Proof Sketch.

Let $\Sigma = \{0, 1\}$ and $m, n \geq 2$. Consider the language

$$L_{\mathrm{L}} = \{m \times n \text{ words where the first column/last row are all 1s and all other symbols are 0s}\}. \qquad \square$$

n-Way vs. Typewriter

Theorem

$L_{2DFA\text{-}2W} \subset L_{TDFA}$ *and* $L_{2NFA\text{-}2W} \subset L_{TFA}$.

Proof Sketch.

Let $\Sigma = \{0, 1\}$. Consider the language

$$L_{1s} = \{2 \times 2 \text{ words having 1s at positions } (0, 1) \text{ and } (1, 0)\}. \qquad \square$$

Theorem

$L_{2DFA\text{-}2W} \subset L_{TDFA}$ *and* $L_{2NFA\text{-}2W} \subset L_{TFA}$.

Proof Sketch.

Let $\Sigma = \{0, 1\}$. Consider the language

$$L_{1s} = \{2 \times 2 \text{ words having 1s at positions } (0, 1) \text{ and } (1, 0)\}. \qquad \square$$

Theorem

$L_{TDFA} \subset L_{2DFA\text{-}3W}$ *and* $L_{TFA} \subset L_{2NFA\text{-}3W}$.

Proof Sketch.

Let $\Sigma = \{0, 1\}$. Consider the language

$$L_{\text{stairs}} = \{n \times n \text{ words having a staircase pattern of 1s from top-left to bottom-right}\}. \qquad \square$$

Theorem

$L_{TFA} = L_{BFA}$ ($= L_{BDFA}$).

Proof Sketch.

It is possible to show that a nondeterministic typewriter automaton can simulate the computation of a returning automaton and vice versa. Thus, they recognize the same class of languages.
We then use the fact that boustrophedon automata and returning automata recognize the same class of languages to complete the proof. □

Summary of Recognition Properties

NCMA 2022

$$
\begin{tikzcd}
 & L_{\text{2NFA-4W}} & & & & \\
L_{\text{2DFA-4W}} \arrow[ur] & & & & & \\
 & L_{\text{2NFA-3W}} \arrow[uu] & & & & \\
L_{\text{2DFA-3W}} \arrow[uu] \arrow[ur] & & & & & \\
 & L_{\text{TFA}} \arrow[uu] & = & L_{\text{BFA}} \arrow[uull] & = & L_{\text{BDFA}} \\
L_{\text{TDFA}} \arrow[uu] \arrow[ur] & & & & & \\
 & L_{\text{2NFA-2W}} \arrow[uu] & & & & \\
L_{\text{2DFA-2W}} \arrow[uu] \arrow[ur] & & & & &
\end{tikzcd}
$$

Table of Contents

- All of the following closure properties were previously known.

	2DFA-3W	2NFA-3W	2DFA-2W	2NFA-2W	BFA
$\cup$	✗	✓	✗	✓	✓
$\cap$	✗	✗	✗	✗	✓
$\neg$	✓	✗	✓	✗	✓
$\ominus / \varobar$	✗	✓$_{\ominus}$ ✗$_{\varobar}$	✗	✗	✓$_{\ominus}$ ✗$_{\varobar}$
R	✗	✓	✗	✗	✓
$\circlearrowright$	✗	✗	✗	✗	✗

- One option to show closure is to use the fact that typewriter automata lie "in between" two-way and three-way 2D automata, and adapt the relevant proofs.

Theorem

L_{TFA} is closed under union. L_{TDFA} is closed under complement.

Closure for Typewriter Automata

- One option to show closure is to use the fact that typewriter automata lie "in between" two-way and three-way 2D automata, and adapt the relevant proofs.

Theorem

L_{TFA} is closed under union. L_{TDFA} is closed under complement.

- More easily, we can get closure as a consequence of the equivalence between L_{TFA} and L_{BFA}.

Theorem

L_{TFA} is closed under intersection, complement, row concatenation, and reversal.

Closure for Typewriter Automata

- One option to show closure is to use the fact that typewriter automata lie "in between" two-way and three-way 2D automata, and adapt the relevant proofs.

Theorem

L_{TFA} is closed under union. L_{TDFA} is closed under complement.

- More easily, we can get closure as a consequence of the equivalence between L_{TFA} and L_{BFA}.

Theorem

L_{TFA} is closed under intersection, complement, row concatenation, and reversal.

Remark

We know that $L_{\text{2NFA-2W}} \subset L_{\mathrm{TFA}} \subset L_{\text{2NFA-3W}}$, yet L_{TFA} is closed under more operations than either nondet. 2D automaton model!

Summary of Closure Properties

	2DFA-3W	2NFA-3W	2DFA-2W	2NFA-2W	TDFA	TFA
$\cup$	✗	✓	✗	✓		✓
$\cap$	✗	✗	✗	✗		✓
$\neg$	✓	✗	✓	✗	✓	✓
$\ominus/\oplus$	✗	$✓_{\ominus}$ $✗_{\oplus}$	✗	✗		$✓_{\ominus}$
R	✗	✓	✗	✗		✓
$\circlearrowright$	✗	✗	✗	✗		

* only positive closure results highlighted

Table of Contents

Conclusions

- Typewriter automata present a conceptually simpler model of computation relative to the standard 2D automaton model.
- Typewriter automata have deep connections to other non-classical models, such as boustrophedon/returning automata.
- Typewriter automata recognize languages "in between" two-way and three-way 2D automata, and they recognize the same class of languages as boustrophedon automata.
- Typewriter automata are closed under many common 2D language operations: union, intersection, complement, row concatenation, and reversal.

Future Work

- These are all very preliminary results!
- It would be worth studying further details of the typewriter automaton model.
  - State complexity bounds for closed operations
  - Decidability and complexity of decision problems
  - Connections/extensions to boustrophedon/returning automata
- One can also consider applications of typewriter automata.
  - Image processing applications
  - Converting 2D data to a compressible 1D string
  - Processing OCR data
  - Edge finding, segment mapping, etc.

References

[1] H. Fernau, M. Paramasivan, M. L. Schmid, and D. G. Thomas. Scanning pictures the boustrophedon way. In R. P. Barneva, B. B. Bhattacharya, and V. E. Brimkov, editors, *Proc. of IWCIA 2015*, volume 9448 of *LNCS*, pages 202–216, Berlin Heidelberg, 2015. Springer-Verlag.

[2] H. Fernau, M. Paramasivan, M. L. Schmid, and D. G. Thomas. Simple picture processing based on finite automata and regular grammars. *J. Comput. System Sci.*, 95:232–258, 2018.

[3] H. Fernau, M. Paramasivan, and D. G. Thomas. Regular array grammars and boustrophedon finite automata. In H. Bordihn, R. Freund, B. Nagy, and G. Vaszil, editors, *Short Papers of NCMA 2016*, pages 55–63, Vienna, 2016. Institut für Computersprachen, TU Wien.

[4] Taylor J. Smith. Two-dimensional typewriter automata. In H. Bordihn, G. Horváth, and G. Vaszil, editors, *Short Papers of NCMA 2022*, pages 38–45, Debrecen, 2022. Faculty of Informatics, University of Debrecen.