Regular Grammars for Array Languages

Henning Fernau, Meenakshi Paramasivan, and D. Gnanaraj Thomas

God completes the favors of wisdom, from the city of Trier

In this sign you will conquer

NCMA, Prague, Czech Republic, August 17-18, 2017

Map

- Regular Matrix Grammars
- (Regular : Regular) Array Grammars
- Isometric Regular Array Grammars
- Relation b/w language families

Map

- Regular Matrix Grammars
- (Regular : Regular) Array Grammars
- Isometric Regular Array Grammars
- Relation b/w language families

Regular Matrix Grammars

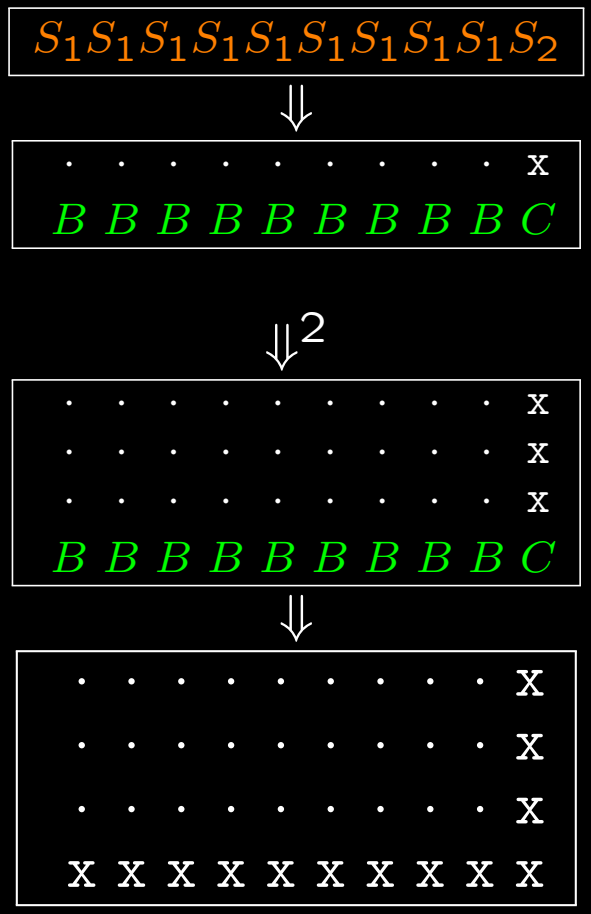
A *two-dimensional right-linear grammar (2RLG)* [G & R 97] is defined by a 7-tuple $G = (V_h, V_v, \Sigma_I, \Sigma, S, R_h, R_v)$, where:

- V_h is a finite set of *horizontal nonterminals*;
  V_v is a finite set of *vertical nonterminals*;
- $\Sigma_I \subseteq V_v$ is a finite set of *intermediates*;
  Σ is a finite set of *terminals*;
- $S \in V_h$ is a *starting symbol*;
- R_h is a finite set of *horizontal rules* of the form $V \to AV'$ or $V \to A$, where $V, V' \in V_h$ and $A \in \Sigma_I$;
- R_v is a finite set of *vertical rules* of the form $W \to aW'$ or $W \to a$, where $W, W' \in V_v$ and $a \in \Sigma$.

Original definition of a 2RLG (under the name of regular matrix grammar (RMG)) and properties of the corresponding class of picture languages, traditionally called *regular matrix languages* (RML), [Siromoney et al 72]. We use $\mathcal{L}_\Sigma(\mathrm{RMG})$ for convenience.

Example

$S \Rightarrow S_1A \Rightarrow^8 S_1S_1S_1S_1S_1S_1S_1S_1S_1A \Rightarrow$



A *(RMG)* $G = (V_h, V_v, \Sigma_I, \Sigma, S, R_h, R_v)$, where:

- $V_h = \{S, A\}$; $V_v = \{S_1, B, S_2, C\}$;
- $\Sigma_I = \{S_1, S_2\} \subseteq V_v$; $(V_h \cap \Sigma_I = \emptyset)$
  $\Sigma = \{\cdot, \mathtt{x}\}$; $S \in V_h$;
- $R_h = \{S \to S_1A,\ A \to S_1A,$
  $A \to S_2\}$;
- $R_v = \{S_1 \to \cdot B,\ B \to \cdot B,\ B \to \mathtt{x},$
  $S_2 \to \mathtt{x}C,\ C \to \mathtt{x}C, C \to \mathtt{x}\}$.

Useful Unary Operations

If W = $\begin{matrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{matrix}$ then

T(W) = $\begin{matrix} a_{1,1} & a_{2,1} & \cdots & a_{m,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{m,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,n} & a_{2,n} & \cdots & a_{m,n} \end{matrix}$ and H(W) = $\begin{matrix} a_{m,n} & \cdots & a_{m,2} & a_{m,1} \\ \vdots & \ddots & \vdots & \vdots \\ a_{2,n} & \cdots & a_{2,2} & a_{2,1} \\ a_{1,n} & \cdots & a_{1,2} & a_{1,1} \end{matrix}$

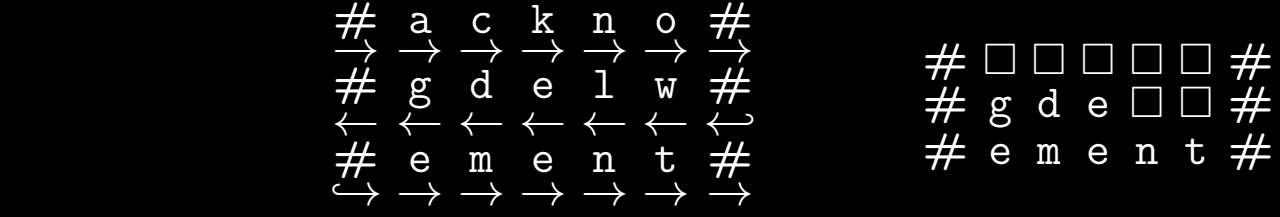
These operations are lifted to array languages and even to families of array languages in a natural way.

$T(\mathcal{L}_{\Sigma}(\mathrm{RMG}))$ = Any Guess ?

Boustrophedon Finite Automata

- Introduced by [H. Fernau et al 2015].

- Finite automata working on rectangular-shaped arrays (pictures). This automaton model reads the input, line after line, alters the direction, when the boundary is encountered.



Theorem 1. $T(\mathcal{L}_{\Sigma}(\mathsf{RMG})) = \mathcal{L}_{\Sigma}(\mathsf{BFA})$.

Map

- Regular Matrix Grammars
- (Regular : Regular) Array Grammars
- Isometric Regular Array Grammars
- Relation b/w language families

(Regular : Regular) Array Grammars

A *(Regular : Regular) Array Grammar* (R:R)AG is defined by a 8-tuple $G = (S, V_N, V_I, \Sigma, P_N, P_I, \pi, \tau)$, where:

- V_N is a nonterminal alphabet with a distinctive start symbol $S \in V_N$;
  V_I is an intermediate alphabet, disjoint from V_N;
  Σ is a terminal alphabet, disjoint from $V_N \cup V_I$;

- P_N is a set of non-terminal rules
  either of the form $A \to XB$ (right-linear),
  or of the form $A \to BX$ (left-linear), where $A, B \in V_N$ and $X \in V_I$;

- P_I is a set of rules of the form $A \to X$, with $A \in V_N$ and $X \in V_I$;

- $\pi : V_I \to \mathcal{L}_\Sigma(\mathsf{RMG}) \cup \mathcal{L}_\Sigma(\mathsf{BFA})$;

- $\tau : P_N \to \{\ominus, \oplus\}$ such that $\tau(p_1) = \tau(p_2)$
  implies that p_1 is right-linear if and only if p_2 is right-linear.

(Regular : Regular) Array Grammars

A *(Regular : Regular) Array Grammar* (R:R)AG is defined by a 8-tuple $G = (S, V_N, V_I, \Sigma, P_N, P_I, \pi, \tau)$, where:

- V_N is a nonterminal alphabet with a distinctive start symbol $S \in V_N$;
  V_I is an intermediate alphabet, disjoint from V_N;
  Σ is a terminal alphabet, disjoint from $V_N \cup V_I$;

- P_N is a set of non-terminal rules
  either of the form $A \to XB$ (right-linear),
  or of the form $A \to BX$ (left-linear), where $A, B \in V_N$ and $X \in V_I$;

- P_I is a set of rules of the form $A \to X$, with $A \in V_N$ and $X \in V_I$;

- $\pi : V_I \to \mathcal{L}_\Sigma(\mathsf{RMG}) \cup \mathcal{L}_\Sigma(\mathsf{BFA})$;

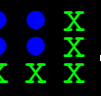
- $\tau : P_N \to \{\ominus, \oplus\}$ such that $\tau(p_1) = \tau(p_2)$
  implies that p_1 is right-linear if and only if p_2 is right-linear.

We denote the corresponding class of picture languages, as $\mathcal{L}((\mathsf{R} : \mathsf{R})\mathsf{AG})$ which was called as (Regular : Regular) Array Languages (R:R)AL [Siromoney et al 73].

Example

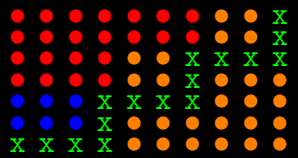
A (R:R)AG generating the staircase of x's of a fixed proportion is defined as $G = (S, V_N, V_I, \Sigma, P_N, P_I, \pi, \tau)$, where

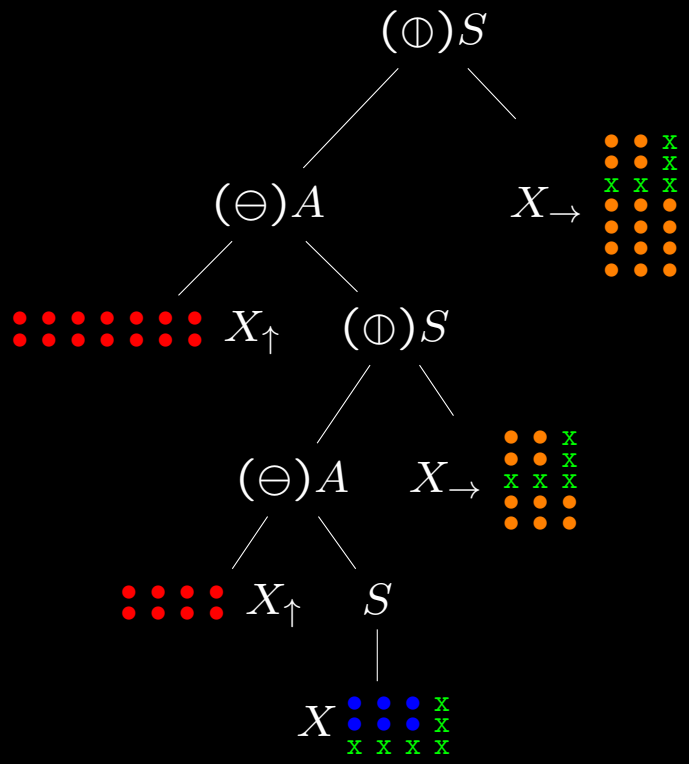
- $V_N = \{S, A\}$, $V_I = \{X_\uparrow, X_\rightarrow, X\}$, $\Sigma = \{\mathrm{x}, \bullet\}$,
  $P_N = \{S \to AX_\rightarrow, A \to X_\uparrow S\}$, $P_I = \{S \to X\}$,

- $\pi : V_I \to \mathcal{L}_\Sigma(\mathrm{RMG}) \cup \mathcal{L}_\Sigma(\mathrm{BFA})$ is given by
  $\pi(X_\uparrow) = \{(\bullet\bullet\bullet)^n \oplus (\bullet) \mid n \geq 1\}$,
  $\pi(X_\rightarrow) = \{(\bullet\bullet\mathrm{x}\,\bullet\bullet\mathrm{x}\,\mathrm{x}\,\mathrm{x}\,\mathrm{x}) \ominus (\bullet\bullet\bullet)_n \mid n \geq 1\}$ and $\pi(X) =$  .

- $\tau : P_N \to \{\ominus, \oplus\}$ is given by
  $\tau(S \to AX_\rightarrow) = \oplus$ and $\tau(A \to X_\uparrow S) = \ominus$.
  Here $\tau(S \to AX_\rightarrow) \neq \tau(A \to X_\uparrow S)$.

Staircase Example

To obtain  the corresponding tree:



Two Types of (R:R)AL

Definition 2. *An (R:R)AG G with $|\tau(P_N)| = 2$ is called* $\ominus$-left, $\oplus$-right

- *if $A \to BX \in P_N$ then $\tau(A \to BX) = \ominus$,*
- *if $A \to XB \in P_N$ then $\tau(A \to XB) = \oplus$.*

$\ominus$-left, $\oplus$-right (R:R)AG describes the language family $\mathcal{L}_{\ominus-\ell,\oplus-r}((\mathrm{R} : \mathrm{R})\mathrm{AG})$.

Definition 3. *An (R:R)AG G with $|\tau(P_N)| = 2$ is called* $\oplus$-left, $\ominus$-right

- *if $A \to BX \in P_N$ then $\tau(A \to BX) = \oplus$,*
- *if $A \to XB \in P_N$ then $\tau(A \to XB) = \ominus$.*

$\oplus$-left, $\ominus$-right (R:R)AG describes the language family $\mathcal{L}_{\oplus-\ell,\ominus-r}((\mathrm{R} : \mathrm{R})\mathrm{AG})$.

Note: Staircase Example is generated by some $\oplus$-left, $\ominus$-right (R:R)AG.

Results

Lemma 4.
$\mathcal{L}((\mathsf{R} : \mathsf{R})\mathsf{AG}) = \mathcal{L}_{\ominus-\ell,\oplus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG}) \cup \mathcal{L}_{\oplus-\ell,\ominus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG})$.

Lemma 5. *(a) A regular string language corresponds to a language of single-row arrays generated by RMG.*
(b) A regular string language corresponds to a language of single-column arrays generated by RMG.

Corollary 6. *(a) A regular string language corresponds to a language of single-row arrays accepted by BFA.*
(b) A regular string language corresponds to a language of single-column arrays accepted by BFA.

Remark 7. *A language of single-row (column) arrays is in both families* $\mathcal{L}_{\ominus-\ell,\oplus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG})$ *and* $\mathcal{L}_{\oplus-\ell,\ominus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG})$.

Results

News: $\mathcal{L}(\mathsf{RMG})$ is not closed under $\ominus$, $\mathcal{L}(\mathsf{BFA})$ is not closed under $\oplus$.

Consider the classes $\mathcal{L}(\mathsf{RMG}) \ominus \mathcal{L}(\mathsf{RMG})$ and $\mathcal{L}(\mathsf{BFA}) \oplus \mathcal{L}(\mathsf{BFA})$ and the rules $S \to X_1 A$ and $A \to X_2$ with
either $\pi(X_1), \pi(X_2) \in \mathcal{L}(\mathsf{RMG})$ and $\tau(S \to X_1 A) = \ominus$
or $\pi(X_1), \pi(X_2) \in \mathcal{L}(\mathsf{BFA})$ and $\tau(S \to X_1 A) = \oplus$.

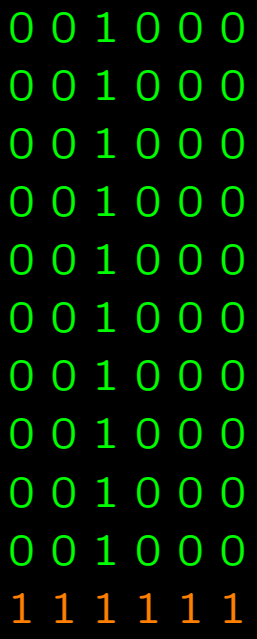
$(\mathcal{L}(\mathsf{RMG}) \ominus \mathcal{L}(\mathsf{RMG})) \subsetneq \mathcal{L}_{\oplus-\ell,\ominus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG})$
and $(\mathcal{L}(\mathsf{BFA}) \oplus \mathcal{L}(\mathsf{BFA})) \subsetneq \mathcal{L}_{\ominus-\ell,\oplus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG})$.

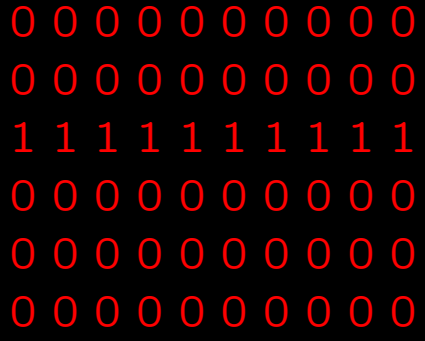
Lemma 8.

$$(\mathcal{L}(\mathsf{RMG}) \ominus \mathcal{L}(\mathsf{RMG})) \cup (\mathcal{L}(\mathsf{BFA}) \oplus \mathcal{L}(\mathsf{BFA}))$$

$$\subsetneq \mathcal{L}_{\ominus-\ell,\oplus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG}) \cap \mathcal{L}_{\oplus-\ell,\ominus-r}((\mathsf{R} : \mathsf{R})\mathsf{AG}).$$

Results





Lemma 9. $(\mathcal{L}(\mathrm{RMG}) \ominus \mathcal{L}(\mathrm{RMG})) \subsetneq (\mathcal{L}(\mathrm{RMG}) \ominus \mathcal{L}(\mathrm{RMG}) \ominus \mathcal{L}(\mathrm{RMG}))$.

Corollary 10. $(\mathcal{L}(\mathrm{BFA}) \oplus \mathcal{L}(\mathrm{BFA})) \subsetneq (\mathcal{L}(\mathrm{BFA}) \oplus \mathcal{L}(\mathrm{BFA}) \oplus \mathcal{L}(\mathrm{BFA}))$.

Results

Recall: For $k, k' \in \mathbb{N}$, by W^k we denote the k-fold column-concatenation of the given array W, by W_k we denote the k-fold row-concatenation of W.

Theorem 11. *For each $k \geq 1$, we have:*
$\mathcal{L}(\mathsf{RMG})_k \subsetneq \mathcal{L}(\mathsf{RMG})_{k+1}$*, as well as* $\mathcal{L}(\mathsf{BFA})^k \subsetneq \mathcal{L}(\mathsf{BFA})^{k+1}$.

As (fixed) finite catenations of a single type can be expressed both by $\mathcal{L}_{\ominus-\ell,\oplus-r}((\mathsf{R}:\mathsf{R})\mathsf{AG})$ and by $\mathcal{L}_{\oplus-\ell,\ominus-r}((\mathsf{R}:\mathsf{R})\mathsf{AG})$, we conclude:

Corollary 12. *For any $k \geq 1$,*
$\mathcal{L}(\mathsf{RMG})_k \cup \mathcal{L}(\mathsf{BFA})^k \subsetneq \mathcal{L}_{\ominus-\ell,\oplus-r}((\mathsf{R}:\mathsf{R})\mathsf{AG}) \cap \mathcal{L}_{\oplus-\ell,\ominus-r}((\mathsf{R}:\mathsf{R})\mathsf{AG})$.

Results

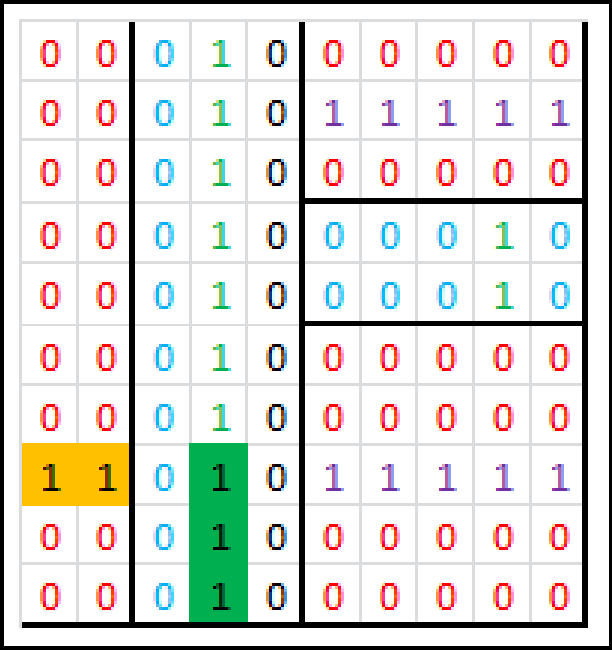
Theorem 13. $\mathcal{L}((\mathrm{R} : \mathrm{R})\mathrm{AG})$ *is not closed under union.*

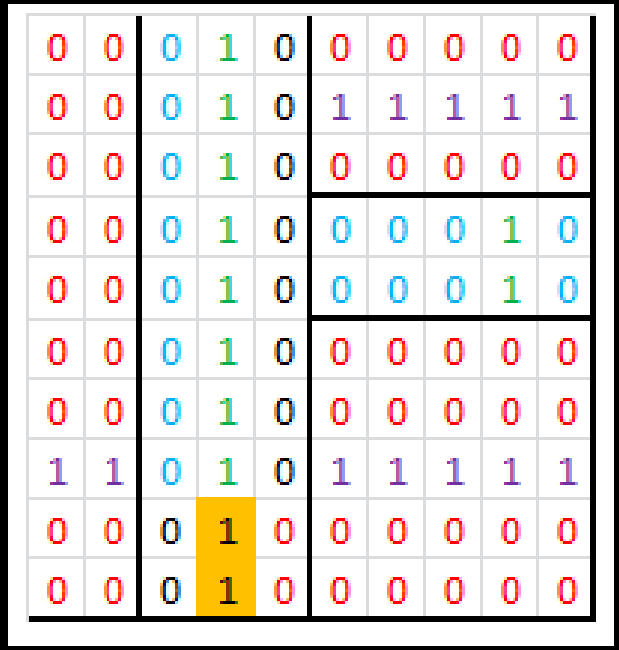
Proof Idea:

Recall $L = \{0\}^{+}_{+} \oplus \{1\}_{+} \oplus \{0\}^{+}_{+}$

0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0
0 0 1 0 0 0

Let $\widehat{L} = (T(L) \ominus L \ominus T(L)) \oplus L \oplus T(L)$. $\widehat{L}$ can be described by some $\oplus$-left, $\ominus$-right (R:R)AG (see end of page 125). Note that $H(\widehat{L}) = T(L) \oplus L \oplus (T(L) \ominus L \ominus T(L))$.





Lemma 14. $\mathcal{L}_{\ominus-\ell,\oplus-r}((\mathrm{R}:\mathrm{R})\mathrm{AG})$ *and* $\mathcal{L}_{\oplus-\ell,\ominus-r}((\mathrm{R}:\mathrm{R})\mathrm{AG})$ *are closed under union.*

Map

- Regular Matrix Grammars
- (Regular : Regular) Array Grammars
- Isometric Regular Array Grammars
- Relation b/w language families

Regular Array Grammars

Isometric Approach: Σ^{++}
Non-Rectangular Arrays

Non-Isometric Approach: $\Sigma^{\substack{+\\+}}$
Rectangular Arrays

Isometric Regular Array Grammars

An *isometric regular array grammar (IRAG)* is defined by a 5-tuple $G = (N, \Sigma, P, S, \#)$, where:

- N - *nonterminal alphabet*; Σ - *terminal alphabet*;
  P - set of *rules*; $S \in N$ - *start symbol*; $\#$ - blank symbol;

- Every rule from P is of the form

$$\#A \to Ba \quad , \quad A\# \to aB \quad , \quad \begin{matrix} \# \\ A \end{matrix} \to \begin{matrix} B \\ a \end{matrix} \quad , \quad \begin{matrix} A \\ \# \end{matrix} \to \begin{matrix} a \\ B \end{matrix} \quad , \text{ or } \quad A \to a \quad ,$$

  where $A, B \in N$ and $a \in \Sigma$.

Example of an IRAG

IRAG using all four types of movements: Consider the array language L the set of all square pictures of diagonal lines from the upper left corner to the lower right corner where the elements in the diagonal are 1 and the other elements are 0.

The IRAG $G = (N, \Sigma, P, S, \#)$ such that $L_{Rect}(G) = L$ is defined as:

$N = \{S, A, B, C, H, E, F\}$, $\Sigma = \{0, 1\}$, $P = P_1 \cup P_2 \cup P_3$ where

$P_1 = \{1 : \begin{matrix}\# \\ S\end{matrix} \to \begin{matrix}A \\ 1\end{matrix}$, $2 : \begin{matrix}\# \\ A\end{matrix} \to \begin{matrix}A \\ 0\end{matrix}$, $3 : \#A \to B\,0$, $4 : \#B \to B\,0$, $5 : \begin{matrix}B \\ \#\end{matrix} \to \begin{matrix}1 \\ F\end{matrix}$,
$6 : \begin{matrix}C \\ \#\end{matrix} \to \begin{matrix}0 \\ C\end{matrix}$, $7 : C\# \to 0\,H$, $8 : H\# \to 0\,H$, $9 : \begin{matrix}\# \\ H\end{matrix} \to \begin{matrix}E \\ 0\end{matrix}$,
$a : \begin{matrix}\# \\ E\end{matrix} \to \begin{matrix}A \\ 1\end{matrix}$, $b : \begin{matrix}F \\ \#\end{matrix} \to \begin{matrix}0 \\ C\end{matrix}$, $c : S \to 1$ $\}$, $P_2 = \{d : F \to 0\}$ and $P_3 = \{f : E \to 1\}$.

$n = 5$:

$$\begin{array}{ccccccccc}
1 & \leftarrow_4 & 0 & \leftarrow_4 & 0 & \leftarrow_4 & 0 & \leftarrow_3 & 0 \\
\downarrow_5 & & & & & & & & \uparrow_2 \\
0 & & 1 & \leftarrow_4 & 0 & \leftarrow_3 & 0 & & 0 \\
\downarrow_b & & \downarrow_5 & & & & \uparrow_2 & & \uparrow_2 \\
0 & & 0 & & 1_f & & 0 & & 0 \\
\downarrow_6 & & \downarrow_b & & \uparrow_9 & & \uparrow_a & & \uparrow_2 \\
0 & & 0 & \rightarrow_7 & 0 & & 1 & & 0 \\
\downarrow_6 & & & & & & \uparrow_9 & & \uparrow_1 \\
0 & \rightarrow_7 & 0 & \rightarrow_8 & 0 & \rightarrow_8 & 0 & & 1
\end{array}$$

$n = 6$:

$$\begin{array}{ccccccccccc}
1 & \leftarrow_4 & 0 & \leftarrow_4 & 0 & \leftarrow_4 & 0 & \leftarrow_4 & 0 & \leftarrow_3 & 0 \\
\downarrow_5 & & & & & & & & & & \uparrow_2 \\
0 & & 1 & \leftarrow_4 & 0 & \leftarrow_4 & 0 & \leftarrow_3 & 0 & & 0 \\
\downarrow_b & & \downarrow_5 & & & & & & \uparrow_2 & & \uparrow_2 \\
0 & & 0 & & 1 & \leftarrow_3 & 0 & & 0 & & 0 \\
\downarrow_6 & & \downarrow_b & & \downarrow_5 & & \uparrow_a & & \uparrow_2 & & \uparrow_2 \\
0 & & 0 & & 0_d & & 1 & & 0 & & 0 \\
\downarrow_6 & & \downarrow_6 & & & & \uparrow_9 & & \uparrow_a & & \uparrow_2 \\
0 & & 0 & \rightarrow_7 & 0 & \rightarrow_8 & 0 & & 1 & & 0 \\
\downarrow_6 & & & & & & & & \uparrow_9 & & \uparrow_1 \\
0 & \rightarrow_7 & 0 & \rightarrow_8 & 0 & \rightarrow_8 & 0 & \rightarrow_8 & 0 & & 1
\end{array}$$

More Results

Lemma 15. *A language of single-row (column) arrays generated by some IRAG corresponds to a regular string language.*

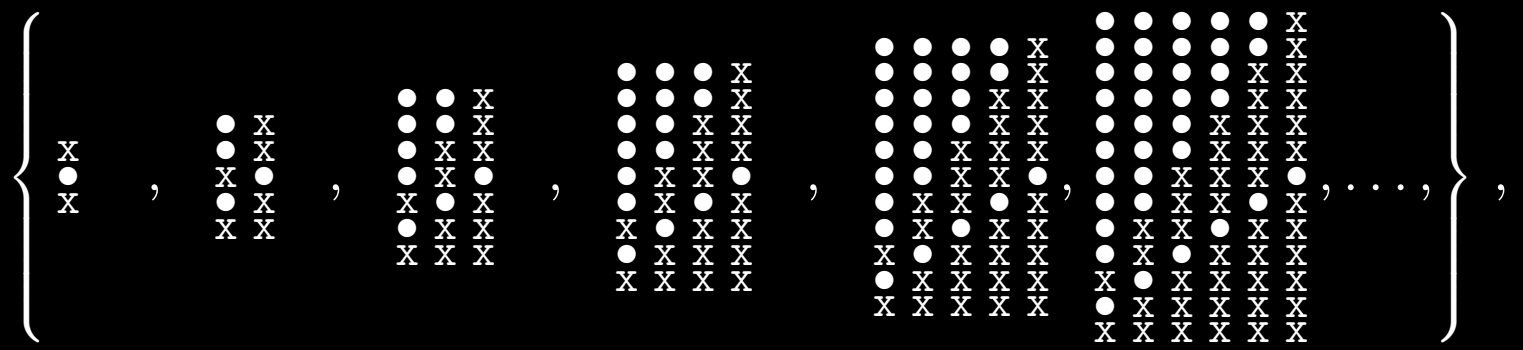
Definition 16. Semi-holes *are some blank symbol surrounded by three terminal symbols and one blank symbol on its four sides, or by two terminal symbols and two blank symbols.* Holes *are some blank symbols completely surrounded by terminal symbols.*

Lemma 17. *Each BFA can be simulated by some self-delimiting IRAG.*

Lemma 18. *Each BFA can be simulated by some IRAG that scans a picture with a fixed left border, starting in any corner and ending in a different corner of the array, assuming that the picture is big enough.*

Example for IRAG

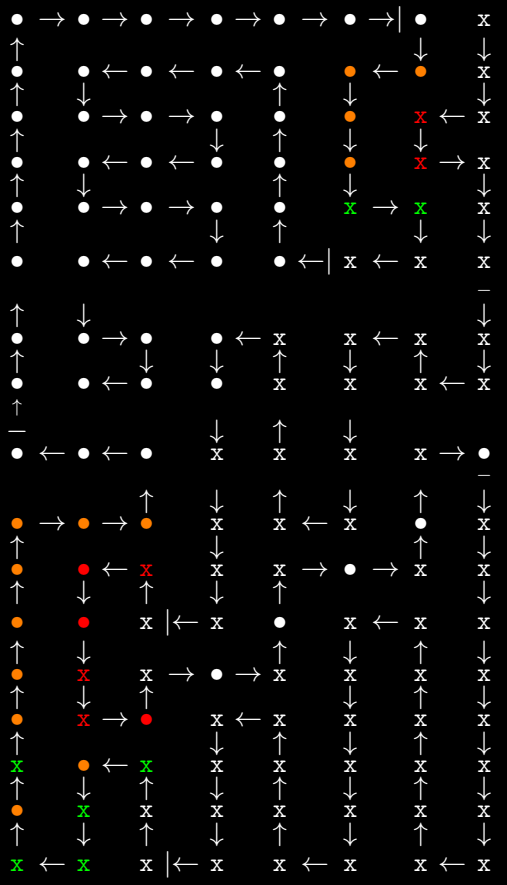
Consider the array language

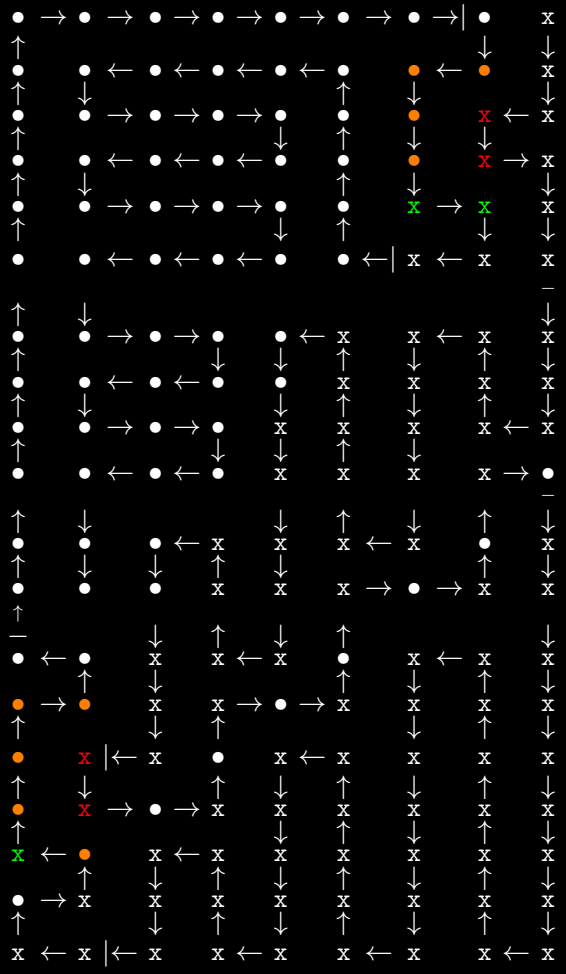
$L =$ 

as argued by [Siromoney et al 73] there is no (R:R)AG that can generate this language L.

Next Slide: Attempt to define an IRAG for L.

How to scan like IRAG: the picture M_8 (on the left) and M_9 (on the right)





More Results

Theorem 19. $\mathcal{L}((\mathsf{R}:\mathsf{R})\mathsf{AG}) \subsetneq \mathcal{L}_{Rect}(\mathsf{IRAG})$.

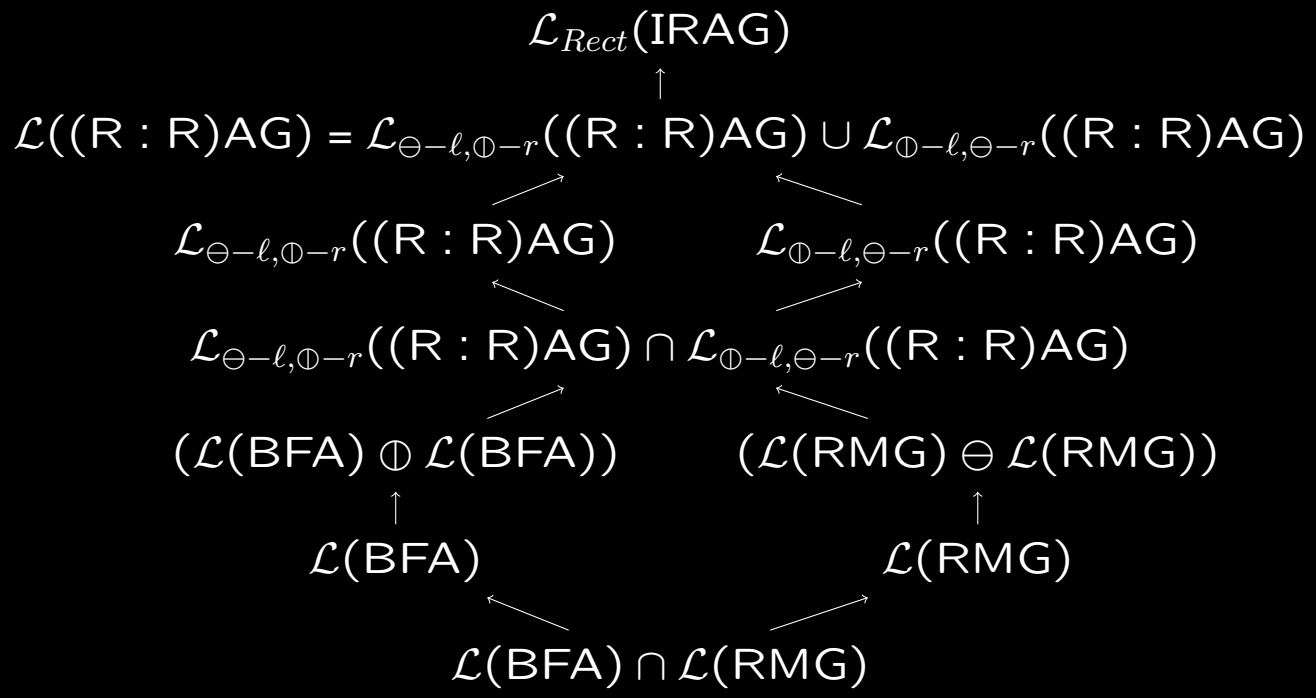
Theorem 20. *For a language of single-row (column) arrays L, the following are equivalent.*

- L *corresponds to a regular string language.*
- $L \in \mathcal{L}(\mathsf{RMG}) \cap \mathcal{L}(\mathsf{BFA})$.
- $L \in \mathcal{L}_{\ominus-\ell,\oplus-r}((\mathsf{R}:\mathsf{R})\mathsf{AG}) \cap \mathcal{L}_{\oplus-\ell,\ominus-r}((\mathsf{R}:\mathsf{R})\mathsf{AG})$.
- L *is generated by some (R:R)AG.*
- L *is generated by some IRAG.*

Map

- Regular Matrix Grammars
- (Regular : Regular) Array Grammars
- Isometric Array Languages and Regular Grammars
- Relation b/w language families

Relation b/w language families



X X X X X · X · · · X · · X X X · · X · · · X · X · · · X

X · · · X · · X X X · · X · · · X

· X X X · · X · · · X · X X X X ·