A parsing technique for TRG languages

Daniele Paolo Scarpazza
Politecnico di Milano
<daniele.scarpazza@elet.polimi.it>

October 15th, 2004

Abstract

In this presentation, we propose a parsing algorithm for TRG languages which is an extension of the Cocke-Kasami-Younger (CKY) algorithm. Our algorithm exhibits a polynomial time complexity.

Agenda:

- a brief review of the original CKY algorithm;
- issues in the extension of CKY algorithm to TRGs;
- a formal description of the steps composing the algorithm;
- the derivation of time complexity;
- a fully-developed example illustrating the algorithm at work.

A brief review of the CKY algorithm

- The CKY algorithm is a bottom-up parsing technique for context-free grammars; the simplest formulation operates on grammars in the Chomsky normal form (CNF);
- let $G = (V, \Sigma, P, S)$ be a CNF context-free grammar;
- let $x \in \Sigma^*$ be a string and $|x| = n$ its length;
- let x_i denote the i-th symbol of x;
- given G and x, the algorithm decides in $O(n^3)$ time whether $x \in (G)$;

```
Input : $G, x$;
Output : $x \in (G)$?;
for all $r = 0$ to $n$ do
    $V_{r,1} \leftarrow \{A \mid (A \rightarrow n_r) \in P\}$
end for
for all $c = 2$ to $n$ do
    for all $r = 1$ to $n - c + 1$ do
        $V_{r,c} \leftarrow \{\}$
        for all $r = 1$ to $n - c + 1$ do
            if $(A \rightarrow BC) \in P,\ B \in V_{r,k},\ C \in V_{r+k,c-k}$ then
                $V_{r,c} \leftarrow V_{r,c} \cup \{A\}$
            end if
        end for
    end for
end for
if $S \in V_{1,n}$ then
    output yes
else
    output no
end if
```

An example illustrating the CKY algorithm

Grammar:

$$\begin{aligned} S &\rightarrow SS \\ S &\rightarrow AA \\ S &\rightarrow b \\ A &\rightarrow AS \\ A &\rightarrow AA \\ A &\rightarrow a \end{aligned}$$

Recognition table:

	$V_{r,c}$	$c:1$	2	3	4
a	$r:1$	A	A,S	A,S	A,S
a	2	A	A	A	$\cdot$
b	3	S	S	$\cdot$	$\cdot$
b	4	S	$\cdot$	$\cdot$	$\cdot$

String: $x = aabb$

Preliminaries: coordinates and rectangles

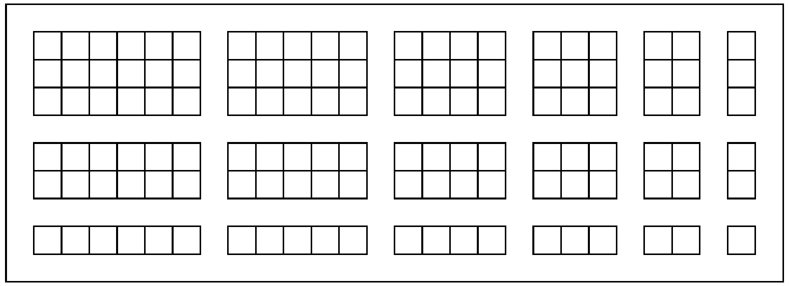
- A *rectangle*, indicated as $r \times c$, is the following set of 2D integer coordinates: $r \times c = \{(1,1),(1,2),\ldots,(1,c),(2,1),(2,2),\ldots,(2,c),\ldots,(r,1),(r,2),\ldots,(r,c)\}$; example: $2 \times 3 = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3)\}$.
- A *subrectangle*, indicated as $r \times c + (a,b)$, is the following set of couples:
$$r \times c + (a,b) = \left\{ \begin{array}{llll} (a,b), & (a,b+1), & \ldots & (a,b+c-1), \\ (a+1,b), & (a+1,b+1), & \ldots & (a+1,b+c-1), \\ \ldots & \ldots & \ldots & \ldots \\ (a+r-1,b), & (a+r-1,b+1), & \ldots & (a+r-1,b+c-1) \end{array} \right\};$$
example: $2 \times 3 + (4,5) = \{(4,5),(4,6),(4,7),(5,5),(5,6),(5,7)\}$.
- Note: rectangles and subrectangles are sets; usual set operators $\cup, \cap, -, =, \neq, \supset$ and $\subset$ are defined. Operands $\cup, \cap, -$ on rectangles yield sets which are not, in general, subrectangles. Example: $(2 \times 3 + (4,5)) \cap (2 \times 3 + (5,6)) = (1 \times 2 + (5,6))$.
- A new notation to indicate subpictures: given p and q, we say that

$$p[r \times c + (i,j)] = q \qquad \text{iff} \qquad q \trianglelefteq_{(i,j)} p \ \wedge\ |q| = (r,c);$$

$$\text{example:} \qquad p = \begin{pmatrix} a & d & g & j & m \\ b & e & h & k & n \\ c & f & i & l & o \end{pmatrix} \quad \Rightarrow \quad p[2 \times 3 + (1,2)] = \begin{pmatrix} d & g & j \\ e & h & k \end{pmatrix}.$$

Extending the CKY algorithm to TRGs (1/3)

- Cells in the CKY recognition matrix represent all the possible *substrings* of the input string;
- for TRGs, we need to represent all the possible *subpictures* of the input picture;
- the extension of the recognition matrix is the *tableau*, where each cell represents a possible subpicture.
- Formally: A *tableau* is a matrix of variable-size matrices. A $m \times n$ tableau T contains $m \times n$ matrices, each denoted by $T_{i,j}, \forall (i,j) \mid 1 \leq i \leq m, 1 \leq j \leq n$. Matrix $T_{i,j}$ must have size $(m-i+1, n-j+1)$. The notation $T[i \times j + (a,b)]$ indicates the (a,b) element of matrix $T_{i,j}$, or $(T_{i,j})_{a,b}$.
- Example: the following figure shows a 3×6 tableau.



Extending the CKY algorithm to TRGs (2/3)

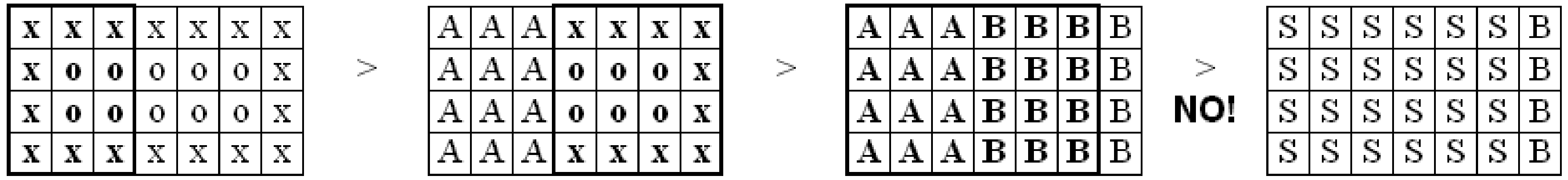
- In the CKY recognition matrix, a symbol A in cell $V_{r,c}$ indicates that substring $x_r \ldots x_{r+c-1}$ is recognized as generated by non-terminal A;
- in TRGs, rules are *isometric*, and either fixed-size or variable-size; *variable-size* rules imply the recognition of $LOC_{u,eq}$ languages;
- as a consequence, the algorithm requires more information in a tableau cell than just A;
- more precisely, elements appearing in cell $T[i \times j + (a,b)]$ must provide information on which rules are recognizable rule on subpicture $i \times j + (a,b)$ and which tiles are missing.
- Example: element $(R_3, \{t_{3,2}, t_{3,3}, t_{3,5}\}, \ldots)$ in a tableau cell indicates that the subpicture corresponding to that tableau cell only uses tiles present in the right-part of rule R_3, and tiles $t_{3,2}, t_{3,3}$, are $t_{3,5}$ not used in the picture.
- An element $(R_i, \{\}, \ldots)$ indicates that the subpicture belongs to $LOC_{u,eq}$ of rule R_i, thus can be recognized as a sentential form of rule R_i.

Extending the CKY algorithm to TRGs (3/3)

- Due to the formal definitions of TRGs, application areas in a derivation must be disjoint, or included the latter in the former in couples. More precisely, if application areas are ordered $\alpha_1, \ldots, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+j}, \ldots$:

$$(\alpha_i \cap \alpha_{i+j} = \emptyset) \vee (\alpha_i \supseteq \alpha_{i+j}) \ .$$

- During bottom-up recognition, the reversed version of the above principle must be enforced. A violation is illustrated in the following figure:



- To enforce the above principle, we use *multipictures*, i.e. pictures containing couples (symbol, area). In the above example, each B would be replaced by $(B, 4 \times 4 + (1,4))$. The last step would be illegal, because

$$(4 \times 6 + (1,1)) \not\supseteq (4 \times 4 + (1,4)), \qquad (4 \times 6 + (1,1)) \cap (4 \times 4 + (1,4) \neq \emptyset).$$

Preliminaries: Multipictures and compatibility

- A *multipicture* is a matrix $m \times n$ where each cell is a set of couples (I, r) where $I \in (\Sigma \cup N)$, and r is a subrectangle $\subseteq m \times n$; our algorithm uses one multipicture; each cell of it initially contains the pixel of the input picture; then, for every recognized application area, corresponding cells are filled with the nonterminal;
- Given a picture p and a multipicture M, both of size $m \times n$ (and alphabet respectively I and $2^{I \times \mathcal{S}(m \times n)}$), the *compatibility* between p and M (denoted as $p :: M$) is the set of pictures p', of size $m \times n$ and alphabet $I \times \mathcal{S}(m \times n)$, such that $\forall i, k \in \{1 ... m\} \forall j, l \in \{1 ... n\}$:
$$p'_{i,j} = (p_{i,j}, \alpha_{i,j}) \in M_{i,j}$$
and $\forall i, k \in \{1 ... m\} \forall j, l \in \{1 ... n\}$:
$$(\alpha_{i,j} = \alpha_{k,l} \wedge p_{i,j} = p_{k,l}) \vee (\alpha_{i,j} \cap \alpha_{k,l} = \emptyset).$$
- We say that p is *compatible* with M iff $p :: M \neq \emptyset$.

p

x	x	x	x	x	x
x	o	o	o	o	x
x	o	o	o	o	x
x	x	x	x	x	x

>

p'

A	A	A	x	x	x
A	A	A	o	o	x
A	A	A	o	o	x
A	A	A	x	x	x

M

x, 1x1+(1,1)	x, 1x1+(1,2)	x, 1x1+(1,3)	x, 1x1+(1,4)	x, 1x1+(1,5)	x, 1x1+(1,6)
x, 1x1+(2,1)	o, 1x1+(2,2)	o, 1x1+(2,3)	o, 1x1+(2,4)	o, 1x1+(2,5)	x, 1x1+(2,6)
x, 1x1+(3,1)	o, 1x1+(3,2)	o, 1x1+(3,3)	o, 1x1+(3,4)	o, 1x1+(3,5)	x, 1x1+(3,6)
x, 1x1+(4,1)	x, 1x1+(4,2)	x, 1x1+(4,3)	x, 1x1+(4,4)	x, 1x1+(4,5)	x, 1x1+(4,6)

>

M'

x, 1x1+(1,1) A, 4x3+(1,1)	x, 1x1+(1,2) A, 4x3+(1,1)	x, 1x1+(1,3) A, 4x3+(1,1)	x, 1x1+(1,4)	x, 1x1+(1,5)	x, 1x1+(1,6)
x, 1x1+(2,1) A, 4x3+(1,1)	o, 1x1+(2,2) A, 4x3+(1,1)	o, 1x1+(2,3) A, 4x3+(1,1)	o, 1x1+(2,4)	o, 1x1+(2,5)	x, 1x1+(2,6)
x, 1x1+(3,1) A, 4x3+(1,1)	o, 1x1+(3,2) A, 4x3+(1,1)	o, 1x1+(3,3) A, 4x3+(1,1)	o, 1x1+(3,4)	o, 1x1+(3,5)	x, 1x1+(3,6)
x, 1x1+(4,1) A, 4x3+(1,1)	x, 1x1+(4,2) A, 4x3+(1,1)	x, 1x1+(4,3) A, 4x3+(1,1)	x, 1x1+(4,4)	x, 1x1+(4,5)	x, 1x1+(4,6)

Example: pictures (p, p') and corresponding multipictures (M, M') in the first step of a recognition chain. Please consider the sub-multipicture $M'[2x2 + (1,1)]$, bordered in the figure. For it, compatibilities with two tiles are reported:

$$M'[2x2+(1,1)] \quad :: \quad \begin{array}{cc} A & A \\ A & A \end{array} = \left\{ \begin{array}{cc} (A, 4{\times}3{+}(1,1)) & (A, 4{\times}3{+}(1,1)) \\ (A, 4{\times}3{+}(1,1)) & (A, 4{\times}3{+}(1,1)) \end{array} \right\}$$

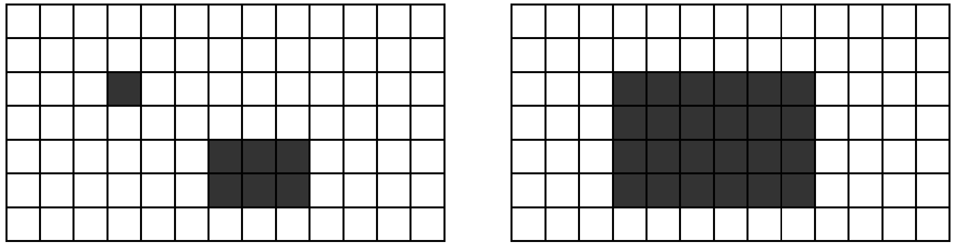
$$M'[2x2+(1,1)] \quad :: \quad \begin{array}{cc} x & x \\ x & o \end{array} = \left\{ \begin{array}{cc} (x, 1{\times}1{+}(1,1)) & (x, 1{\times}1{+}(1,2)) \\ (x, 1{\times}1{+}(2,1)) & (o, 1{\times}1{+}(2,2)) \end{array} \right\}$$

Preliminaries: Copy rules and subrectangles

- If $G = (\Sigma, N, S, R)$ is a TRG, and $A, B \in N$, a rule in the form

$$A \rightarrow \left\{ \begin{array}{cc} B & B \\ B & B \end{array} \right\}$$

  is said to be a *copy rule*; it can be easily proved that for every TRG G with copy rules, $\exists G' \mid \mathcal{L}(G) = \mathcal{L}(G')$ and G' is free from copy rules; with no loss of generality, our algoritm operates on TRG grammars free from copy rules.
- We introduce the function $Sr(\cdot)$ which, given a set of coordinates, returns the smaller subrectangle containing them. Example:



$$Sr(1 \times 1 + (3,4) \cup 2 \times 3 + (5,7)) = 4 \times 6 + (3,4)$$

Our algorithm – hypotheses

- Let a TRG $G(\Sigma, N, S, R)$ and a picture $p \in \Sigma^{(m,n)}$ be given. The following algorithm determines whether $p \in \mathcal{L}(G)$ or not.
- let G be free from copy rules;
- let all rules in R be variable-size rules (the extension of the algorithm to fixed-size rules is trivial);
- let the rules in R be numbered as R_1, R_2, …, and their right-hand sides as ω_1, ω_2, …; let the tiles appearing in ω_i be numbered as $t_{i,1}, t_{i,2}, \ldots$;
- the algorithm operates by constructing a $m \times n$ tableau T, and a $m \times n$ multipicture M; initially, $T[r \times c + (i,j)] = \emptyset \quad \forall r, c, i, j$; and $M_{i,j} = \{(p_{i,j}, 1 \times 1 + (i,j))\}$;
- the following steps are applied until *fixed point* is reached;

Our algorithm – step # 1

Step 1: update each cell $T[2 \times 2 + (i,j)]$, adding elements: if

$$R_e = (Q \rightarrow \omega) \in R, \qquad t \in \omega, \qquad t' \in (t :: M[2 \times 2 + (i,j)]),$$

$$t' = \begin{pmatrix} (t_{1,1}, \alpha_{1,1}) & (t_{1,2}, \alpha_{1,2}) \\ (t_{2,1}, \alpha_{2,1}) & (t_{2,2}, \alpha_{2,2}) \end{pmatrix}$$

then:

$$(R_e, \omega - t, Sr(\alpha_{1,1} \cup \alpha_{1,2} \cup \alpha_{2,1} \cup \alpha_{2,2})) \in T[2 \times 2 + (i,j)] \ ;$$

Where the function $Sr(\cdot)$ yields the smallest subrectangle including the given elements.

Our algorithm – step # 2

Step 2: update T adding elements:
$\forall(r,c) \mid 2 < r \leq m, 2 < c \leq n$ in lexicographical order: if

$$(R_e, \omega_1, \alpha_1) \in T[r{\times}(c{-}1){+}(i,j)] \quad \wedge \quad (R_e, \omega_2, \alpha_2) \in T[r{\times}(c{-}1){+}(i,j+1)] \text{ or}$$
$$(R_e, \omega_1, \alpha_1) \in T[(r{-}1){\times}c{+}(i,j)] \quad \wedge \quad (R_e, \omega_2, \alpha_2) \in T[(r{-}1){\times}c{+}(i+1,j)]$$

then

$$(R_e, \omega_1 \cap \omega_2, Sr(\alpha_1 \cup \alpha_2)) \in T[r \times c + (i,j)] \ ;$$

Comment: elements of two horizontally adjacent tableau cells $m \times n$ can be merged into a corresponding $m \times (n+1)$ cell. Similarly for vertically adjacent cells. Given the same rule, the set of missing tiles is the intersection of the sets of missing tiles, and the scope is the $Sr(\cdot)$ of the union of scopes.

Our algorithm – step # 3

Step 3: update M adding elements: for each element such that

$$(R_e, \{\}, \alpha) \in T[r \times c + (i,j)] \;\mid\; \alpha \subseteq r \times c + (i,j),\; R_e = (A \rightarrow ...)$$

add elements:

$$\forall (i,j) \in r \times c + (i,j) \qquad (A, (r \times c + (i,j)) \in M_{i,j}\ ;$$

Comment: whenever all the tiles of subpicture $r \times c + (i,j)$ appear in the right part of rule R_e, and no tile is missing $(R_e, \{\}, \ldots)$ and the scope α of all the recognized symbols is inside the current subpicture, then recognize the subpicture as application area of rule R_e. Update multipicture, adding a couple $(A, r \times c + (i,j))$ in every cell of sub-multipicture $r \times c + (i,j)$.

Our algorithm – final step

Final step (to be executed when fixed point is reached):
if $\forall(i,j) \in r \times c \qquad (S, m \times n + (1,1)) \in M_{i,j}$ then:

$$\text{declare} \qquad p \in \mathcal{L}(G)$$

else

$$\text{declare} \qquad p \notin \mathcal{L}(G).$$

Comment: the recognition of a rule which has the starting symbol S as the left-hand side, and the whole picture as application area ($m \times n + (1,1)$), indicates that the picture is recognized. In short, the multipicture must contain element $(S, m \times n)$ in every cell.

If such area is not recognized and a fixed point is reached, the picture is not in the language.

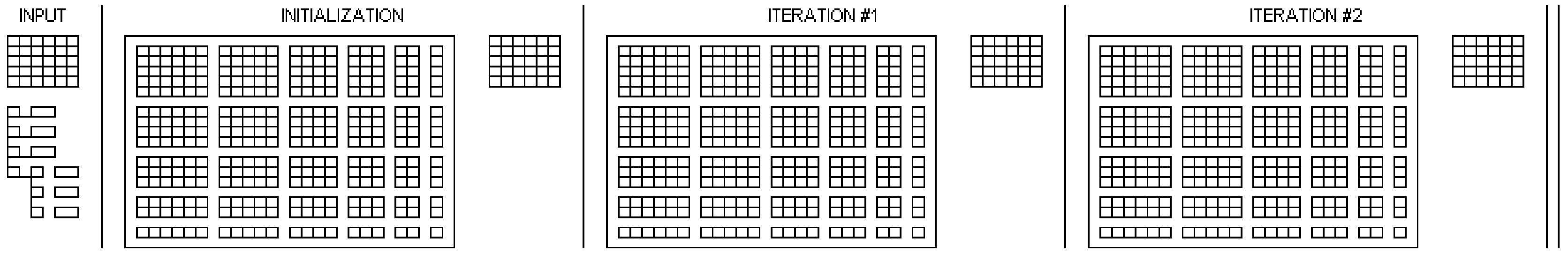
Our algorithm – informal comments

- The algorithm scans the tableau and the multipicture several times; at each scan, one or more disjoint application areas are recognized;
- each tableau cell $T[r \times c + (i, j)]$ is filled with triples (R_e, ω, α) where if rule $R_e = A \rightarrow \omega_e$ exists, then
$$\omega = \omega_e - B_{2,2}(M[r \times c + (i, j)]).$$
- When $(R_e, \emptyset, \alpha) \in T[r \times c + (i, j)]$, then exactly all the tiles in rule R_e are used in subpicture $M[r \times c + (i, j)]$; moreover
- α is the smallest subrectangle including the activation areas of the symbols in the multipicture which comply with rule R_e.
- If α is not smaller or equal to the subrectangle $r \times c + (i, j)$ the rule is discarded, since it would violate the theorem of disjointness of application areas.
- Every time a rule $R_e = (A \rightarrow \ldots)$ is recognized over an activation area q, a couple (A, q) is added to every cell of $M[q]$, so that the next iteration of the algorithm can use the newly added symbol to recognize larger activation areas.

Time complexity

- if the input picture has dimensions $m \times n$, the input size is $N = m \cdot n$; time complexity is derived as a function of the size of the input N;
- the complexity is determined by the most complex step (number 2);
  - this step fills a $m \times n$ tableau which has $\frac{m(m-1)n(n-1)}{4} \leq k_1 N^2$ cells;
  - for each cell, it compares couples of elements of exactly two other cells;
  - elements are in the form $(R_e, \omega, \alpha) \Rightarrow$ they can be at most $|R| \cdot max_i |\omega_{R_i}| \cdot |\mathcal{S}(m \times n)|$, where $k_2 = |R| \cdot max_i |\omega_{R_i}|$ is fixed, while $|\mathcal{S}(m \times n)| \leq k_1 N^2$;
  - therefore, the time complexity of each comparison in step 2 is $\leq (k_1 k_2)^2 N^4$, and the overall complexity of step 2 is $\leq k_1 (k_1 k_2)^2 N^6$.
- Steps 1–3 are iterated until fixed point. At every iteration, one or more application areas are detected. Thanks to the exclusion of copy rules, an application area can appear at most once in a derivation.
- Therefore # of iterations $\leq$ # application areas $\leq k_1 N^2$.
- Therefore, the time complexity of the whole algorithm is $O(N^8)$.

A full example

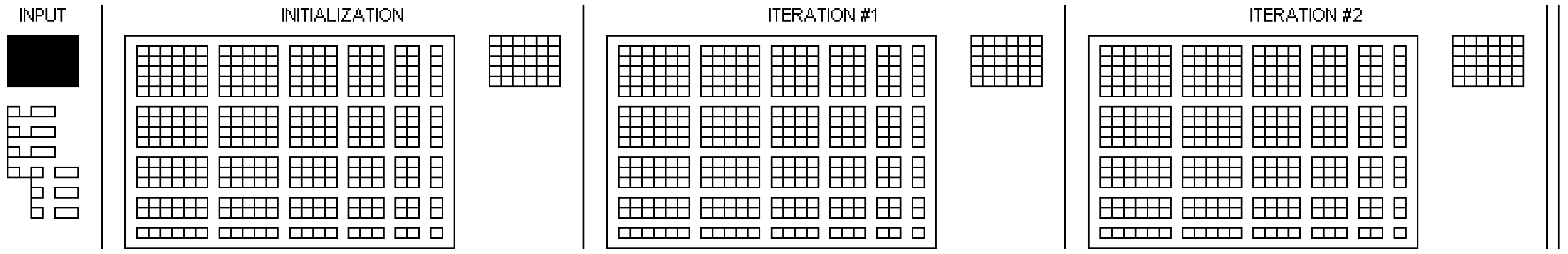


In the next slides we fully develop an example, with the help of the above storyboard. Elements updated in each slide will be marked in black. Elements will be illustrated in this order:

- the input picture and the input grammar;
- the result of initialization on the tableau T and on the multipicture M will be shown;
- the effects of the first iteration;
- the effects of the second iteration (and conclusion).

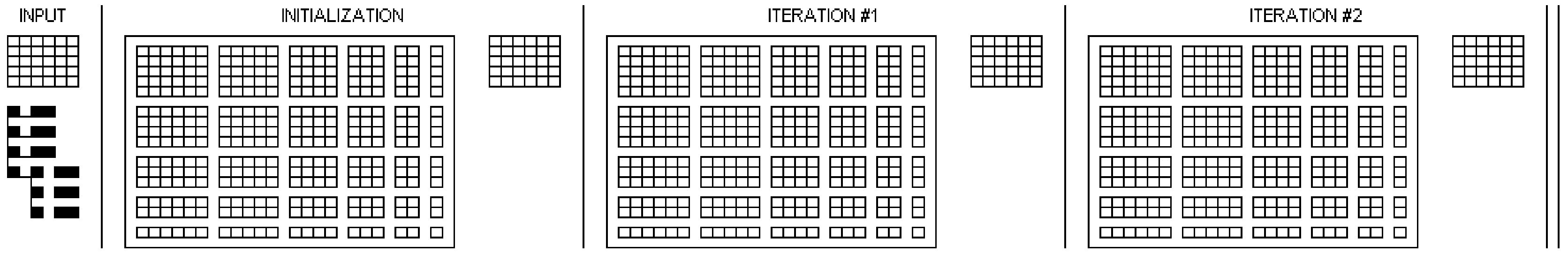
For sake of clarity, T and M at initialization, iteration 1 and 2 are represented as separate entities.

A full example: the input picture



$$p = \begin{array}{|c|c|c|c|c|c|}\hline x & x & x & x & x & x \\ \hline x & o & o & o & o & x \\ \hline x & o & o & o & o & x \\ \hline x & o & o & o & o & x \\ \hline x & x & x & x & x & x \\ \hline \end{array}$$

A full example: the input grammar (1/2)



$$G = (\Sigma, N, S, R)$$
$$\Sigma = \{x, o\}$$
$$N = \{S, A, B\}$$

$$R_1 \quad : \quad S \rightarrow B_{2,2}\begin{pmatrix} A & A & B & B \\ A & A & B & B \end{pmatrix}$$

$$R_2 \quad : \quad A \rightarrow B_{2,2}\begin{pmatrix} x & x & x \\ x & o & o \\ x & o & o \\ x & x & x \end{pmatrix}$$

$$R_3 \quad : \quad A \rightarrow B_{2,2}\begin{pmatrix} x & x & x \\ o & o & x \\ o & o & x \\ x & x & x \end{pmatrix}$$

A full example: the input grammar (2/2)

For ease of reference, we give names to the tiles in each right-hand side.

$$R_1 \quad : \quad S \rightarrow B_{2,2}\begin{pmatrix} A & A & B & B \\ A & A & B & B \end{pmatrix} = \left\{ t_{1,1} = \begin{pmatrix} A & A \\ A & A \end{pmatrix}, t_{1,2} = \begin{pmatrix} A & B \\ A & B \end{pmatrix}, t_{1,3} = \begin{pmatrix} B & B \\ B & B \end{pmatrix} \right\}$$

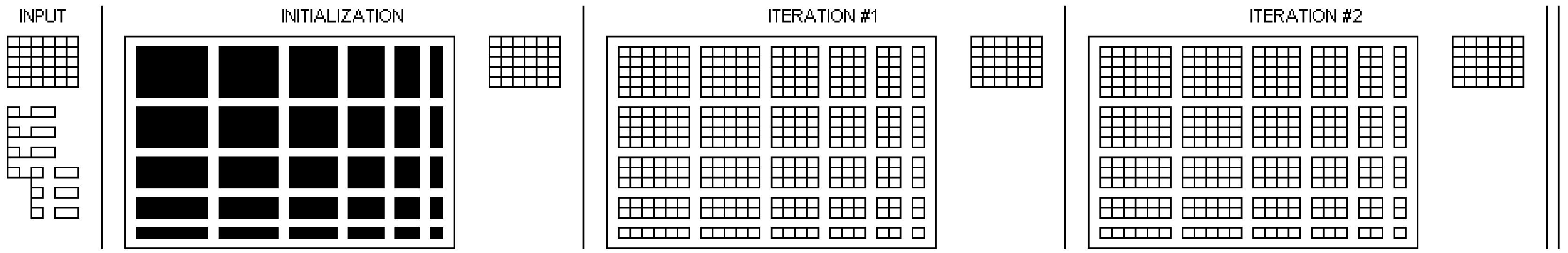
$$R_2 \quad : \quad A \rightarrow B_{2,2}\begin{pmatrix} x & x & x \\ x & o & o \\ x & o & o \\ x & x & x \end{pmatrix} = \left\{ t_{2,1} = \begin{pmatrix} x & x \\ x & o \end{pmatrix}, t_{2,2} = \begin{pmatrix} x & x \\ o & o \end{pmatrix}, t_{2,3} = \begin{pmatrix} x & o \\ x & o \end{pmatrix}, \right.$$

$$\left. t_{2,4} = \begin{pmatrix} x & o \\ x & x \end{pmatrix}, t_{2,5} = \begin{pmatrix} o & o \\ x & x \end{pmatrix}, t_{2,6} = \begin{pmatrix} o & o \\ o & o \end{pmatrix} \right\}$$

$$R_3 \quad : \quad A \rightarrow B_{2,2}\begin{pmatrix} x & x & x \\ o & o & x \\ o & o & x \\ x & x & x \end{pmatrix} = \left\{ t_{3,1} = \begin{pmatrix} x & x \\ o & x \end{pmatrix}, t_{3,2} = \begin{pmatrix} x & x \\ o & o \end{pmatrix}, t_{3,3} = \begin{pmatrix} o & x \\ o & x \end{pmatrix}, \right.$$

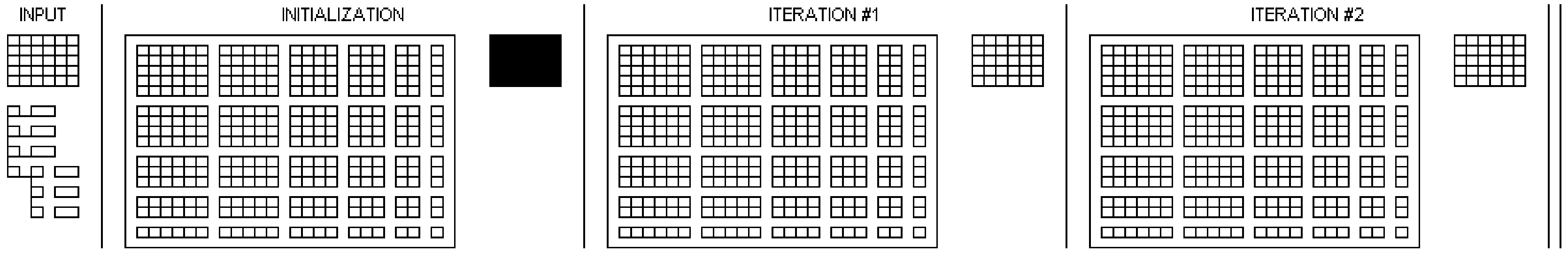
$$\left. t_{3,4} = \begin{pmatrix} o & x \\ x & x \end{pmatrix}, t_{3,5} = \begin{pmatrix} o & o \\ x & x \end{pmatrix}, t_{3,6} = \begin{pmatrix} o & o \\ o & o \end{pmatrix} \right\}$$

A full example: tableau initialization



- At initialization time, the tableau is empty.
- Being the input picture dimensions equal to 6×5, also the tableau T will have 6×5 cells.
- The first cell, $T_{1,1}$, will be a 6×5 matrix. Cells in this matrix represent each 1×1 tile in the input picture.
- $T_{1,2}$, will be a 6×4 matrix, i.e. with one less column, and so on, up to $T_{1,6}$, which is a 6×1 matrix. Same considerations apply to the second and following rows.
- The last cell, $T_{5,6}$, is a matrix composed by a single cell, representing the whole input picture.

A full example: multipicture initialization

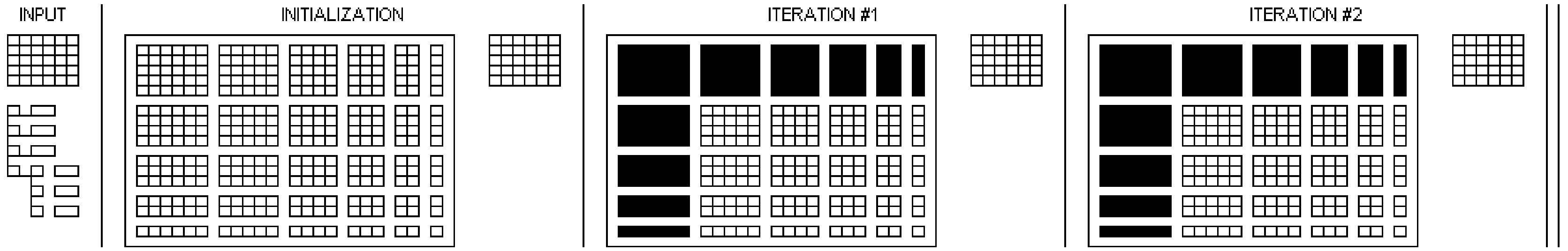


The multipicture is initially set to the contents of the picture. The *scope* of each terminal symbol is set to the 1×1 cell where the symbol belong.

$M =$

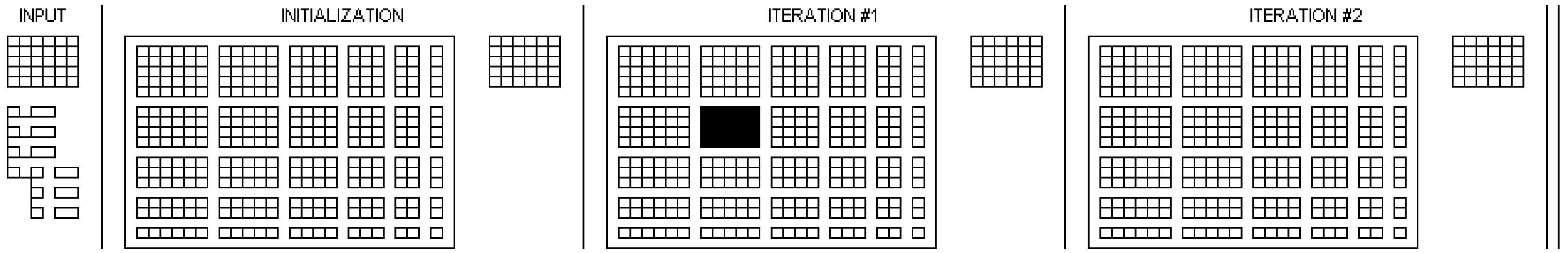
$(x, 1{\times}1{+}(1,1))$	$(x, 1{\times}1{+}(1,2))$	$(x, 1{\times}1{+}(1,3))$	$(x, 1{\times}1{+}(1,4))$	$(x, 1{\times}1{+}(1,5))$	$(x, 1{\times}1{+}(1,6))$
$(x, 1{\times}1{+}(2,1))$	$(o, 1{\times}1{+}(2,2))$	$(o, 1{\times}1{+}(2,3))$	$(o, 1{\times}1{+}(2,4))$	$(o, 1{\times}1{+}(2,5))$	$(x, 1{\times}1{+}(2,6))$
$(x, 1{\times}1{+}(3,1))$	$(o, 1{\times}1{+}(3,2))$	$(o, 1{\times}1{+}(3,3))$	$(o, 1{\times}1{+}(3,4))$	$(o, 1{\times}1{+}(3,5))$	$(x, 1{\times}1{+}(3,6))$
$(x, 1{\times}1{+}(4,1))$	$(o, 1{\times}1{+}(4,2))$	$(o, 1{\times}1{+}(4,3))$	$(o, 1{\times}1{+}(4,4))$	$(o, 1{\times}1{+}(4,5))$	$(x, 1{\times}1{+}(4,6))$
$(x, 1{\times}1{+}(5,1))$	$(x, 1{\times}1{+}(5,2))$	$(x, 1{\times}1{+}(5,3))$	$(x, 1{\times}1{+}(5,4))$	$(x, 1{\times}1{+}(5,5))$	$(x, 1{\times}1{+}(5,6))$

A full example: iteration #1



All tableau cells corresponding to $1 \times n$ and $n \times 1$ cells are never used by the algorithm, and always remain empty.

A full example: iteration #1, step #1



Add elements to cells $T[2 \times 2 + (i, j)]$ in the tableau: informally, for each of the 2×2 tiles t in the multipicture which appear in the right-hand size of a rule, add a $(R_i, \omega_i - t, 2 \times 2 + (i, j))$ entry.

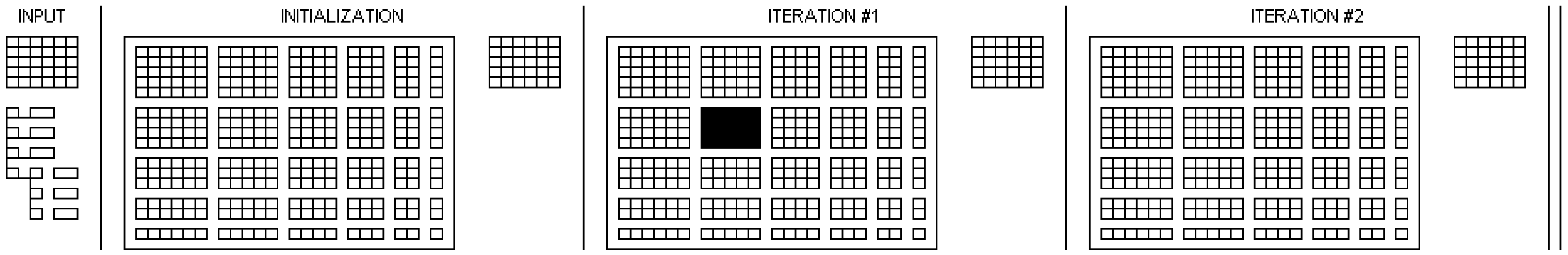
For the first cell, $T[2 \times 2 + (1, 1)]$:

$$M = \begin{array}{|c|c|c} \hline (\mathbf{x}, \mathbf{1}{\times}\mathbf{1}{+}(\mathbf{1}, \mathbf{1})) & (\mathbf{x}, \mathbf{1}{\times}\mathbf{1}{+}(\mathbf{1}, \mathbf{2})) & \dots \\ \hline (\mathbf{x}, \mathbf{1}{\times}\mathbf{1}{+}(\mathbf{2}, \mathbf{1})) & (\mathbf{o}, \mathbf{1}{\times}\mathbf{1}{+}(\mathbf{2}, \mathbf{2})) & \dots \\ \hline \dots & \dots & \dots \end{array}$$

$$\Downarrow$$

$$T_{2,2} = \begin{array}{|c|c} \hline (\mathbf{R_2}, \{\mathbf{t_{2,2}}, \mathbf{t_{2,3}}, \mathbf{t_{2,4}}, \mathbf{t_{2,5}}, \mathbf{t_{2,6}}\}, \mathbf{2}{\times}\mathbf{2}{+}(\mathbf{1}, \mathbf{1})) & \dots \\ \hline \dots & \dots \end{array}$$

A full example: iteration #1, step #1 (continued)



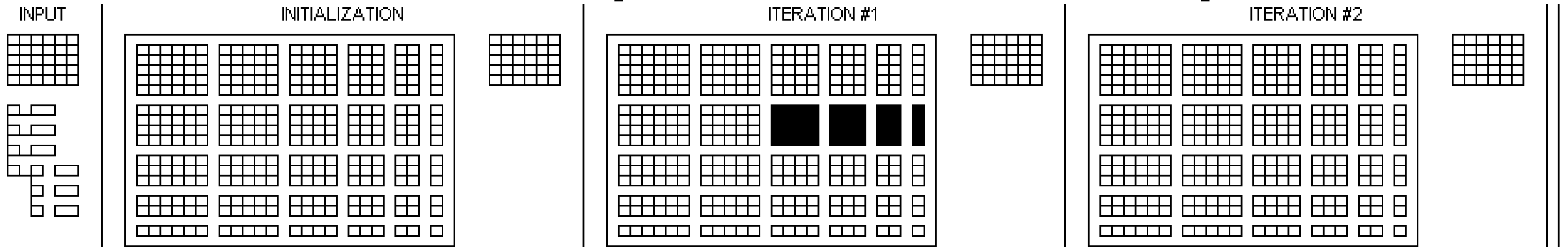
Same for the second cell, $T[2 \times 2 + (1,2)]$. Please note that the corresponding tile appears in the right-hand side of two rules, these two elements are added to the tableau:

$$M = \begin{array}{|c|c|c|c}
\hline
(x, 1{\times}1{+}(1,1)) & (\mathbf{x}, \mathbf{1{\times}1{+}(1,2)}) & (\mathbf{x}, \mathbf{1{\times}1{+}(1,3)}) & \dots \\
\hline
(x, 1{\times}1{+}(2,1)) & (\mathbf{o}, \mathbf{1{\times}1{+}(2,2)}) & (\mathbf{o}, \mathbf{1{\times}1{+}(2,3)}) & \dots \\
\hline
\dots & \dots & \dots &
\end{array}$$

$$\Downarrow$$

$$T_{2,2} = \begin{array}{|c|c|c}
\hline
(R_2, \{t_{2,2}, t_{2,3}, t_{2,4}, t_{2,5}, t_{2,6}\}, 2{\times}2{+}(1,1)) & (\mathbf{R_2}, \{\mathbf{t_{2,1}}, \mathbf{t_{2,3}}, \mathbf{t_{2,4}}, \mathbf{t_{2,5}}, \mathbf{t_{2,6}}\}, \mathbf{2{\times}2{+}(1,2)}) & \dots \\
 & (\mathbf{R_3}, \{\mathbf{t_{3,1}}, \mathbf{t_{3,3}}, \mathbf{t_{3,4}}, \mathbf{t_{3,5}}, \mathbf{t_{3,6}}\}, \mathbf{2{\times}2{+}(1,2)}) & \dots \\
\hline
\dots & \dots &
\end{array}$$

A full example: iteration #1, step #2



For all the matrices in the tableau from $T_{2,3}$ to $T_{2,6}$, each cell can be filled by "merging" the contents of a couple of cells in the tableau matrix immediately on the left:

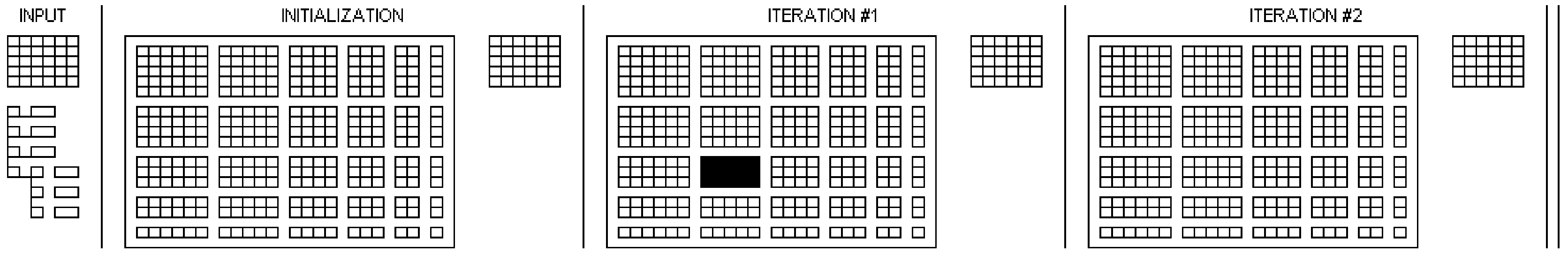
$$\left.\begin{array}{l} (R_e, \omega_1, \alpha_1) \in T[r{\times}(c{-}1){+}(i,j)] \,\wedge \\ (R_e, \omega_2, \alpha_2) \in T[r{\times}(c{-}1){+}(i,j+1)] \end{array}\right\} \Rightarrow (R_e, \omega_1 \cap \omega_2, Sr(\alpha_1 \cup \alpha_2)) \in T[r \times c + (i,j)]$$

$$T_{2,2} \quad = \quad \begin{array}{|c|c|c} \hline (\mathbf{R_2}, \{\mathbf{t_{2,2}}, \mathbf{t_{2,3}}, \mathbf{t_{2,4}}, \mathbf{t_{2,5}}, \mathbf{t_{2,6}}\}, \mathbf{2{\times}2{+}(1,1)}) & \begin{array}{c} (\mathbf{R_2}, \{\mathbf{t_{2,1}}, \mathbf{t_{2,3}}, \mathbf{t_{2,4}}, \mathbf{t_{2,5}}, \mathbf{t_{2,6}}\}, \mathbf{2{\times}2{+}(1,2)}) \\ (R_3, \{t_{3,1}, t_{3,3}, t_{3,4}, t_{3,5}, t_{3,6}\}, 2{\times}2{+}(1,2)) \end{array} & \begin{array}{c} \dots \\ \dots \end{array} \\ \hline \dots & \dots & \end{array}$$

$$\Downarrow$$

$$T_{2,3} \quad = \quad \begin{array}{|c|c|} \hline (\mathbf{R_2}, \{\mathbf{t_{2,3}}, \mathbf{t_{2,4}}, \mathbf{t_{2,5}}, \mathbf{t_{2,6}}\}, \mathbf{2{\times}3{+}(1,1)}) & \dots \\ \hline \dots & \dots \end{array}$$

A full example: iteration #1, step #2 (continued)



Also cells in matrix $T_{3,2}$ can be derived from cells in $T_{2,2}$:

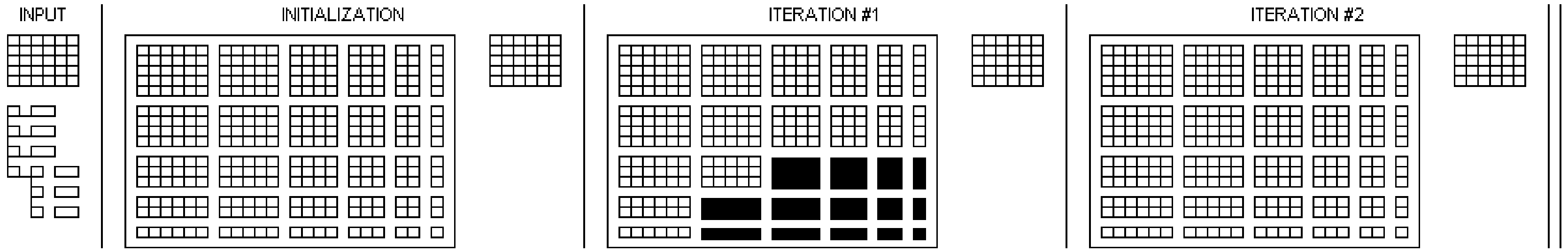
$$\left.\begin{array}{l}(R_e,\omega_1,\alpha_1)\in T[(r-1)\times c+(i,j)] \quad \wedge\\ (R_e,\omega_2,\alpha_2)\in T[(r-1)\times c+(i+1,j)]\end{array}\right\} \Rightarrow (R_e,\omega_1\cap\omega_2,Sr(\alpha_1\cup\alpha_2))\in T[r\times c+(i,j)]$$

$$T_{2,2} \quad = \quad \begin{array}{|c|c|c}
\hline
(\mathbf{R_2},\{\mathbf{t_{2,2}},\mathbf{t_{2,3}},\mathbf{t_{2,4}},\mathbf{t_{2,5}},\mathbf{t_{2,6}}\},\mathbf{2}\times\mathbf{2}+(\mathbf{1},\mathbf{1})) & (R_2,\{t_{2,1},t_{2,3},t_{2,4},t_{2,5},t_{2,6}\},2\times2+(1,2)) & \ldots \\
 & (R_3,\{t_{3,1},t_{3,3},t_{3,4},t_{3,5},t_{3,6}\},2\times2+(1,2)) & \ldots \\
\hline
(\mathbf{R_2},\{\mathbf{t_{2,1}},\mathbf{t_{2,2}},\mathbf{t_{2,4}},\mathbf{t_{2,5}},\mathbf{t_{2,6}}\},\mathbf{2}\times\mathbf{2}+(\mathbf{2},\mathbf{1})) & \ldots & \\
\hline
\ldots & \ldots &
\end{array}$$

$$\Downarrow$$

$$T_{2,3} \quad = \quad \begin{array}{|c|c|}
\hline
(\mathbf{R_2},\{\mathbf{t_{2,2}},\mathbf{t_{2,4}},\mathbf{t_{2,5}},\mathbf{t_{2,6}}\},\mathbf{3}\times\mathbf{2}+(\mathbf{1},\mathbf{1})) & \ldots \\
\hline
\ldots & \ldots
\end{array}$$

A full example: iteration #1, step #2 (continued)

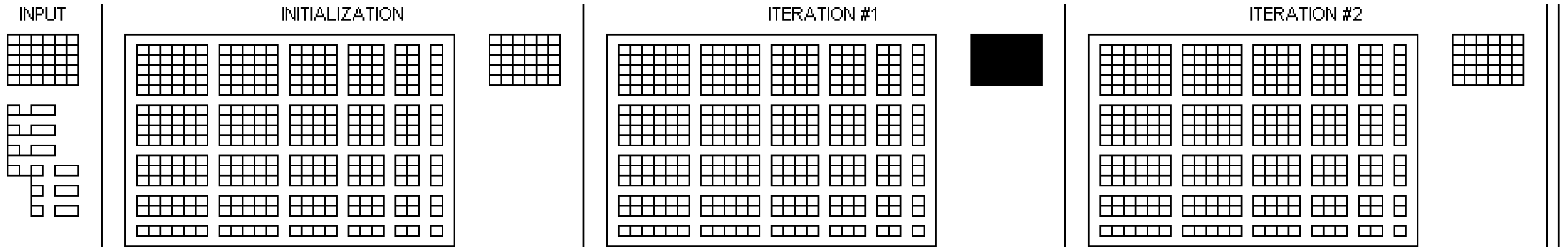


The contents of the remaining cells in the tableau can be derived by applying the same horizontal and vertical merging rules shown in the last two slides:

$$\left.\begin{array}{l}(R_e,\omega_1,\alpha_1)\in T[r\times(c-1)+(i,j)]\ \wedge\\(R_e,\omega_2,\alpha_2)\in T[r\times(c-1)+(i,j+1)]\end{array}\right\}\quad\Rightarrow\quad(R_e,\omega_1\cap\omega_2,Sr(\alpha_1\cup\alpha_2))\in T[r\times c+(i,j)]$$

$$\left.\begin{array}{l}(R_e,\omega_1,\alpha_1)\in T[(r-1)\times c+(i,j)]\ \wedge\\(R_e,\omega_2,\alpha_2)\in T[(r-1)\times c+(i+1,j)]\end{array}\right\}\quad\Rightarrow\quad(R_e,\omega_1\cap\omega_2,Sr(\alpha_1\cup\alpha_2))\in T[r\times c+(i,j)]$$

A full example: iteration #1, step #3



$(\mathbf{R2}, \emptyset, \mathbf{5{\times}3{+}(1,1)})$	$(R2, \{t_{2,1}, t_{2,3}, t_{2,4}, \} 5{\times}3{+}(1,2))$ $(R3, \{t_{3,1}, t_{3,3}, t_{3,4}\}, 5{\times}3{+}(1,2))$	$(R2, \{t_{2,1}, t_{2,3}, t_{2,4}\}, 5{\times}3{+}(1,3))$ $(R3, \{t_{3,1}, t_{3,3}, t_{3,4}\}, 5{\times}3{+}(1,3))$	$\mathbf{R3}, \emptyset, \mathbf{5{\times}3{+}(1,4)})$

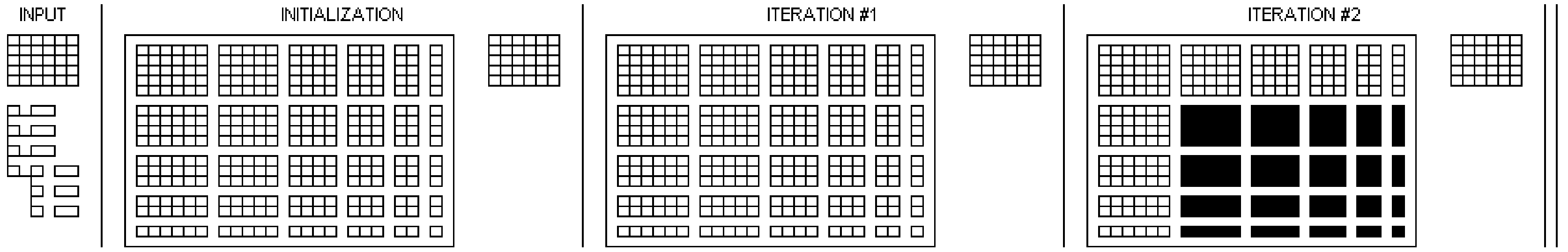
Let's consider the final state of tableau cell $T_{5,3}$: rules $R2$ and R_3 were recognized:

- an entry $(A, 5{\times}3{+}(1,1))$ is added to all the pixels in the $5{\times}3{+}(1,1)$ subpicture of the multipicture M;
- an entry $(B, 5{\times}3{+}(1,4))$ is added to all the pixels in the $5{\times}3{+}(1,4)$ subpicture of the multipicture M.

The final state of the multipicture at the end of iteration 1 is shown in the next slide.

$$M = \begin{array}{|c|c|c|c|c|c|}
\hline
\begin{matrix}(x,1\times1+(1,1))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\end{matrix} &
\begin{matrix}(x,1\times1+(1,2))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(1,3))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times4+(1,3))\\(B,5\times5+(1,1))\end{matrix} &
\begin{matrix}(x,1\times1+(1,4))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(1,5))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(1,6))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} \\
\hline
\begin{matrix}(x,1\times1+(2,1))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\end{matrix} &
\begin{matrix}(o,1\times1+(2,2))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(o,1\times1+(2,3))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times4+(1,3))\\(B,5\times5+(1,1))\end{matrix} &
\begin{matrix}(o,1\times1+(2,5))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(o,1\times1+(2,5))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(2,6))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} \\
\hline
\begin{matrix}(x,1\times1+(3,1))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\end{matrix} &
\begin{matrix}(o,1\times1+(3,2))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(o,1\times1+(3,3))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times4+(1,3))\\(B,5\times5+(1,1))\end{matrix} &
\begin{matrix}(o,1\times1+(3,4))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(o,1\times1+(3,5))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(3,6))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} \\
\hline
\begin{matrix}(x,1\times1+(4,1))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\end{matrix} &
\begin{matrix}(o,1\times1+(4,2))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(o,1\times1+(4,3))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times4+(1,3))\\(B,5\times5+(1,1))\end{matrix} &
\begin{matrix}(o,1\times1+(4,4))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(o,1\times1+(4,5))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(4,6))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} \\
\hline
\begin{matrix}(x,1\times1+(5,1))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\end{matrix} &
\begin{matrix}(x,1\times1+(5,2))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(5,3))\\(A,5\times3+(1,1))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times4+(1,3))\\(B,5\times5+(1,1))\end{matrix} &
\begin{matrix}(x,1\times1+(5,4))\\(A,5\times4+(1,1))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(5,5))\\(A,5\times5+(1,1))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} &
\begin{matrix}(x,1\times1+(5,6))\\(B,5\times3+(1,4))\\(B,5\times4+(1,3))\\(B,5\times5+(1,2))\end{matrix} \\
\hline
\end{array}$$

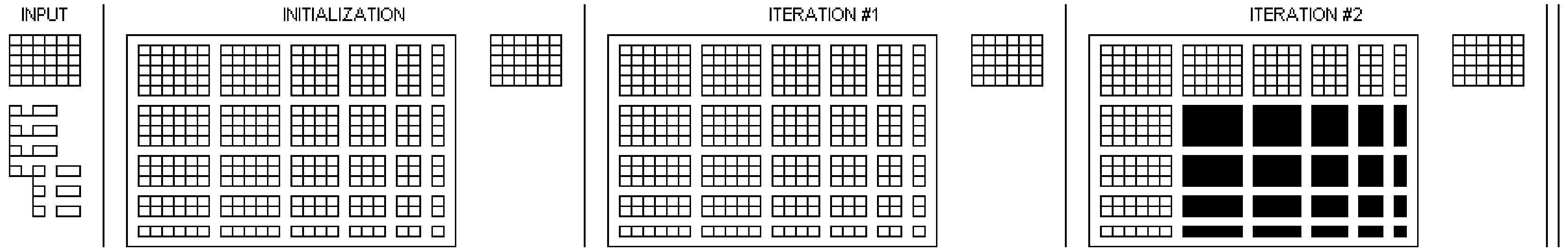
A full example: iteration #2, step #1



Step #1 is applied on the new multipicture. Entire subrectangles of A's and B's are recognized by rule R_1.

$$M = \begin{array}{c|c|c|c} \cdots. & \cdots & \cdots & \cdots \\ \hline \cdots & \begin{array}{c}\cdots\\(A,5{\times}3+(1,1))\\(A,5{\times}4+(1,1))\\(A,5{\times}5+(1,1))\\\cdots\end{array} & \begin{array}{c}\cdots\\(A,5{\times}3+(1,1))\\(A,5{\times}4+(1,1))\\(A,5{\times}5+(1,1))\\\cdots\end{array} & \cdots \\ \hline \cdots & \begin{array}{c}\cdots\\(A,5{\times}3+(1,1))\\(A,5{\times}4+(1,1))\\(A,5{\times}5+(1,1))\\\cdots\end{array} & \begin{array}{c}\cdots\\(A,5{\times}3+(1,1))\\(A,5{\times}4+(1,1))\\(A,5{\times}5+(1,1))\\\cdots\end{array} & \cdots \\ \hline \cdots & \cdots & \cdots & \cdots \end{array} \Rightarrow T_{2,2} = \begin{array}{c|c|c} \cdots. & \cdots & \cdots \\ \hline \cdots & \begin{array}{c}\cdots\\(R_1,\{t_{1,2},t_{1,3}\},5{\times}3+(1,1))\\(R_1,\{t_{1,2},t_{1,3}\},5{\times}4+(1,1))\\(R_1,\{t_{1,2},t_{1,3}\},5{\times}5+(1,1))\\\cdots\end{array} & \begin{array}{c}\cdots\\\cdots\end{array} \\ \hline \cdots & \cdots & \cdots \end{array}$$

A full example: iteration #2, step #2

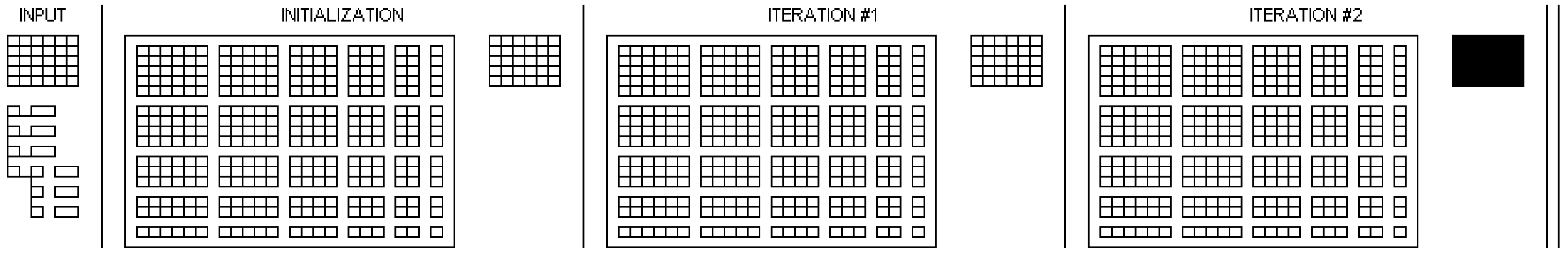


At the end of step 2, cell $T_{5,6}$ contains the following value:

$$T_{2,2} = \boxed{\ (R_1, \emptyset, 5{\times}6{+}(1,1))\ }\ .$$

Thus, an application area for rule R_1 was recognized over the entire multipicture.

A full example: iteration #2, step #3



- An element $(S, 5{\times}6+(1,1))$ is added to every cell in the multipicture belonging to the rectangle $5{\times}6+(1,1)$, which is the whole picture.
- The picture is therefore recognized.
- The final state of the multipicture is shown in the next slide.

$M =$

$(x, 1\times1+(1,1))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(1,2))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(1,3))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(1,4))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(1,5))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(1,6))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$
$(x, 1\times1+(2,1))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(2,2))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(2,3))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(2,5))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(2,5))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(2,6))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$
$(x, 1\times1+(3,1))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(3,2))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(3,3))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(3,4))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(3,5))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(3,6))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$
$(x, 1\times1+(4,1))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(4,2))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(4,3))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,1))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(4,4))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(o, 1\times1+(4,5))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$	$(x, 1\times1+(4,6))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$ $(B, 5\times5+(1,2))$ $(S, 5\times6+(1,1))$
$(x, 1\times1+(5,1))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$	$(x, 1\times1+(5,2))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$	$(x, 1\times1+(5,3))$ $(A, 5\times3+(1,1))$ $(A, 5\times4+(1,1))$	$(x, 1\times1+(5,4))$ $(A, 5\times4+(1,1))$ $(A, 5\times5+(1,1))$	$(x, 1\times1+(5,5))$ $(A, 5\times5+(1,1))$ $(B, 5\times3+(1,4))$	$(x, 1\times1+(5,6))$ $(B, 5\times3+(1,4))$ $(B, 5\times4+(1,3))$

The end

Thank you for your attention.

Questions are welcome.