Syntactical Pattern Recognition with 2-Dimensional Array Grammars and Array Automata

Rudi FREUND
Faculty of Informatics
Vienna University of Technology
Wien, Austria
Overview

Offline Character Recognition

- Preprocessing
  - normalization
  - elimination of noisy pixels
  - thinning
- Syntactic Analysis

Variants of Array Grammars/Automata for the Syntactic Analysis

Related Theoretical Results

Summary
Character Recognition – Preprocessing

Normalization

The scanned characters first are normalized to fill out a 320×400 grid in order to get comparable patterns. Then noisy pixels are eliminated. After noise elimination, the resulting arrays on the 320×400 grid are mapped on a 20×25 grid.
Character Recognition – Preprocessing

Elimination of Noisy Pixels

Algorithms are based on parallel array grammars

eliminate pixel if number of pixels in the 8-neighbourhood = 0,
iterate this algorithm until no more changes
Character Recognition – Preprocessing
Thinning

Algorithms are based on parallel array grammars. Reduction to a unitary skeleton (e.g., J. H. Sossa, An improved parallel algorithm for thinning digital patterns, Pattern Recognition Letters 10 (1989), pp. 77-80).
Character Recognition – Preprocessing Result

Unitary skeleton on a 20 x 25 grid
Variants of Array Grammars/Automata for the Syntactic Analysis (SSPR’96)

Bounded Parallelism
Bounded Parallelism
Bounded Parallelism
Bounded Parallelism
Bounded Parallelism
Bounded Parallelism
Bounded Parallelism/Prescribed Teams

array productions in team are applied in parallel
Bounded Parallelism/Prescribed Teams
derivation modes

array productions in team are applied in parallel; the teams themselves may be applied in different derivation modes (variants of co-operation as in co-operating distributed grammar systems):

=\(k\), >\(k\), <\(k\), *, t (maximal derivation mode)

internally hybrid modes:
(\(t,=k\)), (\(t,>k\)), (\(t,<k\)), (\(>m,<k\))
Bounded Parallelism/Prescribed Teams
finite index restriction/pattern analysis

Finite index restriction:
by applying the array productions of a team in parallel, all non-terminal symbols in the current sentential form must be affected

Analysis of a given pattern:
whenever a terminal symbol is generated, it must coincide with the symbol in the pattern (character) to be analysed
Syntactic Pattern Analysis – Distance from Ideal Cluster

During the analysis of a given pattern, its distance from the ideal cluster of arrays representing a specific character is computed.

Features to be taken into account:

- deviation of lines
- gaps in lines
- superfluous/missing lines
- superfluous (remaining) pixels
Variants of Array Grammars/Automata for the Syntactic Analysis (SSPR’98, k-head finite automata)

k-head finite automata

The k heads in a k-head finite array automaton are the counterparts of the k non-terminal symbols in array grammars with prescribed teams and finite index. In each step, every head has to move. The automaton has a „head sensing ability“, i.e., two heads can never occupy the same position. Moreover, a position carrying a terminal symbol in the array to be parsed can only be visited once by one of the k heads (and then is marked as „forbidden position“ for the rest of the parsing procedure).
Variants of Array Grammars/Automata for the Syntactic Analysis (regulated array grammars of finite index)

Variants of Array Grammars/Automata for the Syntactic Analysis (SSPR 2000)

Hybrid Systems

Pre-selection by neural network
(for a given pattern, only a few pre-selected array grammars have to analyse it)

Application of teams controlled by additional mechanism (programmed, matrix)

Look-ahead instead of inefficient backtracking
(larger neighbourhoods)
Related Theoretical Results

Regulated Array Grammars of Finite Index

Restricted by the finite index condition, with the control mechanisms of using variants of context-free array productions in prescribed teams, matrices or a control graph, the corresponding families of generated array languages coincide, even in the appearance checking case. The corresponding models of k-head automata accept the same families of array languages, too. With respect to k, we obtain infinite hierarchies for dimensions $n > 1$. 
THANK YOU
FOR YOUR ATTENTION
GRAZIE, MERCI, KIITOS, DÍKY, DANKE!