Syntactic Pattern Recognition summary

L. Stogiolaris, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 2.8.1

Artificial Intelligence & Information Analysis Lab

Syntactic Pattern Recognition

- **Introduction**
  - **Syntactic Pattern Recognition System**
  - Preprocessing Techniques
- String-Based Models
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - Attributed grammars
  - Stochastic grammars
- Graph-Based Models
  - Graph matching algorithms
- Applications

Introduction to Pattern Recognition

- Pattern recognition techniques are among the most important tools used in the field of machine intelligence.
- Pattern recognition can be defined as the categorisation of input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail.
- A pattern is essentially an arrangement. It may be defined as a quantitative or structural description of an object or some other entity of interest.
- A pattern class is a set of patterns that share some common properties.

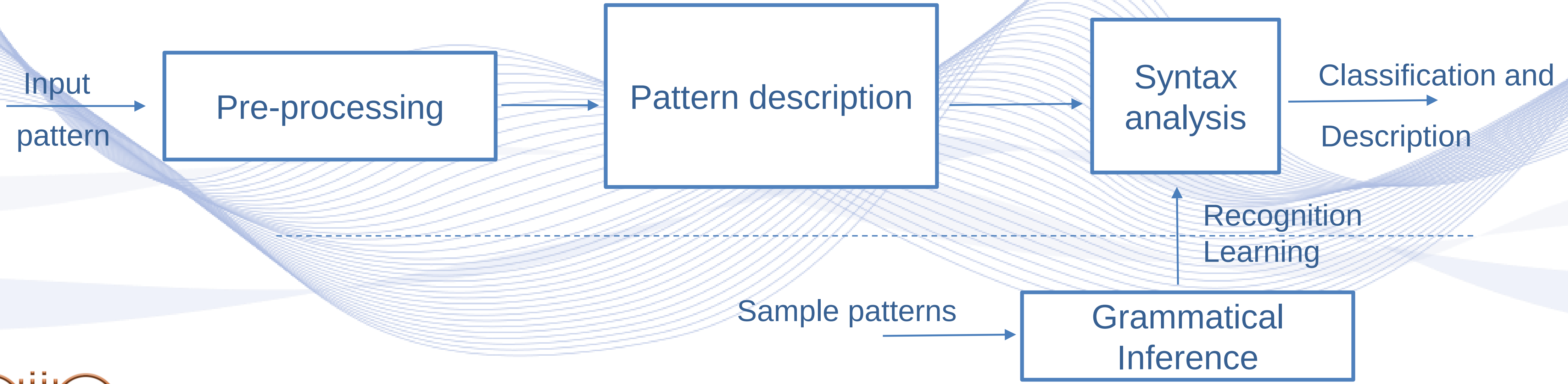
Introduction to Syntactic Pattern Recognition

- Syntactic pattern recognition is based on concepts from formal language theory, the origins of which may be traced to the middle 1950s with the development of mathematical models of grammars by Noam Chomsky.
- Syntactic pattern recognition is based on symbolic data structures like strings, trees, graphs, or arrays for pattern representation.
- A parser is used as recognition engine.
- Approaches to syntactic pattern recognition system design may be divided into two principal categories :
  - the decision theoretic approach,
  - the syntactic approach

Syntactic Pattern Recognition System

A syntactic pattern recognition system can be considered as consisting of three major parts:

- Preprocessing
- Pattern description ,
- Syntax analysis.



Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - **Preprocessing Techniques**
- String-Based Models
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - Attributed grammars
  - Stochastic grammars
- Graph-Based Models
  - Graph matching algorithms
- Applications

Preprocessing Techniques

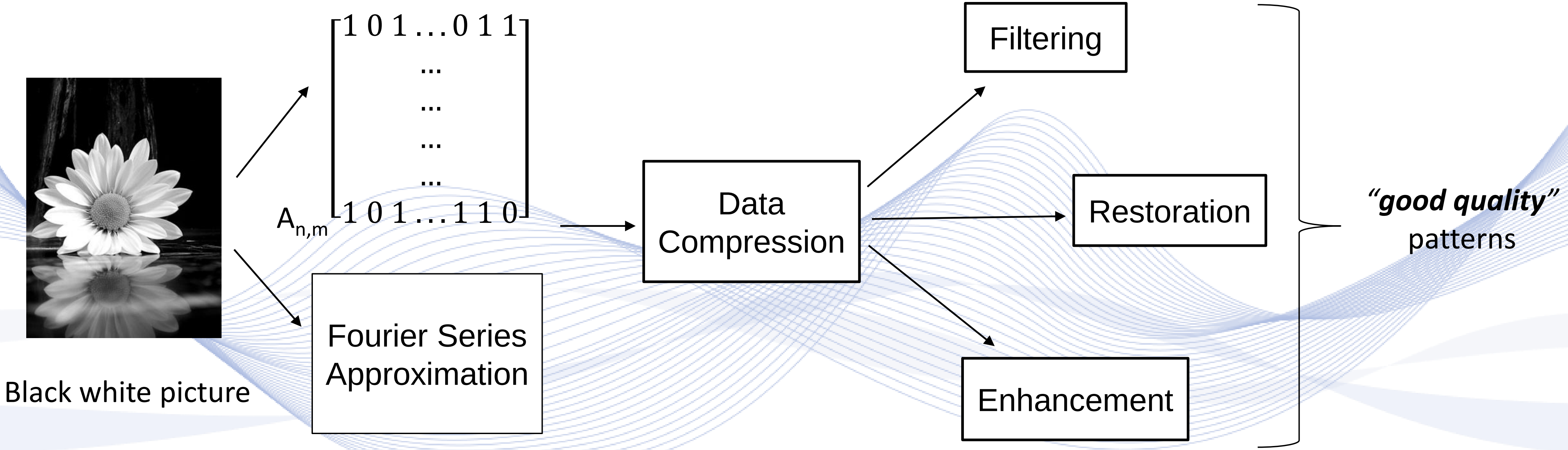
There are several stages of pre-processing:

- pattern encoding ,
- filtering which is used to clean noise ,
- restoration to restore the degradation ,
- enhancements to improve the quality of coded patterns.

- After preprocessing each preprocessed pattern is represented by a language-like structure such as string, graph etc.

Preprocessing Techniques

Example of pre-processing phase before the pattern description phase:



Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - Preprocessing Techniques
- **String-Based Models**
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - Attributed grammars
  - Stochastic grammars
- Graph-Based Models
  - Graph matching algorithms
- Applications

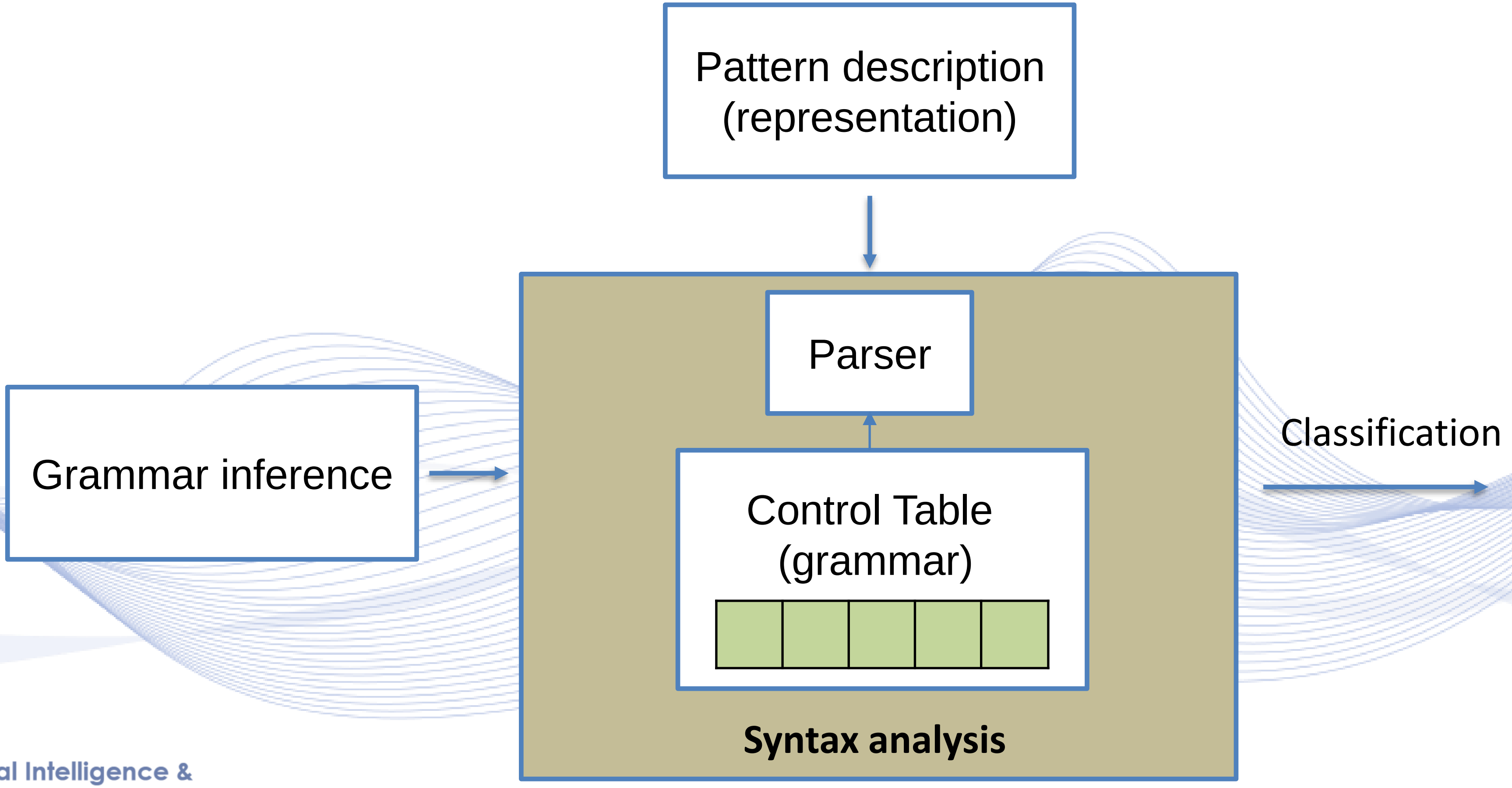
String-Based Models

- The principal requirement in designing a syntactic pattern recognition system is the development of a ***grammar*** capable of generating a given class of patterns.

- A parser is used as a recognition engine
  - ***Top-down parser*** : tree derivation construction from the root to the leaves, applying the grammar productions.
  - ***Bottom-up parser*** : tree derivation construction from the leaves to the root, applying the grammar productions in backward direction.

String-Based Models

- Syntax analysis scheme using String-Based models



Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - Preprocessing Techniques
- String-Based Models
  - **Formal Grammars**
    - **Context-sensitive grammars**
    - **Context-free grammars**
    - **Regular grammars**
  - Attributed grammars
  - Stochastic grammars
- Graph-Based Models
  - Graph matching algorithms
- Applications

Formal Grammars

Some key concepts :

- There are two categories of sets of interest in formal language theory:
  - finite sets, which have a finite number of elements and
  - countably infinite sets.
- An alphabet is a finite set of symbols, such as the binary set { 0,1 }.
- A string *x* over alphabet V is a string of finite length formed with symbols from V.
- The length of *x* , denoted by |*x*|.
- The ***concatenation*** of *x* with *y* is the sentence *xy* with length |*xy*| = |*x*|+|*y*| .

Artificial Intelligence & Information Analysis Lab

Formal Grammars

- The empty string, denoted by λ or ε , is the sentence with no symbols, so $|\lambda|=0$.
- The empty string doesn't match with the null string φ **,** which nulifies when used in concatenation.
- For any alphabet V , the countably infinite set of all sentences over V , including λ, is the **closure** of V , denoted by V^* .
  - The positive closure of V is the set $V^+ = V – \{ \lambda \}$.
- Given alphabet V = { a,b } these sets are :
  - $V^* = \{ \lambda ,a, b, aa, ab, ba, bb, aaa \ldots\}$,
  - $V^+ = \{ \lambda ,a, b, aa, ab, ba, bb, aaa \ldots\}$,
- A language is a set of sentences over an alphabet; i.e., a language over alphabet V is a finite countably infinite subset of V^* .

Artificial Intelligence & Information Analysis Lab

Context-sensitive grammars (type 1)

The productions in the P set of type 1 grammar are restricted to the form :

$$z_1 A z_2 \rightarrow z_1 b z_2$$

where $A \in V_N$, z_1, z_2, $b \in V^*$ and $b \neq \lambda$.

So the A can replaced by b in the context z_1, z_2. It also implies that :

$|z_1 A z_2| \rightarrow |z_1 b z_2|$ or $|A| < |b|$.

The languages generated by context-sensitive grammars are called ***type 1 languages*** i.e., $\{ 0^n 1 0^n \mid n = 1,2,3,4 \ldots \}$.

Regular grammars (type 3)

The productions in the P set of type 3 grammar are restricted to the form :

$$A \rightarrow aB \qquad \text{or} \qquad A \rightarrow b$$

where A,B $\in V_N$ and *a*, *b* $\in V_T$.

A,B, *a*,*b* are all single symbols.

For example , the language $\{0^m,1^n \mid m,n =1,2,\dots ,\}$ is a regular language while the language $\{0^n,1^n \mid n =1,2,\dots ,\}$ not.

The languages generated by regular grammars are called ***regular of finite-state languages.***

String Based-Model Application

Let consider a two-class pattern problem and the patterns of these classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are composed of features from a set of terminals V_T.

So each pattern may be consider as a sentence since it is composed of terminals.

Let $\boldsymbol{G}$ be a grammar such that its language $L(\boldsymbol{G})$ consist of patterns which belongs to $\mathcal{C}_1$.

Thus , any incoming pattern can be classified in $\mathcal{C}_1$ if it belongs to $L(\boldsymbol{G})$, otherwise it will be classified to class $\mathcal{C}_2$.

String Based-Model Application

Example:

Considering a context-free grammar $\boldsymbol{G}$ = { V_N,V_T,P,S }

where V_T={ *a*,*b* }, V_N={ S,A } and the production set is

P = {

S → *aSb* ,

S → *b*

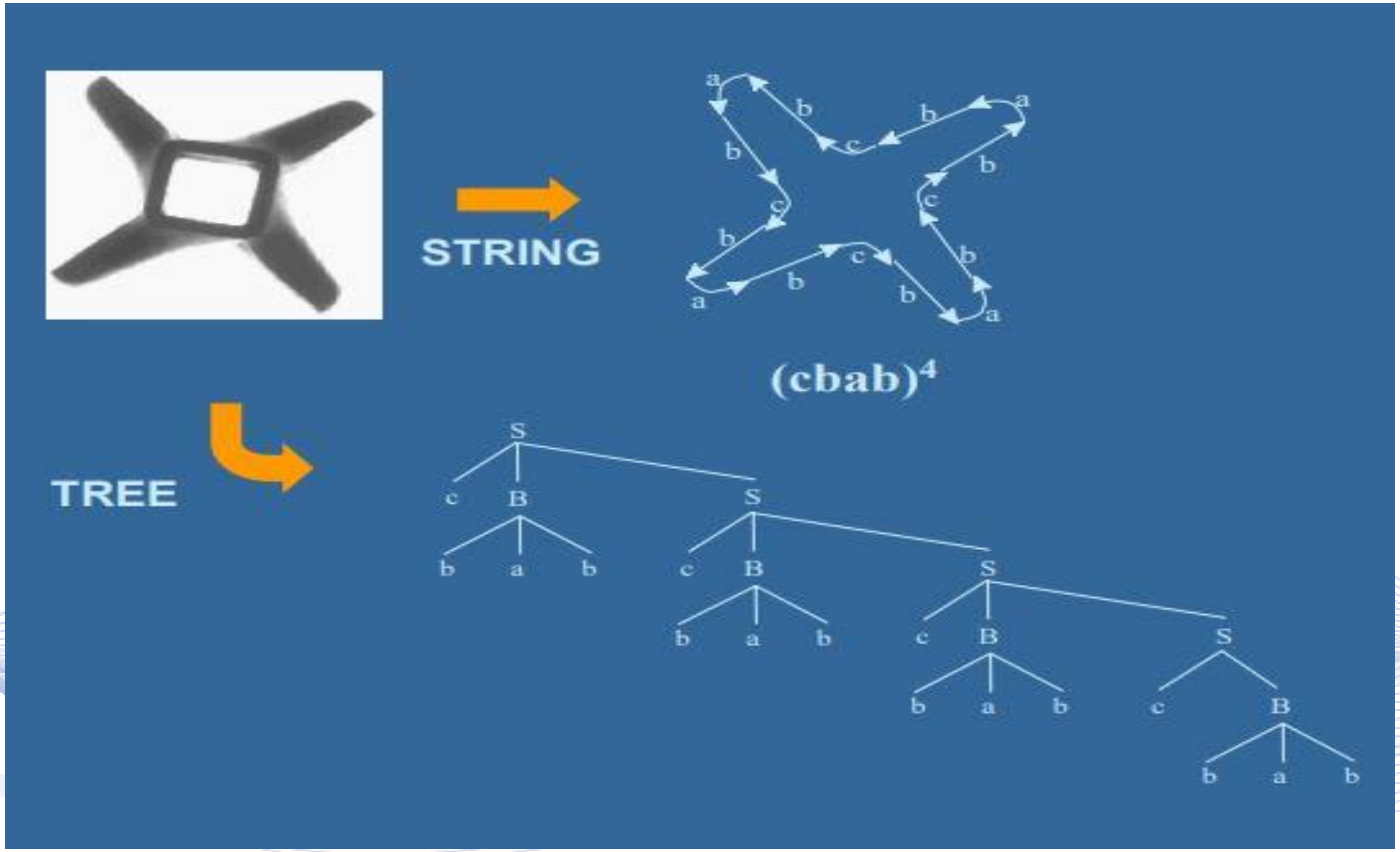
}.

So , the corresponding language $L(\boldsymbol{G})$ consists of the strings

{b; $a^n b^n$+1, n ≥ 1}.

Artificial Intelligence & Information Analysis Lab

String Based-Model Application



Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - Preprocessing Techniques
- String-Based Models
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - **Attributed grammars**
  - Stochastic grammars
- Graph-Based Models
  - Graph matching algorithms
- Applications

Attributed grammars

- An attribute grammar is a context-free grammar augmented with attributes, semantic rules, and conditions.

- For each syntactic category $X \in N$ in the grammar, there are two finite disjoint sets $I(X)$ and $S(X)$ of inherited and synthesized attributes .

  For $X = S$, the start symbol, $I(X) = \emptyset$.

- In contrast to purely syntactic grammars, attribute grammars are

capable of describing features that are not easily represented by finite symbols.

Attributed grammars

An attribute grammar *AG* is a five-tuple

$$AG = (\boldsymbol{G}, SD, AD, R, C)$$

where

$$\boldsymbol{G} = (V_N, V_T, P, S) \text{ is the underlying context-free grammar.}$$

- SD denote a semantic domain consisting of a set of types (e.g., integers or coordinates) and a set of functions operating on the types.
- AD is a set of attributes associated with each symbol occurring in the productions in P. Each attribute is of a certain type.
- R is a set of attribute evaluation rules associated with each production $p \in P$.
- C is a set of semantic conditions associated with each $p \in P$, which are predicates defined on the attribute values.

Attributed grammars

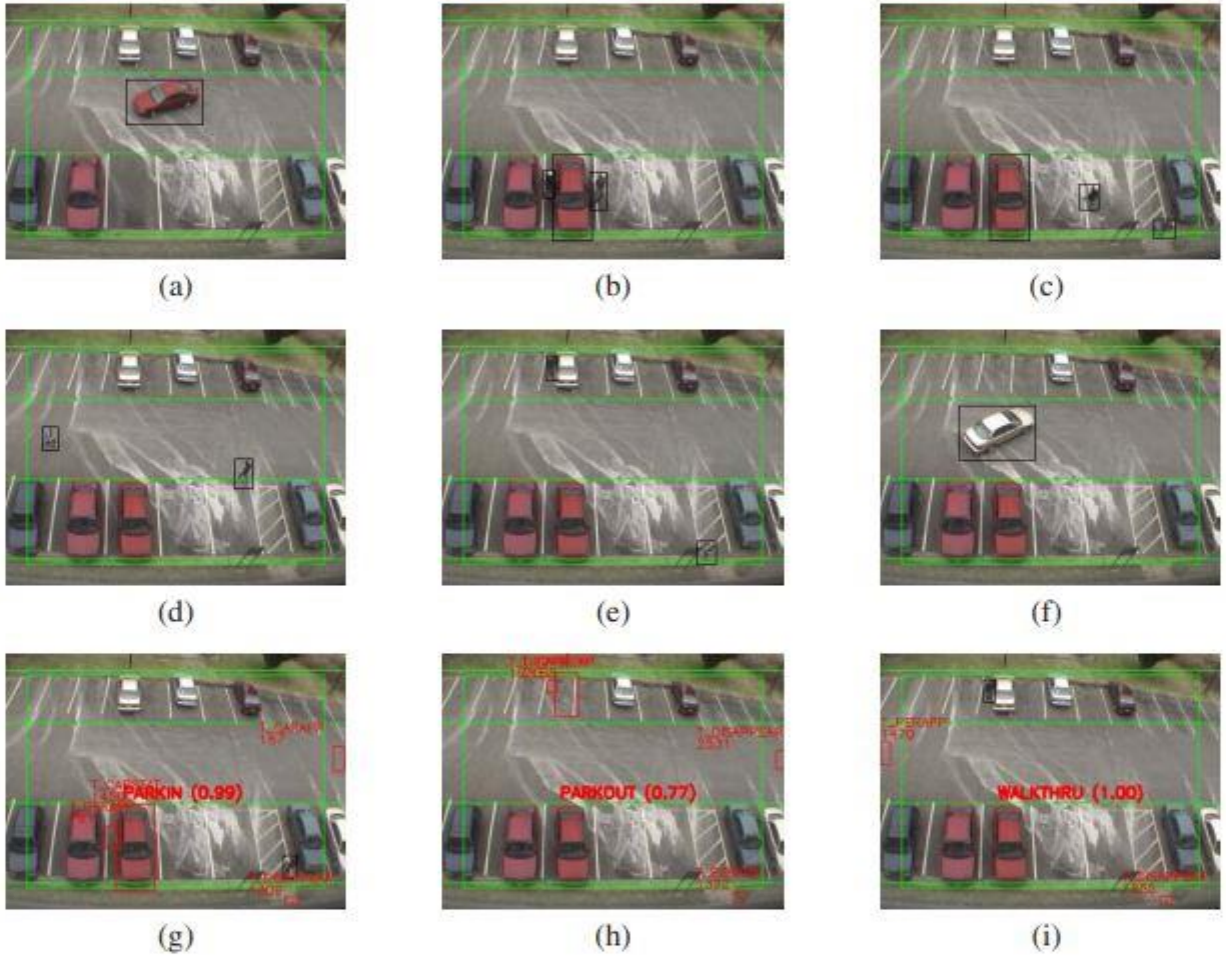


Figure 1. Normal events in the parking lot. (a)–(f): Key frames from the video. (g)–(i): Recognition results.

Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - Preprocessing Techniques
- String-Based Models
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - Attributed grammars
  - **Stochastic grammars**
- Graph-Based Models
  - Graph matching algorithms
- Applications

Stochastic grammars

Due to measurement noise and some ambiguity regarding the characteristics of the pattern classes, it is necessary to consider a stochastic model of grammar and stochastic languages.

Definition:

A stochastic grammar is a set $G_s\ \{V_T, V_N, P, Q, S\}$

where V_T, V_N, P and S

are as explained in preliminaries, and Q is a set of probabilities associated with the given production.

String-based Models

- Strings are a useful class of data structures as they allow the efficient implementation of many operations.
- Symbolic representations allow to describe spatial, temporal, conceptual, … relationships between primitives, and also their hierarchical structure
- On the other hand, they are limited in their representational power because they are intrinsically one-dimensional.
- To overcome this problem, graphs have been used as a more general method.

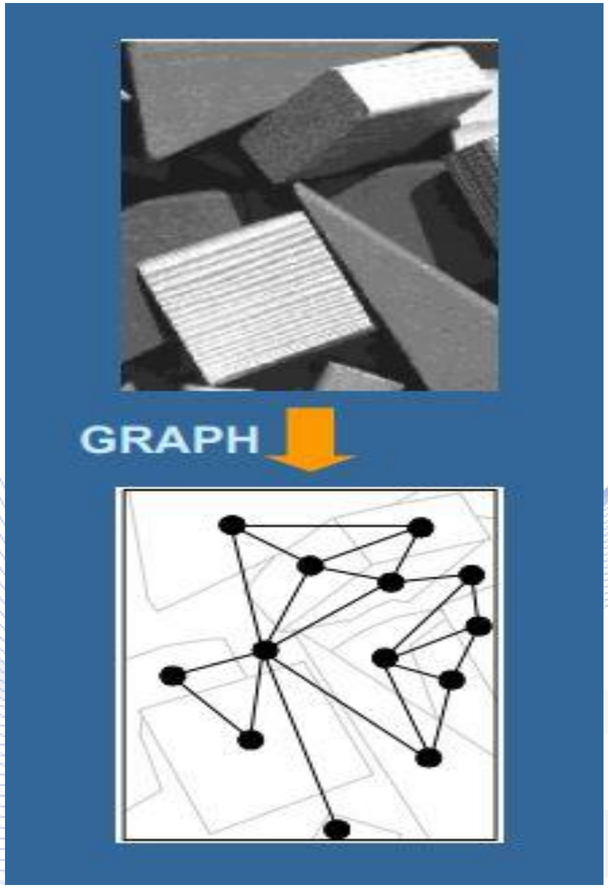
Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - Preprocessing Techniques
- String-Based Models
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - Attributed grammars
  - Stochastic grammars
- **Graph-Based Models**
  - Graph matching algorithms
- Applications

Graph-Based Models

- If a pattern is given in terms of a graph, then the nodes usually represent simpler *subpatterns* and the edges indicate relations between those *subpatterns*.

- The relations may be of spatial, temporal, or any other type

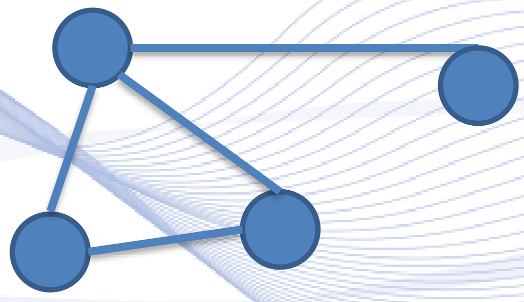


Artificial Intelligence & Information Analysis Lab

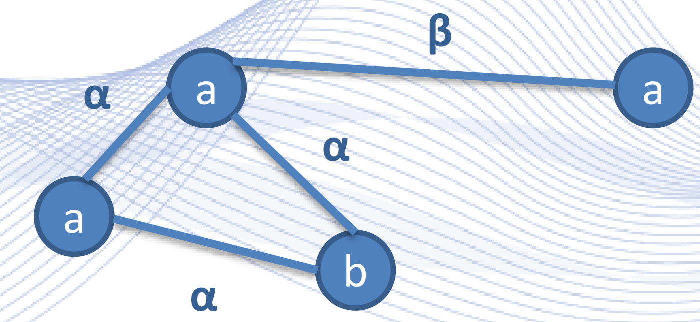
Graph-Based Models

A labeled graph is a graph $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{m}, \mathrm{u})$ where m and u are respectively, two labeling functions for vertices and edges.

Given an unknown pattern represented by a graph G, the problem of recognition is to determine if the graph G belongs to a language $L(\mathbf{G})$ generated by a graph grammar G.



Unlabeled graph



Labeled graph

Graph isomorphism and subgraph isomorphism

- A graph isomorphism between $G = (V, E, L_V, L_E)$ and $G' = (V', E', L_V', L_E')$ is a bijective mapping $f : V \rightarrow V'$ such that the following conditions are satisfied:

$$L_V(v) = L_V'(f(v)) \text{ for all } v \in V.$$

For any edge $e = (v_i, v_j) \in E$ there exists an edge

$$e' = (f(v_i), f(v_j)) \in E' \text{ such that } L_E(e) = L_E'(e').$$

For any edge $e' = (v_i', v_j') \in E'$ there exists an edge

$$e = (f_1(v_i'), f_1(v_j')) \in E \text{ such that } L_E'(e') = L_E(e).$$

An injective mapping $f : V \rightarrow V'$ is a subgraph isomorphism if there exists a subgraph $G'' \subseteq G'$ such that f is an isomorphism between the graphs G and G''.

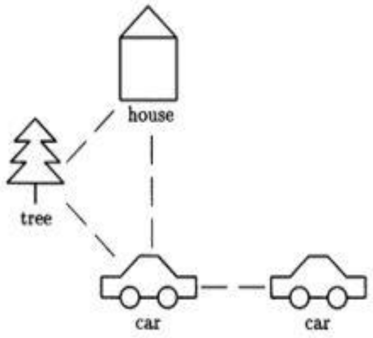


Fig. 2 An example pattern

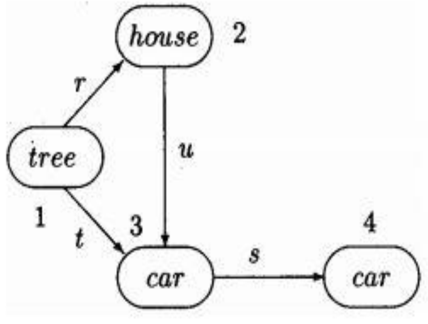


Fig. 3 An *IE* graph representing the example pattern shown in Fig. 2

Interpreted graph

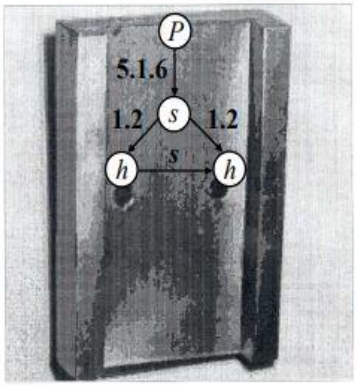
Definition :

Let $H = (V, E, \Sigma, \Gamma, \varphi)$ be an IE graph,

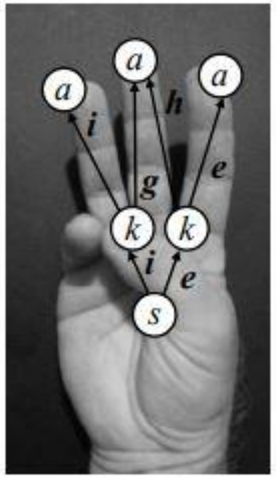
where $V = \{1, \dots, k\}$ is the set of its nodes,

$I(i)$, $i = 1, \dots, k$ is the characteristic description of the form of the node i.

A string $I(1) \dots I(k)$ is called a characteristic description of the graph H.



a)

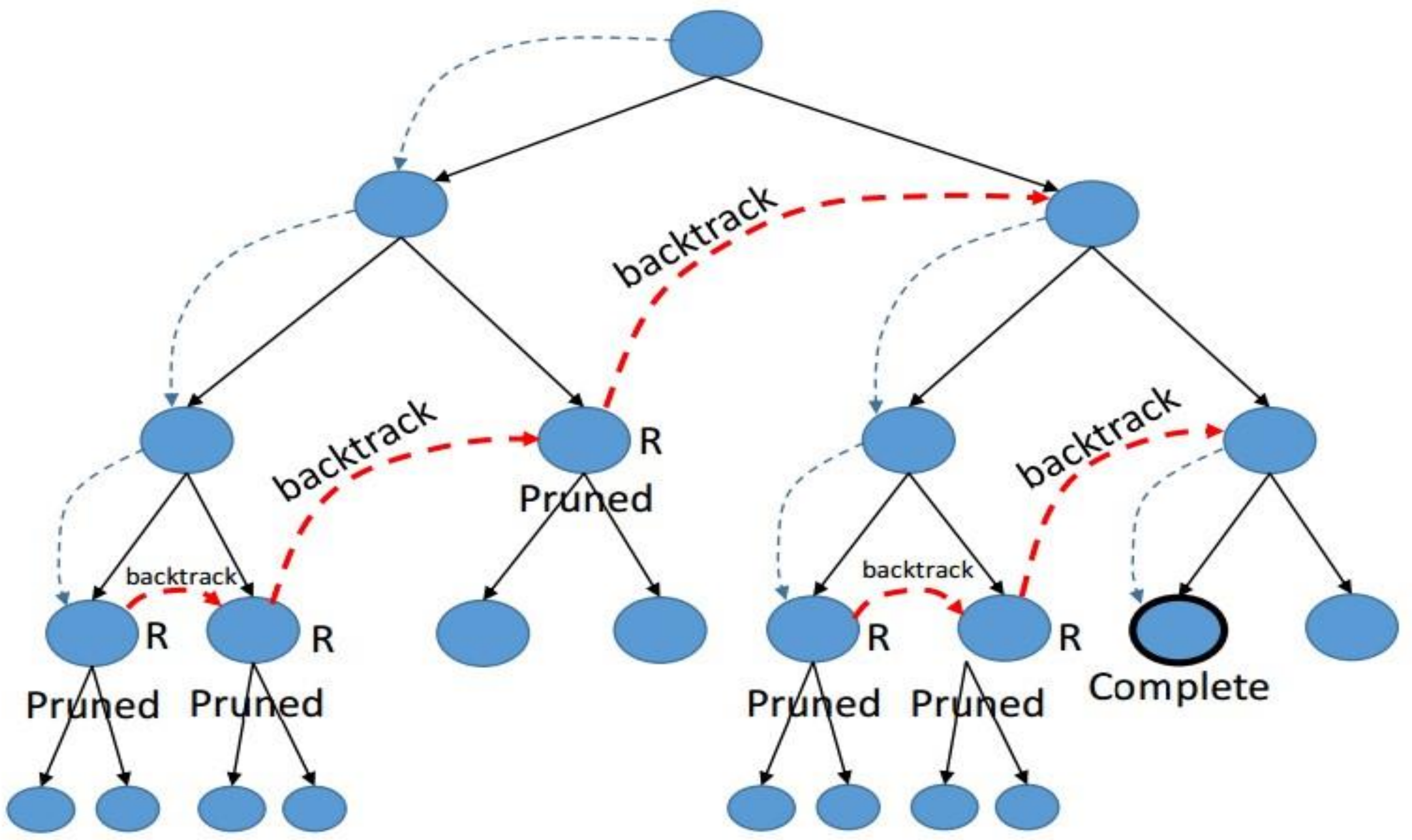


b)

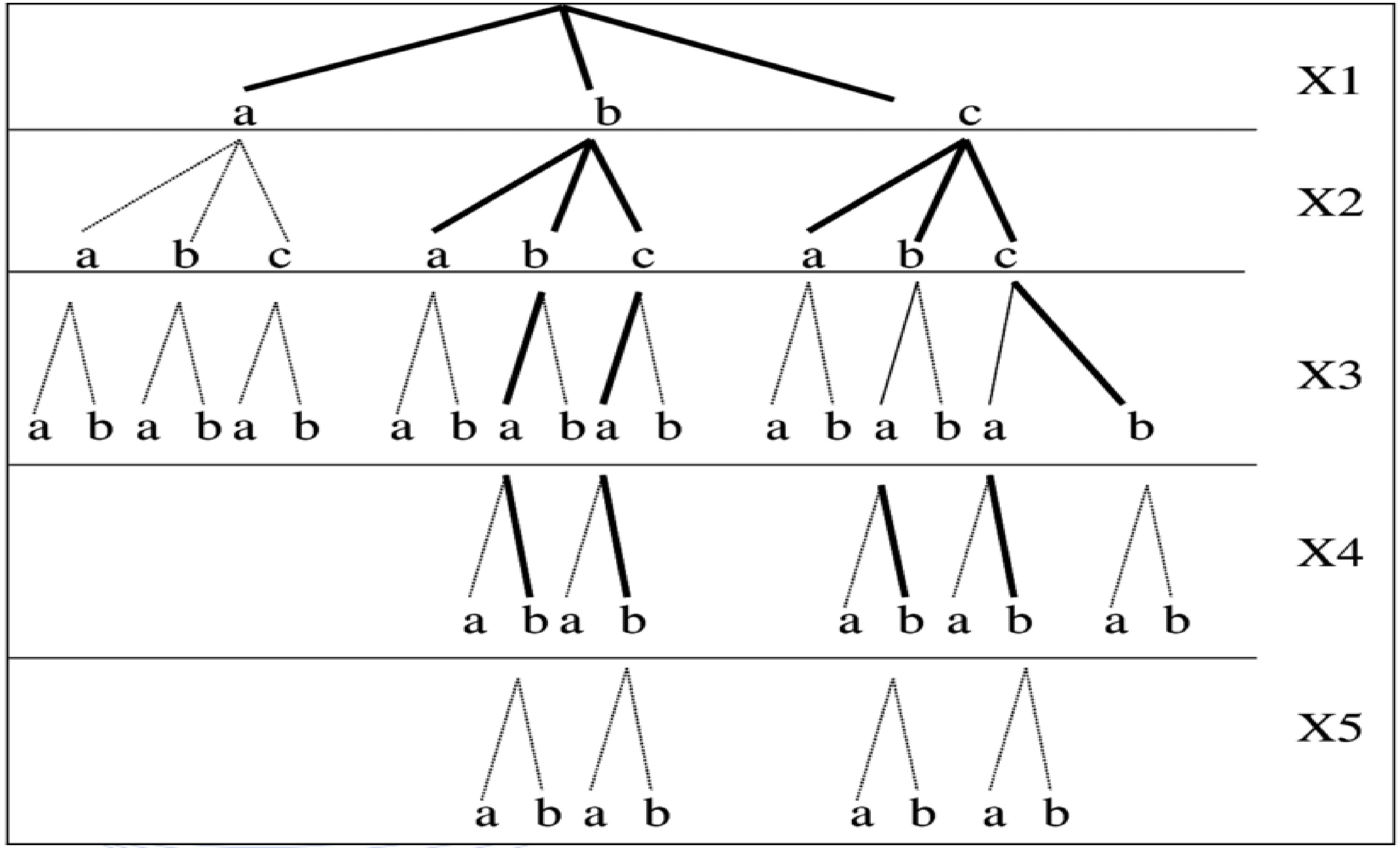
Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - Preprocessing Techniques
- String-Based Models
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - Attributed grammars
  - Stochastic grammars
- Graph-Based Models
  - **Graph matching algorithms**
- Applications

Backtracking Checking



Forward Matching



Syntactic Pattern Recognition

- Introduction
  - Syntactic Pattern Recognition System
  - Preprocessing Techniques
- String-Based Models
  - Formal Grammars
    - Context-sensitive grammars
    - Context-free grammars
    - Regular grammars
  - Attributed grammars
  - Stochastic grammars
- Graph-Based Models
  - Graph matching algorithms
- **Applications**

Applications

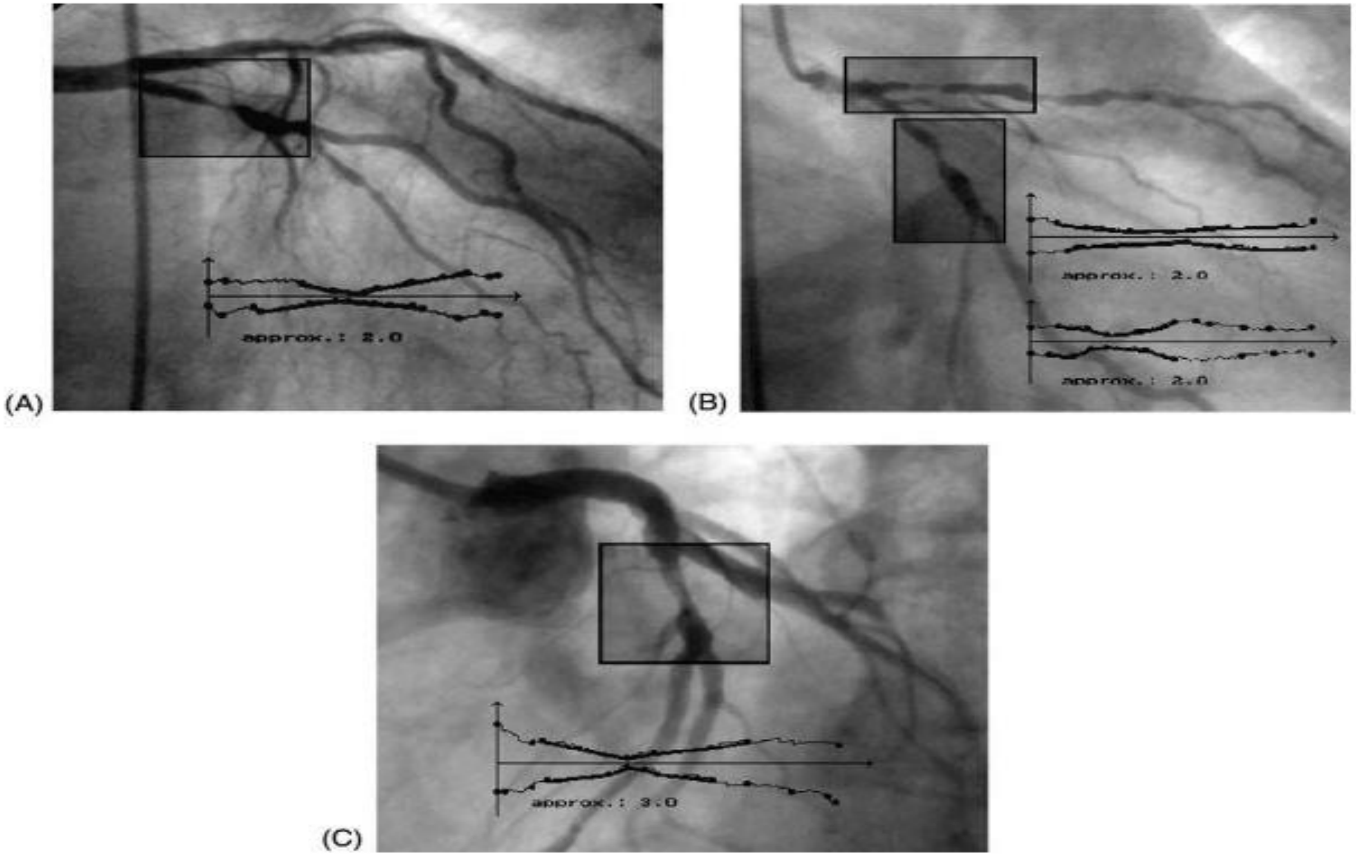


Fig. 3. (A–C) Results of recognition of disease symptoms with the use of syntactic methods of image recognition in the case of coronary arteries.

Applications

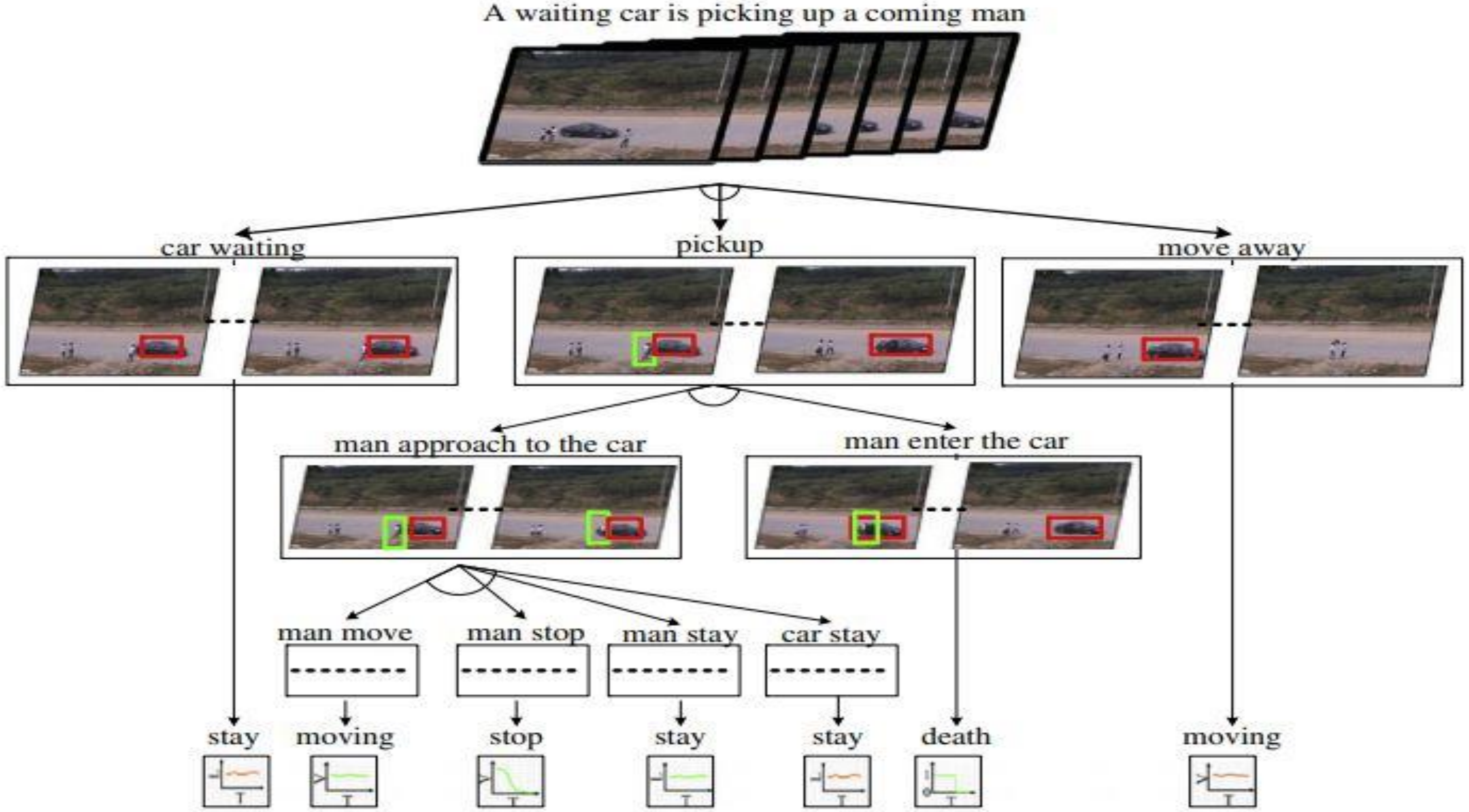


Fig. 1. Event parse graph to represent one specific semantic event. An event is divided into three event components, with temporal constraints (sequential order). And one of these components is further decomposed into two sub-components with sequential order as well. Finally these event components can be explained by event primitives in the lowest level, which can be computed via related tracked objects features (visibility, location, velocity), as shown in the bottom of this figure.

Artificial Intelligence &
Information Analysis Lab

Applications

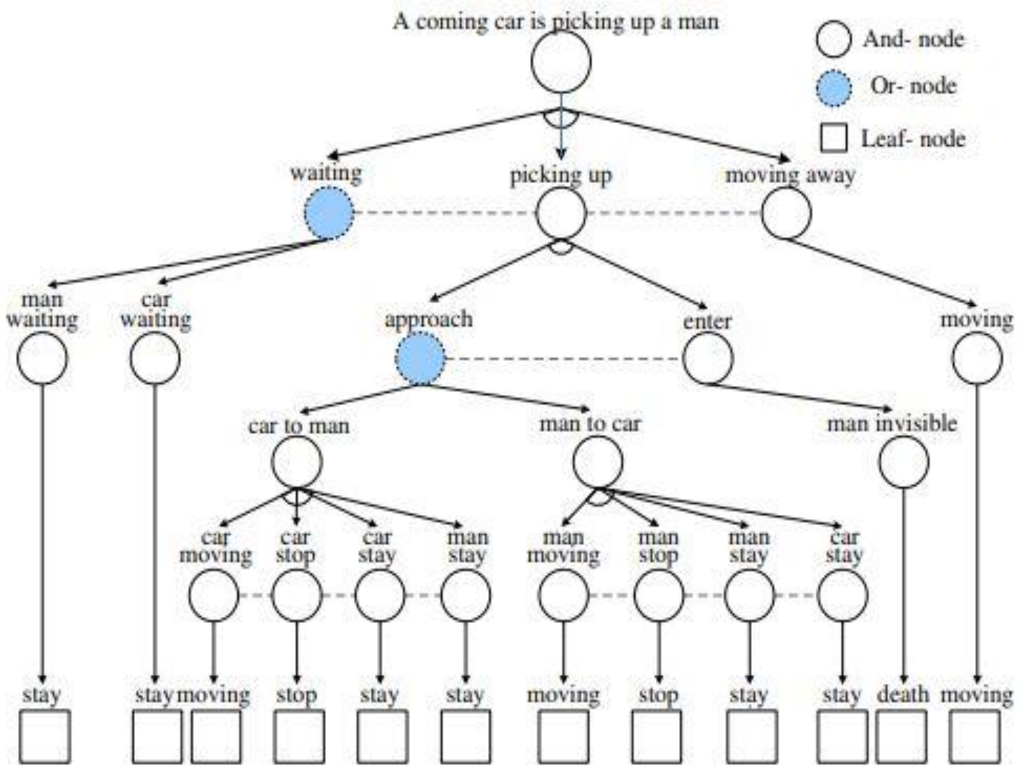


Fig. 2. Event And–Or graph to represent all possible variety of a semantic event. In this figure, an event "a coming car is picking up a man" is modeled with an event And–Or graph, in which the And-nodes and Or-nodes denote meaningful event components. Each And-node is compositional and Or-node is a choice. The And-nodes and Or-nodes (event components) can be decomposed into a set of Leaf-nodes (atomic event primitives) with temporal-spatial constraints.

Artificial Intelligence & Information Analysis Lab

Bibliography

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr