Chapter 4

Crisp Rule Based Systems

4.1 Introduction

-Production systems
-Facts & Templates
-Production rules
-The inference process
-Advantages of production systems
-Disadvantages of production systems
-Expert systems

4.2 Production Systems

-A production system consists of
–Working memory (facts memory)
–Production rules memory
–Inference engine
-Working memory
–working memory of the system
–stores the facts currently being dealt with
–facts also called working memory elements
-Production rules memory
–stores the set of productions (rules) of the system
–long term memory of the system
-Inference Engine
–control mechanism of the system
–matches facts from working memory with productions
–selects rules to execute

4.3 Facts

-Production rules assess and manipulate facts
-Facts are propositions about the objects dealt with by the system
-Facts are represented within templates
`( <object or relation> <attribute_1> <attribute_2>...<attribute_k> )`

4.3.1 Templates

-examples `(is_a <relationship> <name1><name2>)` a template
`(is_a father John Mary)` a fact
`(<car_par><parameter><value>)` a template
`(car_par  temperature 135)` a fact
`(<car_status><system> functioning <status>)` a template
`(car_status breaks functioning slowly )` a fact
`(car_status cooling functioning overheating)` a fact
`(car_status gauge functioning OK)` a fact
`(<lecture_attribute><property><value>)` a template
`(Presentation is Dull)` a fact

4.4 Production Rules

-Productions are transformation rules
–Gives one string from another string
—e.g.,
—AB ¿ CD
- Useful for things like compilers, as well as Production Systems
-A production rule consists of two parts
–left and right
–left side also known as
-condition
–antecedent
-right side also known as
–conclusion
–action
–consequent
-Antecedent part of the rule describes the facts or conditions that must exist for the rule to fire
-Consequent describes
–the facts that will be established, or
–the action that will be taken

```
IF (conditions) THEN (actions)
IF (antecedents) THEN (consequents)
```

-Condition elements can be:
–a negation of a fact (means absence of this fact);

–i.e. logical NOT
–e.g. `If NOT (sky_is cloudy)`
–expressions with variables or wild cards; a wild card is a variable which can be satisfied by any value; e.g. temperature ¿ 120.
-Examples

```
IF   (Gauge is OK)   AND     [TEMPERATURE] > 120
        THEN Cooling system is in the state of overheating

IF (Presentation is Dull) AND (Voice is Monotone)
       THEN Lecture is boring

       IF   (Gauge is OK)   AND     [TEMPERATURE] < 100
        THEN Cooling system is functioning normally

IF NOT (Presentation is Dull) AND (Voice is Lively)
   THEN Lecture is Great
```

4.5 The Inference Process

-Antecedent Matching
–matches facts in working memory against antecedents of rules
–each combination of facts that satisfies a rule is called an instantiation
–each matching rule is added to the conflict set or agenda
-One rule at a time fires
-Rule must be selected from the Agenda
-Some selection strategies:
–recency
–triggered by the most recent facts
-specificity
-rules prioritised by the number of condition elements
–matches rules with fewer instantiations
-Rule selection:
–refraction
—once a rule has fired, cannot fire again for a period of time
—certain number of cycles or permanently
-salience
–based on priority number attached to each rule
-random
–choose a rule at random from the agenda
-Execution of the rule
–can modify working memory
—add facts
—remove facts
—alter existing facts
–alter rules
–perform an external task
-Inference work repetitively

–match facts
–perform inference
–fire rules
–modify facts
–repeat
-continues until no more productions in the agenda

4.6 Advantages of Production Systems

-Universal computational mechanisms
–can represent any computational process, including a production system
-Universal function approximators
–can approximate any function given a sufficient number of rules
-Intuitive
–close to how humans articulate knowledge
–easy to get rules out of a client / user
-Readable
–rules are easy to read
-Explanatory power
–can clearly show how a conclusion was reached
-Expressive
–well crafted rules can cover a wide range of situations
–few rules are needed for some problems
-Modular
–each rule is a separate piece of knowledge
–rules can be easily added or deleted

4.7 Disadvantages of Production Systems

-Handling uncertainty
–how to deal with inexact values?
–if a value is unknown, how to represent it?
–concepts like tall, short, fat, thin
-Sequential
–one rule at a time fires
–results can depend on which rule fires first
-Elucidating the rules
–how to get the rules in the first place?
–who wants to write down 5,000 rules?
-Completeness of the rules
–do the rule cover every possibility?
–what happens if they dont?

4.8 Expert Systems

-Production systems are sometimes called expert systems
-They are not the same thing, however
-An expert system may use a production system, but a production system is not always in an expert system
-These are information systems for solving a specific problem which provides an expertise similar to those of experts in the problem area.
-An ES contains expert knowledge.
-A typical ES architecture consists of:
–knowledge base module;
–working memory (database) module (for the current data);
–inference engine
–user interface (possibly a NLI, menu, GUI, etc)
–explanation module
-Knowledge Base Module
–stores the domain knowledge
–analogous to the production rules module of production systems
–may be production rules
–may be another model, such as a neural network
-Working memory
–same as production systems
-Inference engine
–controls the functioning of the entire system
–inference mechanism can be forward or backward chaining
-Explanation Module
–explains the reasoning made by the system
–describes the HOW and WHY of actions taken
–HOW it has inferred a fact or conclusion
–WHY it has taken the action it has

4.9 The Inference Process

-Inference can be forward or backward chaining
-Chaining is a line of inference / reasoning

4.9.1 Forward Chaining

-Forward chaining
–starts from known facts
–fires rules to infer a conclusion

-Example:

```
the sky is clear
the temperature is warm
the wind is light
THEREFORE the weather is good
```

-Good for when the goal is not known by the user

4.9.2 Backward Chaining

-Backward chaining
–starts with a conclusion to be proven
–fires rules that can establish that conclusion
-Example:

```
The weather is good
THEREFORE
is the sky clear?
is the temperature warm?
is the wind light?
```

-Goal oriented
-Good for user interaction

4.10 Summary

-Production systems are rule based
-Production systems can be universal computational mechanisms and function approximators
-Provide Readable, explainable systems
-Dont handle uncertainty well
-Expert systems solve problems in one domain
-Can be based on PS, or other models
-Encapsulate domain knowledge for use
-Problems with acquiring domain knowledge