City University London

Module H800: Symbolic Artificial Intelligence

Component Guide: Rule Based (Expert) Systems

Andrew Tuson (email: andrewt@soi.city.ac.uk)

Contents

Chapter 1

Component Synopsis

Lecturer: Andrew Tuson

This component will aim to give the student an introduction to: rule based KBS/ES; the underlying knowledge representation issues; the architecture upon with they are based; some representative applications; KBS design issues; and tools (shells) for their construction. This component consists of four lectures and one supporting tutorial.

1.1 Learning Outcomes

The learning outcomes from this component will result in the student being able to accomplish the following:

1.1.1 Basic KR and Production Rule KBSs

- Note the role of logic (propositional and first-order) in AI knowledge representation (KR).
- Understand and apply production rules as a KR method, be aware of how they relate to logic-based KR, and their (dis)advantages;
- Understand and apply decision trees as a KR method, their relation to production rules, and their (dis)advantages;
- Describe the components/architecture of a production system;
- Illustrate the procedural workings of this architecture;
- Illustrate and describe the forward (data-directed), backward (goal-directed), and bi-directional chaining search options and be aware of the factors behind an appropriate choice.
- Understand, describe, and illustrate the purpose and use of the following conflict resolution strategies: recency, refraction, specificity.
- Be aware of efficiency issues, with particular reference to RETE networks;
- Describe and be aware of the application of a number of production system-based KBSs, for example: MYCIN, XCON, and MUD.
- Note that some method for explanation and uncertainty handling is often required, and the utility of meta-knowledge in KBS/ES systems;

1.1.2 KBS Design and Implemenation Issues

- Describe the stages of the KBS design process, understand their role and the issues that arise at each stage.
- Be aware of the 'knowledge acquisition bottleneck', and why KA is hard in practice.
- Be aware of and outline a number of knowledge acquisition strategies, their (dis)advantages, and the issues behind the choice of an appropriate KA method;

- Be aware of KBS validation and verification issues, with regard to representative metrics.
- Recall and describe the components of a ES/KBS;
- Be aware of the choices of programming languages for KBS development and their (dis)advantages.
- Appreciate the role of ES/KBS shells and the services they can offer;
- Be familiar with the basics of the ESTA shell sufficent for the purposes of the module coursework.

1.2 Component Materials

Many of the above topics should be covered in any decent AI text, though you may want to read other texts for alternative coverage. In addition, the slides will be given out at the start of the lecture (with supplementary handouts). The exact coverage required for the exam will be, of course, defined by the lecture (so turn up!).

Chapter 2

Component Tutorials

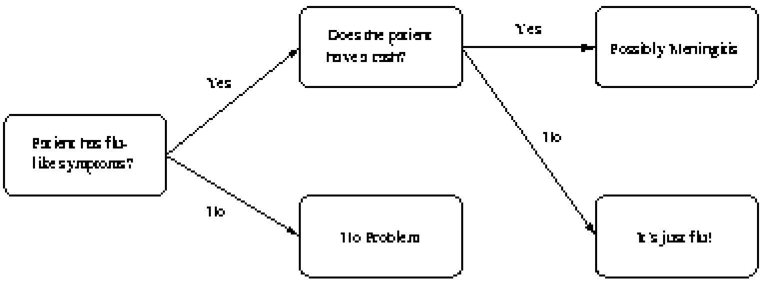
The tutorial exercises for each of the two weeks of this component are given below.

2.1 Tutorial 1: Rules, Decision Trees, and Production Systems

These exercises show help you to get used to some of the mechanics of knowledge representation and inference with the above methods.

2.1.1 Rules and Decision Tree (10 minutes)

Convert the (simple) decision tree below into equivalent production rules.



What process did you use to acheive this? Comment on the form of the rules you obtain.

2.1.2 Production Systems Exercise (40 minutes)

Adapted from Kahney et al 'Knowledge Engineering'

Consider the following set of production system rules for identifying a fault with a car's starter motor:

```
defrule rule1
if
car_not_starting &
    battery_is_ok &
    car_has_petrol
then
starter_problem.
end.

defrule rule2
if
car_lights_on
then
battery_is_ok
end.
```

with the following state of working memory:

```
car_not_starting
car_has_petrol
car_lights_on
```

and the query:

```
?- deduce starter_problem
```

Please split up into small groups and attempt the following:

1. Work out a trace of what happens when these rules are processed by backward-chaining;
2. Repeat this for a forward-chaining system;
3. Extend the above set of rules to illustrate the effect of one or more of the following conflict resolution strategies: refraction, recency, and specifity.

2.2 Tutorial 2: Coursework Checkpoint

This week, please bring the checkpoint report to the tutorial. The tutor needs to make a note of the following: your name, your email address, your course and year, and a descriptive title of your proposed system. The tutor will try to give some general feedback and answer any queries you may have. If it turns put that some proposed systems are very similiar then Andrew Tuson will get in touch later to resolve this.

Appendix A

Tutor's Notes

A.1 Tutorial 1: Rules, Decision Trees, and Production Systems

Outline answers to the tutorial exercises are below.

A.1.1 Rules and Decision Trees

The method is as follows. Take one of the branches of the first (root) node. The decision made there becomes one of conditions of the rule. From there each of the additional decisions you make while traversing the tree also becomes an extra condition. Join all of the conditions with **and** to complete the rule and have the leaf node become the consequent of the rule.

Repeat this procedure for each of the different way of traversing the decision tree, to get all of the rules.

In the example considered here the rules thus produced are:

```
IF not has_flu_symptoms THEN no_problem

IF has_flu_symptoms AND has_rash THEN meningitis

IF has_flu_symptoms AND not has_rash THEN flu
```

The rule set may resemble a look-up table, and thus some rules may be usefully rearranged to better reflect how an expert would write them. Also the procedural information implicit in the decision tree has been lost. Finally if the decision tree is deep, then a rule will have many preconditions (and thus could be broken down to reflect stages in reasoning).

A.1.2 Production Systems Exercise

See the photocopied solution of the backward chaining system. The forward chaining example should follow from this. It may also be useful to get the students to recall when backward or forward chaining should be used.

In the case of (3), this is more open ended. I suggest that you get the students to recall what these conflict resolution strategies are, and then add rules such that a different result occurs with different choices of conflict resolution strategy. An example would be where two diagnoses with very similar symptoms (they have only one different) occur. In this case, if the discriminating rule is removed or tampered with, then it may be possible that the conflict resolution strategy affects the answer. Rule sets where changes in the conflict resolution strategy lead to a different path to the same answer would also be fine.

A.2 Tutorial 2: Coursework Checkpoint

Nothing specific. Get their details down and back to me (Andrew Tuson). Also try to provide feedback and helpful suggestions when possible. It may be a good idea to get the students involved in making suggestions.