CHAPTER 6

CONCLUSIONS
The present work has been concerned with exploring the various avenues of parallelization in Hierarchical Censored Production Rules. We have discussed the design and implementation of a Parallel Specificity model and Parallel Censored Production Rules system. This chapter presents a brief summary of the results obtained in this thesis and discusses directions for future work.

6.1 Summary of Results

Main characteristics of HCPRs are the specificities which connect the related chunks of knowledge in an hierarchical manner. In this thesis we have first of all identified the various characteristics of the specificities and the generalities of the rule, which support inherent parallelism. All related specificites and generalities of the rule were found to be independent of each other having a disjoint set of rules associated with them, implying that they can be evaluated asynchronously with minimum synchronization overheads. Further all the specificities of a rule are mutually exclusive, meaning that only one of them can be true at any instance of time. This allows the preemption of redundant tasks which leads to better utilization of available resources.

We exploited these characteristics of specificities to develop a Parallel Specificity model for a shared memory architecture. The measurements were carried out for a knowledge base of about 1500 rules, with different set of facts each time. The model is found to highly dependent upon the knowledge base. The speedups were found to be better for cases where the knowledge can be represented in the form of a large hierarchy, and the specific cases that happen to be true are present towards the end of the specificity list. The speedups in
such cases are almost linear for the number of processors integral multiple and less than or equal to the number of tasks to be evaluated in parallel.

Another model that we have developed in this thesis is the Parallel Censored Production Rules model, in which the premises and the censors of an individual HCPR are handled in parallel. The speedups obtained in the model are found to be independent of the knowledge base. The results indicate linear speedup performance of upto 7.3 fold for a small number of processors (<10) and a reasonably large number of rules. The speedups obtained were linear with almost 100 percent processor utilization for upto six processors after which the speedups becomes very gradual and the processor utilization decreases.

Both the above models are simulated for shared memory architecture using a parallel programming language MultiPascal.

6.2 Suggestion for future work

Although many issues regarding the use of parallelism in implementing a parallel HCPRs model have been addressed in this thesis, many more remain to be addressed. This section discusses some such issues.

The Parallel Censored Production Rules Model is divided into two phases of execution, viz., the evaluation phase and the propagation phase. The propagation of the results is done only after all the rules upto the specific censor chaining depth have been evaluated. The propagation is done breadth first from the leaves of the rule tree to the root. This approach does not take care of the condition when during the evaluation process, any censor or any premise value turns out to be such that the overall decision becomes false. In such a situation, then there is no need to continue with the evaluation of the rule. It would be interesting to develop a parallel censors model, which takes
care of the preemption of tasks also and compares its performance with the present Parallel Censored Production Rules model, developed in this thesis.

The systems have been developed for shared memory architecture. It would be a good idea to see their scope for distributed memory architectures. Implementation on distributed memory architecture system, would mean distributing the knowledge base into independent sets, such that they can be handled in parallel without much message passing.

In this thesis we have developed two separate models for handling specificities and censored production rules. It would be interesting to combine these two models to develop a complete Parallel Hierarchical Censored Production Rules model. This would mainly mean developing proper processor allocation schemes, such that these can be utilized efficiently.