The pattern and subject trees; (c) an index into an array of resource vectors; and (d) cost, latency and delay cost data. Information in this table is used by the code generator when performing instruction scheduling, register allocation and code selection. Other generated tables contain declarations, information about the virtual machine, classes and elements, auxiliary latencies, patterns, etc.

Two important target-dependent functions are the one that returns the general purpose register set for a given type and the one that returns register overlapping information and register sizes. This information is used for the creation of pseudo-registers and register allocation.

6. Conclusions

The paper described a code generator generator, named GGCO. The most important contribution is the formal specification of the processor dependent part of the code generator by means of a special purpose language. Its formal definition may be seen as a code generator system which provides, from the machine description a mapping from a processor description to its code generator, and whose final result is a program piece in C, which corresponds to the machine dependent part of the code generator for the described architecture.

The system prototype is not yet fully implemented. At the moment, for example, facilities to support register windows for the Sun SPARC station permit only one window. Additional studies must be done before the incorporation of the register allocation algorithm presented in Section 3.2 into the system, in order to evaluate and compare it with the algorithm proposed by Bradlee for scheduling and register allocation. Gross and Hennessy's algorithm must still be implemented to fill the delay slots with valid instructions during scheduling. The implementation of the proposed solution to attend the priority schema present in some superscalar architecture is still yet to be accomplished. On the other hand, all above facilities are already available in LDA.

Referencias