The exam was too easy for this cohort. Two students got 100%, and several more got close to that.

Q1. This was done satisfactorily. Parts a) -- c) were essentially about difference lists: candidates either understood them or they didn't. Part a) was probably the trickiest of these parts. Part d) was general Prolog programming. One alarmingly common error was to try to embed `function calls' in Prolog, as if it were Java. How anyone can have got through all the labs and still write `ack(M,N,Ans):- ack(M-1, ack(M,N-1))' is beyond me. Those who did were punished.

Q2. This was moderately well done, though, oddly enough, most errors occurred in the basic formalization (Part (a)). It was as if candidates had learned all the technical machinery without stopping to think what any of this stuff actually meant. In general, I did not take marks off subsequent parts if they were correct but based on erroneous answers to earlier parts (e.g. mixing up an `and' and an `implies' in Part (a).) However, I did take marks off in Part (d) if there was a correct resolution proof based on over-strong (erroneous) formalizations of the premises: candidates should have noticed that something was wrong in this case.

Q3. Most candidates did Parts (a) and (b) well. Whoever taught formal languages in the first year had obviously done a good job. The last part proved more of a challenge. Most candidates realized that high-caste Numerian was a non-context-free language, and that, therefore, variables were needed to pass information around the parse-tree. But very few implemented it correctly.

Q4. This was attempted by many candidates, almost all of whom aced it. Simple as that.