p8 q13
MJCG

Solution notes for SV2.2 2000

Syntax and semantics of CTL is bookwork. I would expect most candidated to reproduce the definitions from the notes, which are couched in terms of higher order logic. However, I'd be happy with other (e.g. more textbook-like) presentations.

LTL (Linear Temporal Logic) differes syntactically in not having path quantifiers A and E. It differes semantically in that a CTL formula is evaluated with respect to a state (the root of a branching-time tree), whereas LTL formulas are predicates on traces (infinite sequences of states).

if req goes high then it will stay high until ack goes high and then go low on the next cycle

```
AG[req ==> (A[req U ack] /\ AG[ack ==> AX ~req])]
```

if ever req is high and started is low then sometime later error will become permanently high

```
AG[req /\ ~started ==> AF error]
```

Model checking consists of checking properties (often written in temporal logic, e.g. CTL or LTL) of particular finite state automata which model computing systems. The models are usually compiled from higher level descriptions in higher level languages (e.g. bespoke model checking languages like SMV or hardware description languages). The model checking process uses stsate enumeration algorithms to check the properties, often based on fixed-point calculations. Symbolic model checking uses algorithms that manipulate symbolic representations of boolean formulae, often based on binary decision diagrams (BDDs).

A theorem prover can be used to verify arbitrary properties of arbitrary systems. There is no restriction on the models being finite state. However, theorem provers offer limited automation and require manual guidance. Often this is very tricky and time consuming. Model checking is restricted to finite state systems (mainly) but it is largely automatic. Model checking algorithms can find counterexamples when properties fail to check, which is very useful for debugging. Because model checking is automatic it is gaining popularity in industry, more so than theorem proving which is generally considered too hard to use (there are a few notable exceptions)