The expressive powers of LTL and CTL

CTL allows explicit quantification over paths, and in this respect it is more expressive than LTL, as we have seen. However, it does not allow one to select a range of paths by describing them with a formula, as LTL does. In that respect, LTL is more expressive. For example, in LTL we can say 'all paths which have a p along them also have a q along them,' by writing $F\ p \rightarrow F\ q$. It is not possible to write this in CTL because of the constraint that every F has an associated A or E. The formula $AF\ p \rightarrow AF\ q$ means something quite different: it says 'if all paths have a p along them, then all paths have a q along them.' One might write $AG\ (p \rightarrow AF\ q)$, which is closer, since it says that every way of extending every path to a p eventually meets a q, but that is still not capturing the meaning of $Fp \rightarrow F\ q$. CTL* is a logic which combines the expressive powers of LTL and CTL, by dropping the CTL constraint that every temporal operator (X, U, F, G) has to be associated with a unique path quantifier (A, E). It allows us to write formulas such as

- $A[(p\ U\ r) \lor (q\ U\ r)]$: along all paths, either p is true until r, or q is true until r.
- $A[X\ p \lor X\ X\ p]$: along all paths, p is true in the next state, or the next but one.
- $E[G\ F\ p]$: there is a path along which p is infinitely often true.

These formulas are not equivalent to, respectively, $A[(p \lor q)\ U\ r)]$, $AX\ p \lor AX\ AX\ p$ and $EG\ EF\ p$. It turns out that the first of them can be written as a (rather long) CTL formula. The second and third do not have a CTL equivalent.

The syntax of CTL* involves two classes of formulas:

- state formulas, which are evaluated in states:
  $\varphi ::= \mid p \mid (\neg\varphi) \mid (\varphi \land \varphi) \mid A[\alpha] \mid E[\alpha]$
  where p is any atomic formula and α any path formula; and
- path formulas, which are evaluated along paths:
  $\alpha ::= \varphi \mid (\neg\alpha) \mid (\alpha \land \alpha) \mid (\alpha\ U\ \alpha) \mid (G\ \alpha) \mid (F\ \alpha) \mid (X\ \alpha)$ where φ is any state formula.

  This is an example of an inductive definition which is mutually recursive: the definition of each class depends upon the definition of the other, with base cases p and T.