Why We Don't Always Require Fairness

Why don't we require condition 3 to hold for all algorithms? Why must we add a weak fairness formula to the specification? We wouldn't want to allow a program to stop at any point in its execution.

An algorithm is not a program. An algorithm is a specification, which is an abstract model of a system. A statement in an algorithm might model the user providing input to the system. We probably don't want to require the user to provide input. We want to allow behaviors in which the system waits forever for the input—that is, behaviors ending in a state in which the system is waiting for input. We want the algorithm to satisfy a modified version of condition 3 that allows behaviors to end in certain states. We will see later how to express such a condition in PlusCal.

Weak fairness of the next-state action is just one of many possible fairness requirements that we might want of an algorithm. Some of them can be expressed in the PlusCal code, but no sensible language could express them all that way. For some, we must write the code with no fairness requirement and conjoin the desired fairness formula to the specification produced by the translator.