

nr's theory:

1. An equation can be solved if it has a side with no unevaluated (i.e., unbound) logical variables.
2. Once an equation is solved, all its logical variables are evaluated (bound).
3. A guard can be evaluated "like pattern matching" if there is an order in which all its equations can be solved.

Now consider a graph of equations, not variables. There is an edge from equation A to equation B if solving A makes B solvable. That may be the dependency graph you are looking for.

guard ::= exp list

exp ::= value | eq; e | \exists x. exp | fail | e1 **■** e2 | v1 v2 | oneexp | allexp

eq ::= exp | equation

equation ::= x = exp

A term eq is either an ordinary expression e, or an equation v = e; this syntax ensures that equations can only occur to the left of a ";".

BUILDING A GUARD IS THE TOPOLOGICAL SORT.

the guard is the decision tree.

BIG NOTE:

In verse,

1 ; 2 will give you 2 results. we can't have 2 results in ml. so- can you eliminate the form e1 ; e2 ?