Introducing Three New Predicates and Functions

Author: Joshua Harkness

Predicates

self(A,B): This predicate would find values where B equals one or zero, and multiply or add B to A respectively to keep A's value the same. It will then return the proper expression.

half(A,G): Returns true if the number given is half the goal.

oneLess(A,G): This predicate returns true when one number is one less than the goal.

Functions

locateSpeed: The locateSpeed function would calculate the speed at which a human would be able to depict which heuristic is applicable for a given problem. This function could be called in the hex function to add accuracy to the human executability.

extraNums: This function would take into consideration how many numbers can be ignored in each heuristic. The more numbers that can be ignored, the greater the score will be towards human executability. An example of this is when the heuristic multiplies the remaining numbers by zero. This function will also be useful inside of the hex function.

compSpeed: This function will consider the computation speed of a given heuristic. An increasing number of predicates likely leads to more computation time. Another factor that would increase computation time is how concise the predicates are.

Complete Language

Predicates

sameP(): Returns true if all the numbers are the same.

zeroP(): Returns an expression which equates to zero.

oneP(): Returns the proper expression if the one can be equated from the numbers.

goalP(): Returns an expression that equates to the goal from the numbers.

numbersP(): This predicate contains the extra numbers in a heuristic.

onemoreP(A,B): Returns the proper true if one number is one more than another number.

twomoreP(A,B): Returns the proper expression if one number is two higher than another number.

twoP(A,B): Returns the proper expression of two numbers equating to two.

self(A,B): This predicate would find values where B equals one or zero, and multiply or add B to A respectively to keep A's value the same. It will then return the proper expression.

half(A,G): Returns true if the number given is half the goal.

oneLess(A,B): This predicate returns true when one number is one less than the goal.

Functions

hex: Determines the human executability of a given heuristic. Based on a 0 -1 scale, the score depicts how easily a human would be able to use the given heuristic.

app: Determines the applicability of a given heuristic when all possibilities for number combinations are considered.

locateSpeed: The locateSpeed function would calculate the speed at which a human would be able to depict which heuristic is applicable for a given problem. This function could be called in the hex function to add accuracy to the human executability.

extraNums: This function would take into consideration how many numbers can be ignored in each heuristic. The more numbers that can be ignored, the greater the score will be towards human executability. An example of this is when the heuristic multiplies the remaining numbers by zero. This function will also be useful inside of the hex function.

compSpeed: This function will consider the computation speed of a given heuristic. An increasing number of predicates likely leads to more computation time. Another factor that would increase computation time is how concise the predicates are.