Understanding how people approach constraint modelling and solving – Codes

Anonymous

1 Codes

The codes for annotating the participant videos are split into 3 categories. These categories are Visual Elements (VE), Memory related visual elements (MEM) (both in Table 1), and Solving Approaches (SA) (split over multiple tables).

The majority of visual codes are the same as in [1]. We added a few codes that are more specific to people solving, rather than explaining the problem. In addition to these elements we note a few more codes which were found while the participants were solving. We thus leave it to the reader to look at the codes in [1] and only list in Table 1 the additional ones we identified.

The solving approaches category is split into solving approaches that can be universally found in all problems (SA1.X, Table 2) given to the participants and ones that are specific to the problems. SA2.X relates to the Knapsack problem (Table 3), SA3.X relates to Sudoku (Table 4) , SA4.X relates to Subset sum (Table 5), SA5.X relates to the Scheduling problem (Table 6), SA6.X relates to Magic Square (Table 7)and SA7.X relates to Word Crypto (Table 8).

In Sudoku solving a lot of the strategies are based on the positioning of values in a Block, Column or Row (BCR). It is also common for the participants to focus on identifying as few positions in BCRs in which a value can be placed.

References

[1] Xu Zhu, Miguel A. Nacenta, Özgür Akgün, and Peter Nightingale. How people visually represent discrete constraint problems. *IEEE Trans. Vis. Comput. Graph.*, 26(8):2603–2619, 2020.

Code	Name	Description
VE1.9	Recreating Instance	Use of the visual element given in the problem description or recreation of the representation used in the previous part of the experiment.
VE1.10	Lists& Sequences	Explicit lists or sequences of elements or numbers. This can be seen as a single column vector, or single row.
VE1.11	Crossing out	Line(s) through a single or more characters, or scribbles over a larger area.
MEM1.1	(Sorted) Lists	Use of lists or tables to aid memory of all possible variables, values, objects etc.
MEM1.2	Table	Use of multi-column/multi-row tables with an ordering on the objects.
MEM2	Grouping	Objects are noted based on proximity (VE 1.8) or surrounded by boxes/circles (VE 1.1.3) and gestures towards that grouping using the lasso gesture (VE 2.2.2).
MEM3	Temporary solution	Noting and pointing out a temporary solution. This solution could be incorrect or partially valid. A temporary solution is one where there is a majority of variable and value assignments, and is written out as such.
MEM4.1	Visual representation of globally possible values	Note (or a mark) of possible values that can be assigned to a variable. The assessment of the possibilities was made with all variables or values in consideration.
MEM4.2	Visual representation of locally possible values	Note (or a mark) of possible values that can be assigned to a variable. The assessment of the possibilities was made based on a subset of either. This subset can be based on proximity.

Table 1: VE and MEM Code table

Code	Name	Description
SA1.1.1	Backtracking remembering (no)goods	The participant is found to backtrack if they have reached a point in their solution where they believe not to be able to continue on and they undo one (or more) of their previous steps, but keep some of their assignments. They have learned something about the variables and values that either is a good thing, (good clause) or something that cannot ever happen (nogood clause).
SA1.1.2	Backtracking forgetting (no)goods	Same as SA1.1.1 but the participant does not remember anything they might have learned.
SA1.1.3	Backtracking same strategy	Same as SA1.1.1 or SA1.1.2 and observing that the participant is continuing with the strategy that they have used so far (until this step in their solving process).
SA1.1.4	Backtracking different strategy	Same as SA1.1.1 or SA1.1.2 and observing that the participant is using a different solving approach.
SA1.2.1	Restart remembering (no)goods	The participant has decided to give up on their current partial (or full) solution. It is irrelevant whether that solution is correct. They then start the solving from the beginning and remember something they about the variables and values.
SA1.2.2	Restart forgetting (no)goods	Same as SA1.2.1 but they do not remember anything.
SA1.2.3	Restart same strategy	Same as SA1.2.1 or SA1.2.2 and the participant use the same strategy as before they gave up.
SA1.2.4	Restart different strategy	Same as SA1.2.3, except that they pursue a different approach.
SA1.3	Random Partial Assignment	A strategy of randomly assigning a few variable value pairs.
SA1.4	Determine (no)goods	The participant might not settle on a specific value/variable assignment, but they will find a subset of the domain that is either good or bad for a given variable or for the whole problem. The participants will usually queue this solving strategy by saying *this is a good thing* or *this is a bad thing* to have.
SA1.5	Propagation	The participant follows a variable/value assignment with checks to see if or how it influences other variables in the puzzle, and fills in any assignments from that.

Table 2: Universal SA Code table

Code	Name	Description
SA2.1	By value	The set of objects will be sorted into a highest to lowest value list and then a subset of the highest valued objects will be chosen.
SA2.2	By weight	The set of objects will be sorted into a heaviest to lightest weight list and then a subset of the heaviest valued objects will be chosen.
SA2.3	By value/weight ratio	The set of object will be sorted into a highest to lowest value to weight ratio. A subset of the highest ratios will be chosen.
SA2.4	By weight/value ratio	The set of object will be sorted into a lowest to highest weight to value ratio. A subset of the lowest ratios will be chosen.
SA2.5	By weight value product	The set of object will be sorted into a highest to lowest weight by value product. A subset of the highest products will be chosen.

Table 3: Knapsack specific SA Code table

Code	Name	Description
SA3.1	BCR elimination	Based on the positions of a value in blocks, columns and rows the participants excludes cells which that value could contain, until they find a set of positions in a BCR where it is possible to fill in that value.
SA3.2	BCR only possible value	Following BCR the participant concludes that there is no other value that can be filled into given cell in that BCR.
SA3.3	BCR only possible cell	Following BCR the participant concludes that there is no other cell in that BCR where a given value can be filled in.
SA3.4	BCR last cell	In a BCR there is only one cell free.
SA3.5	Position in Blocks	Similar to the BCR elimination, but focusing on just a block by block basis. The participant eliminates cells in a block into which a given value cannot appear.
SA3.6	Cell focus	Rather than focusing on a whole BCR the participant focusses on a given cell and finds what values that cell could validly contain.
SA3.7	Follow most recent value	The participant follows the most recently entered value to see if it has any propagational impact in other cells, mainly on the same value.

Table 4: Sudoku specific SA Code table

Code	Name	Description
SA4.1	Brute force by values	The participant finds all possible values for sets of one sign (or both) and assigns some of the elements of the other sign to it till the sum matches (or checks the sums of all possibilities till it works).
SA4.2	Brute force by set size	The participant creates all sets of a given set size and checks which have the correct sum. This set enumeration can be done explicitly or implicitly.
SA4.3	Sort negative vs positive	The participant sorts the values into a table of elements, ordered by absolute value, with one column containing only positive values and the other only negative values.
SA4.4	Match By	The participant creates sets in a given way and then matches them up to get the right final sum. Such matchings happened through comparing *SA 4.4.1 sets of absolute values*, or a *SA 4.4.2 single number against single numbers (or pairs) of the other sign.*

Table 5: Subset sum specific SA Code table

Code	Name	Description
SA5.1	Propagate busy per person	The participant marks in their schedule when people are busy (or when they are available).
SA5.2	Propagate meetings	The participant marks all (or some) options for each meeting. This could be dependent on other assigned meeting or not.
SA5.3	Unique options first	The participant fixes the meetings with an unique option first.
SA5.4	Most constrained first	The participant fixes the most constrained meeting or person first.

Table 6: Scheduling specific SA Code table

Code	Name	Description
SA6.1	Column/Diagonal/Row Equations	The participant creates the equations for the columns, diagonals and rows, with or without the knowledge of their sum.
SA6.2	Find sum value	The participants attempt to find the value of the column, diagonal and row sums in different manners. Some use the *SA 6.2.1 average value of triples*, others *SA 6.2.2 match triples* in an ordered fashion. Others yet again find the sum of all together and weight it into triples (*SA 6.2.3* $(1+\cdots+9)/3$), while others have seen this problem before and have *SA 6.2.4 previous knowledge* of the sum.
SA6.3	Random partial assignment	This code is more specialised than the universal category SA1.3 Random Partial Assignments. The participants approached the random partial assignments with some additional knowledge and stayed within the realms of it. The participant either *SA 6.3.1 RPA with sum knowledge* where they found different triples which summed to the same value, or *SA 6.3.2 RPA in equations* where they have the equations for the Columns/Diagonals/Rows and fill the variables randomly there.
SA6.4	Spread extremal values over Columns/Diagonals/Rows	The participants fill extremal values (lowest or highest) into the grid but spreading them over non-clashing columns, diagonals or rows.
SA6.5	Centre Cell	The participants focus on the centre cell, and either *SA 6.5.1 attempt* to figure out what belongs, or know that a "neutral" middle value (in this case 5) belongs there, we call this *SA 6.5.2 middle in middle*. Another way they fill the centre cell is with a high or low value (*SA 6.5.3 extrema in middle*).
SA6.6	Permuting	This strategy usually follows on after all variable have been assigned a value, but the participant finds that some of the sums are still not the same or not what they expected. This shuffling of variables is done over *SA 6.6.1 columns/rows* or over *SA 6.6.2 equations*.
SA6.7	Extrema in corner	The participant fills the highest or lowest values into the corner cells.

Table 7: Magic Square specific SA Code table

Code	Name	Description
SA7.1	Column Equations	The participant creates equations on a column by column sum basis. This is usually followed by an attempt to minimise the number of variables, through the use of algebraic substitutions.
SA7.2	ENR relations	A set of relations or equations between the E,N, and R variables is created.
SA7.3	Last column nogoods	The participant creates clauses or assignments from the right most column.

Table 8: Word Crypto specific SA Code table