A proposal for a collaborative Sudoku solver based on tkeden

A simple collaborative approach

The obvious way to attempt to solve Sudoku puzzles collaboratively is to have several people simultaneously inspecting a puzzle as it is being solved. When one person spots what they believe to be a valid inference they can then point this out to the others, who can check the reasoning. If all participants are satisfied that a valid deduction has been made about the value of a square, this value can then be entered.

This approach to collaborative solving probably works most effectively when the puzzle is moderately easy. Some puzzles can be solved without too much difficulty by using very basic rules of inference - the main problem is that timely identification of where a rule applies involves an element of chance. Having several people inspecting the grid at once improves the chances of a key inference being spotted quickly.

Examples of basic inference rules that can be used to enter a value v in place p:

- Rule 1: v is the only value that doesn't conflict directly with the combined contents of its row, column and square
- Rule 2: p is the only place where v can be placed within a specific col, row or square
- Rule 3: the values other than v necessarily have to be placed elsewhere in a row, col or square

In the most direct application of such rules, the premise of a rule is established by the explicit presence of digits on the grid, but sometimes a more indirect argument can be used. For instance, the knowledge that a value v must appear in row x or row y in column z makes it impossible to place the value v in any other row in column z, even though we may not actually know precisely in which row value v appears in column z.

One possible application for dtkeden would be to enable several people to try to solve a puzzle collaboratively using this basic approach. Some of the key concerns in that context would be:

- It isn't necessarily the case that there are many different independent simple inferences that can be made in any particular puzzle situation. This means that all solvers may well need to be thinking about the same issue if they are to have any chance of contributing directly to the solution.
- In the process of searching for a key inference, each individual solver acquires knowledge that is not as easily communicable as 'the value in this square is x'.
- It is very hard to recover from mistakes even in individual solution of a puzzle - unless you have carefully documented the order in which you make entries for instance. In collaborative solving, it is even harder to undo an error. This suggests that yet more emphasis needs to be put on checking any proposed inference. The result may be that the benefits of having many minds at work are outweighed by the need for participants to reconstruct and check each other's thinking.

An additional meta-concern in using sudoku to study EM for collaborative working in this fashion is that solving the puzzle by entering numbers into the grid does not represent model-building in more than a superficial sense. A conceptually richer extension of this approach would involve solvers-acting-as-modellers identifying basic rules and introducing actions to implement these rules. Such work could truly involve model-building and be distributed in such a way that it was not duplicated. Its impact would be to do more than tackle the solution a specific puzzle - it would involve on-the-fly development of a semi-automatic puzzle-solving environment that might be expected to lead to the fully automatic solution of simple puzzles.

An alternative approach

An interesting question is whether there is any way of collaboratively solving sudoku puzzles that could significantly outperform a solitary solver. In this context, it is appropriate to have in mind sudoku puzzles that are sufficiently 'hard' that they admit no routine solution using a small repertoire of standard basic rules. The status of this category of puzzles is unclear: it is not obvious whether there is a comprehensive set of rules that can be feasibly applied to solve any sudoku puzzle, or whether there are puzzles for which there is no better solution than brute force search. However, sudoku composers are disposed to classify their puzzles by difficulty (as hard, medium or easy), and regard a sudoku puzzle as well-posed only if there is a sequence of deductions that can be used to progress towards a solution without backtracking. The soft nature of these concepts is rather striking - from a computational perspective, brute force solution presumably may be just as effective whether the puzzle is hard, medium or easy, and what is deemed to be "a useable deduction" has to be defined with reference to something other than abstract logical entailment. For example, a computer program that "helps" the solver by providing the correct value of a blank square at random may give no clue as to how to arrive at that particular value by logical steps each of which the solver can directly apprehend. These observations motivate a qualification of the concept of 'solving a sudoku puzzle' that embraces more than coming up somehow with a complete grid that satisfies the required constraints: solving the puzzle must be understood to mean *providing a step-by-step solution with an associated step-by-step logical justification*.

The broad idea behind the proposed collaborative solution strategy is to exploit a *Human Computing* style of interaction supported by modelling with dependencies using `tkeden`. Working with three solvers, for instance, each solver will first study the given puzzle independently and focus on the constraints that govern the disposition of 3 digits. (So that e.g. A considers how the digits 1, 4 and 7 are disposed; B the digits 2, 5 and 8; C the digits 3, 6 and 9.) Each will record the product of this study by making definitions in a pre-existing script. This script will include nine observables (called OK1, OK2, etc.), one for each of the digits 1 to 9, that indicate whether a particular configuration of digits on the grid is consistent with the constraints on disposition of the corresponding digit. The idea is that the products of private reflection on the grid will be pooled in a collaborative phase in which the objective is to take one or more logical steps towards a solution. Phases of individual and collaborative work on this pattern are expected to alternate as many times as are required to arrive at a complete solution. It is unclear whether this strategy will be effective, and if so how much iteration will be required.

At the current stage of development, the basic sudoku model due to Karl King has been extended so that the nine observables OK1, OK2, ..., OK9 have been specified in terms of a list of constraints ('checkrules') to be supplied by solvers A, B and C. A typical constraint takes the form of a statement of logical equivalence, such as "inserting the digit 8 in square h1 is equivalent to inserting 8 in i5 is equivalent to inserting 8 in b6 is equivalent to *either* inserting 8 in a8 *or* in a9". Another constraint might take the form "*either* inserting 8 in h1 *or* in i1 is equivalent to *either* inserting 8 in d2 *or* in f2". To encode both these rules governing the location of 8 in the grid, the following `tkeden` definition must be introduced:

```
checkrules8 is [[[h1], [i5], [b6], [a8, a9]], [[h1, i1], [d2,f2]]];
```

Combining checkrules for digits in a single model poses no problem, but the framework within which these are used in collaboration and negotiation when completing the puzzle is yet to be explored. A key issue would be identifying and handling any errors in the specification of checkrules.