Conceptualizing Procedural Knowledge Targeted at Students of Different Skill Levels

Martin Možina, Matej Guid, Aleksander Sadikov, Vida Groznik, Jana Krivec, and Ivan Bratko
Contact email: martin.mozina@fri.uni-lj.si
University of Ljubljana, Faculty of Computer and Information Science, Slovenia

Abstract. Conceptualizing procedural knowledge is one of the most challenging tasks of building systems for intelligent tutoring. We present a novel algorithm that enables teachers to accomplish this task (semi)automatically. Furthermore, it is desired to adapt the level of conceptualization to the skill level of particular students. We argue that our algorithm facilitates such adaptation in a straight-forward fashion. We demonstrate this feature of the algorithm with a case study.

1 Conceptualization of Procedural Knowledge

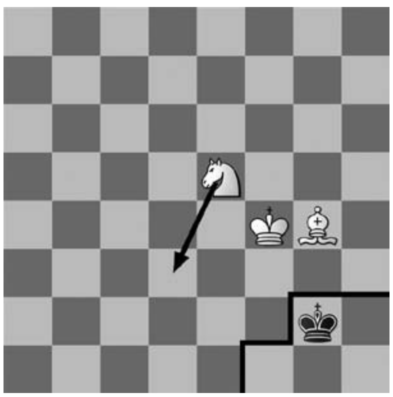
In symbolic problem solving domains (like physics, mathematics, or games like chess), a particular domain is defined with a basic domain theory and a solution to be achieved. The task is to find a sequence of steps that bring us from the beginning state of the problem (definition of the problem) to the goal state (the solution). The basic domain theory (or basic declarative knowledge of the domain) is usually simple and easy to remember and, in principle, sufficient for solving problems; e.g. knowing rules of chess could in theory enable optimal play. However, finding a solution using only declarative knowledge would require far too extensive searching. A human student is incapable of searching very deep, therefore we need to teach him also the procedural knowledge – how to solve problems.

The “complete” procedural knowledge would be a function mapping from each problem state to an action that leads to the solution. For example, in chess endgames a tablebase specifies best moves for all possible positions. Tablebases can be used easily because they only require trivial amount of search. But now the problem is the space complexity – it is impossible for humans to memorize such tablebases that typically contain millions of positions. There is a way, however, that enables humans to solve problems in such chess endgames quite comfortably. Humans use some intermediate representation of the domain that lies between the basic domain theory and the ‘”complete” procedural knowledge. We call such an intermediate representation a “conceptualized domain”.

We propose a goal-oriented conceptualization of domains. A goal-oriented rule has the following structure: *IF preconditions THEN goal (depth).* The rule's *preconditions* and *goal* are both expressed in terms of attributes used for describing states. The term *preconditions* specifies applicability of the rule, while a *goal* specifies the values of attributes in the state to be achieved. The *depth* property of a rule is the maximum allowed number of steps in achieving the *goal*. We developed an interactive algorithm that combines specialized minimax search with the ABML principle [2] for (semi) automatic construction of such rules, where the teacher and the algorithm in turns improve the model. The *depth* parameter is set prior to learning and can be used to dictate the complexity of learned rules. A higher *depth* will result in fewer rules with more general goals and vice-versa. Due to space limitations, we will skip the description of the algorithm (for details see [1]) and demonstrate its idea on a case study.

2 Case Study: KBNK Chess Endgame

KBNK (king, bishop, and knight vs. a lone king) is regarded as the most difficult of the elementary chess endgames. Most books mention only a basic strategy, however, it is hardly enough for successfully checkmating the opponent. Our aim was to conceptualize procedural knowledge in this domain for chess players at club level. Our chess teacher evaluated that they are able to calculate chess variations about 3 moves (6 plies) ahead.



Computer: "I suggest the following goal: the distance between black king and the edge of the board should decrease. However, it does not seem to work well in this position. *What goal would you suggest for white in this position? What are the reasons for this goal to apply in this position?"*

The teacher gave the following answer: "Pushing black king to the edge of the board is fine. However, I find the following goal to be more instructive for the student: *Build a barrier and squeeze the defending king into the corner. Currently such barrier is not yet established.* The move expected from the student is 1.Ne5-d3 achieving the goal."

Figure 1: Interaction between computer and teacher: explanation of a critical example.

An example interaction between the method and the teacher is shown in Fig. 1. The teacher is presented with a critical example, i. e., the example where the current set of rules suggested a bad goal ("push black king to the edge of the board" can be achieved, but is not leading to solution). The teacher was therefore asked to provide a better goal for this position, which was then used in the construction of a new set of goal-based rules. The process was completed when all critical examples were explained by the expert.

The final rules[1] were presented to three chess teachers (among them a selector of Slovenian women's squad and a selector of Slovenian youth squad) to evaluate their appropriateness for teaching chess-players. They all agreed on the usefulness of the presented concepts and found the derived strategy suitable for educational purposes at the level targeted for. Among the reasons to support this assessment was that the instructions "clearly demonstrate the intermediate subgoals of delivering checkmate."

References

[1] Možina M., Guid M., Sadikov A., Bratko I.: Goal-Based Rule Learning. Technical report, University of Ljubljana, 2009, *http://www.ailab.si/matej/KBNK/GBRL.pdf*.

[2] Možina, M., Žabkar, J., Bratko, I. Argument based machine learning. *Artificial Intelligence*, 2007, 171(10-15), pp. 922-937.

[1]The complete rule-based model for KBNK and example games containing automatically generated instructions can be found in a web appendix at *http://www.ailab.si/matej/KBNK/*.