Introduction to AI
Exercise at week 10

April 12th, 2020

Introduction Many directions of research in Artificial Intelligence proceed by drawing parallels with human cognitive ability, matching and, possibly, surpassing it. In this sense Artificial Intelligence and Cognitive Science are tightly knit together. The research into cognitive aspect of problem-solving often involves constructing a *cognitive model*, which can be viewed as an AI engine. In such cases however, the priority of efficient design, so common in AI engineering, are sacrificed for more faithful representation of human reasoning. Moreover, human behaviour often displays erratic and non-logical patterns, which both Cognitive Science and Artificial Intelligence often need to account for. In this exercise session we invite you to analyse the well-known game of Mastermind in terms of logical reasoning involved in the game. One of the take-away messages from this exercise, apart from practising the techniques introduced in the course, is that finding an efficient or correct solution to the problem at hand (so-called *normative perspective*), can be drastically different from human experience of attempting to solve a problem (so-called *descriptive perspective*).

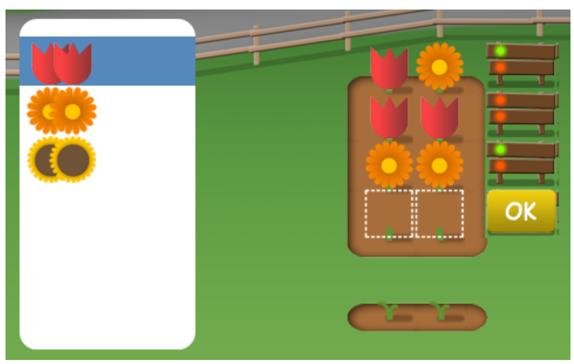
Mastermind is a code-breaking game for two players. It consists of a decoding board, code pegs of k colors, and feedback pegs of red and white (see Figure 1). There are two players, the

Figure 1: The modern Mastermind Game game with pegs was invented in 1970 by Mordecai Meirowitz, but the game resembles an earlier pen and paper game called Bulls and Cows. The figure shows the board version of Mastermind game as it is known today (source: Wikipedia).

code-maker, who chooses a secret pattern of ℓ code pegs (color duplicates are allowed), and the code-breaker, who guesses the pattern in a given n rounds. Each round consists of the code-breaker making a guess by placing a row of ℓ code pegs, and of the code-maker providing the feedback: a red peg for each code peg of correct color and position, and a white peg for each peg of correct color but wrong position. Guesses and feedbacks continue to alternate until either the code-breaker guesses correctly, or n incorrect guesses have been made. The code-breaker wins if she obtains the solution within n rounds; the code-maker wins otherwise.

Mastermind is an *inductive inquiry* game that involves *trials of experimentation and evaluation.* The game has been used to investigate the acquisition of complex skills and strategies in the domain of reasoning about others [6]. Existing mathematical results on Mastermind focus on finding strategies that allow winning the game in the smallest number of rounds [see 2, 3, 4, 5]. The deductive reasoning processes involved in Mastermind has been studied in [1]. I also upload the **paper and slides of my lecture in Discrete Mathematics Course 2018**, for those interested in how to AI engine designed to solve the game can be used to draw predictions and explanations of human behaviour.

Ex. 1 In this exercise you are invited to consider a simpler version of the Mastermind game, called Deductive Mastermind, proposed in [1]. Figure shows an example of a Deductive Mastermind game. It consists of a decoding board and the domain of flowers to choose from while constructing the solution. The goal of the game is to guess the correct sequence of flowers on the basis of the clues, consisting of earlier conjecture with feedback given on the decoding board. Thus each row of flowers forms a conjecture that is accompanied by a feedback on the small board on the right side: one green dot for each flower of the correct color and position, one orange dot for each flower of correct color but in a wrong position, and one red dot for each flower that does not appear in the correct secret sequence at all.



(a) What is the solution to this game? Give an appropriate sequence of flowers.

Answer: Sunflower followed by a daisy.

(b) Show, using the resolution technique, that your solution indeed follows from the premises. Hint: Translate each pair (conjecture, feedback) and your solution into propositional logic formulas, using propositions such as s_1 := 'sunflower in the first position'. Convert each of the resulting formulas into their CNF-form. Resolve.

Answer: First assign propositional symbols to each atomic fact, for instance: s_1 for the sunflower in the first position, t_2 for the tulip in the second position, etc. We take d for the orange daisy.

Our knowledge base must contain the basic assumptions of the game (background knowledge): (a) there must be one and (b) can only be one flower in a given position.

(a1) $(t_1 \vee s_1 \vee d_1)$

(a2) $(t_2 \vee s_2 \vee d_2)$

(b1) $(\neg t_1 \vee \neg s_1) \wedge (\neg s_1 \vee \neg d_1) \wedge (\neg d_1 \vee \neg t_1)$

(b2) $(\neg t_2 \vee \neg s_2) \wedge (\neg s_2 \vee \neg d_2) \wedge (\neg d_2 \vee \neg t_2)$

Then we formalise premises given in the game:

1. $(t_1 \wedge \neg d_2) \vee (d_2 \wedge \neg t_1) \equiv (t_1 \vee d_2) \wedge (\neg d_2 \vee \neg t_1)$
2. $\neg t_1 \wedge \neg t_2$
3. $(d_1 \wedge \neg d_2) \vee (d_2 \wedge \neg d_1) \equiv (d_1 \vee d_2) \wedge (\neg d_2 \vee \neg d_1)$

The conclusion is:

4. $(s_1 \wedge d_2)$

Our KB is now consists now of the formulas in (a), (b) and (1-3). In other to check if $KB \models (4)$, we add it's negation, (4'): $\neg(s_1 \wedge d_2)$, to KB. Was your answer entailed by the premises?

Next, we transform the formulas in KB into their respective CNFs, obtaining the following set of clauses:

(a1) $t_1 \vee s_1 \vee d_1$

(a2) $t_2 \vee s_2 \vee d_2$

(b11) $\neg t_1 \vee \neg s_1$

(b12) $\neg s_1 \vee \neg d_1$

(b13) $\neg d_1 \vee \neg t_1$

(b21) $\neg t_2 \vee \neg s_2$

(b22) $\neg s_2 \vee \neg d_2$

(b23) $\neg d_2 \vee \neg t_2$

1.1 $t_1 \vee d_2$

1.2 $\neg d_2 \vee \neg t_1$

2.1 $\neg t_1$

2.2 $\neg t_2$

3.1 $d_1 \vee d_2$

3.2 $\neg d_2 \vee \neg d_1$

4'. $\neg s_1 \vee \neg d_2$

One possible resolution sequence is:

$$r(r(r(2.1, 1.1), 4'), r(r(r(2.1, 1.1), 3.2), r(2.1, a1)))$$

(c) [Open Question] Reflect on how the resolution-based method for solving the problem differed from your initial way of solving the problem. What methods would you use to implement your introspective method of solving the problem?

(d) [*] Describe a general procedure for translating any pair $(conjecture, feedback)$ for an arbitrary number of flowers k and arbitrary number of positions ℓ in the code. (Remember about the orange feedback, absent in the previous exercise!)

Answer: Each DMM game consists of a sequence of conjectures.

A *conjecture* of length ℓ (consisting of ℓ pins) over k colors is any sequence given by a total assignment, $h : \{1, \ldots, \ell\} \to \{c_1, \ldots, c_k\}$. The *goal sequence* is a distinguished conjecture, $goal : \{1, \ldots, \ell\} \to \{c_1, \ldots, c_k\}$.

Every non-goal conjecture is accompanied by a feedback that indicates how similar h is to the given goal assignment. The three feedback colors, green, orange, and red will be represented by letters g, o, and r.

Let h be a conjecture and let *goal* be the goal sequence, both of length ℓ over k colors. The *feedback f for h with respect to goal* is a sequence

$$\overbrace{g \ldots g}^{a}\ \overbrace{o \ldots o}^{b}\ \overbrace{r \ldots r}^{c} = g^a o^b r^c,$$

where $a, b, c \in \{0, 1, 2, 3, \ldots\}$ and $a + b + c = \ell$. The feedback consists of:

- exactly one g for each $i \in G$, where $G = \{i \in \{1, \ldots, \ell\} \mid h(i) = goal(i)\}$.
- exactly one o for every $i \in O$, where

  $O = \{i \in \{1, \ldots, \ell\}\backslash G \mid \text{there is a } j \in \{1, \ldots, \ell\}\backslash G, \text{ s. t. } i \neq j \text{ and } h(i) = goal(j)\}.$
- exactly one r for every $i \in \{1, \ldots, \ell\}\backslash(G\cup O)$.

As literals of our Boolean formulae we take $h(i) = goal(j)$, where $i, j \in \{1, \ldots \ell\}$. They can be viewed as propositional variables $p_{i,j}$, for $i, j \in \{1, \ldots \ell\}$. With respect to sets G, O, and R that induce a partition of $\{1, \ldots, \ell\}$, we define $\varphi^g_G, \varphi^o_{G,O}, \varphi^r_{G,O}$, the propositional formulae that correspond to different parts of the feedback, in the following way:

- $\varphi^g_G := \bigwedge_{i\in G} h(i){=}goal(i) \wedge \bigwedge_{j\in\{1,\ldots,\ell\}\backslash G} h(j){\neq}goal(j)$,
- $\varphi^o_{G,O} := \bigwedge_{i\in O}(\bigvee_{j\in\{1,\ldots,\ell\}\backslash G, i\neq j} h(i) = goal(j))$,
- $\varphi^r_{G,O} := \bigwedge_{i\in\{1,\ldots,\ell\}\backslash(G\cup O), j\in\{1,\ldots\ell\}\backslash G, i\neq j} h(i) \neq goal(j)$.

Observe that there will be as many substitutions of each of the above schemes of formulae, as there are ways to choose the corresponding sets G and O. To fix the domain of those possibilities we set $\mathbb{G} := \{G | G \subseteq \{1, \ldots, \ell\} \wedge card(G){=}a\}$, and, if $G \subseteq \{1, \ldots, \ell\}$, then $\mathbb{O}^G = \{O | O \subseteq \{1, \ldots, \ell\}\backslash G \wedge card(O){=}b\}$. Finally, we can set $Bt(h, f)$, the Boolean translation of (h, f), to be given by:

$$Bt(h, f) := \bigvee_{G\in\mathbb{G}} (\varphi^g_G \wedge \bigvee_{O\in\mathbb{O}^G} (\varphi^o_{G,O} \wedge \varphi^r_{G,O})).$$

References

[1] Nina Gierasimczuk, Han L. J. van der Maas, and Maartje E. J. Raijmakers. An analytic tableaux model for deductive mastermind empirically tested with a massively used online learning system. *Journal of Logic, Language and Information*, 22(3):297–314, Jul 2013.

[2] R. W. Irving. Towards an optimum Mastermind strategy. *Journal of Recreational Mathematics*, 11:81–87, 1978.

[3] Donald E. Knuth. The computer as master mind. *Journal of Recreational Mathematics*, 9(1):1–6, 1977.

[4] Barteld Kooi. Yet another Mastermind strategy. *ICGA Journal*, 28(1):13–20, 2005.

[5] Mami Koyama and Tony Lai. An optimal Mastermind strategy. *Journal of Recreational Mathematics*, 25:251–256, 1993.

[6] R. Verbrugge and L. Mol. Learning to apply theory of mind. *Journal of Logic, Language and Information*, 17(4):489–511, 2008.