Towards a Formalization of Mental Model Reasoning for Syllogistic Fragments

Yutaro Sugimoto[1], Yuri Sato[2] and Shigeyuki Nakayama[1]

[1] Department of Philosophy, Keio University, Tokyo, Japan
[2] Interfaculty Initiative in Information Studies, The University of Tokyo, Japan
{sugimoto,nakayama}@abelard.flet.keio.ac.jp,sato@iii.u-tokyo.ac.jp

Abstract. In this study, Johnson-Laird and his colleagues' mental model reasoning is formally analyzed as a non-sentential reasoning. Based on the recent developments in implementations of mental model theory, we formulate a mental model reasoning for syllogistic fragments in a way satisfying the requirement of formal specification such as mental model definition.

1 Introduction

Recently, non-sentential or diagrammatic reasoning has been the subject of logical formalization, where diagrammatic reasoning is formalized in the same way as sentential reasoning is formalized in modern logic (e.g., [11]). In line with the formal studies of diagrammatic logic, we present a formalization of *mental model* reasoning, which was introduced by [6], as a cognitive system of reasoning based on non-sentential forms.

The mental model theory has been about cognitive-psychological theory, providing predictions of human performances and explanations of cognitive processes. Meanwhile, the theory has been attracted attention from various research fields including AI, logic, and philosophy beyond the original field (e.g., [2, 5, 8]) considering it can be taken as an applied theory based on the mathematical and logical notions such as models and semantics. It has been discussed not only empirical plausibility but also a formal specification of mental model reasoning.

The problem we focus on is that the definition of "mental model" is not provided properly within the explanations of mental model theory. It is a key to understand the full system and a step to give formal specifications of the theory. Recently, Johnson-Laird and his colleagues' several implementation works were made public[1], revealing the detailed procedures of the theory. However formal specifications or definitions requested here are still not included in their programs. An appropriate way to address the problem is to formulate the theory in accordance with their programs satisfying the requirements of the formal specification such as mental model definition. Our view is consistent with the seminal study in [1], who took the first step towards formalization of mental model reasoning while presenting a computer programs of it.

The theory was originally formulated for categorical syllogisms [6], therefore we begin our formalization project in the domain of syllogisms only. Particularly, we focus on the more recent version in [3] and the corresponding computer program [9].

[1] See their laboratory's webpage: http://mentalmodels.princeton.edu/programs

Before the formal work, we provide a brief overview of mental model theory with its illustrations of solving processes of syllogistic reasoning. The basic idea underlying the mental model theory is that people interpret sentences by constructing mental models corresponding to situations and make inferences by constructing counter-models. Mental models consist of a finite number of tokens, denoting the properties of individuals. For example, the sentence, "*All A are B*," has a model illustrated on the leftmost side of Fig.1, where each row represents an individual. Here, a row consisting of two

[a] b	c −b	[a] b	[a] b c
[a] b	c −b	[a] b	[a] b c
	b	−b c	−b c
	b	−b c	−b c
All A are B	*Some C are not B*	*Integrated model*	*Alternative model*
1st premise	*2nd premise*		

Fig. 1: Solving processes in mental model theory for a syllogistic task.

tokens, a and b, refers to an individual which is *A* and *B*. Furthermore, the tokens with square brackets, [a], express that the set containing them is exhaustively represented by these tokens and that no new tokens can be added to it. By contrast, a sequence of tokens without square brackets can be extended with new tokens so that an alternative model is constructed. However, such an alternative model is not taken since *parsimonious descriptions* are postulated to be preferred (chap. 9 of [7]). In a similar way, the sentence "*Some C are not B*" has a model illustrated on the second from the left of Fig.1. Here, a row having a single token, b, refers to an individual which is *B* but not *C*. Furthermore, the same thing can be also represented by the use of the device of "–" denoting negation. A row consisting of two tokens, c and –b, refers to an individual which is *C* but not *B*.

The right side of Fig. 1 shows a model integration process with these two premises. In this process, the two models in the left side of Fig.1 are integrated into a single model by identifying the tokens of set *B*. After the integration process, a searching process for counterexamples is performed, and alternative models are constructed. In this case of Fig.1, an alternative models is constructed from the integrated model by adding new tokens (i.e., token c). Since each tokens of set *A* are corresponding to tokens of set *C*, one of tentative conclusions "*Some A are not C*" is refuted. Hence, this tentative conclusion can be considered a *default assumption*, i.e., it can be specified as a conclusion by default and it can be revised later if necessary (chap. 9 of [7]). Instead, by observing that some tokens of set *C* are disjoint from the tokens of set *A*, one can extract a valid conclusion "*Some C are not A*" from the alternative model.

In the next section, we provide a formalization of mental model theory including the features: *parsimonious descriptions* and *default assumption*. We note here that our work does not intend to provide a normative and sophisticated version of mental model theory. Hence our work is not in line with the stance as taken in [4, 5], where the features above, postulated in mental model theory, are less focused.

2 A Mental Model Reasoning System

We provide a formalization for a mental model (syllogistic) reasoning system. Since the prototype program [9] is fully implemented by Common Lisp, it lacks static type information [12] and mental models are not defined explicitly. In order to treat the system formally, types serve significant role. Firstly, we describe the system as a finite state transition machine and provide type information to main procedures. Fig.4 shows the transitions from one state (model) to another by following processes: (1) constructing mental models of premises, (2) integrating premise models into an initial model, (3) drawing a tentative conclusion from an initial model, (4) constructing alternative model by falsification, and (5) responding a final conclusion.

2.1 Mental Model Construction

Though actual mental models are constructed implicitly in human cognition, the computational (syntactical) representation for mental models is constructed explicitly by the interpreter which converts semi-natural syllogistic language into computational representation for mental models. Accordingly, we first define a formal language for (semi-natural) syllogistic language by extended BNF following [9]. See Fig.2.

⟨*sentence*⟩ ::= ⟨*np*⟩ ⟨*pred*⟩
  | ⟨*np*⟩ ⟨*negpred*⟩
  | ⟨*neg-np*⟩ ⟨*pred*⟩
⟨*np*⟩ ::= ⟨*quant*⟩ ⟨*term*⟩
⟨*neg-np*⟩ ::= ⟨*neg-quant*⟩ ⟨*term*⟩
⟨*pred*⟩ ::= ⟨*cop*⟩ ⟨*term*⟩
⟨*negpred*⟩ ::= ⟨*cop*⟩ ⟨*neg*⟩ ⟨*term*⟩
⟨*term*⟩ ::= A | B | C
⟨*quant*⟩ ::= All | Some
⟨*neg-quant*⟩ ::= No
⟨*neg*⟩ ::= not
⟨*cop*⟩ ::= are

Fig. 2: Grammar for syllogistic language

⟨*token*⟩ ::= ⟨*atom*⟩
  | ⟨*lsqbracket*⟩ ⟨*atom*⟩ ⟨*rsqbracket*⟩
  | ⟨*neg*⟩ ⟨*atom*⟩
  | ⟨*nil*⟩
⟨*lsqbracket*⟩ ::= [
⟨*rsqbracket*⟩ ::=]
⟨*atom*⟩ ::= a | b | c
⟨*neg*⟩ ::= -
⟨*nil*⟩ ::=

Fig. 3: Grammar for mental model tokens

Next we give a definition for mental model units for syllogistic reasoning as follows:
A *mental model* is a *class of models*[2] s.t. $m \times n$ matrix of *tokens* where $m \geq 2$ and $3 \geq n \geq 1$. A *row* or an *individual* of a mental model is a finite array of tokens (*model*) where each atoms occur at most once. A *column* or a (*property*) of a mental model is a finite array of tokens where tokens contain any different atoms cannot co-occur. If square bracketed tokens occur in a column, only negative atoms can be added. Fig. 3 is the vocabulary and grammar for mental model *token*s. Since the detail of language translation is not our current concern, we do not give a specification for the language interpreter[3]. Alternatively, we give examples of translations. Let X,Y denote terms A, B, C. The four types of syllogistic sentences can be translated to mental models as follows:

All X are Y	Some X are Y	No X are Y	Some X are not Y
$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
$[x]\ y$ $[x]\ y$	$x\ y$ x $\quad y$	$[x]\ -y$ $[x]\ -y$ $\quad [y]$ $\quad [y]$	$x\ -y$ $x\ -y$ $\quad y$ $\quad y$

2 For a treatment of a mental model as a class of models, see [2].

3 For the detail of typical formal language transformation processes, see e.g. [10].

2.2 Integrating Premises into Initial Model

We give a description for the integration process of premises into an initial model via mid-term tokens (Fig.5). The integration process can be considered nearly as having functional type $f : P \to P \to M$ (P is a type of premisses: P_1, P_2).

Reordering and Switching Since syllogisms have several "figures" according to the order of premises and term arrangements, the actual integration procedure should occur after reordering terms and switching premises as preprocesses. This preprocess has the following four patterns:

(1) If the term order of P_1 is AB and P_2 is BC, nothing happens.
(2) If the term order of P_1 is BA and P_2 is CB, starts with P_2.
(3) If the term order of P_1 is AB and P_2 is CB, swaps second model round and adds it.
(4) If the term order of P_1 is BA and P_2 is BC, swaps first model then adds second model.

Finding a middle atom The procedure of *finding a middle atom: a* can be considered as having a functional type $g : P \to P \to a$. The actual implementation for this is a similar to set intersection operation for the affirmative tokens (tokens which do not contain negatives). For example, when two premises are as Fig.1, $\{a,a,b,b\} \cap \{c,c,b,b\} = b$.

Match The procedure of matching premises P_1, P_2, and middle atom a could have functional type $rec : P \to P \to a \to M$. This recursive procedure calls `join` as sub procedure to join the premises to an integrated model.

Join This recursive procedure takes a mid atom and two individuals, and joins two individuals together setting new mid to exhausted if one or other was exhausted in first individual or second individual. This procedure could have a recursive functional type: $rec : a \to Indiv \to Indiv \to Indiv$.

2.3 Drawing a Conclusion from a Model

Drawing a conclusion (Fig.6) is a procedure which takes an integrated (initial) model and dispatches whether it contains negative token or not. It then dispatches further based on the predicates (all-isa, some-isa, no-isa, and some-not-isa) and returns *corresponding answers*.[4] If the predicates return #f, then it returns "No Valid Conclusion." The followings are sub procedures of `conclude`:

all-isa takes a model which has end terms X, Y and returns the answer "All X are Y" iff all subjects are objects in individuals in model. This has a functional type: *all-isa* : $M \to A$. For example, if a model M : $\begin{smallmatrix} [a] & b & c \\ [a] & b & c \end{smallmatrix}$ is given, where end terms are A and C, then returns the answer "All A are C."

some-isa takes a model which has end terms X, Y and returns the answer "Some X are Y" iff at least one individual in model contains positive occurrences of both subject and object atoms. This has a functional type: *some-isa* : $M \to A$. For example, if a model : $\begin{smallmatrix} [a] & [b] & c \\ [a] & [b] & \\ & & c \end{smallmatrix}$ is given when end terms are A and C then returns the answer "Some A are C".

[4] Notice: since possible conclusions have term order: Subj-Obj and Obj-Subj, `conclude` is executed twice respectively. For simplicity, we omit the second execution of `conclude`.

no-isa takes a model which has end terms X, Y and returns “No X are Y” iff no subject end term is object end term in any individuals in model. This has a functional type: *no-isa* : $M \to A$. For example, if a model M : $\begin{array}{ccc} [a] & -b & \\ [a] & -b & \\ & [b] & [c] \\ & [b] & [c] \end{array}$ is given when end terms are A and C then returns the answer “No A are C.”

some-not-isa takes a model which has end terms X, Y and returns “Some X are not Y” iff at least one subject occurs in individuals without object. This has a functional type: *some-not-isa*: $M \to A$. For example, if a model M : $\begin{array}{ccc} [a] & b & c \\ [a] & b & c \\ & -b & c \\ & -b & c \end{array}$ is given when end terms are A and C then returns the answer “Some A are not C.”

2.4 Constructing Alternative Model

Once the mental model theory constructs an initial model and draws a tentative conclusion, the theory, according to the rules, tries to construct an alternative model in order to refute the conclusion (i.e., default assumption). The process of falsification (Fig.7) takes a model and dispatches whether it contains negative token or not. Then based on the predicates (breaks, add-affirmative, moves, and add-negative) it tries to modify the model. If succeeded, returns an alternative model and call `conclude` again. If failed, the recursive call of this procedure terminates. Here are main constructs of `falsify`:

breaks has a functional type: *breaks* : $M_1 \to M_2$. *breaks* finds an individual containing two end terms with non-exhaustive mid terms, divides it into two, then returns new (broken) model or returns *nil*. For example, if M_1 is $a\ b\ c$, then *breaks*: $a\ b\ c \;\to\; \begin{array}{ccc} a & b & \\ & b & c \end{array}$.

add-affirmative has a functional type: $add^+ : M_1 \to M_2$. If add^+ succeeds, then it returns a new model $M_2$with added item (added model), else it returns *nil* if conclusion is not A-type (“All X are Y”) or if there is no addable subject item.

For example, if M_1 is $\begin{array}{ccc} [a] & [b] & c \\ [a] & [b] & c \end{array}$, then add^+: $\begin{array}{ccc} [a] & [b] & c \\ [a] & [b] & c \end{array} \to \begin{array}{ccc} [a] & [b] & c \\ [a] & [b] & c \\ & & c \end{array}$.

moves has a functional type: *moves* : $M_1 \to M_2$. If there are exhausted end items not connected to other end items or their negs (i.e E-type (“No X are Y”) conclusion), and if the other end items are exhausted or O-type (“Some X are not Y”) conclusion, then it joins them. Otherwise joins one of each and returns *nil* if the first end item cannot be moved even if a second one can be.

E.g., if M_1 is $\begin{array}{ccc} [a] & -b & \\ [a] & -b & \\ & [b] & -c \\ & [b] & -c \\ & & [c] \\ & & [c] \end{array}$, then *moves*: $\begin{array}{ccc} [a] & -b & \\ [a] & -b & \\ & [b] & -c \\ & [b] & -c \\ & & [c] \\ & & [c] \end{array} \to \begin{array}{ccc} [a] & -b & [c] \\ [a] & -b & [c] \\ & [b] & -c \\ & [b] & -c \end{array}$. When this procedure is called by `falsify`, `neg-braking` (similar procedure to `breaks`) is also called as an argument.

add-negative has functional type: $add^- : M_1 \to M_2$. It returns a new model with added item (add-neged model), or returns *nil* if conclusion is not O-type or if there is no addable subject item. E.g., if M_1 is $\begin{array}{ccc} [a] & b & \\ [a] & b & \\ & -b & c \\ & -b & c \end{array}$, then add^-: $\begin{array}{ccc} [a] & b & \\ [a] & b & \\ & -b & c \\ & -b & c \end{array} \to \begin{array}{ccc} [a] & b & c \\ [a] & b & c \\ & -b & c \\ & -b & c \end{array}$.

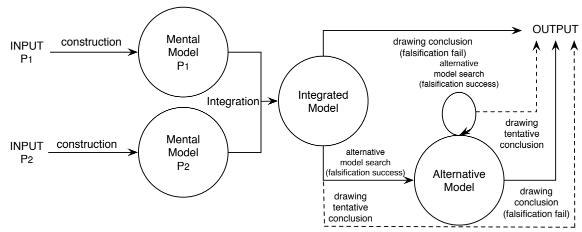

Fig. 4: Finite state transition machine diagram for syllogisms

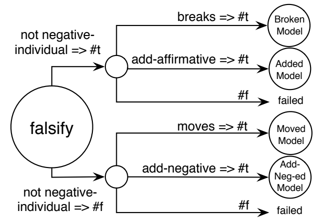

Fig. 7: Falsification process

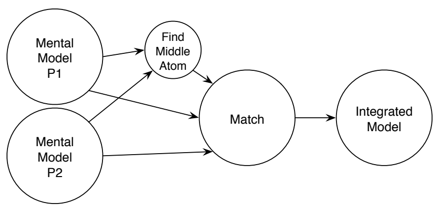

Fig. 5: Integration process

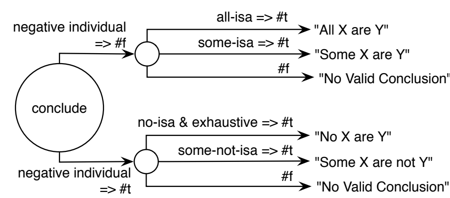

Fig. 6: Drawing conclusion process

Acknowledgements. This study was supported by MEXT-Supported Program for the Strategic Research Foundation at Private Universities (2012-2014) and Grant-in-Aid for JSPS Fellows (25·2291).

References

1. Bara, B.G., Bucciarelli, M., & Lombardo, V. (2001). Model theory of deduction: A unified computational approach. *Cognitive Science*, 25, 839–901.
2. Barwise, J. (1993). Everyday reasoning and logical inference. *Behavioral and Brain Sciences*, 16, 337–338.
3. Bucciarelli, M. & Johnson-Laird, P.N. (1999). Strategies in syllogistic reasoning. *Cognitive Science*. 23, 247–303.
4. Clark, M. (2010). *Cognitive Illusions and the Lying Machine*. Ph.D Thesis, Rensselaer P.I.
5. Hintikka, J. (1987). Mental models, semantical games, and varieties of intelligence. In *Matters of Intelligence* (pp. 197–215), Dordrecht: D. Reidel.
6. Johnson-Laird, P.N. (1983). *Mental Models*. Cambridge, MA: Harvard University Press.
7. Johnson-Laird, P.N., & Byrne, R. (1991). *Deduction*. Hillsdale, NJ: Erlbaum.
8. Lowe, E.J. (1993). Rationality, deduction and mental models. In *Rationality* (pp. 211–230), Taylor & Frances/Routledge.
9. Mental Models and Reasoning Lab. Syllogistic reasoning code [Computer program]. Retrieved Oct.10, 2012, from `http://mentalmodels.princeton.edu/programs/Syllog-Public.lisp`.
10. Mitchell, J.C. (1996). *Foundations for Programming Languages*. Cambridge, MIT Press.
11. Shin, S.-J.(1994). *The Logical Status of Diagrams*. New York: Cambridge University Press.
12. Steele, Jr. G.L. (1990). *Common LISP: The Language (2nd ed.)*. Newton, MA: Digital Press.