Redefining the Current Discourse Space Model as a Recursive Monadic architecture

Yoichiro Hasebe
Doshisha University, yhasebe@mail.doshisha.ac.jp

Keywords: current discourse space, monad, recursion, discourse, cognitive grammar, large language model, artificial intelligence, generative pre-trained transformer

This study aims to model the construction process of dialogic discourse. The goal is twofold. The first step is to redefine Langacker's (2001, 2012) notion of the current discourse space (CDS) as a recursive structure that incorporates the idea of "monads" from functional programming (FP). The second step is to simulate CDS as a monadic recursive structure in a computer program using a GPT-based text completion API.

Langacker (2001: 151) proposed considering linguistic structures as "instructions to modify the current discourse space in particular ways." In this view, every time a speech event occurs, CDS is updated in such a way that the entire discourse domain, which contains not only the concepts of the speaker, the hearer, and the subject matter but also the context and the shared knowledge, is passed on to subsequent speech events.

Following the argument of Hasebe (2021), this study maintains that CDS has a monadic structure and aims to redefine CDS as such. The concept of a monad is based on the category theory of mathematics and is widely used in the FP paradigm in computer science. A monad is often described as a "value wrapped in an environment" (e.g. Hutton 2016), and a structure that satisfies the conditions for being a monad is characterized by its ability to execute operations continuously and sequentially while maintaining and updating the structure given as the environment (Wadler 1995; Petricek 2018). The conditions are as follows: There is a procedure (*unit*, 1a) that wraps the target value *x* in the environment; there is a procedure (*map*, 1b) that "lifts" a function *f* to another function *f'* that deals with the value wrapped in an environment; and there is a procedure (*join*, 1c) that flattens a doubled layer of environments. It is shown that the process of updating CDS includes all these procedures and satisfies the requirements as a monad, and that CDS can be regarded as a type of "state monad" illustrated in Fig. 1.

(1) a. *unit* :: $a \rightarrow \boxed{a}$

b. *map* :: $(a \rightarrow b) \rightarrow (\boxed{a} \rightarrow \boxed{b})$

c. *join* :: $\boxed{\boxed{a}} \rightarrow \boxed{a}$

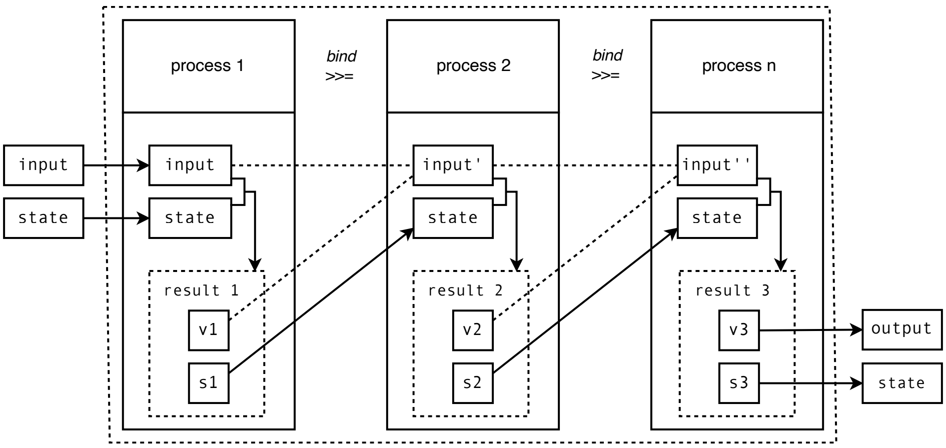

Fig. 1: State Monad

Redefining CDS as a monadic structure has practical advantages. For instance, the idea of CDS as a monadic structure can be used as a design pattern to implement a natural language computer interface like ChatGPT. Hasebe (2023) developed "Monadic Chat," a framework to provide an interactive interface to conduct a natural language conversation with large language models using the text completion API of OpenAI. This framework enables developers to easily create a chat-style AI application program. In addition, by providing an extra implementation code for two component structures (“accumulator” and “reducer”), the composition and function of “context” and “shared knowledge” within CDS can be freely configured. The latter functionality allows for a computational simulation of processes such as incremental context building (Harder 1996; Langacker 2008), stack-based focus/memory management (Chafe 1994), and compression/abstraction of concepts (Fauconnier & Turner 2000; Barsalou 2005). The basic architecture of the software, which is essentially a state monad, is illustrated in Fig. 2.

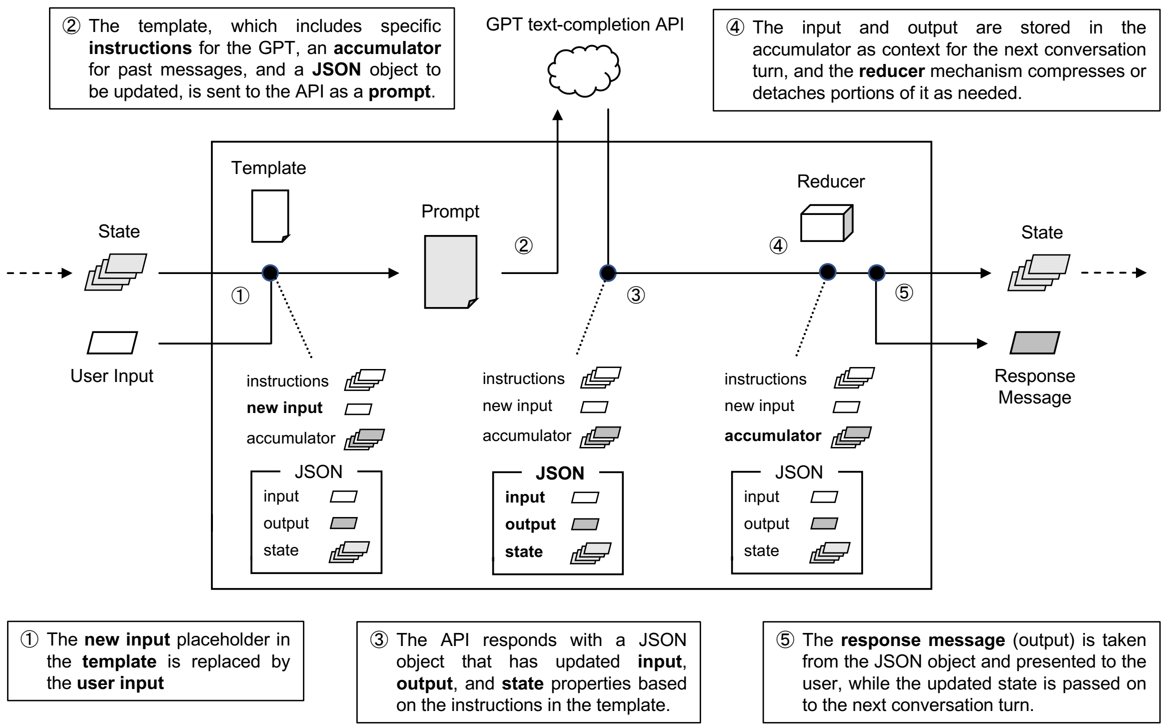

Fig. 2: Basic Architecture of Monadic Chat

References

Barsalou, Lawrence W. 2005. Abstraction as dynamic interpretation in perceptual symbol systems. In Lisa Gershkoff-Stowe & David H. Rakison (eds.), *Building Object Categories in Developmental Time*, 389–431. Mahwah, NJ: Lawrence Erlbaum.

Chafe, Wallace L. 1994. *Discourse, Consciousness, and Time: The Flow and Displacement of Conscious Experience in Speaking and Writing*. Chicago: University of Chicago Press.

Fauconnier, Gilles & Mark Turner. 2000. Compression and global insight. *Cognitive Linguistics* 11(3-4). 283–304.

Harder, Peter. 1996. *Functional Semantics: A Theory of Meaning, Structure and Tense in English*. Berlin: Mouton de Gruyter.

Hasebe, Yoichiro. 2021. *An Integrated Approach to Discourse Connectives as Grammatical Constructions*: Kyoto University PhD Dissertation.

Hasebe, Yoichiro. 2023. Monadic Chat: Framework for managing context with text completion API. In *Proceedings of the 29th Annual Meeting of the Association for Natural Language Processing*, 3138–3143. `https://www.anlp.jp/proceedings/annual_meeting/2023/pdf_dir/Q12-9.pdf`.

Hutton, Graham. 2016. *Programming in Haskell*. Cambridge: Cambridge University Press 2nd edn.

Langacker, Ronald W. 2001. Discourse in Cognitive Grammar. *Cognitive Linguistics* 12(2). 143–188.

Langacker, Ronald W. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford: Oxford University Press.

Langacker, Ronald W. 2012. Interactive cognition toward a unified structure. *International Journal of Cognitive Linguistics* 3(2). 95–125.

Petricek, Tomas. 2018. What we talk about when we talk about monads. *The Art, Science, and Engineering of Programming* 2(3). `http://programming-journal.org/2018/2/12`.

Wadler, Philip. 1995. Monads for functional programming. In Johan Jeuring & Erik Meijer (eds.), *Advanced Functional Programming*, 24–52. Berlin: Springer.