How to do it with LPS (Logic-based Production System)

Robert Kowalski[1], Fariba Sadri[1] and Miguel Calejo[2]

[1] Imperial College London, UK
[2] interprolog.com

Abstract. LPS is a logic and computer language in which computation performs actions, to make goals true, using beliefs about what is already true.

Keywords: LPS, logic programming, reactive rules, goals, beliefs.

LPS [6, 7] aims to bridge the gap between logical and imperative computer languages. The logical nature of LPS is reflected in its view of computation as performing actions to make **goals** true. Goals include both **reactive rules** of form *if antecedent then consequent* and **constraints**. The imperative nature of LPS is reflected in the use of reactive rules, to make *consequents* true whenever *antecedents* become true, and the use of constraints to prevent unwanted actions.

LPS includes **beliefs** as logic program clauses of the form *conclusion if conditions*. Clauses have an imperative interpretation as procedures, which decompose the problem of determining whether a *conclusion* is true or of making a *conclusion* true, to the sub-problem of determining or making the *conditions* true.

LPS also includes **destructive change of state**, to construct a model, to make goals true. Models in LPS are similar to possible worlds in modal logic. But models in LPS are classical, non-modal structures, with an explicit representation of time and events.

LPS shares the notion of computation as model-generation with the modal language MetaTem [2]. However, MetaTem uses frame axioms to construct possible worlds, and simulates logic programs by reactive rules. LPS shares the use of destructive change of state with Transaction Logic [3]. However, Transaction Logic uses a possible world semantics, and simulates reactive rules by logic programs.

LPS is a direct descendant of abductive logic programming (ALP) [5]. Abduction in ALP is used both to generate external event to explain past observations, and (as in LPS) to generate actions to achieve future states of affairs.

LPS can be viewed as a BDI (Belief, Desire, Intention) language, in which beliefs are logic programs, and desires are reactive-rules. LPS does not have a separate “intention” component. Intentions are replaced by search strategies for exploring the search space generated by goals and by beliefs that reduce goals to subgoals.

BDI languages are sometimes formalised in modal logics, but all practical implementations are procedural representations. In practical BDI languages, *plans* are used to represent both logic programs and reactive rules. This dual use of *plans* is one of the main reasons practical BDI languages do not have a logical semantics.

LPS is a programming, database and AI language; but its AI character has been deliberately scaled down, for practical purposes. LPS shares this unifying, but practical orientation with Prolog, Datalog, LogiQL from LogicBloX [1] and Yedalog at

Google [4]. But, while these languages are all based on logic programming, LPS is based on the use of destructive change of state to make goals true. Logic programs still play an important role, but one that is subordinate to goals.

LPS provides novel solutions to classic computing problems. For example, a program for bubble sort, driven by a reactive rule that swaps items at adjacent locations if they are out of order, will swap the items at all out-of-order adjacent locations simultaneously. A constraint prevents an item from being swapped with two different neighbours at the same time. In case there are several alternative ways to satisfy the goals, LPS chooses one of the alternatives and commits to it arbitrarily.

The same general strategy applies to the dining philosophers problem, specified by the goal that all philosophers must eventually dine. The solution is a logic program that defines dining as picking up two adjacent forks simultaneously, eating, and then putting down the two forks simultaneously. Logic programs also define the laws governing change of state, namely that putting down a fork initiates its availability, and picking up a fork terminates its availability. Finally, the action of picking up a fork is constrained, by specifying that a fork can be picked up only if it is available, and that two philosophers cannot pick up the same fork simultaneously.

LPS can also be viewed as a scaled-down model of human thinking. The representation of goals in LPS is supported by their similarity to condition-action rules in production systems, which are one of the most influential computational models of human thinking. The representation of beliefs as logic programs in LPS is supported both by psychological studies of human reasoning, such as [8, 10], and by normative models, such as [8, 9].

The online, open-source prototype of LPS is accessible from lps.doc.ac.uk.

References

1. Aref M. *et al* (2015). Design and implementation of the LogicBlox system. *ACM SIGMOD International Conference on Management of Data ACM,* 1371-1382.
2. Barringer, H. *et al*. (1996). *The imperative future: principles of executable temporal logic*. John Wiley & Sons, Inc.
3. Bonner, A.J. and Kifer, M. (1994). An overview of transaction logic. *Theoretical Computer Science, 133(2),* 205-265.
4. Chin B, *et al* (2015) Yedalog: Exploring knowledge at scale. *Leibniz International Proceedings in Informatics (Vol. 32)*. Dagstuhl.
5. Kakas, A., Kowalski, R. and Toni, F. (1998) The Role of Logic Programming in Abduction. *Handbook of Logic in Artificial Intelligence and Programming,* Oxford University Press, 235-324.
6. Kowalski, R. and Sadri, F. (2015). Reactive Computing as Model Generation. *New Generation Computing, 33, 1,* 33-67.
7. Kowalski, R. and Sadri, F. (2016). Programming in logic without logic programming. TPLP, 16, 269-295.
8. Kowalski, R. (2011) *Computational Logic and Human Thinking – How to be Artificially Intelligent.* C.U.P. Also: https://www.doc.ic.ac.uk/~rak/papers/newbook.pdf
9. Pereira, L. M. and Saptawijaya, A. (2016). *Programming Machine Ethics*. Springer.
10. Stenning, K. and Van Lambalgen, M. (2012). *Human reasoning and cognitive science*. MIT Press.