asprune: Solving the Path Counting Problem with Answer Set Programming

Irumi Sugimori, Makoto Nagayama, Manaka Tomita, Shuji Kosuge, and Mutsunori Banbara[2][0000−0002−5388−727X]

[1] School of Informatics, Nagoya University
{irumi.sugimori,makoto.nagayama,manaka.tomita}@nagoya-u.jp
[2] Graduate School of Informatics, Nagoya University
{kosugeshuji003,banbara}@nagoya-u.jp

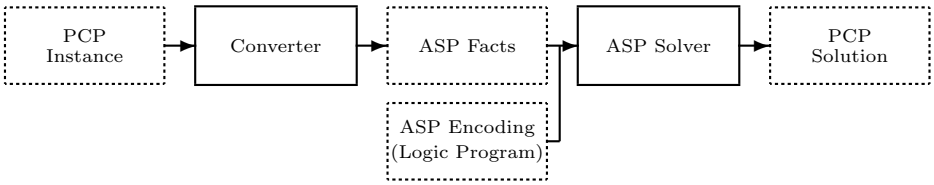

Fig. 1. Architecture of the *asprune* solver

The first international competition on graph counting algorithms (ICGCA 2023 [1]) is focusing on *the Path Counting Problem* (PCP). The task of PCP is to count all paths on a graph, subject to a given set of constraints. Theoretically, this problem is known to be #P-complete. Answer Set Programming (ASP; [1, 3, 4]) is a declarative programming paradigm, combining a rich modeling language with high performance solving capacities. ASP is well suited for modeling combinatorial (optimization) problems, and has been successfully applied in diverse areas of artificial intelligence and other fields over the last decade.We develop an ASP-based solver *asprune* for the ICGCA 2023 competition. The architecture of *asprune* is shown in Fig. 1. The *asprune* solver reads a PCP instance expressed in an extended DIMACS format and converts it into ASP facts. In turn, these facts are combined with an ASP encoding for PCP solving, which can subsequently be solved by any off-the-shelf ASP solvers, in our case *clingo* [2]. The declarative approach of ASP has several advantages. First, combinatorial problems can be modeled as first-order logic programs and then solved by general-purpose ASP solvers, rather than dedicated implementations. Second, ASP allows for easy extensions to additional constraints. And finally, *clingo* allows for enumerating solutions of the problems, as well as counting them.

[1] https://afsa.jp/icgca/

References

1. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
2. Gebser, M., Kaminski, R., Kaufmann, B., Lindauer, M., Ostrowski, M., Romero, J., Schaub, T., Thiele, S.: Potassco User Guide. University of Potsdam, second edition edn. (2015), `http://potassco.org`
3. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, K. (eds.) Proceedings of the Fifth International Conference and Symposium of Logic Programming (ICLP'88). pp. 1070–1080. MIT Press (1988)
4. Niemelä, I.: Logic programs with stable model semantics as a constraint programming paradigm. Annals of Mathematics and Artificial Intelligence **25**(3-4), 241–273 (1999)