Predicate invention and reuse

Andrei Diaconu [1] , **Andrew Cropper**[1] , **Rolf Morel** [1]
[1]University of Oxford
{emails}@ox.ac.uk

Abstract

Abstract here.

1 Introduction

Inductive programming (IP) [Gulwani *et al.*, 2015] - also known as program synthesis or example based learning - is a field that lies at the intersection of several computer science topics (machine learning, artificial intelligence, algorithm design) and is a form of automatic programming. IP tries to find a target program starting with an incomplete specification and tries to generalize that into a program. Usually, that incomplete specification is represented by examples, so we can informally define inductive programming to be the process of creating programs from examples using a limited amount of background information. We give an example of what an IP system might produce, given a task:

2 Framework

2.1 Invention

2.2 Reuse

3 Invention and reuse in ILP systems

3.1 Metagol

3.2 ILASP

3.3 Aleph

3.4 Popper

4 Experiments

4.1 Problems where reuse helps

4.2 Problems where reuse does not help

5 Related work

6 Conclusions and further work

References

[Gulwani *et al.*, 2015] Sumit Gulwani, José Hernández-Orallo, Emanuel Kitzelmann, Stephen H. Muggleton, Ute Schmid, and Benjamin G. Zorn. Inductive programming meets the real world. *Commun. ACM*, 58(11):90–99, 2015.