J. Roland Olsson - How to Invent Functions (1999)

Tom Rochette <tom.rochette@coreteks.org>

August 30, 2025 — 861fb9d0

0.1 Context

0.2 Learned in this study

0.3 Things to explore

- Using EP theory, how does one program "mutate" into another?

1 Overview

- Adenine, cytosine, guanine and thymine are the four basic building blocks of programs (binary blocks)
- Short interspersed nuclear elements (SINEs) = small helper functions
- Abstraction: encapsulating a small program within a function, provided the function accepts an argument
- During large scale program evolution, abstraction is essential for at least the following two reasons:
  1. The user of an automatic programming system should not be required to define all needed help functions. Instead, the user should define a small number of primitives whereas the system automatically constructs a possibly large number of help functions.
  2. The system can construct a help function exactly where it is needed and avoid having a too large scope for the function.
- A form of scope restriction actually seems to exist in DNA since repeats often occur in localized regions
  - For example, the clustering of multiple copies of genes encoding ribosomal RNA in humans
- ADATE uses a so-called cost limit l that says how many children programs are to be produced from a given parent program. An abstraction is assigned a cost c which indicates that l/c programs are to be based on the abstraction
- Discriminate against bodies containing if-tests that do not depend on any parameter

2 See also

3 References

- Olsson, J. Roland. "How to invent functions." European Conference on Genetic Programming. Springer Berlin Heidelberg, 1999.