What is a (Logic) Program

March 19, 2020

Abstract

1 Introduction

the chapter moves roughly around the divide "constructive vs. classical" semantics of programs.

(JB) proposal to starts from the Strachey-Scott view / denotational semantics as a way to ground the different definitons of computation. semantics as a way of distinguishing between the two paradigms (connection to languages group in the volume)

CH isomorphims: -) Introduction to pure lambda-calculus as an abstract functional programming language; -) Possibility of defining "pathological" programs in pure lambda-calculus (like looping programs or never ending ones); -) Use of a type system based on the logical implicational fragment in order to get always terminating programs (i.e. total functions). Correspondence between simple typed lambda-calculus and the implicational fragment in natural deduction; -) Extension of the Curry-Howard correspondence to first-order logic and to arithmetic. Use of new functional programming instruction in order to decorate the rules of natural deduction. -) Introduction of dependent type systems and extension of Curry-Howard correspondence to second order logic. This could be interesting to show that some of the "pathological" programs of pure lambda-calculus can now be typed (in the Girard-Reynolds system F), e.g. the program (x x).

operational vs. denotational interpretation
typing vs. realizability interpretation
natural deduction intuitionistic logic constructive tt
Edgar look at actors

2 philosophical questions

1. intensional vs extensional?: Is there a way to identify programs on an extensional way (i.e. by an equivalence relation) Or is the notion of program

a purely intensional one?

2. CTT?

3. conceptual order between computation and logic

4. invariants: what are the abstract conditions on a given information dynamics for meaning to emerge

5. pathological vs. correctness