Study Notes

Theory of Functional Programming

Emma Bach

March 27, 2025

Contents

Preface

These notes are not based on any particular lecture I attended. Instead, they were created during self-study of theoretical topics related to functional programming and the λ-calculus.

My primary source is the "Handbook of Logic in Computer Science", edited by Samson Abramsky, Dov Gabbay and Thomas Maibaum, in particular volume 2, which deals with computational structures [2]. Additional sources I found helpful include the video "What is plus times plus?" by YouTube channel 2Swap, which is a wonderful introduction to the λ-calculus [1].

Chapter 1

Term Rewriting Systems

1.1 Abstract Reduction Systems

Definition 1.1.1.

1. An **abstract reduction system** is a structure $\mathcal{A} = \langle A, \{\to_\alpha\}_{\alpha \in I} \rangle$, consisting of a set of terms A and a set of binary relations $\to_\alpha \in A^2$, called the **reduction relations** or the **rewrite relations**. As shorthand, the relation $\to_\alpha$ is occasionally written simply as α. An abstract reduction system with only one relation $\to$ is also known as a **replacement system** or a **transformation system**.

2. If $a \to_\alpha b$, then b is known as a **one step (α-) reduct** of a, meaning that b can be created by applying the reduction $\to_\alpha$ to a once.

Note that we let $\to_\alpha$ be any arbitrary relation, without additional demands of transitivity, reflexivity, or anything else of the sort.

Example 1.1.2. As an example, we could consider a term rewriting system that reduces sums of natural numbers. To write this system as an abstract reduction system, we can define A and $\to$ as follows:

- Let A be the smallest set of terms such that:
  - Every natural number $n \in \mathbb{N}$ is a term.
  - If t_1 and t_2 are terms, then $t_1 + t_2$ is a term.
- We define the following reduction relations $\to_0$ and $\to_1$:
  - If n and m are natural numbers and $k = n + m$, then $n + m \to_0 k$
  - If $t_1 \to_0 k$, then $t_1 + t_2 \to_1 k + t_2$

We could now apply these reduction rules to reduce a sum:

$$10 + 5 + 2 + 7 \to_1 15 + 2 + 7 \to_1 17 + 7 \to_0 24$$

The term $15 + 2 + 7$ is a one step 1-reduct of the term $10 + 5 + 2 + 7$, while the term $17 + 7$ is a one-step 1-reduct of $15 + 2 + 7$ and therefore a two step 1-reduct of $10 + 5 + 2 + 7$.

Definition 1.1.3. Given a reduction relation $\to_\alpha$, we are often interested in investigating new relations created by extending $\to_\alpha$. Some relations that are of interest to us include:

- The **transitive reflexive closure** $\twoheadrightarrow_\alpha$, defined as the smallest transitive relexive relation containing $\to_\alpha$. To give a more explicit definition, we have $a \twoheadrightarrow_\alpha b$ iff:
  1. $a \to_\alpha b$, or
  2. $a = b$, or
  3. $\exists n : \exists c_1, \ldots, c_n : a \to_\alpha c_1 \to_\alpha \ldots \to_\alpha c_n \to_\alpha b$

  In most of the literature, this closure would be denoted as $\to_\alpha^*$. However, the book uses $\twoheadrightarrow_\alpha$ because it makes transitive diagrams involving the relation more legible.

- The **convertibility relation** $=_\alpha$ generated by $\to_\alpha$, defined as the smallest equivalence relation containing $\to_\alpha$. Once again, to give a more explicit definition:

  We have $a =_\alpha b$ iff. $\exists u_1, \ldots, u_n \in A$ such that $a = u_1$, $b = u_n$, and for each u_i, we have either $u_i \to_\alpha u_{i+1}$ or $u_{i+1} \to_\alpha u_i$.

  Note that even though $\twoheadrightarrow_\alpha$ is the smallest transitive reflexive relation that contains $\to_\alpha$ and $=_\alpha$ is the smallest equivalence (transitive, reflexive, and symmetric) relation that contains $\to_\alpha$, $=_\alpha$ is **not** equivalent to the symmetric closure of $\twoheadrightarrow_\alpha$, as can be seen in Example 1.1.4.

Example 1.1.4. Let $\to_\alpha = \{(a, c), (b, c), (c, d)\}$. Then:

1. $\twoheadrightarrow_\alpha = \{(a, a), (b, b), (c, c), (d, d), (a, c), (a, d), (b, c), (b, d), (c, d)\}$
2. The symmetric closure of $\twoheadrightarrow_\alpha$ would be $\twoheadrightarrow^s_\alpha = \{(a, a), (b, b), (c, c), (d, d), (a, c), (c, a), (a, d), (d, a), (b, c), (c, b), (b, d), (d, b), (c, d), (d, c)\}$. Note that this relation is no longer transitive, since it includes (a, c) and (c, b), but not (a, b).
3. $=_\alpha$ would include every single pair of these elements.

In short, if $a \to_\alpha c \leftarrow_\alpha b$, then we have $a =_\alpha b$, but not necessarily $a \twoheadrightarrow^s_\alpha b$.

Bibliography

[1] 2swap. What is plus times plus? https://www.youtube.com/watch?v=RcVA8Nj6HEo. Accessed: Spring 2025.

[2] S. Ambramsky, Dov M. Gabbay, and T.S.E.Maibaum, editors. *Handbook of Logic in Computer Science, Volume 2: Background: Computational Structures.* Oxford Science Publications, 1992.