SNE – The Neurion editor

Hans Hüttel Michael Koldsgaard Thomas Munk

August 2021

1 Introduction

This document outlines the design decisions underlying SNE.

2 Handles and environments

Every value in the knowledge base is bound by a name, which we call it its *handle*. Handles are unique; the collection of values known is therefore an *environment* in the usual sense of program semantics.

The environment is represented as a database.

We distinguish between internal and external handles. Internal handles are used by the database and are necessarily unique. External handles are used by the programmer and may not be unique.

There are various approaches to representing internal handles. deBruijn indexes are examples of internal handles. Another possibility is to use sequences of atomic handles that denote nested scopes such that e.g. $a_3.a_7.a_2$ denotes the value with atomic handle a_2 found within the scope with atomic handle a_7, that is found within the scope with atomic handle a_3.

3 Edit actions

We distinguish between edit actions that modify values, actions that modify the environment and actions that modify annotations.

3.1 Editing values

The editor supports the following atomic edit actions for values.

- Introduce an AST
- Introduce AST nodes
- Delete AST nodes

These correspond to the atomic actions found in the editor calculus of [1]. Composite actions can be defined using the control structures in the editor calculus. An example is that deletion can be defined as substituing a by the hole [ ].

3.2 Editing annotations

Values can be annotated. Most notably, values can be annotated with types. Type annotations can be derived using type inference, but we can also explicitly type a value by declaring the type of its handle.

4 How edit actions modify the data

Håndtag skal have et tidsstempel – eller ”den ældste committede”/”nyeste committede”

En save-knap.

5 The implementation

(men kan det understøttes pænt i Elm?)

** Det gode ved den nuværende implementation: Få primitiver ** Det mindre gode: Mange specialiseringer (pga. manglende ad hoc-polymorfi)

6 Future work

Re-implement everything in NPL.

References

[1] Christian Godiksen, Thomas Herrmann, Hans Hüttel, Mikkel Korup Lauridsen, and Iman Owliaie. A type-safe structure editor calculus. In Sam Lindley and Torben Æ. Mogensen, editors, *Proceedings of the 2021 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation, PEPM@POPL 2021, Virtual Event, Denmark, January 18-19, 2021*, pages 1–13. ACM, 2021.