

Seamless, Correct, and Generic Programming over Serialised Data

Overview

- Introduction
- Data types
- Serialized representations
- Pointers and Buffers
- Serializing
- Results & Conclusion

Idris2

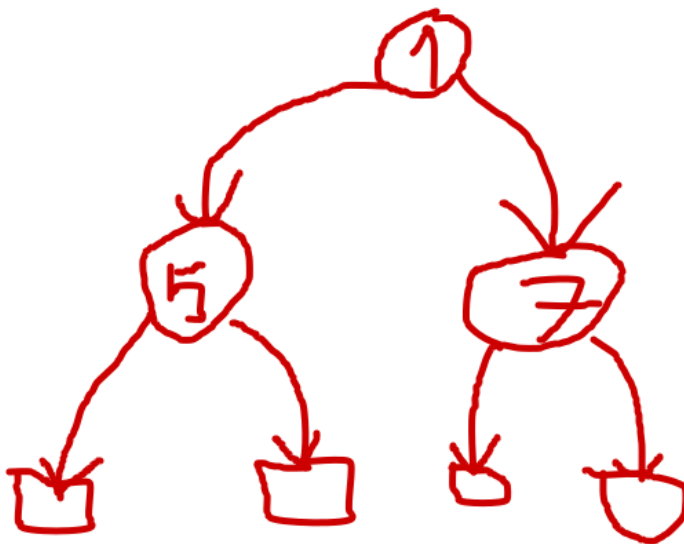
The code in this paper are written in Idris2.

- Dependent Types
- Totality

Dependent types

The type is influenced by the value. For example we can have the length of a list inside the type. We can have functions that depend on types.

Induction tree



In the Iris2, how does it store this tree? One idea: We have a value that describes the kind of the constructor, values and pointers to pointer to another constructor. We want to end up having a solution that uses buffers, which is very fast.

Data types

- Descriptor:
- Constructor: Give a name to a description
- Data: List of constructors, such that we can build a tree or list

Serialized representations

The format they have chosen is a hex file. The header describes the way the data is build up.

Pointers and Buffers

Given some fixed point over data we have location and a datatype.

The type family singleton represents all types that are described uniquely by their variables.

Now we can a view function that takes a type an a pointer and returns what is stored in the location. The main constructor is the (#) to combine serialization processes.

Conclusion

In this language it is impossible to have faulty pointer arithmetic, because the datatype does not allow it.