Type inference for GADTs (OhMyGADT)

TLC Final Project Proposal

Nikhil Mehta
nm3077@columbia.edu

Alexis Gadonneix
ag4625@columbia.edu

April 2023

1 Problem Description

Throughout our project, we are going to follow [Pey+06] in order to implement a language and a type system with Generalized Abstract Data Types (GADTs).

GADTs are more powerful than data constructors, type constructors, and type functions in Haskell. They are somewhat similar to data classes, but let you have multiple arguments arguments (it is actually possible to turn them into full-featured GADTs with GHC extensions).

Let's look at a simple example with a Haskell-like syntax:

```
Simple a where
    Lit :: Int -> Term Int
    Add :: Term Int -> Term Int -> Term Int
    IsZ :: Term Int -> Term Bool
    If  :: Term Bool -> Term a -> Term a -> Term a
```

A possible `eval` function is easy to write:

```
eval :: Simple a -> a
eval (Lit i) = i
eval (Add x y) = (eval x) + (eval y)
eval (IsZ x) = (eval x) == 0
eval (If c x y) = if (eval c) then (eval x) else (eval y)
```

Here are some key ideas / remarks about GADTs:

- The constructors' signatures are explicit
- Their argument types are arbitrary, which is different from a regular Haskell type constructor.
- They can be seen as functions with polymorphic types
- And the more interesting part: they allow for pattern matching with type refinement. For example, if we try to pattern-match on `Lit i`, the type of `i` can be refined to `Int`!

2 Goals

- Implement a simple language, starting from Hindley-Milner's lambda calculus and adding pattern matching and type annotations.
- Understand the syntax-directed typing rules of the paper.
- Implement a type inference algorithm for our language. It should be pretty similar to Hindley-Milner.
- Understand the limitations of this type system.

References

[Pey+06] Simon Peyton Jones et al. “Simple Unification-Based Type Inference for GADTs”. In: *SIGPLAN Not.* 41.9 (Sept. 2006), pp. 50–61. ISSN: 0362-1340. DOI: `10.1145/1160074.1159811`. URL: `https://doi.org/10.1145/1160074.1159811`.