Limits of Redex Trails

Olaf Chitil

July 11, 2000

1 Implementation through Transformation

1.1 Higher-order types

The transformation of types does not work for higher-order types, that is, types with parameters of a kind beyond `*`. For example, how should `data Comb f a b = C (f a b)` be transformed? Both `Comb (->) Int Bool` and `Comb (,) Int Bool` are instances of the type.

1.2 Non-simple contexts and classes

The non-simple contexts of Haskell 98 may contain types that are not just type variables. Would these types need to be transformed as well? Would the resulting contexts still be valid?

2 Trusting

It would be nice if all reductions of untrusted functions were registered in the redex trail, even if some functions are trusted. Unfortunately this is currently not the case, for example, suppose

```
filter test [1,2,3]
```

evaluates to `[]`. If `filter` is trusted, then the parent redex of `[]` is `filter test [1,2,3]`. The redexes `test 1`, `test 2`, `test 3` and their values `False`,`False`,`False` are not part of the redex trail. The problem is that the values of these redexes are not part of the final value. The problem seems related to the required special handling of projections.