ask

probably some other people and Conor McBride

October 10, 2020

1 Introduction

The `ask` system is a programming language and proof assistant (not) implemented (yet) as an idempotent textfile transducer. You feed your pathetic blethering through `ask` and it will build parts of your development which are obviously missing and comment out and upon parts of your development which are obviously bogus. It must be idempotent in that if you feed it its own output, there should be nothing new to add or take away.

The `ask` programming language is a total fragment of Haskell with strictly positive unindexed inductive datatypes, higher order function types, and higher rank type schemes. The `ask` proof language is some form of predicate logic with configurable rules.

2 Propositions

The keyword `prop` introduces the declaration of a new proposition former. `Prop` is the type of propositions.

We define a relation as follows

```
prop R τ ... where
  intro
  ⋮
```

with each $intro_i$ given as

```
R p ... ← I p' ... where
  condition
  ⋮
```

There is no need to name an introduction rule if its conclusion anti-unifies with the conclusion of all other introduction rules.

Relation symbols may be Constructor symbols or any infix symbol (like types when TypeOperators is enabled.) Intro rules must be Constructor symbols or :infix.

For examples,

```
prop Prop && Prop where
  a && b where
    a
    b
```

```
prop Prop || Prop where
  a || b ← Left where
    a
  a || b ← Right where
    b
```

Of course, conditions may themselves have hypotheses.

```
prop Prop → Prop where
  a → b where
    b where
      a
```