Orphan Instance Lang (OIL): A Programming Language supporting Orphan Instances

Asbjørn Rysgaard Eriksen[1]

[1] Aalborg University, Fredrik Bajers Vej 7, 9220 Aalborg Øst, Denmark
mail@bliztle.com

Abstract. This paper presents Orphan Instance Lang (OIL), a new programming language designed to allieviate the challenges posed by orphan instances in functionally inspired programming languages.

Keywords: Orphan Instances · Functional Programming · Syntax · Semantics · Rust

1 First Section

My awesome paper …

2 Syntax

Syntax is heavily inspired by Rust, with the addition of the `use` keyword to disambiguate in the case of multiple, conflicting implementations of a trait for a type.

Ident is a simple identifier, QIdent is a qualified identifier (`intance.field`), and TIdent is a type identifier (`module::function`).

```
    Prog ::= Def*
     Def ::= Mod
           | Struct
           | Impl
           | Trait
           | Fn
     Mod ::= mod Ident {Def*}
  Struct ::= struct Ident {Struct′*}
 Struct′ ::= Ident : TIdent;
    Impl ::= impl Ident {Fn*}
           | impl Ident for Ident {Fn*}
           | impl Ident of Ident for Ident {Fn*}
   Trait ::= trait Ident {Trait′*}
  Trait′ ::= fn Ident () -> Ident
      Fn ::= fn Ident () -> TIdent {ExprStat}
   Ident ::= [a-zA-Z_][a-zA-Z_0-9]*
  QIdent ::= Ident
           | QIdent . Ident
  TIdent ::= Ident
           | TIdent :: Ident

ExprStat ::= ExprLet
           | ExprBlock
           | ExprInvoke
           | ExprIf
           | ExprUse
    Expr ::= Expr Op Expr′
           | Expr′
   Expr′ ::= Expr′Op′Expr″
           | Expr″
   Expr″ ::= Expr″Op″Expr‴
           | Expr‴
      Op ::= == | != | <
           | <= | > | >=
     Op′ ::= + | -
     Op″ ::= * | /
   Expr‴ ::= [0-9]+ | [0-9]*\.[0-9]+
           | (Expr)
           | QIdent
           | ExprBlock
           | ExprInvoke
           | ExprIf
```

References