

# FUN — Untyped

Grigore Roşu and Traian Florin Şerbănuţă ({grosu,tserban2}@illinois.edu)

University of Illinois at Urbana-Champaign

## Abstract

...

## MODULE FUN-UNTYPED-SYNTAX

```
SYNTAX  Exp ::= Int
          | Bool
          | Id
          | (Exp) [bracket]
          | Exp Exp [strict]
          | & Id
          | @ Exp [strict]
          | callcc Exp [strict]
          | cons Exp Exp [strict]
          | Exp * Exp [strict]
          | Exp / Exp [strict]
          | Exp % Exp [strict]
          | Exp + Exp [strict]
          | Exp - Exp [strict, prefer]
          | Exp < Exp [strict]
          | Exp <= Exp [strict]
          | Exp > Exp [strict]
          | Exp >= Exp [strict]
          | Exp == Exp [strict]
          | Exp != Exp [strict]
          | ! Exp [strict]
          | Exp && Exp [strict(1)]
          | Exp || Exp [strict(1)]
          | Exp := Exp [strict]
          | let Bindings in Exp
          | letrec Bindings in Exp [prefer]
          | if Exp then Exp else Exp [strict(1)]
          | try Exp catch (Id)Exp
          | Exp ; Exp [strict(1)]
          | fun Cases
          | {Exps} [strict]
          | [Exps] [strict]
          | [Exps | Exp]
```

SYNTAX Exps ::= List{Exp, “,”} [strict]

SYNTAX Ids ::= List{Id, “,”}

SYNTAX Case ::= Exp -> Exp [binder]

SYNTAX Cases ::= List{Case, “[”}

SYNTAX Binding ::= Exp = Exp

SYNTAX Bindings ::= List{Binding, “and”}

RULE 
$$\frac{P1 \ P2 \rightarrow E}{P1 \rightarrow \text{fun } P2 \rightarrow E}$$
 [macro]

RULE 
$$\frac{F \ P = E}{F = \text{fun } P \rightarrow E}$$
 [macro]

RULE 
$$\frac{[E, Es \mid T]}{[E \mid [Es \mid T]]} \quad \text{requires } Es \neq_K \bullet_{Exps}$$
 [macro]

```
SYNTAX  Id ::= head
          | tail
          | null?
          | ref
          | throw
```

```
SYNTAX  Id ::= $h
          | $t
          | $l
          | $x
          | $k
          | $v
```

RULE 
$$\frac{\text{head}}{\text{fun } [ \$h \mid \$t ] \rightarrow \$h}$$
 [macro]

RULE 
$$\frac{\text{tail}}{\text{fun } [ \$h \mid \$t ] \rightarrow \$t}$$
 [macro]

RULE 
$$\frac{\text{null?}}{\text{fun } [\bullet_{Exps}] \rightarrow \text{true} \mid [ \$h \mid \$t ] \rightarrow \text{false}}$$
 [macro]

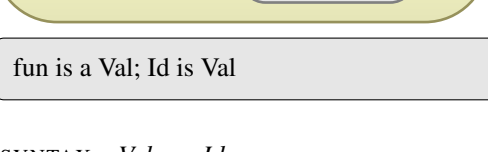
RULE 
$$\frac{\text{try } E \text{ catch } (X)E'}{\text{callcc } (\text{fun } \$k \rightarrow (\text{fun } \text{throw} \rightarrow E) (\text{fun } X \rightarrow \$k \ E'))}$$
 [macro]

SYNTAX Exp ::= mu Case

END MODULE

## MODULE FUN-UNTYPED

CONFIGURATION:



fun is a Val; Id is Val

```
SYNTAX  Val ::= Id
          | Int
          | Bool
          | fun Cases
```

SYNTAX Vals ::= List{Val, “,”}

SYNTAX Exp ::= Val

SYNTAX KResult ::= Val

RULE 
$$\frac{I1 + I2}{I1 +_{Int} I2}$$

RULE 
$$\frac{(I1 - I2)}{I1 -_{Int} I2}$$

RULE 
$$\frac{I1 * I2}{I1 *_{Int} I2}$$

RULE 
$$\frac{I1 / I2}{I1 \div_{Int} I2} \quad \text{requires } I2 \neq_K 0$$

RULE 
$$\frac{I1 \% I2}{I1 \%_{Int} I2} \quad \text{requires } I2 \neq_K 0$$

RULE 
$$\frac{I1 < I2}{I1 <_{Int} I2}$$

RULE 
$$\frac{I1 <= I2}{I1 \leq_{Int} I2}$$

RULE 
$$\frac{I1 > I2}{I1 >_{Int} I2}$$

RULE 
$$\frac{I1 >= I2}{I1 \geq_{Int} I2}$$

RULE 
$$\frac{V1 == V2}{V1 =_K V2}$$

RULE 
$$\frac{V1 != V2}{V1 \neq_K V2}$$

RULE 
$$\frac{! T}{\neg_{Bool}(T)}$$

RULE 
$$\frac{\text{true} \ \&\& \ E}{E}$$

RULE 
$$\frac{\text{false} \ \&\& \ \text{—}}{\text{false}}$$

RULE 
$$\frac{\text{true} \ || \ \text{—}}{\text{true}}$$

RULE 
$$\frac{\text{false} \ || \ E}{E}$$

RULE 
$$\frac{(\text{fun } P \rightarrow E \mid \text{—}) \ V}{E[\text{getMatching}(P, V)]} \quad \text{requires isMatching}(P, V)$$

RULE 
$$\left( \text{fun } P \rightarrow \text{—} \mid C_s \right) V \quad \text{requires } \neg_{Bool} \text{isMatching}(P, V)$$

RULE 
$$\text{decomposeMatching}([H \mid T], [V, Vs])$$
  
$$H, T \quad V, [Vs]$$

RULE 
$$\frac{\text{let } Bs \text{ in } E}{(\text{fun } \{ \text{ids}(Bs) \} \rightarrow E) \ \{ \text{exps}(Bs) \}}$$

SYNTAX Exp ::= mu Case

RULE 
$$\frac{\text{mu } X \rightarrow E}{E[(\text{mu } X \rightarrow E) / X]}$$

RULE 
$$\frac{\text{letrec } F = E \text{ in } E'}{\text{let } F = (\text{mu } F \rightarrow E) \text{ in } E'}$$

RULE 
$$\frac{\text{if true then } E \text{ else —}}{E}$$

RULE 
$$\frac{\text{if false then — else } E}{E}$$

SYNTAX Val ::= {Vals}

SYNTAX Val ::= [Vals]

RULE 
$$\frac{\text{cons } V \ [Vs]}{[V, Vs]}$$

RULE 
$$\frac{\text{ref } V}{L} \quad \frac{\text{store } \bullet_{Map}}{L \mapsto V} \quad \text{requires fresh}(L)$$

RULE 
$$\frac{\text{@ } L}{V} \quad \frac{\text{store}}{L \mapsto V}$$

RULE 
$$\frac{L := V}{V} \quad \frac{\text{store}}{L \mapsto \text{—}} \quad \frac{V}{V}$$

RULE 
$$\frac{V ; E}{E}$$

SYNTAX Val ::= cc (K)

RULE 
$$\frac{\text{callcc } V \curvearrow K}{V \curvearrow \text{cc}(K)}$$

RULE 
$$\frac{\text{cc}(K) \ V \curvearrow \text{—}}{V \curvearrow K}$$

SYNTAX Ids ::= ids (Bindings) [function]

RULE 
$$\frac{\text{ids}(\bullet_{Bindings})}{\bullet_{Ids}}$$

RULE 
$$\frac{\text{ids}(X = \text{—} \text{and } Bs)}{X, \text{ids}(Bs)}$$

SYNTAX Exps ::= exps (Bindings) [function]

RULE 
$$\frac{\text{exps}(\bullet_{Bindings})}{\bullet_{Exps}}$$

RULE 
$$\frac{\text{exps}(\text{—} = E \text{and } Bs)}{E, \text{exps}(Bs)}$$

END MODULE