A type system for SCA language

What follows has been engineered taking *Type Systems* by Luca Cardelli, and *A quick introduction to Type Systems* by David B. MacQueen, as main guidelines . It's a first order (no type abstraction and/or type polymorphism) type system with subtyping ($\mathrm{F}_{1<}$ *family*).

1. BASIC JUDGMENTS

Γ is a well-formed type environment:

$$\Gamma \vdash \diamond$$

`t` is a well-formed type:

$$\Gamma \vdash \texttt{t}$$

e is a well-formed term of type t:

$$\Gamma \vdash e : \texttt{t}$$

2. BASIC TYPES

$$\frac{\Gamma \vdash \diamond}{\Gamma\ \vdash \mathit{unit} : \texttt{Unit}}$$

$$\frac{\Gamma \vdash \diamond, \quad \mathit{const} \in \{\mathit{Basic}\}}{\Gamma\ \vdash \mathit{const}}$$

$$\frac{\Gamma \vdash \diamond}{\Gamma\ \vdash \mathit{integer} : \texttt{Int}}$$

$$\frac{\Gamma \vdash \diamond}{\Gamma\ \vdash \mathit{char} : \texttt{Char}}$$

$$\frac{\Gamma \vdash \diamond}{\Gamma\ \vdash \mathit{float} : \texttt{Float}}$$

$$\frac{\Gamma \vdash \diamond}{\Gamma\ \vdash \mathit{boolean} : \texttt{Boolean}}$$

$$\frac{\Gamma \vdash \diamond}{\Gamma\ \vdash \mathit{string} : \texttt{String}}$$

Where *unit, const, integer, char, float, boolean, string* range over their respective domains.

3. FUNCTIONS

Arrow Operator for Function Application:

$$\frac{\Gamma \vdash \mathtt{t}_1 \qquad \Gamma \vdash \mathtt{t}_2}{\Gamma \ \vdash \mathtt{t}_1 \to \mathtt{t}_2}$$

Function Application (Generic):

$$\frac{\Gamma \vdash e_1\colon \mathtt{t}_1 \to \mathtt{t}_2 \qquad \Gamma \vdash e_2\colon \mathtt{t}_1}{\Gamma \ \vdash e_1 e_2\colon \mathtt{t}_2}$$

Arithmetics, Logics and Ordering stuff:

$$\frac{\Gamma \vdash e_1\colon \mathtt{Boolean} \qquad \Gamma \vdash e_2\colon \mathtt{Boolean}}{\Gamma \ \vdash (e_1 \quad \mathtt{ANDOR} \quad e_2)\colon \mathtt{Boolean}}$$

$$\mathtt{ANDOR} \in \{\&\&, ||\}$$

$$\frac{\Gamma \vdash e\colon \mathtt{Boolean}}{\Gamma \ \vdash !\, e\colon \mathtt{Boolean}}$$

$$\frac{\Gamma \vdash e_1\colon \mathtt{t}_1 \qquad \Gamma \vdash e_2\colon \mathtt{t}_2}{\Gamma \ \vdash (e_1 \quad \mathtt{Comp} \quad e_2)\colon \mathtt{Boolean}}$$

$$\mathtt{t}_{1,2} \in \{\mathtt{Int}, \mathtt{Float}, \mathtt{Char}, \mathtt{String}\}$$

$$\mathtt{Comp} \in \{==, >, <, <=, >=, !=\}$$

$$\frac{\Gamma \vdash e_1\colon \mathtt{t}_1 \qquad \Gamma \vdash e_2\colon \mathtt{t}_2}{\Gamma \ \vdash (e_1 \quad \mathtt{BIN\_OP} \quad e_2)\colon \mathtt{t}_{\mathtt{max}}}$$

$$\mathtt{t}_{\mathtt{max}} = max\{\mathtt{t}_1, \mathtt{t}_2\}$$

(See section below for clarifications about usage of *max* in this context)

$$\mathtt{t}_{\mathtt{1,2,max}} \in \{\mathtt{Int}, \mathtt{Float}\}$$

$$\mathtt{BIN\_OP} \in \{+, -, *, /, \%, \hat{}\,\}$$

$$\frac{\Gamma \vdash e\colon \mathtt{t}}{\Gamma \ \vdash (-\, e)\colon \mathtt{t}}$$

$$\mathtt{t} \in \{\mathtt{Int}, \mathtt{Float}\}$$

$$\frac{\Gamma \vdash \ \mathit{write}\mathtt{T} : \mathtt{T} \to \mathtt{String} \qquad \Gamma \vdash \ e\colon \mathtt{T}}{\Gamma \vdash \ \mathit{write}\mathtt{T}\ e : \ \mathtt{String}}$$

$$\mathtt{T} \in \{\mathtt{Int}, \mathtt{Float}, \mathtt{Char}, \mathtt{String}\}$$

$$\frac{\Gamma \vdash \ \mathit{read}\mathtt{T} : \mathtt{String} \to \mathtt{T} \qquad \Gamma \vdash \ e\colon \mathtt{String}}{\Gamma \vdash \ \mathit{read}\mathtt{T}\ e \ \ : \ \mathtt{T}}$$

$$\mathtt{T} \in \{\mathtt{Int}, \mathtt{Float}, \mathtt{Char}, \mathtt{String}\}$$

4. SUBTYPING

Subtyping is achieved by introducing an order relation over types, " $<$ ", and a type `Top`, which is the supertype of any well-formed type according to " $<$ ". `Top` has a purely theoretical meaning and is not visible to programmers.

$$\frac{\Gamma \vdash \diamond}{\Gamma \ \vdash \ \texttt{Top}}$$

If *any* is any given type, then:

$$\frac{\Gamma \vdash \mathit{any}}{\Gamma \ \vdash \mathit{any} <: \texttt{Top}}$$

$$\frac{\Gamma \vdash \texttt{Int} \quad \Gamma \vdash \texttt{Float}}{\Gamma \ \vdash \texttt{Int} <: \texttt{Float}}$$

Value-Type Subsumption.

$$\frac{\Gamma \vdash e : \texttt{t}_1, \quad \Gamma \vdash \texttt{t}_1 <: \texttt{t}_2}{\Gamma \ \vdash e : \texttt{t}_2}$$

Also, being an order relation, reflexivity and transitivity also hold true for " $<$ "

5. OTHER OPERATORS AND CONSTRUCTS

$$\frac{\Gamma \vdash \ e\colon \mathtt{t}}{\Gamma \ \vdash \ \mathtt{PrefixINCDEC}\, e : \mathtt{t}}$$

$$\mathtt{PrefixINCDEC} \in \{++, --\}$$

$$\mathtt{t} \in \{\mathtt{Int}, \mathtt{Float}\}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{t}}{\Gamma \ \vdash \ e\, \mathtt{PostfixINCDEC}\colon \mathtt{t}}$$

$$\mathtt{PostfixINCDEC} \in \{++, --\}$$

$$\mathtt{t} \in \{\mathtt{Int}, \mathtt{Float}\}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{t} \quad \Gamma \vdash \ f\colon \mathtt{t}}{\Gamma \ \vdash \ e\, \mathtt{OP}\, f : \mathtt{t}}$$

$$\mathtt{OP} \in \{+ =, - =, * =, / =\}$$

$$\mathtt{t} \in \{\mathtt{Int}, \mathtt{Float}\}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{Boolean} \quad \Gamma \vdash \ \mathtt{BODY}}{\Gamma \ \vdash \ \mathtt{if}\ (e)\ \mathtt{then\ \ BODY}}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{Boolean} \quad \Gamma \vdash \ \mathtt{BODY}(1,2)}{\Gamma \ \vdash \ \mathtt{if}\ (e) \ \ \mathtt{BODY1\ \ else\ \ BODY2}}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{Boolean} \quad \Gamma \vdash \ \mathtt{BODY}}{\Gamma \ \vdash \ \mathtt{while}\ (e) \ \ \mathtt{BODY}}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{Int} \quad \Gamma \vdash \ x\colon \mathtt{Int} \quad \Gamma \vdash \ y\colon \mathtt{Int} \quad \Gamma \vdash \ \mathtt{BODY}}{\Gamma \ \vdash \ \mathtt{for}\ (e \leftarrow \ x\colon \mathtt{Int} \quad y\colon \mathtt{Int}) \quad \mathtt{BODY}}$$

6. ARRAYS AND POINTERS

$$\frac{\Gamma \vdash \ e\colon \mathtt{Int} \qquad \Gamma \vdash \ array\colon \mathtt{t}}{\Gamma \ \vdash \ array[e]\colon \mathtt{t}}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{t}}{\Gamma \ \vdash \ \&e\colon \mathtt{PTR\_TO}(\mathtt{t})}$$

$$\frac{\Gamma \vdash \ e\colon \mathtt{PTR\_TO}(\mathtt{t})}{\Gamma \ \vdash \ *e\colon \mathtt{t}}$$

In the above rules t is any type.