Subtypes

Frank McCabe

March 4, 2005

Abstract

A short note about types and sub-types

1 Notation

1.1 Type expressions

`void` The bottom or inconsistent type. Less than all other types.

`top` The top or universal type. Greater than all other types.

`(`$\mathtt{T}_1$`,...,`$\mathtt{T}_n$`)` The tuple type.

`(`$\mathtt{T}_1$`,...,`$\mathtt{T}_n$`)=>`$\mathtt{T}_R$ The function type.

`(`$\mathtt{T}_1$`,...,`$\mathtt{T}_n$`)*` The action procedure type.

`(`$\mathtt{T}_1$`,...,`$\mathtt{T}_n$`){}` The predicate type.

`(`$\mathtt{T}_1$`,...,`$\mathtt{T}_n$`)-->`$\mathtt{T}_R$ The grammar type.

{`$\mathtt{F}_1$`:`$\mathtt{T}_1$`,...,`$\mathtt{F}_n$`:`$\mathtt{T}_n$} A type interface.

`V-T` Universally quantified type.

`Nm[`$\mathtt{T}_1$`,...,`$\mathtt{T}_n$`]` A named type. If n=0, then the square brackets may be dropped.

1.2 Type relations

$\mathtt{T}_1 \preceq \mathtt{T}_2$ $\mathtt{T}_1$ is a sub-type of $\mathtt{T}_2$.

2 Type rules

2.1 Numeric literals

$$\frac{X \text{ is a number literal}}{E \vdash_t X \rightsquigarrow \texttt{number}} \tag{1}$$

2.2 Function application

$$\frac{E \vdash_t F \rightsquigarrow \vec{A_f} => R_f \qquad E \vdash_t \vec{\texttt{A}} \rightsquigarrow \vec{A_t} \qquad E \vdash_t \vec{A_t} \preceq \vec{A_f}}{E \vdash_t \texttt{F(}\vec{\texttt{A}}\texttt{)} \rightsquigarrow R_t} \tag{2}$$