Strictness and Totality Analysis

Kirsten Lackner Solberg*
Hanne Riis Nielson and Flemming Nielson
Computer Science Dept.
Aarhus University, Denmark
e-mail: {kls,hrn,fn}@daimi.aau.dk

A full version of the paper can be found in *Proceedings of SAS'94*, LNCS 864 pages 408 – 422, 1994

Strictness analysis has proved useful in the implementation of lazy functional languages as Miranda, Lazy ML and Haskell: when a function is strict it is safe to evaluate its argument before performing the function call. Totality analysis is equally useful but has not be adopted so widely: if the argument to a function is known to terminate then it is safe to evaluate it before performing the function call.

In this talk we present an inference system for performing strictness *and* totality analysis. We restrict our attention to a simply typed lambda-calculus with constants and a fixpoint operator. The inference system is an extension of the usual type system in that we introduce three annotations on types t:

- $!^{\mathbf{b}}\mathrm{t}$: the value has type t and it definitely $\perp$,
- $!^{\mathbf{n}}\mathrm{t}$: the value has type t and is definitely *not* $\perp$, and
- $!^{\top}\mathrm{t}$: the value has type t and it can be any value.

Annotated types can be constructed using the function type constructor and (top-level) conjunction. As an example a function may have the annotated type $(!^{\mathbf{n}}\mathtt{Int} \to !^{\mathbf{n}}\mathtt{Int}) \wedge (!^{\mathbf{b}}\mathtt{Int} \to !^{\mathbf{b}}\mathtt{Int})$ which means that given a terminating argument the function will definitely terminate and given a non-terminating argument it will definitely not terminate. Thus we capture the strictness as well as the totality of the function. Strictness and totality information can also be combined as in $(!^{\mathbf{b}}\mathtt{Int} \to !^{\mathbf{n}}\mathtt{Int} \to !^{\mathbf{n}}\mathtt{Int}) \wedge (!^{\mathbf{n}}\mathtt{Int} \to !^{\mathbf{b}}\mathtt{Int} \to !^{\mathbf{n}}\mathtt{Int}) \wedge (!^{\mathbf{b}}\mathtt{Int} \to !^{\mathbf{b}}\mathtt{Int} \to !^{\mathbf{b}}\mathtt{Int})$ which will be the annotated type of McCarthy's ambiguity operator.

We give examples of its use and prove the correctness with respect to a natural-style operational semantics.

*Dept. of Math. and Computer Science, Odense University, Denmark