Type Systems—Assignment

GLOBAN 2006

The Global Computing Approach to Analysis of Systems International Summer School at DTU, August 21–25, 2006

Vasco T. Vasconcelos
University of Lisbon

Extending the FTP server. Suppose that we want to extend the functionality of the FTP server, by adding operations to change the current directory and to list the contents of the current directory.

a) Write the new version of the FTP server, by adapting that in the slides.

b) Show the new type for name f.

c) Would the client, as in the slides, be able to interact with the new server, that is, would reduction, →, still go through? Show the reduction steps of process System below, up to the point where the new server (Ftpd′) differs from the old (Ftpd).

$$\text{System} \overset{\text{def}}{=} (\nu f)(\text{Ftpd}'[f] \mid \text{Client}[f])$$

Type checking. In order to type check process System, the Client cannot be typed as in the slides.

a) Present an intuitive justification as to why we need a new type for the client.

Note: since we have not dealt with recursive types, process Actions cannot be recursive. Consider then the following definition for process Actions(f, y), and adapt it to the extended FTP server, FTPd′.

```
y>{
  get: y?(fileName). y![aFile]. y>{bye: Ftpd[f]},
  put: y?(aFile). y>{bye: Ftpd[f]}
}
```

b) Present the non-recursive types for channel x in process Client and channel y in process FTPd′.

c) Present the derivation tree for sequent

$$\Gamma \vdash \text{System},$$

for an appropriate Γ.

Subtyping. In order to avoid re-typing the client process whenever more functionality is added to the server we need a means to allow the type for the extended FTP server to be compatible with that of the client. A possible solution is to say that the new type for the FTP server is a *subtype* of that of the old server.

a) Present a definition of the subtyping relation for session types, inspired on that of input-output types in the slides. You may want to refer to Simon J. Gay and Malcolm J. Hole. Subtyping for session types in the pi calculus. *Acta Informatica*, 42(2–3):191–225, 2005.

b) Adapt the typing rules for sort send and receive, branch and select, and channel send and receive, in order to allow subtyping to come into play.

c) Show that process System above is typable *even when* channel y in Client is assigned the type where the new functionality of the server is not present.

d) Prove Subject-Reduction for the new system, by adapting that in paper Nobuko Yoshida and Vasco T. Vasconcelos. Language Primitives and Type Discipline for Structured Communication-Based Programming Revisited - Two Systems for Higher-Order Session Communication. Technical Report DI/FCUL TR-06-9. Department of Computer Science, University of Lisbon. 2006.

e) What about the Type Safety result? Do we need a new proof? Why?