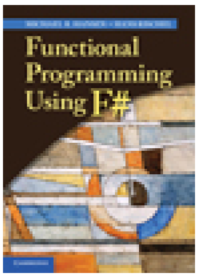

Book DOI: http://dx.doi.org/10.1017/CBO9781139093996

Online ISBN: 9781139093996

Hardback ISBN: 9781107019027

Paperback ISBN: 9781107684065

Chapter DOI: http://dx.doi.org/10.1017/CBO9781139093996.017

Appendix C

The dialogue program from Chapter 13

This appendix contains the complete program for the skeleton program shown in Table 13.6. The reader should consult Section13.5 for further information.

```
type Message = Start of string | Clear | Cancel
               | Web of string | Error | Cancelled

let ev = AsyncEventQueue()

let rec ready() =
  async {urlBox.Text <- "http://"
         ansBox.Text <- ""

         disable [cancelButton]
         let! msg = ev.Receive()
         match msg with
         | Start url -> return! loading(url)
         | Clear     -> return! ready()
         | _         -> failwith("ready: unexpected message")}

and loading(url) =
  async {ansBox.Text <- "Downloading"
         use ts = new CancellationTokenSource()
         Async.StartWithContinuations
             (async {let webCl = new WebClient()
                     let! html = webCl.AsyncDownloadString(Uri url)
                     return html},
              (fun html -> ev.Post (Web html)),
              (fun _    -> ev.Post Error),
              (fun _    -> ev.Post Cancelled),
              ts.Token)

         disable [startButton; clearButton]
         let! msg = ev.Receive()
         match msg with
         | Web html ->
             let ans = "Length = " + String.Format("0:D",html.Length)
             return! finished(ans)
         | Error    -> return! finished("Error")
         | Cancel   -> ts.Cancel()
                       return! cancelling()
         | _        -> failwith("loading: unexpected message")}
```

```
and cancelling() =
  async
    {ansBox.Text <- "Cancelling"

     disable [startButton; clearButton; cancelButton]
     let! msg = ev.Receive()
     match msg with
     | Cancelled | Error | Web  _ -> return! finished("Cancelled")
     | _             ->  failwith("cancelling: unexpected message")}

and finished(s) =
   async {ansBox.Text <- s

          disable [startButton; cancelButton]
          let! msg = ev.Receive()
          match msg with
          | Clear -> return! ready()
          | _     ->  failwith("finished: unexpected message")}
```

Table C.1 *Dialogue program for automaton in Figure 13.4*