A.3 The `NextLevelRefs` program

This section contains the complete `NextLevelRefs` program. For the documentation of the program, we refer to Section 10.9.

```
// NextLevelRefs
// MRH, HR May 15, 2013
// Inputs a WebCat file containing line pairs Title and URL for some
// MSDN web-pages, scans the sub-menu of each of these pages and
// outputs the WebCat file corresponding to the found child pages
.
open System;;
open System.IO;;
open System.Net;;
open System.Collections.Generic;;
open System.Text.RegularExpressions;;
open System.Web;;
open System.Xml;;
open TextProcessing;;

type infoType = ButtonStart | RefInfo of string * string | EndOfFile;;

// nextInfo: XmlReader -> infoType
// steps XmlReader forward to the next "node of interest" or to
// end of Xml file.
//
let rec nextInfo(r:XmlReader) =
  match r.Read() with
  | false -> EndOfFile
  | _     ->
  match r.NodeType with
  | XmlNodeType.Element ->
    match r.Name with
    | "div" when (r.GetAttribute "class" = "toclevel2")
          -> ButtonStart
    | "a" -> let path = r.GetAttribute "href"
             ignore(r.Read())
             RefInfo(r.Value,path)
    |  _  -> nextInfo r
  |  _    -> nextInfo r;;

// cStr: string -> string
// Converts HTML encoded string and
// replaces quotes by apostrophes
//
let regQuote = Regex @"\042";;
let quoteReplace str = regQuote.Replace(str,"'");;
let cStr s = quoteReplace(HttpUtility.HtmlDecode s);;
```

```
// getWEBrefs: string -> (string * string) list
// Scans a MSDN web-page and finds the list of
// pairs of title and uri for child pages
//
let getWEBrefs(uri: string) =
  let baseUri = Uri uri
  let webCl = new WebClient()
  let doc = webCl.DownloadString baseUri
  use docRd  = new StringReader(doc)
  let settings =
      XmlReaderSettings(DtdProcessing = DtdProcessing.Ignore)
  use reader = XmlReader.Create(docRd,settings)
  // getNextRef is used in getRefs to find next ref
  let rec getNextRef() =
    match nextInfo reader with
    | RefInfo(t, path) ->
        let pathUri = Uri(baseUri,path)
        (cStr t, pathUri.AbsoluteUri)
    | _               -> failwith "Button Format Error"
  // getRefs finds remaining refs
  let rec getRefs() =
    match nextInfo reader with
    | ButtonStart -> getNextRef() :: getRefs()
    | EndOfFile   -> []
    | _           -> getRefs()
  getRefs();;

open System;;
open System.IO;;

let outputRef (output:StreamWriter) (title:string, uri:string) =
  output.WriteLine title
  output.WriteLine uri;;

let expandUri (output:StreamWriter) uri =
  let lst = getWEBrefs uri
  List.iter (outputRef output) lst;;

let handleLinePair (output:StreamWriter) (rdr: StreamReader) =
  ignore(rdr.ReadLine())
  expandUri output (rdr.ReadLine());;

[<EntryPoint>]
let main (args: string[]) =
  if Array.length args < 2 then
    failwith "Missing parameters"
  else
  if File.Exists args.[1] then
    failwith "Existing output file"
  else
    use output = File.CreateText args.[1]
    fileXiter (handleLinePair output) args.[0]
    output.Close()
    0;;
```

Table A.3 *The* `NextLevelRefs` *program*