# Forth Family Tree and Timeline

*M. Anton Ertl*
*TU Wien*

This is a paper version of the incomplete family tree and timeline of Forth implementations and standards. The Web version is recommended and is available at http://www.complang.tuwien.ac.at/forth/family-tree/.

There are two graphs:

Tree
: just shows which implementation/standard took ideas from which; the node position does not indicate anything else.

Timeline
: is the same graph, but the x-coordinate of each node reflects the release date (or some other significant date) for the implementation/standard.

The graphs contain two kinds of nodes:

Standards
: (shown in Times Roman font (with serifs)). These are specifications for a language and not executable.

Implementations
: (shown in Helvetiva font (sans serif)). These are executable code.

They contain three kinds of edges:

begot
: (solid lines). One implementation (the child) directly incorporates (or got started by incorporating) significant amounts of code from another implementation (the parent). For standards, text is incorporated instead of code.

conforms-to
: (dashed lines). A Forth implementation conforms to a standard; also known as implementing the standard.

inspired
: (dotted lines). Words or concepts from one Forth implementation were incorporated in an implementation or standard. Only the most significant "inspired" edges are shown, to avoid clutter.

## But implementation X is missing

That’s because you did not send me any data on X. Don’t complain, just send me the information (see below).

## Forth family tree

You can find the paper version of this picture at the end of the paper. In the Web version this is a client-side image map. If your browser supports that feature, clicking on an implementation will get you to a page about that implementation or standard (in many cases; there are not pages for all of them); also, your browser may display a tooltip for these nodes with a few details (typically about authors or features).

## Forth Timeline

You can find the paper version of this picture at the end of the paper. In the Web version this is a client-side image map; the links and tooltips are exactly the same as in the family tree.

## How to submit information (or understand tree.fs)

You can describe the information about an implementation and its relations to other implementations informally; or you can describe it formally as a Forth program (see below). Then email it to anton@mips.complang.tuwien.ac.at, or post it in comp.lang.forth.

For the formal description, you first have to define all the nodes (implementations/standards) involved, in one of the following ways:

```
<year> s" <proper name>" name-implementation <nodename>
<year> s" <proper name>" standard <nodename>
```

Node names have to be C-style names (starts with a letter or _, contains only letters, digits, or _). If the proper name conforms to the restrictions for nodenames, you can use

```
<year> implementation <nodename> \ nodename will be used as proper name
```

Examples:

```
1978 s" fig-Forth" name-implementation figForth
1994 s" ANS Forth" standard Forth94
1996 Implementation Gforth
```

You can add an URL and a little bit of info (for a tooltip) for a node by preceding the node definition with

```
\U <URL>
\I <tooltip text>
```

This information is used in the image maps. Note that you have to define an URL, or the tooltip text will be ignored.

Once you have defined node names, you can use them in edge definitions:

```
<nodename1> begot <nodename2>
<nodename2> conforms-to <nodename1>
<nodename1> inspired <nodename2>
```

If you have any additional information, include them in comments. That's all there is to it.

## How it works

The source file tree.fs runs (on Gforth), and produces tree.dot, the input file for dot (the directed graph layout tool from the graphviz toolbox). Dot can produce a layout for the graph, in various formats, among them Postscript; it also produces y-coordinates for the nodes (which are processed into timeline.fs), which are then taken by tree.fs in another run to produce timeline.neato (basically a version of tree.dot, but with the node positions fixed). This file is then processed by neato (another graphviz tool) to produce the timeline in Postscript format. Currently the GIF and PDF formats are produced from the Postscript formats.

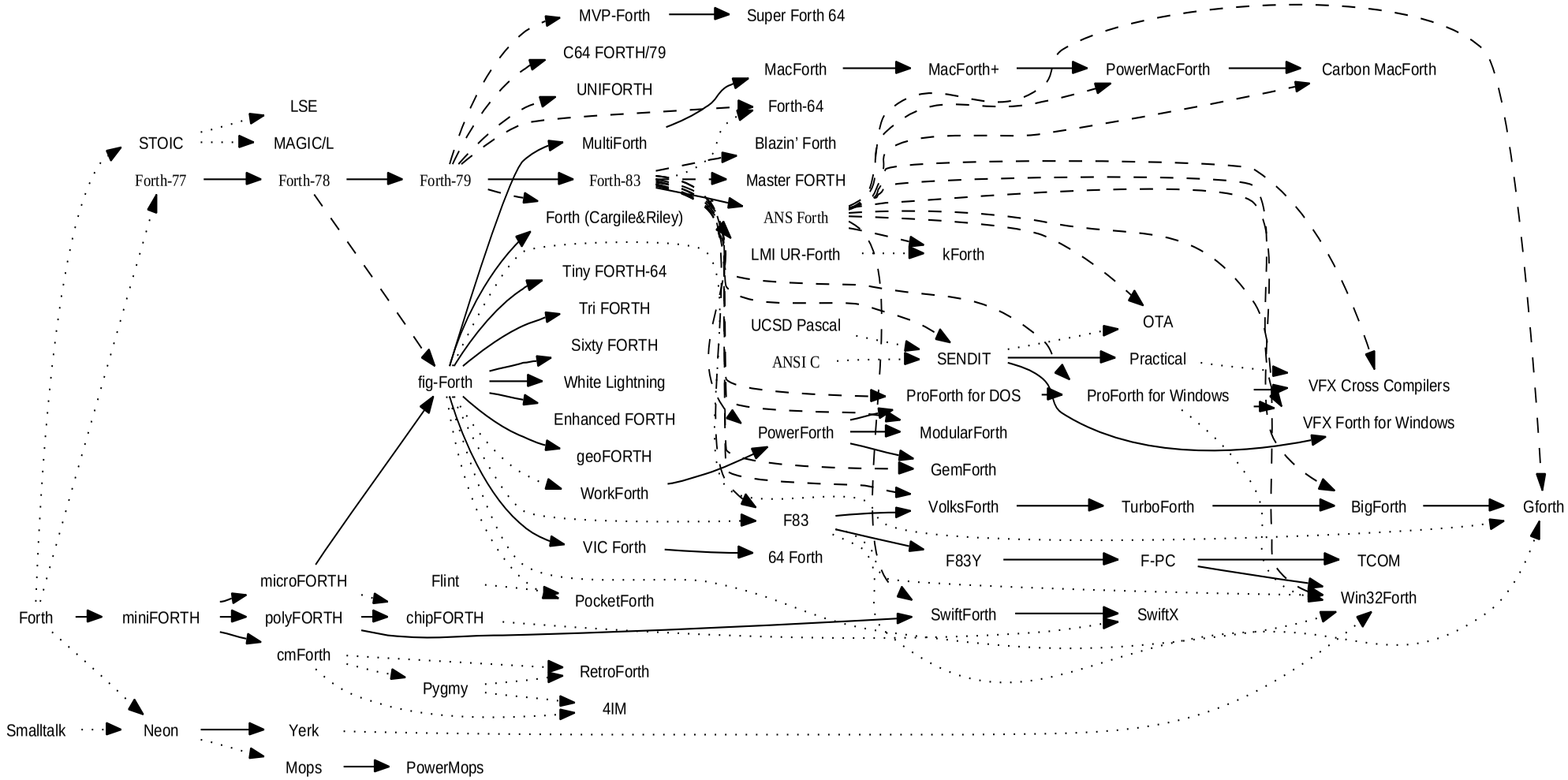
## Doing your own tree and timeline

Just grab the source package, unpack, edit the files for your needs, and say "make". You need GNU make, Gforth, graphviz, Ghostscript, awk, and the netpbm tools with the current Makefile (less if you don't need all the formats).

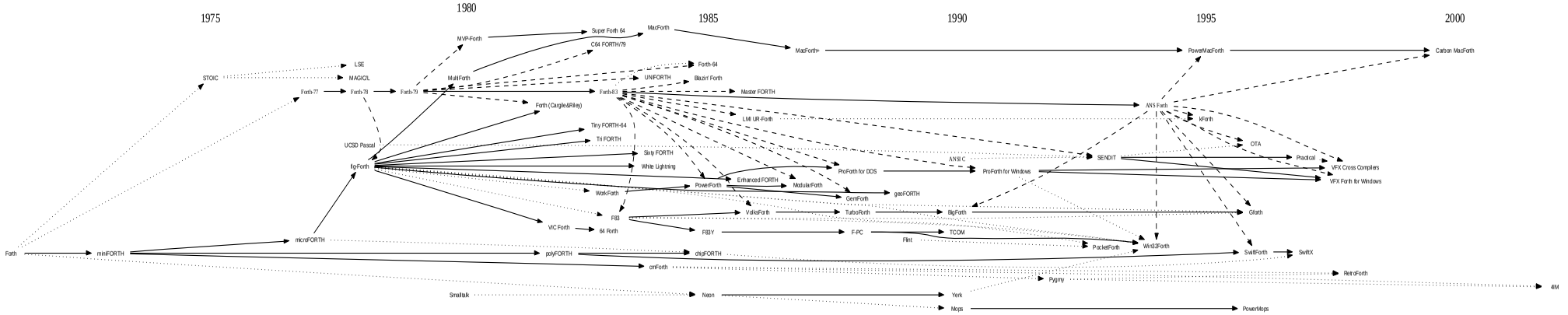
## Acknowledgments

The following persons provided information incorporated in the family tree: Guy Macon, Astrobe, Krishna Myneni, John Doty, Elizabeth Rather, Doug Hoffman, Alex McDonald, George Hubert, Stephen Pelc. Further information came from various sources on the WWW, in particular the HOPL-II Forth paper and C. H. Ting's F83 documentation.

# Family Tree


MVP-Forth
Super Forth 64
C64 FORTH/79
UNIFORTH
MacForth
MacForth+
PowerMacForth
Carbon MacForth
LSE
STOIC
MAGIC/L
Forth-64
MultiForth
Blazin' Forth
Forth-77
Forth-78
Forth-79
Forth-83
Master FORTH
Forth (Cargile&Riley)
ANS Forth
LMI UR-Forth
kForth
Tiny FORTH-64
Tri FORTH
OTA
UCSD Pascal
Sixty FORTH
ANSI C
SENDIT
Practical
fig-Forth
White Lightning
ProForth for DOS
ProForth for Windows
VFX Cross Compilers
Enhanced FORTH
PowerForth
ModularForth
VFX Forth for Windows
geoFORTH
GemForth
WorkForth
VolksForth
TurboForth
BigForth
Gforth
F83
VIC Forth
64 Forth
F83Y
F-PC
TCOM
microFORTH
Flint
PocketForth
Win32Forth
Forth
miniFORTH
polyFORTH
chipFORTH
SwiftForth
SwiftX
cmForth
RetroForth
Pygmy
4IM
Smalltalk
Neon
Yerk
Mops
PowerMops

# Timeline


1975
1980
1985
1990
1995
2000
Forth
STOIC
LSE
MAGIC/L
Forth-77
Forth-78
Forth-79
Forth-83
MVP-Forth
MultiForth
Super Forth 64
C64 FORTH/79
MacForth
MacForth+
PowerMacForth
Carbon MacForth
Forth-64
UNIFORTH
Blazin' Forth
Master FORTH
Forth (Cargile&Riley)
Tiny FORTH-64
Tri FORTH
UCSD Pascal
fig-Forth
Sicky FORTH
White Lightning
LMI UR-Forth
ANS Forth
ANSI C
SEND/IT
OTA
Practical
VFX Cross Compilers
VFX Forth for Windows
ProForth for DOS
ProForth for Windows
Enhanced FORTH
PowerForth
ModularForth
WorkForth
GemForth
geoFORTH
F83
VolksForth
TurboForth
BigForth
Gforth
VIC Forth
64 Forth
F83Y
F-PC
TCOM
Flint
PocketForth
Win32Forth
microFORTH
miniFORTH
polyFORTH
chipFORTH
SwiftForth
SwiftX
omForth
RetroForth
Pygmy
4M
Smalltalk
Neon
Yerk
Mops
PowerMops