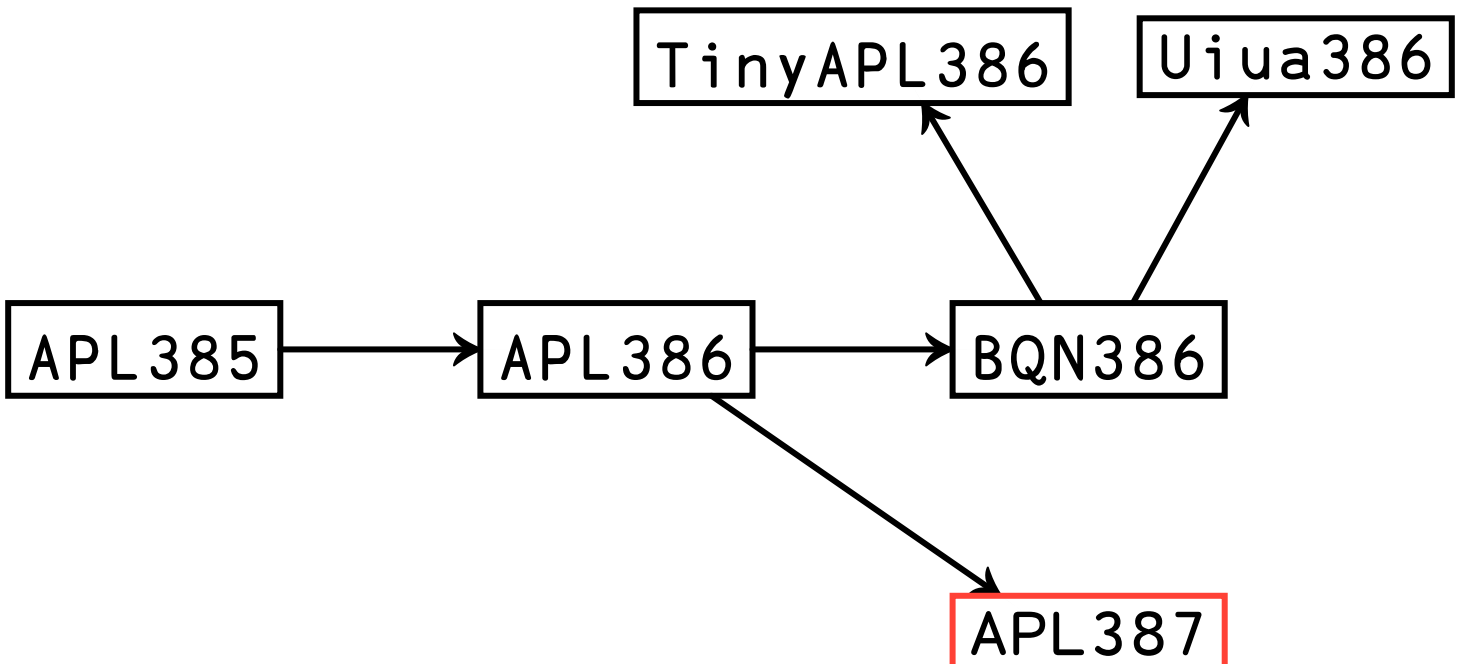
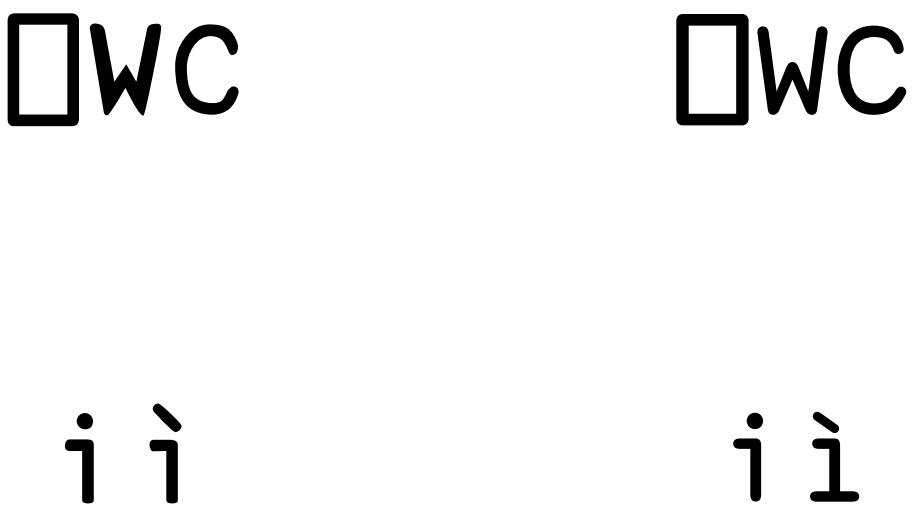
The Next APL Font Family

This work was supported by a grant from The APL Trust US, a nonprofit organization dedicated to advancing the use, development, and understanding of APL.

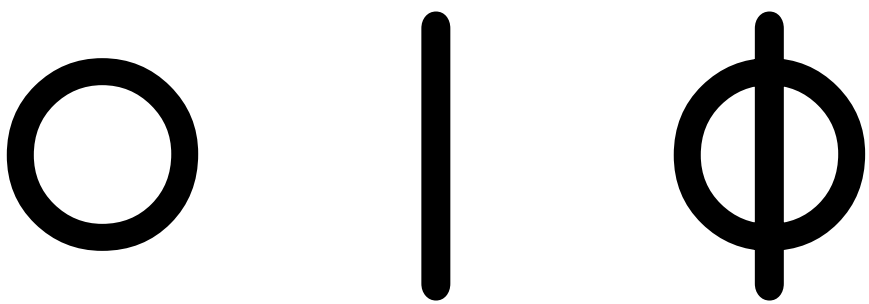
A bit of history



A bit of history



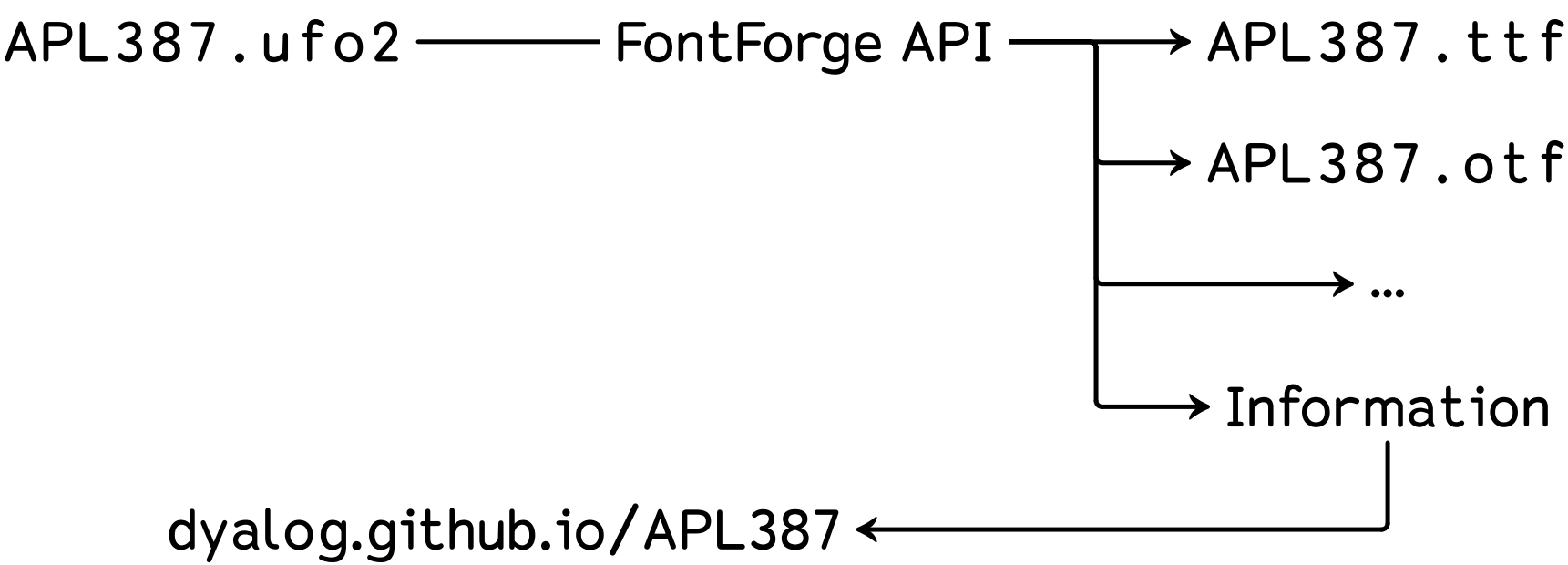
Components



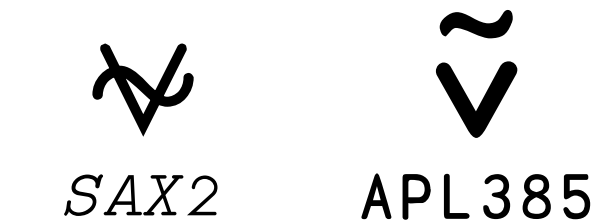
Components

```
<?xml version="1.0" encoding="UTF-8"?>
<glyph name="uni233D" format="1">
  <advance width="600"/>
  <unicode hex="233D"/>
  <outline>
    <component base="part_stile"/>
    <component base="part_circle"/>
  </outline>
</glyph>
```

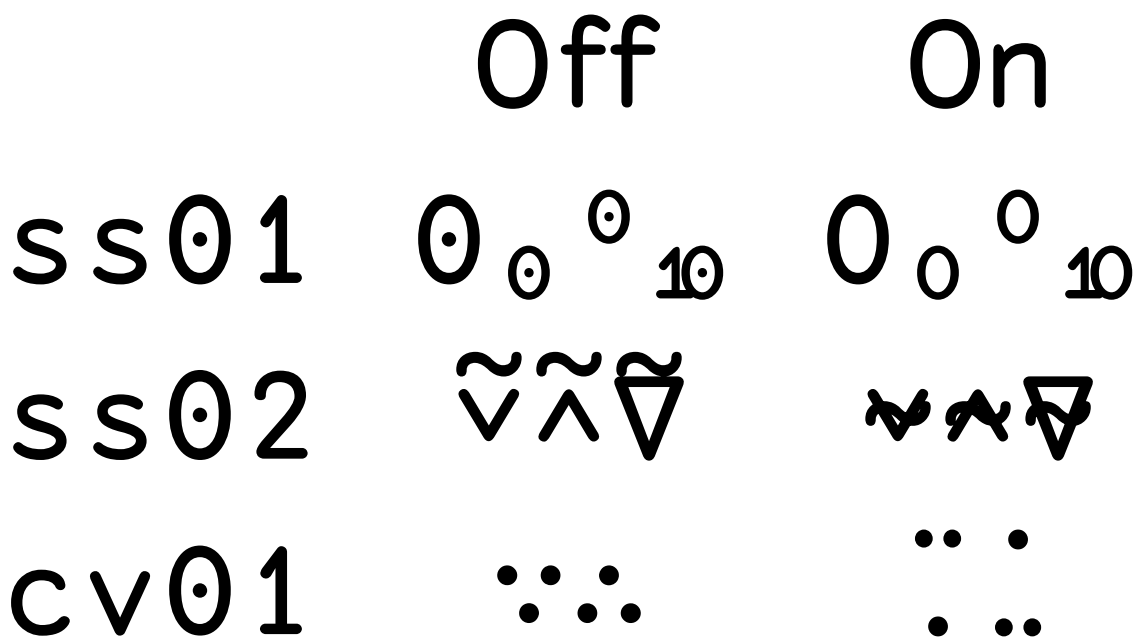
Unified source



Variants

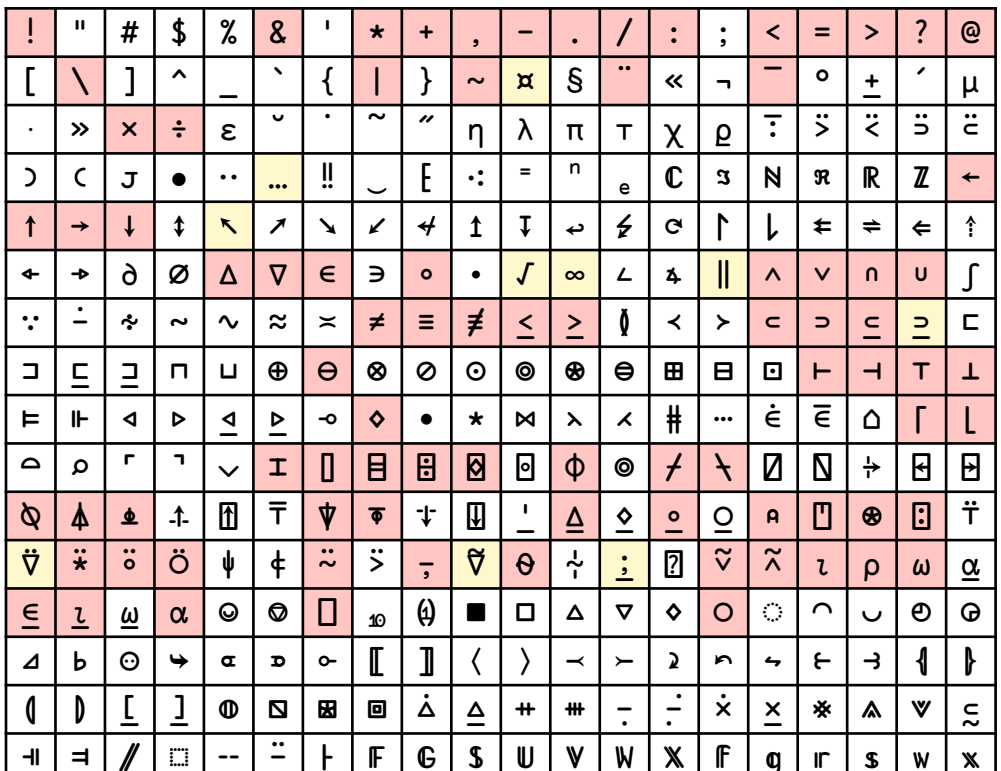


Variants



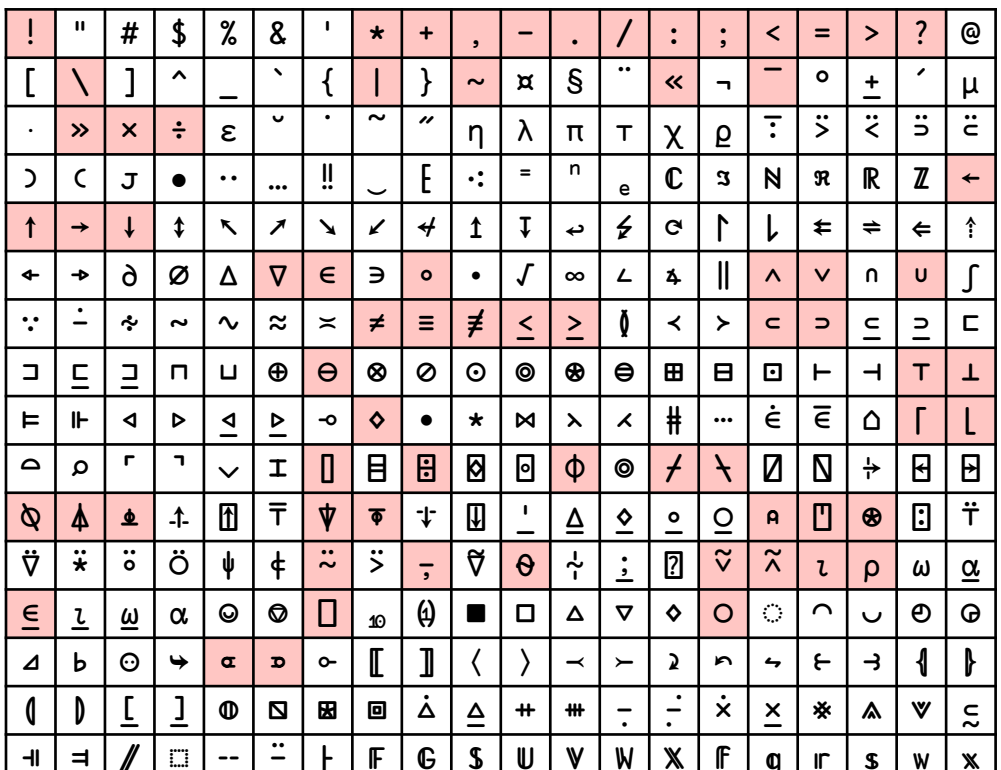
Support

Over 500 symbolic glyphs, including full support for Dyalog APL,



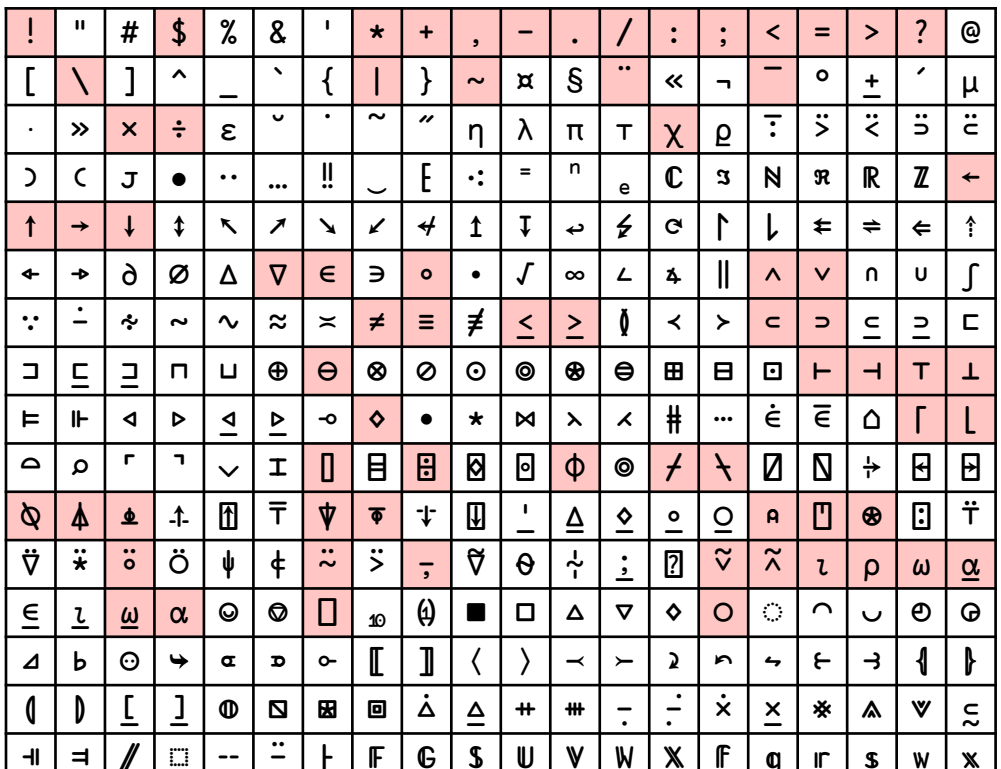
Support

Over 500 symbolic glyphs, including full support for Dyalog APL, APL64,



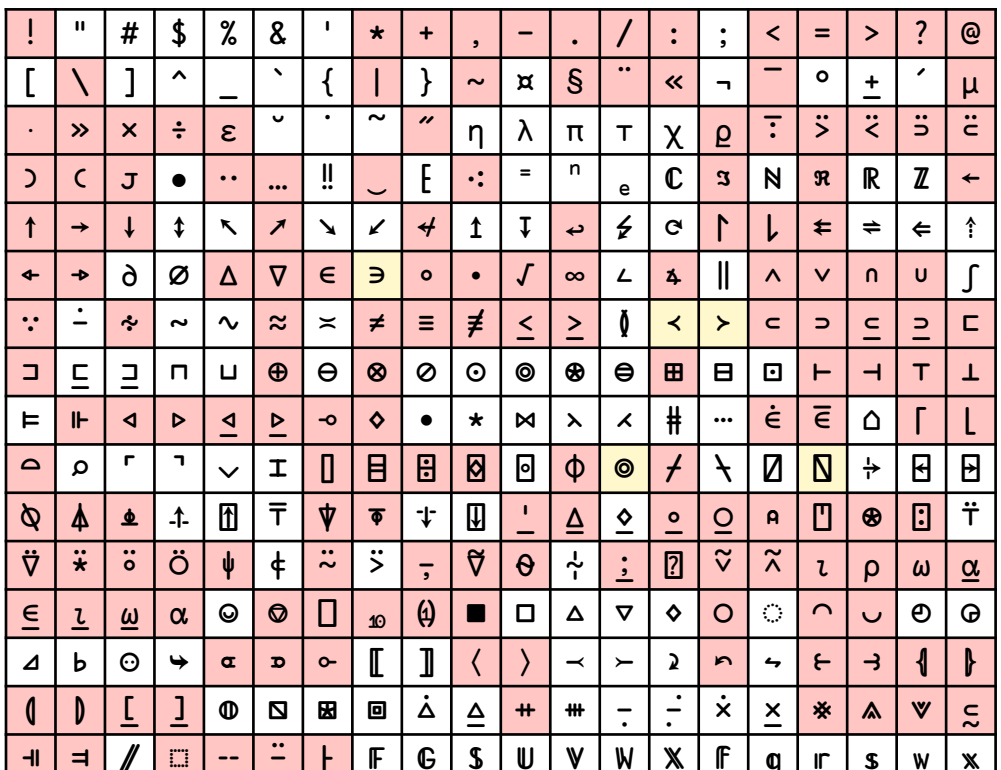
Support

Over 500 symbolic glyphs, including full support for Dyalog APL, APL64, GNU APL,



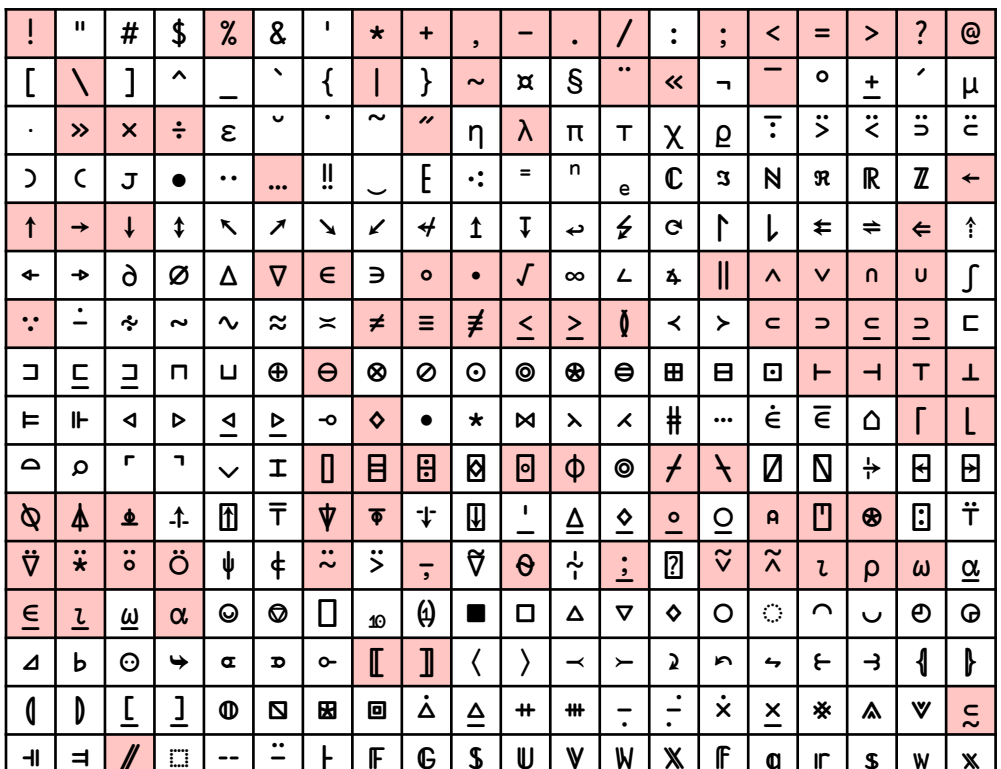
Support

Over 500 symbolic glyphs, including full support for Dyalog APL, APL64, GNU APL, TinyAPL,



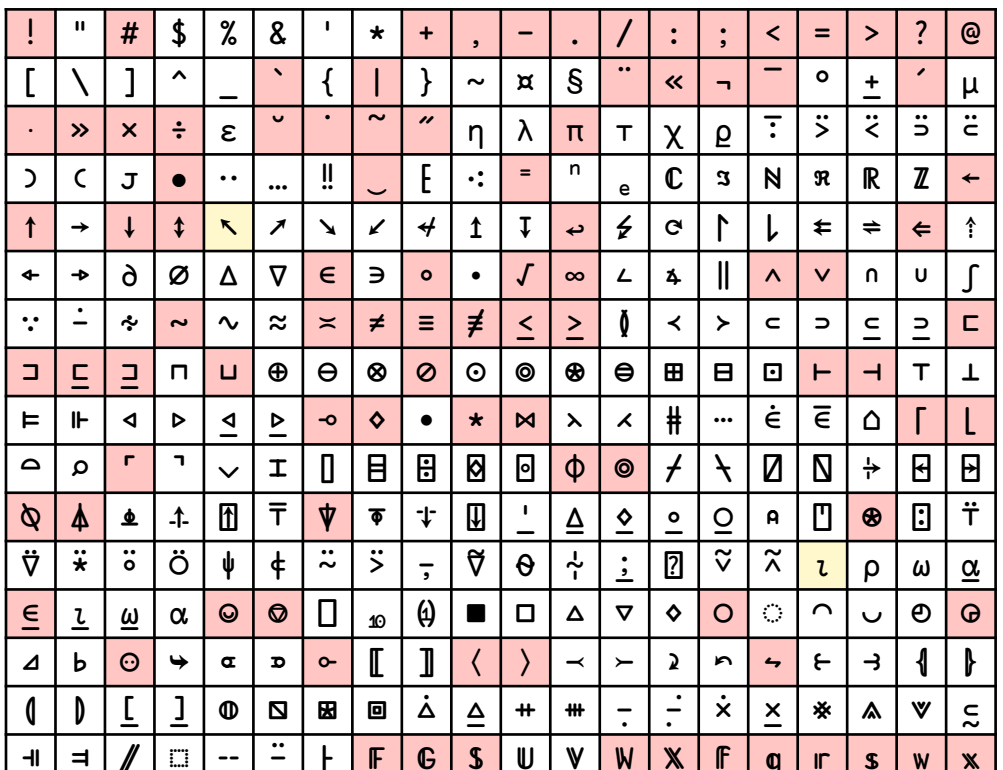
Support

Over 500 symbolic glyphs, including full support for Dyalog APL, APL64, GNU APL, TinyAPL, Kap,



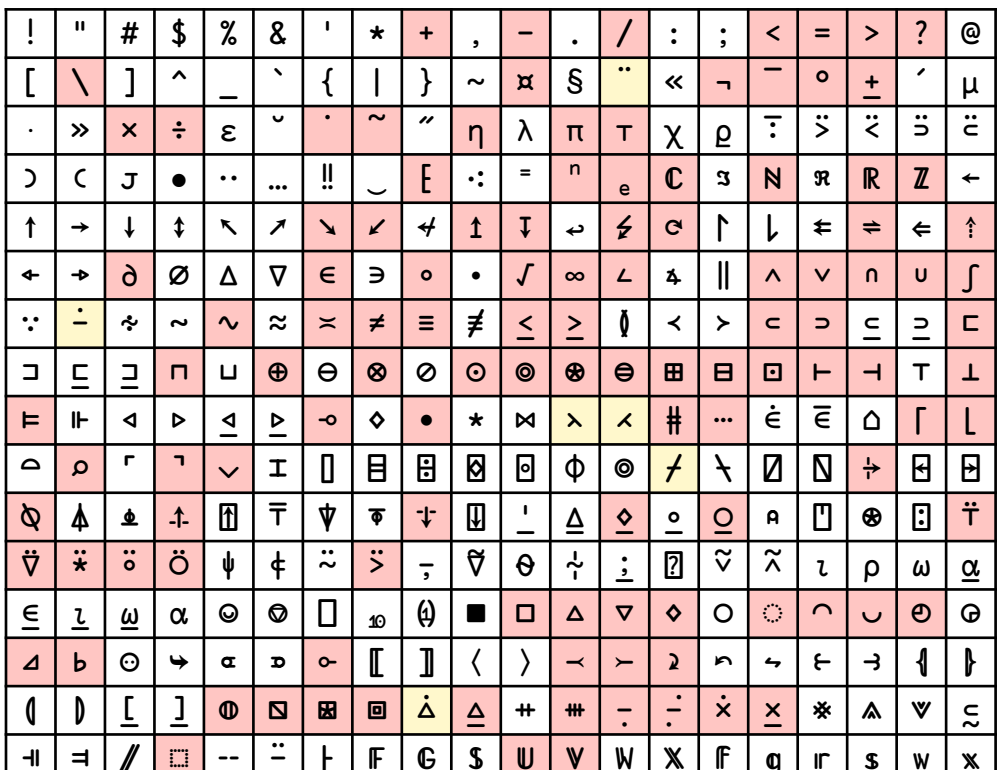
Support

Over 500 symbolic glyphs, including full support for Dyalog APL, APL64, GNU APL, TinyAPL, Kap, BQN,



Support

Over 500 symbolic glyphs, including full support for Dyalog APL, APL64, GNU APL, TinyAPL, Kap, BQN, Uiua,



Support

Over 500 symbolic glyphs, including full support for Dyalog APL, APL64, GNU APL, TinyAPL, Kap, BQN, Uiua, …

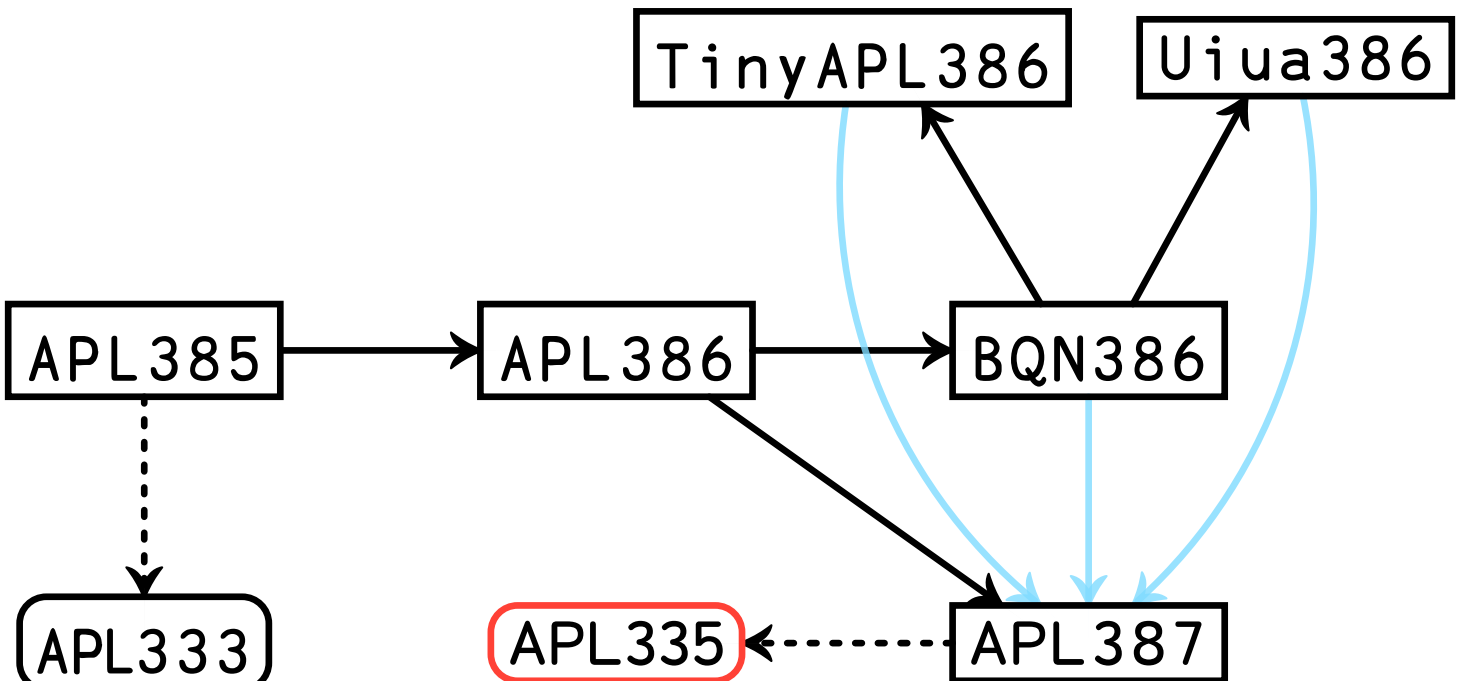


Support

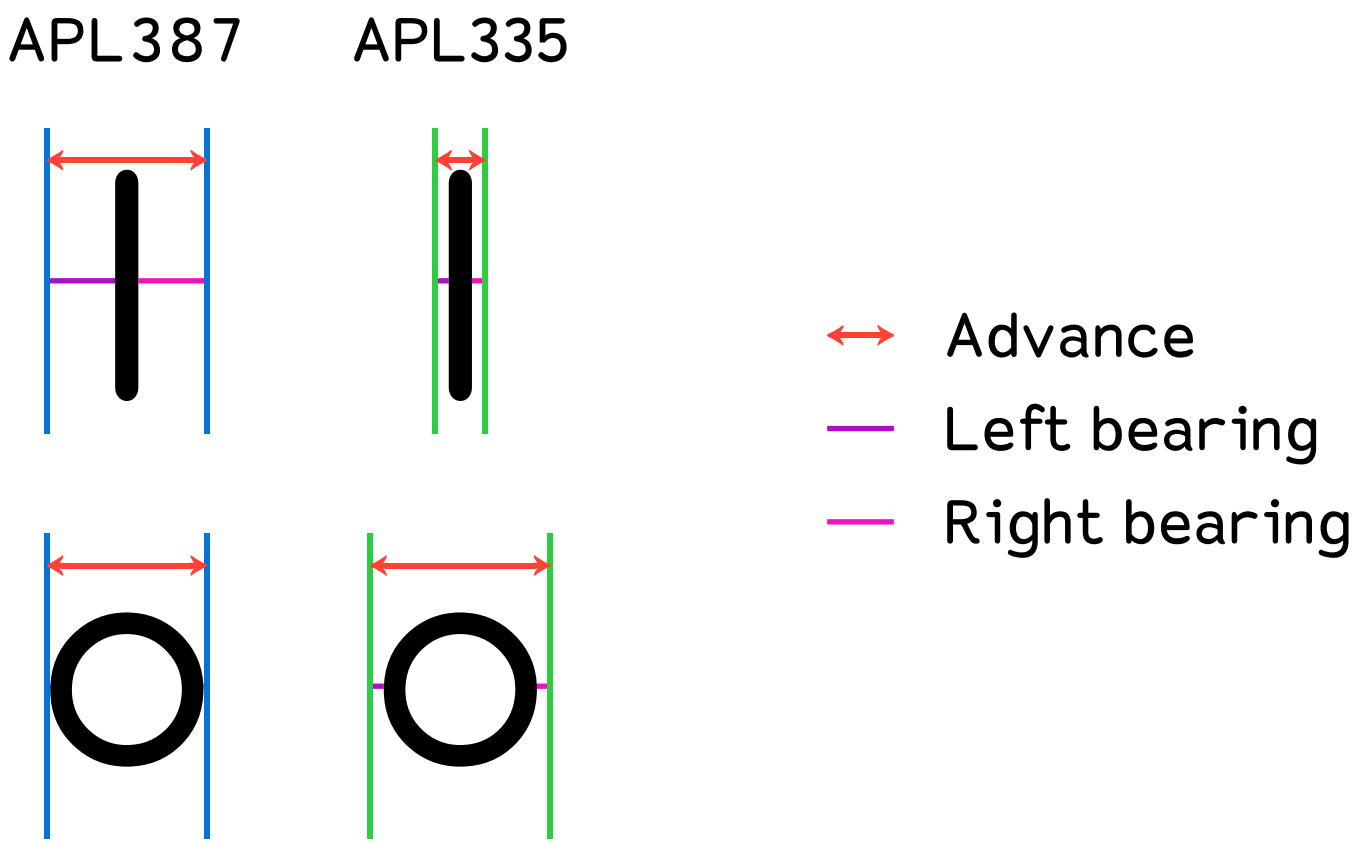
Over 500 symbolic glyphs, including full support for Dyalog APL, APL64, GNU APL, TinyAPL, Kap, BQN, Uiua, …, your notation?



More history



Proportionalization



Proportional substitutions

I M W m w

APL387

I M W m w

APL335

Examples

```
w←⊃(⊃0⍴⍵){
      (e a)←⌽⍺
      T←⍬⍤(0>⊃⍬⍺)
      Pad←⍵⍵⍀(T⊣)⌿⍵⌿(T⊢)
      need←(1+e),1↓⍴⍵
      a=0:(  1↓need⍴0↑⍵)Pad( 1↓need⍴0↑⊢⍵)
      a=1:( ¯1↓need⍴1↑⍵)Pad(¯1↓need⍴1↑⊖⍵)
      a=2:(⊖¯1↓need⍴  ⍵)Pad(¯1↓need⍴  ⊖⍵)
      a=3:(⊖ 1↓need⍴  ⍵)Pad( 1↓need⍴  ⊖⍵)
      a=4:(⊖¯1↓need⍴ ⊖⍵)Pad(¯1↓need⍴   ⍵)
}(¯1⍬⍳≠⍴⍵)/(⍬extra,⍤⍺⊣0),⊂⍵
```

Examples

APL (named after the book A Programming Language) is a programming language developed in the 1960s by Kenneth E. Iverson. Its central datatype is the multidimensional array. It uses a large range of special graphic symbols to represent most functions and operators, leading to very concise code. It has been an important influence on the development of concept modeling, spreadsheets, functional programming, and computer math packages. It has also inspired several other programming languages.

Examples

```
w←⊃(⊃0⍴⍵){
  (e a)←⌈⍺
  T←⍞⍤(0>⊃⍞⍺)
  Pad←⍵⍵⍀(T⊣)⌿⍵⌿(T⊢)
  need←(1+e),1↓⍴⍵
  a=0:(1↓need⍴0↑⍵)Pad(1↓need⍴0↑⍵)
  a=1:(1↓need⍴1↑⍵)Pad(1↓need⍴1↑⊖⍵)
  a=2:(⊖¯1↓need⍴⍵)Pad(¯1↓need⍴⊖⍵)
  a=3:(⊖1↓need⍴⍵)Pad(1↓need⍴⊖⍵)
  a=4:(⊖¯1↓need⍴⊖⍵)Pad(¯1↓need⍴⍵)
}(¯1⍞⍳≠⍴⍵)/(⍞extra,⍤⍺⊣0),⊂⍵
```

Examples

APL (named after the book A Programming Language) is a programming language developed in the 1960s by Kenneth E. Iverson. Its central datatype is the multidimensional array. It uses a large range of special graphic symbols to represent most functions and operators, leading to very concise code. It has been an important influence on the development of concept modeling, spreadsheets, functional programming, and computer math packages. It has also inspired several other programming languages.

What should this family be called?

What should this family be called?

https://forms.gle/SRtE3D4wKVwdVPZk9

https://dyalog.github.io/APL387



Please share your suggestions!