

Universal Slides

Behdad Esfahbod

behdad@google.com

<https://goo.gl/yWYUrd>

https://youtu.be/1K_my0awb0U

IUC 39—October 2015, Santa Clara

HarfBuzz Indic Shaper



- Maintain three diverging Indic shapers
- Font vs shaper bug / feature / responsibility
- Existing implementations too *magical*

⇒ Develop a new shaper “to rule ‘em all”!

Universal shaping

Reducing the cost of shaping support



Previously at IUC

"One Font To Rule Them All"

- Super OpenType/CFF Collection—*Super OTC*
 - All 28 weight/language combinations packed into a single font resource!
 - Saves 10MB 🖱️ *113MB versus 123MB*
 - 458,745 total glyphs



Previously at IUC

“One standard to rule them all”



Previously at IUC



The Fire Hose

Narrow by: [Product](#)

Follow us: [f](#) [t](#) [in](#)

[Subscribe all](#)

[Sign in](#)

Windows' global reach expands with Universal Shaping Engine for language scripts



Sample of Soyombo script showing the consistent vertical descenders.

Posted February 23, 2015

[f](#) 0 [t](#) 0 [in](#) 25

With Windows 10, Windows moves forward in becoming a truly global product with the help of a Universal Shaping Engine used for [complex text layout](#), which is needed for about half of the world's writing systems. Systems now supported using this engine in Windows 10 include English, Hindi, Japanese, Malayalam, and Sinhala.

Featured Posts

Microsoft's first flagship store opens in New York City, showcases innovation



Microsoft is opening its first-ever flagship store in New York ... [Read more »](#)

Read the inside story behind the making of 'Halo 5: Guardians'



Jeremie Sloan's garage in Oregon City teams with life-size "Halo" ... [Read more »](#)

Microsoft researchers and Red Bull Air Race pilot work together to solve future robot navigation challenges



Microsoft researchers are working with Red Bull Air Race pilot ... [Read more »](#)

cluster model—spec

3. Consonant cluster

```
[R] < B | GB > [VS] (CMAbv)* (CMBlw)* (< H B | SUB > [VS] (CMAbv)* (CMBlw)*)*  
      [MPre] [MAbv] [MBlw] [MPst]  
      (VPre)* (VAbv)* (VBlw)* (VPst)*  
      (VMPre)* (VMAbv)* (VMBlw)* (VMPst)*  
      (FAbv)* (FBlw)* (FPst)* [FM]
```

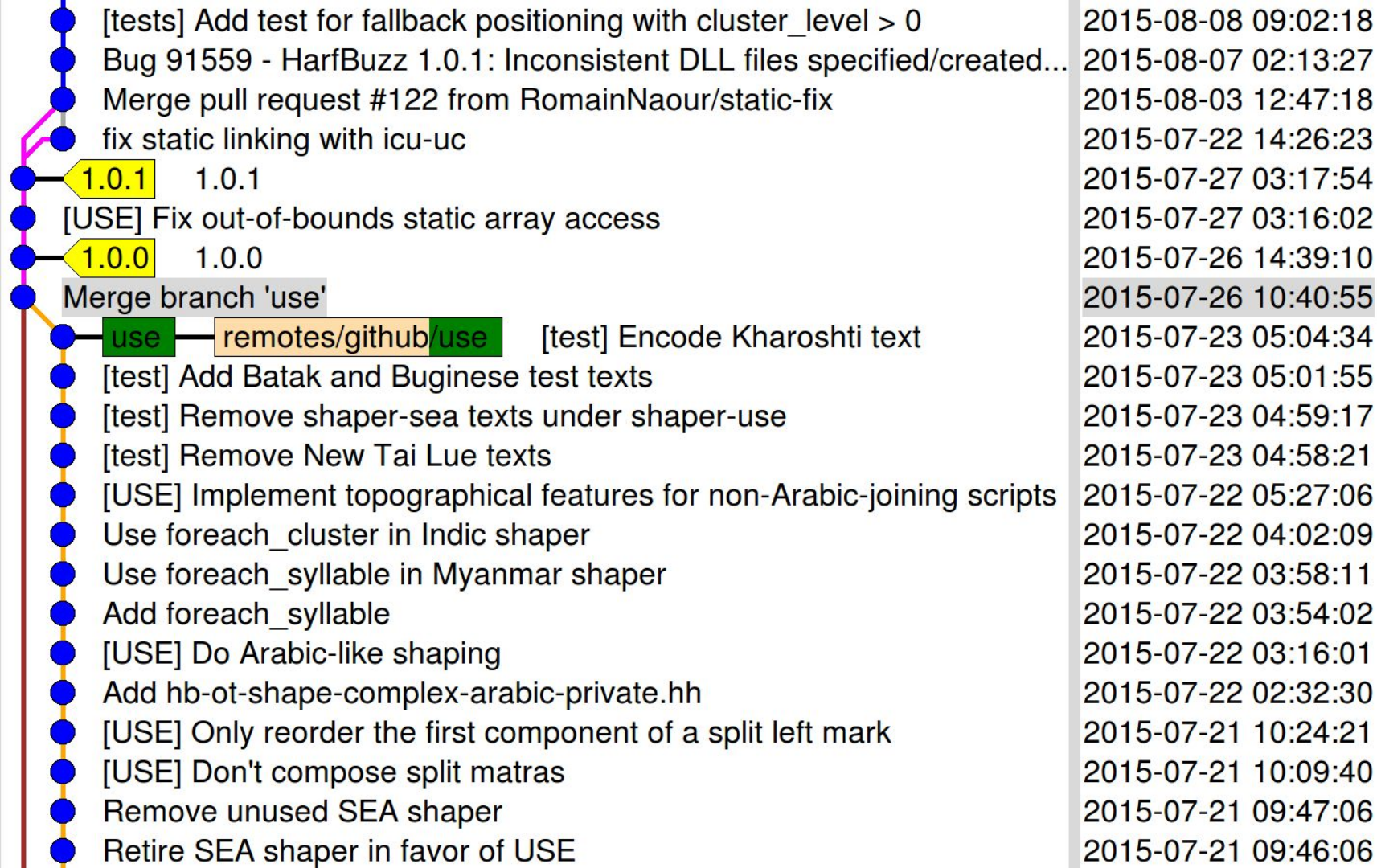
cluster model—code

```
consonant_modifiers = CMABv* CMBlw* ((H B | SUB) VS? CMABv? CMBlw*)*;
medial_consonants = MPre? MABv? MBlw? MPst?;
dependent_vowels = VPre* VAbv* VBlw* VPst*;
vowel_modifiers = VMPre* VMABv* VMBlw* VMPst*;
final_consonants = FAbv* FBlw* FPst* FM?;
```

```
consonant_cluster =
    R? (B | GB) VS?
    consonant_modifiers
    medial_consonants
    dependent_vowels
    vowel_modifiers
    final_consonants
;
```


● Add FLAG_SAFE() for values known to be small-enough	2015-07-21 07:50:02
● Add note re ASSERT_STATIC_EXPR_ZERO()	2015-07-21 07:45:48
● Remove unused macro ASSERT_STATIC_EXPR()	2015-07-21 07:17:27
● Use unsigned in FLAG()	2015-07-21 07:05:35
● [USE] Use use_category() for rphf/pref memory	2015-07-21 06:51:45
● [USE] Hook of rphf and pref custom processing	2015-07-21 06:15:35
● [USE] Add CGJ to table	2015-07-21 03:57:23
● [USE] Set up features	2015-07-21 02:24:32
● [USE] Hook up new scripts to USE shaper	2015-07-21 02:02:04
● [USE] Remove unused Unicode data enums from USE C++ side	2015-07-21 01:31:19
● [USE] Finish converting Unicode positional categories to USE	2015-07-20 10:01:10
● [USE] Map from Unicode data to USE syllabic categories	2015-07-20 09:00:06
● [USE] Start moving Unicode-to-USE mapping into Python code	2015-07-20 07:56:19
● [USE] Fix Number clusters	2015-07-20 07:33:25
● [USE] Start putting together the shaper body	2015-07-20 06:24:55
● [USE] Put a Ragel machine together	2015-07-20 05:30:51
● Minor	2015-07-20 05:30:45
● [USE] Minor optimization of USE table	2015-07-20 04:01:20
● [USE] Add Kharoshti test data from Unicode proposal	2015-07-20 03:57:44
● [USE] Build Universal Shaping Engine data table from Unicode 8 files	2015-07-20 03:46:17
● [USE] Add Universal Shaping Engine to Makefile	2015-07-20 03:32:48
● 0.9.42 0.9.42	2015-07-26 10:29:53

● [USE] Only reorder the first component of a split left mark	2015-07-21 10:24:21
● [USE] Don't compose split matras	2015-07-21 10:09:40
● Remove unused SEA shaper	2015-07-21 09:47:06
● Retire SEA shaper in favor of USE	2015-07-21 09:46:06
● [USE] Only use USE shaper if script system is not DFLT	2015-07-21 09:31:43
● Route misc untested scripts through USE shaper instead of Indic	2015-07-21 09:24:18
● [USE] Implement dotted-circle	2015-07-21 09:14:54
● [USE] Move pref	2015-07-21 08:55:26
● [USE] Fix-up variation selectors and word joiner in table	2015-07-21 08:46:37
● [USE] Use a couple warnings	2015-07-21 08:43:27
● [USE] Move rphf	2015-07-21 08:07:10
● Add FLAG_UNSAFE()	2015-07-21 07:52:15
● Add FLAG_SAFE() for values known to be small-enough	2015-07-21 07:50:02
● Add note re ASSERT_STATIC_EXPR_ZERO()	2015-07-21 07:45:48
● Remove unused macro ASSERT_STATIC_EXPR()	2015-07-21 07:17:27
● Use unsigned in FLAG()	2015-07-21 07:05:35
● [USE] Use use_category() for rphf/pref memory	2015-07-21 06:51:45
● [USE] Hook of rphf and pref custom processing	2015-07-21 06:15:35
● [USE] Add CGJ to table	2015-07-21 03:57:23
● [USE] Set up features	2015-07-21 02:24:32
● [USE] Hook up new scripts to USE shaper	2015-07-21 02:02:04
● [USE] Remove unused Unicode data enums from USE C++ side	2015-07-21 01:31:19



indic

```
behdad:src 0$ wc -l hb-ot-shape-complex-indic*
```

```
1843 hb-ot-shape-complex-indic.cc
```

```
1762 hb-ot-shape-complex-indic-machine.hh
```

```
130 hb-ot-shape-complex-indic-machine.rl
```

```
180 hb-ot-shape-complex-indic-private.hh
```

```
956 hb-ot-shape-complex-indic-table.cc
```

```
4871 total
```

use

```
behdad:src 0$ wc -l hb-ot-shape-complex-use*
```

```
585 hb-ot-shape-complex-use.cc
```

```
548 hb-ot-shape-complex-use-machine.hh
```

```
180 hb-ot-shape-complex-use-machine.rl
```

```
97 hb-ot-shape-complex-use-private.hh
```

```
696 hb-ot-shape-complex-use-table.cc
```

```
2106 total
```

myanmar

```
behdad:src 0$ wc -l hb-ot-shape-complex-myanmar*
```

```
532 hb-ot-shape-complex-myanmar.cc
```

```
400 hb-ot-shape-complex-myanmar-machine.hh
```

```
128 hb-ot-shape-complex-myanmar-machine.rl
```

```
1060 total
```

source code

hb-ot-shape-complex-**indic**:initial_reordering_consonant_syllable()

470 lines ~50% comments

hb-ot-shape-complex-**indic**:final_reordering_syllable()

405 lines ~50% comments

hb-ot-shape-complex-**use**:reorder_syllable()

63 lines 15% comments

hb-ot-shape-complex-**myanmar**:initial_reordering_consonant_syllable()

87 lines 5% comments

machine code

```
behdad:~$ size hb-ot-shape-complex-indic*.o
```

text	data	bss	dec	hex	filename
37220	88	0	37308	91bc	hb-ot-shape-complex-indic.o
9940	0	0	9940	26d4	hb-ot-shape-complex-indic-table.o

```
behdad:~$ size hb-ot-shape-complex-use*.o
```

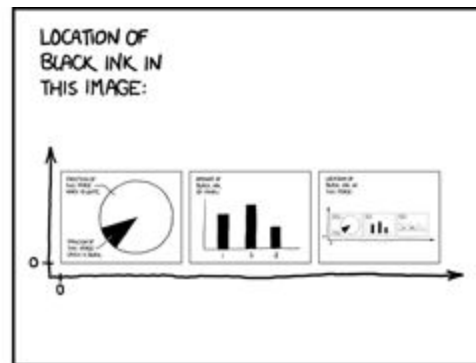
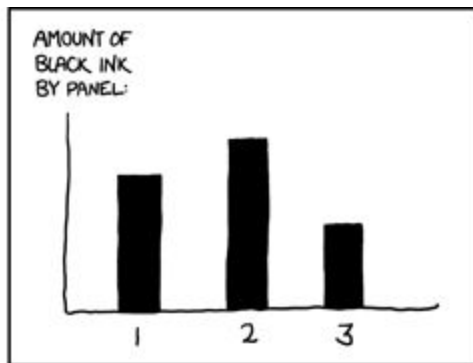
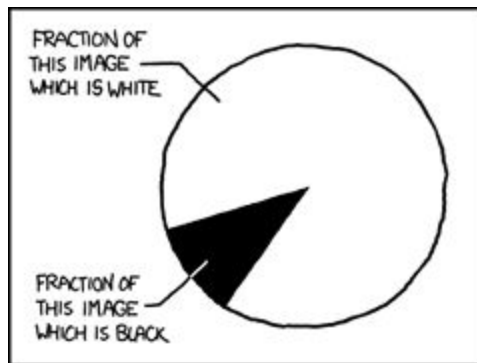
text	data	bss	dec	hex	filename
8238	88	0	8326	2086	hb-ot-shape-complex-use.o
5770	0	0	5770	168a	hb-ot-shape-complex-use-table.o

```
behdad:~$ size hb-ot-shape-complex-myanmar*.o
```

text	data	bss	dec	hex	filename
5545	88	0	5633	1601	hb-ot-shape-complex-myanmar.o

speed

speed



arabic



W3C Internationalization Activity

ArabicScript Layout Task Force

www.w3.org/International/groups/arabic-layout

Behnam Esfahbod (Facebook) <zwnj@fb.com>

Shervin Afshar (Netflix) <safshar@netflix.com>



W3C Internationalization Activity

Arabic Script Layout Task Force

www.w3.org/International/groups/arabic-layout

Behnam Esfahbod (Facebook) <zwnj@fb.com>

Shervin Afshar (Netflix) <safshar@netflix.com>



W3C Internationalization Activity

Arabic Script Layout Task Force

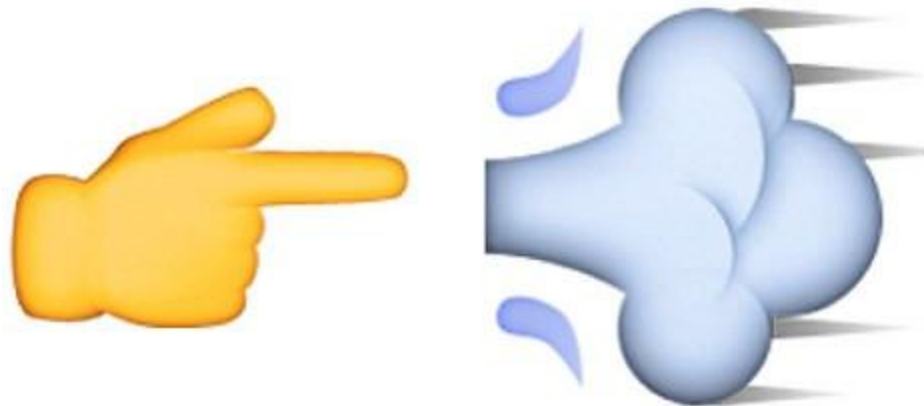
www.w3.org/International/groups/arabic-layout

Behnam Esfahbod (Facebook) <zwnj@fb.com>

Shervin Afshar (Netflix) <safshar@netflix.com>







universal



No emoji found.



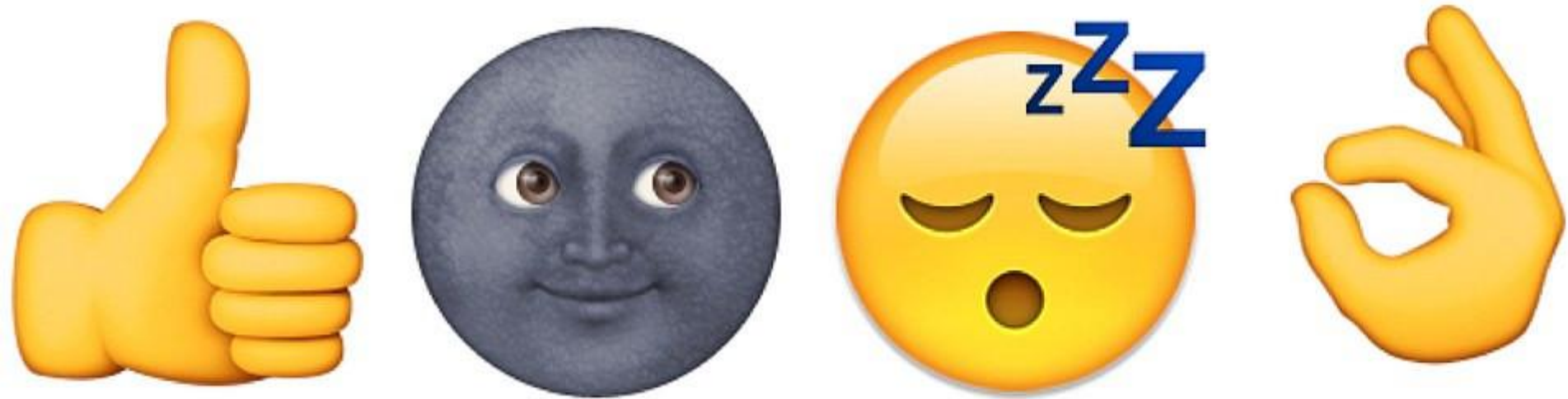
universe



No emoji found.





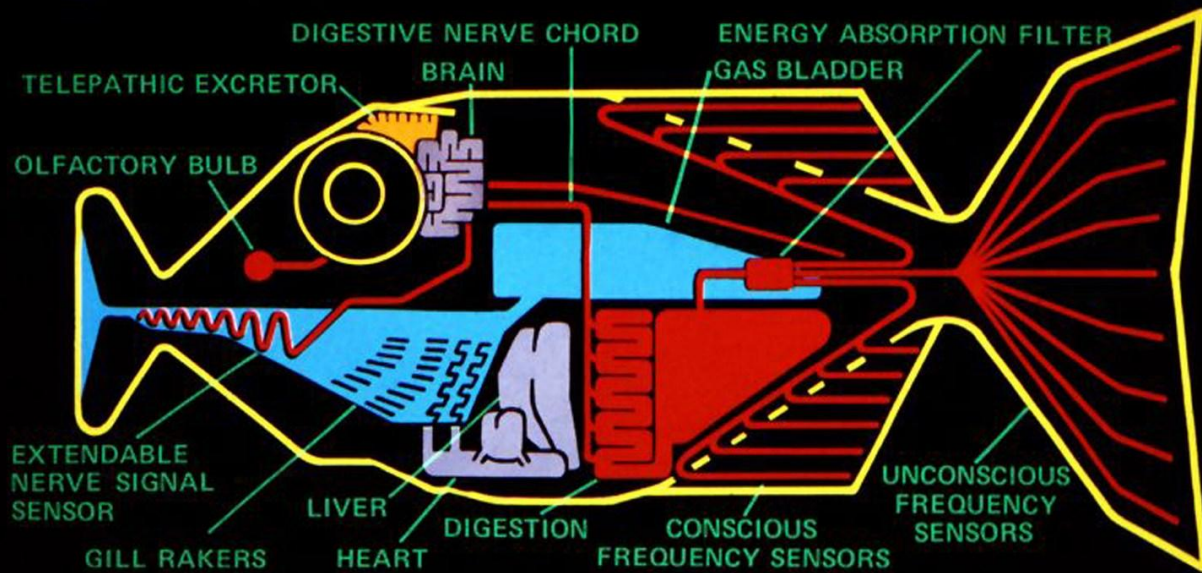




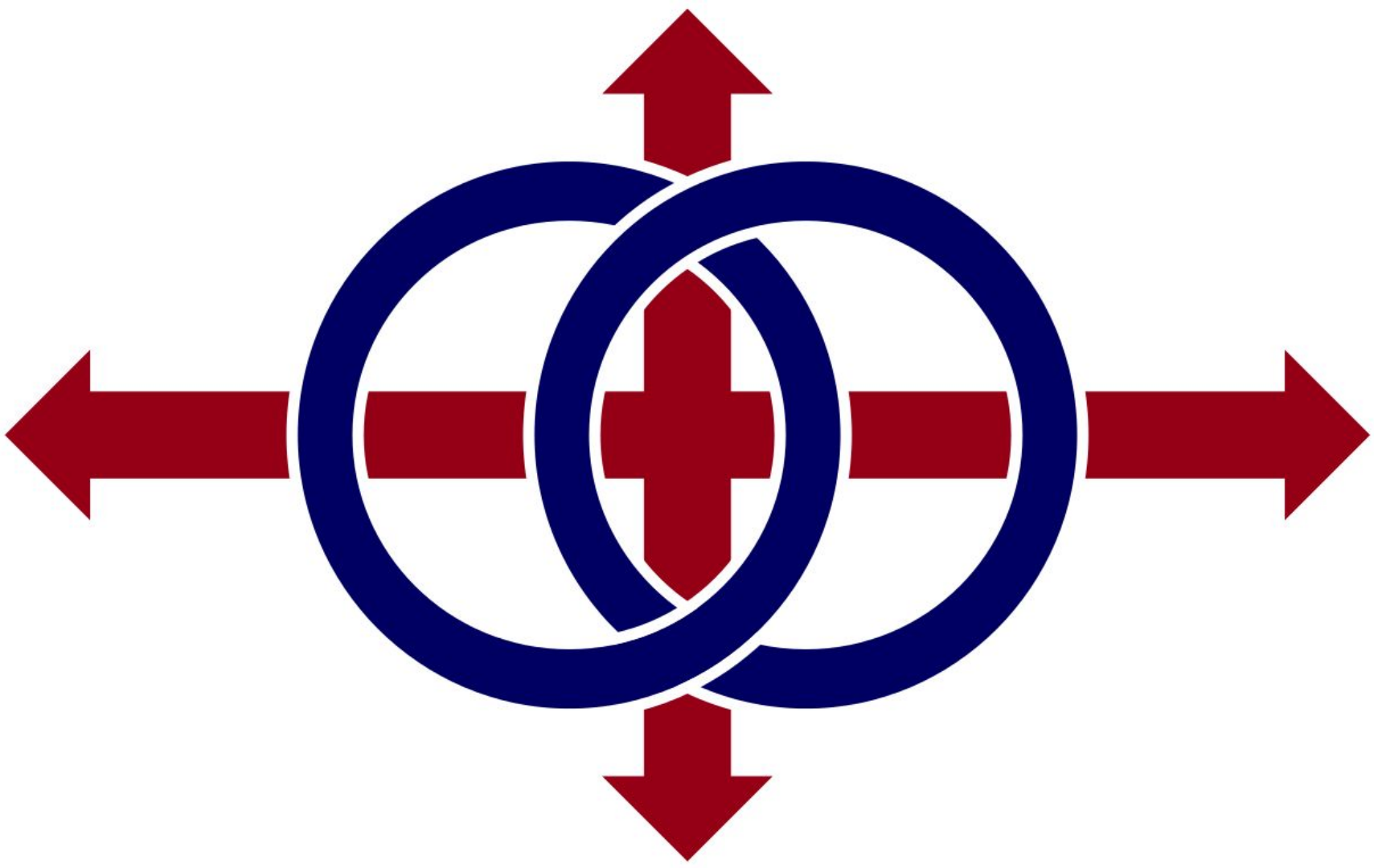


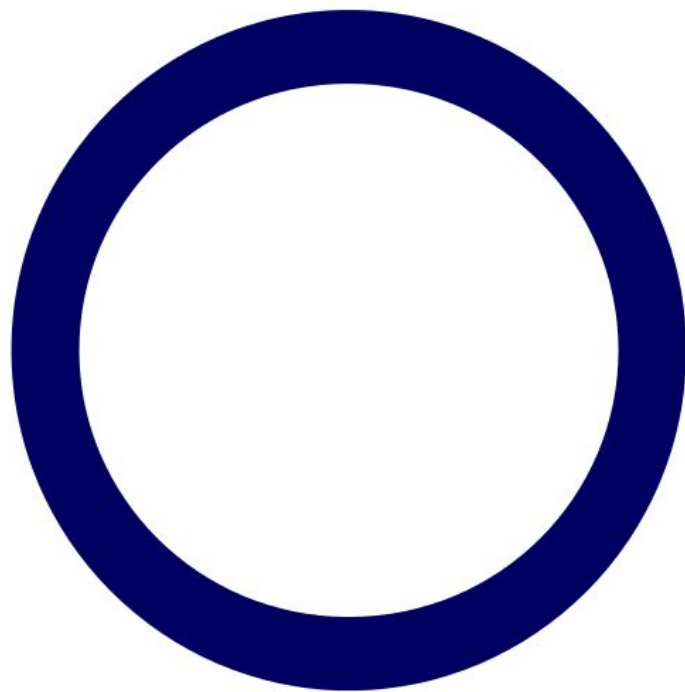
YOU AND YOUR
PEOPLE ARE IN DANGER.

BABEL FISH

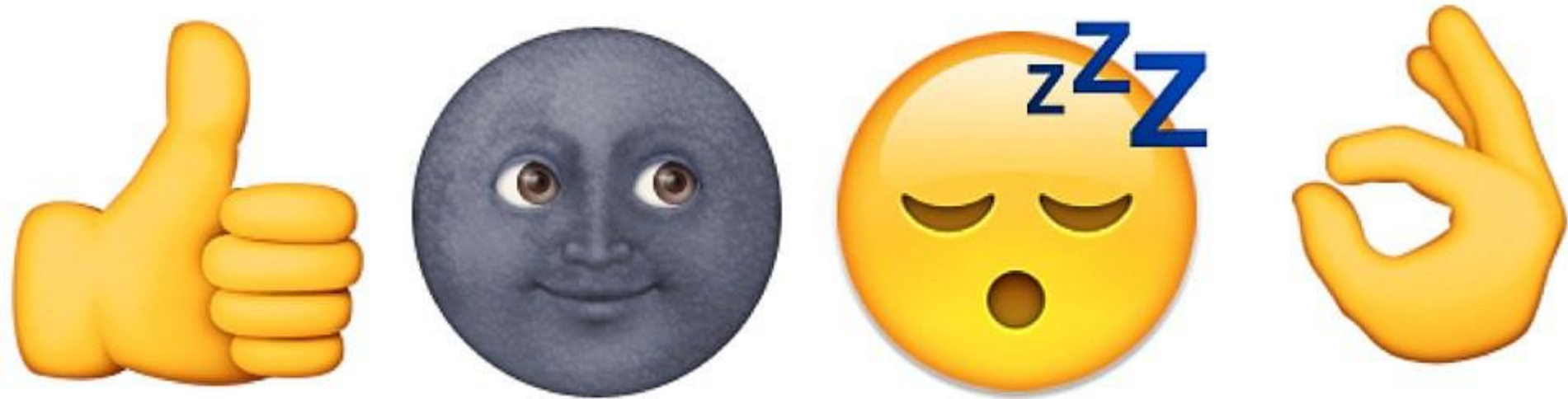


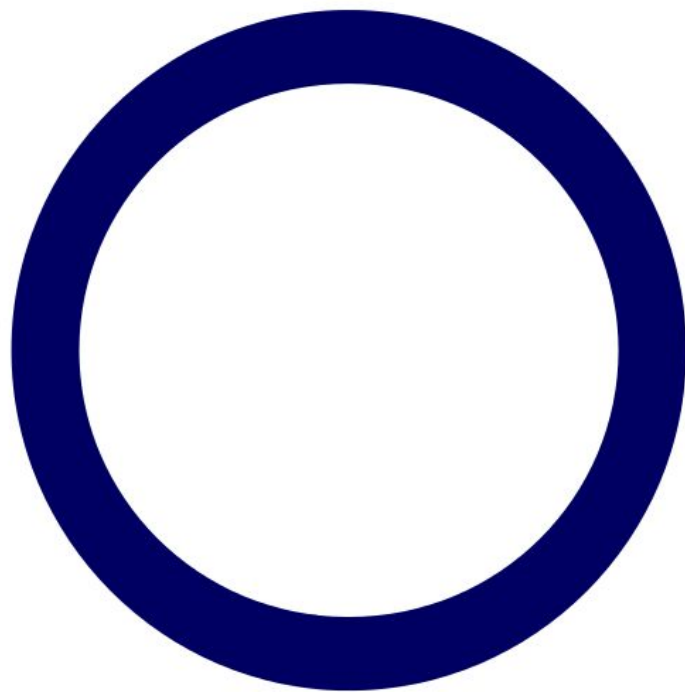
THE BABEL FISH IS SMALL, YELLOW, LEECHLIKE,
AND PROBABLY THE ODDEST THING IN THE UNIVERSE.
IT FEEDS ON BRAIN WAVE ENERGY, ABSORBING ALL











10

11

00

10

01

1

^1

$\wedge 1 \$$

^01\$

^01\$\$

^10\$

10

^01\$

^A01\$

^A01\$

$\wedge \emptyset A 1 \$$

^ 0 A 1 \$

^ 0 1 A \$

^01A\$

^ 0 क 0 \$

^ ०क० \$

^ क 10 \$

10

01 → 10

...to ∞ and beyond...

lookup 1:

$x \ S \ y \rightarrow x \ 0 \ T$

$x \ S \ y \rightarrow x \ 1 \ T$

$x \ S \ y \rightarrow T \ x \ 0$

$x \ S \ y \rightarrow T \ x \ 1$

...

* *' lookup 1

lookup 0:

*

^_ lookup 1

^_ lookup 0

...to ∞ and beyond...

Universal

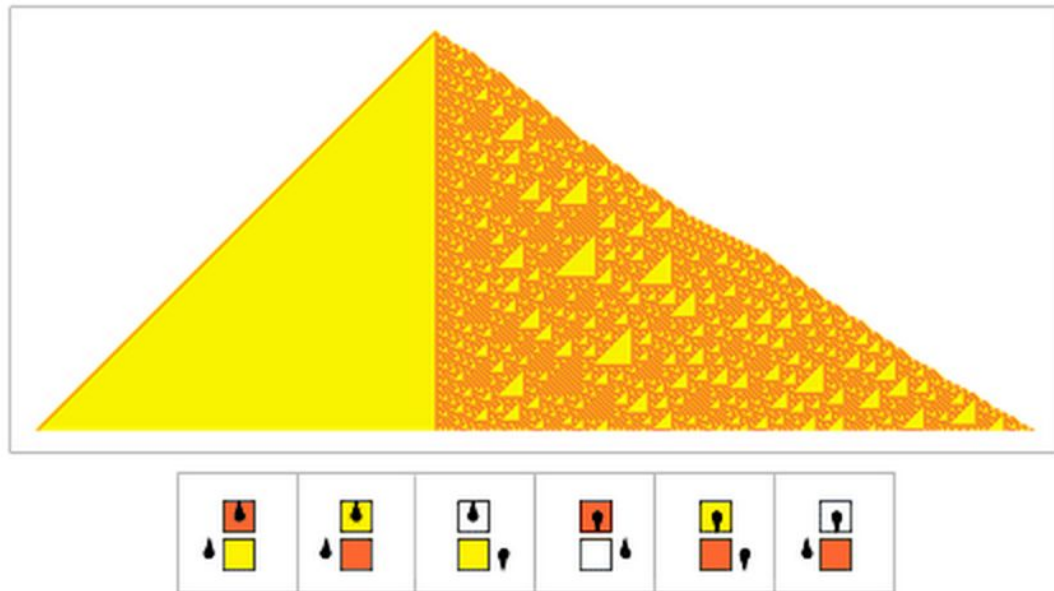


Machine

...but...

We are also at the end of a quest that has spanned more than half a century to find the very simplest universal Turing machine.

Here it is. Just two states and three colors. And able to do any computation that can be done.



We've come a long way since Alan Turing's original 1936 universal Turing machine—taking **four pages** of dense notation to describe.

