boxes and glue

TEXs algorithms re-implemented

TUG 2022
online conference
July 23 2022

boxes & glue
@boxesandglue

Patrick Gundlach
gundlach@speedata.de
@speedata

speedata GmbH
Berlin

speedata
Let's surpass the mainstream

What is »boxes and glue«?

boxes and glue …

… is a collection of software libraries

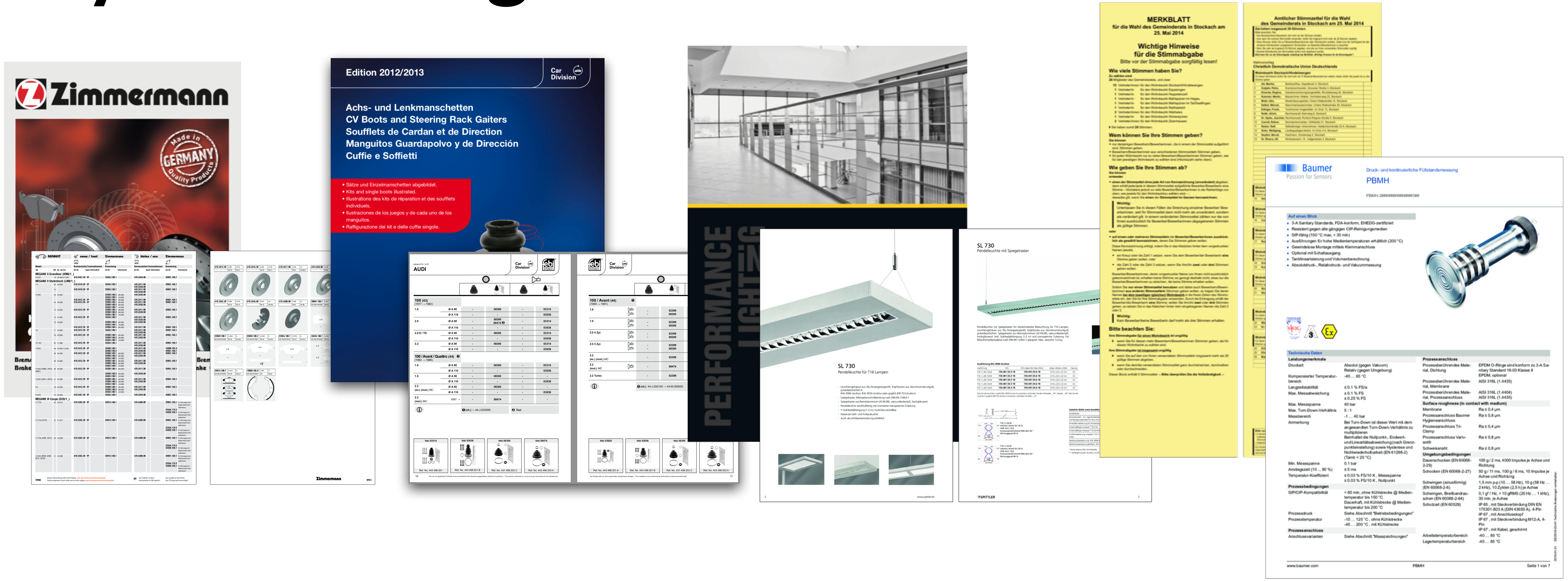
… not a ready-to-run piece of software

… written in the Go programming language

… the attempt to bring TEX's superb typesetting quality to a modern environment

… and of course OpenSource

Why boxes and glue?


Zimmermann

Made in GERMANY Quality Products

Edition 2012/2013

Car Division

Achs- und Lenkmanschetten
CV Boots and Steering Rack Gaiters
Soufflets de Cardan et de Direction
Manguitos Guardapolvo y de Dirección
Cuffie e Soffietti

- Sätze und Einzelmanschetten abgebildet.
- Kits and single boots illustrated.
- Illustrations des kits de réparation et des soufflets individuels.
- Ilustraciones de los juegos y de cada uno de los manguitos.
- Raffigurazione dei kit e delle cuffie singole.

AUDI

PERFORMANCE LIGHTING

SL 730

Pendelleuchte für T16 Lampen

SL 730

Pendelleuchte mit Spiegelraster

MERKBLATT
für die Wahl des Gemeinderats in Stockach am 25. Mai 2014

Wichtige Hinweise
für die Stimmabgabe

Bitte vor der Stimmabgabe sorgfältig lesen!

Wie viele Stimmen haben Sie?

Wem können Sie Ihre Stimmen geben?

Wie geben Sie Ihre Stimmen ab?

Bitte beachten Sie:

Amtlicher Stimmzettel für die Wahl des Gemeinderats in Stockach am 25. Mai 2014

Baumer
Passion for Sensors

Druck- und kontinuierliche Füllstandsmessung

PBMH

Technische Daten

(Fully automatic) catalog production with LuaTEX

How does it work?

Baumer
Passion for Sensors

Druck- und kontinuierliche Füllstandsmessung

PBMH

PBMH-2################O##

Auf einen Blick

- 3-A Sanitary Standards, FDA-konform, EHEDG-zertifiziert
- Resistent gegen alle gängigen CIP-Reinigungsmedien
- SIP-fähig (150 °C max, < 30 min)
- Ausführungen für hohe Medientemperaturen erhältlich (200 °C)
- Gewindelose Montage mittels Klemmanschluss
- Optional mit Schaltausgang
- Tanklinearisierung und Volumenberechnung
- Absolutdruck-, Relativdruck- und Vakuummessung

Technische Daten

Leistungsmerkmale	
Druckart	Absolut (gegen Vakuum) Relativ (gegen Umgebung)
Kompensierter Temperatur-bereich	-40 … 85 °C
Langzeitstabilität	≤ 0.1 % FS/a
Max. Messabweichung	± 0.1 % FS ± 0.25 % FS
Max. Messspanne	40 bar
Max. Turn-Down-Verhältnis	5 : 1
Messbereich	-1 … 40 bar
Anmerkung	Bei Turn-Down ist dieser Wert mit dem angewandten Turn-Down-Verhältnis zu multiplizieren Beinhaltet die Nullpunkt-, Endwert- und Linearitätsabweichung (nach Grenz-punkteinstellung) sowie Hysterese und Nichtwiederholbarkeit (EN 61298-2) (Tamb = 20 °C)
Min. Messspanne	0.1 bar
Anstiegszeit (10 ... 90 %)	≤ 5 ms
Temperatur-Koeffizient	≤ 0.03 % FS/10 K , Messspanne ≤ 0.03 % FS/10 K , Nullpunkt
Prozessbedingungen	
SIP/CIP-Kompatibilität	< 60 min., ohne Kühlstrecke @ Medien-temperatur bis 150 °C Dauerhaft, mit Kühlstrecke @ Medien-temperatur bis 200 °C
Prozessdruck	Siehe Abschnitt "Betriebsbedingungen"
Prozesstemperatur	-10 … 125 °C , ohne Kühlstrecke -40 … 200 °C , mit Kühlstrecke
Prozessanschluss	
Anschlussvarianten	Siehe Abschnitt "Masszeichnungen"

Prozessanschluss	
Prozessberührendes Mate-rial, Dichtung	EPDM O-Ringe sind konform zu 3-A Sa-nitary Standard 18-03 Klasse II EPDM, optional
Prozessberührendes Mate-rial, Membrane	AISI 316L (1.4435)
Prozessberührendes Mate-rial, Prozessanschluss	AISI 316L (1.4404) AISI 316L (1.4435)
Surface roughness (in contact with medium)	
Membrane	Ra ≤ 0,4 µm
Prozessanschluss Baumer Hygieneanschluss	Ra ≤ 0,8 µm
Prozessanschluss Tri-Clamp	Ra ≤ 0,4 µm
Prozessanschluss Variv-ent®	Ra ≤ 0,8 µm
Schweissnaht	Ra ≤ 0,8 µm
Umgebungsbedingungen	
Dauerschocken (EN 60068-2-29)	100 g / 2 ms, 4000 Impulse je Achse und Richtung
Schocken (EN 60068-2-27)	50 g / 11 ms, 100 g / 6 ms, 10 Impulse je Achse und Richtung
Schwingen (sinusförmig) (EN 60068-2-6)	1,5 mm p-p (10 … 58 Hz), 10 g (58 Hz … 2 kHz), 10 Zyklen (2,5 h) je Achse
Schwingen, Breitbandrau-schen (EN 60068-2-64)	0,1 g² / Hz, > 10 gRMS (20 Hz … 1 kHz), 30 min. je Achse
Schutzart (EN 60529)	IP 65 , mit Steckverbindung DIN EN 175301-803 A (DIN 43650 A), 4-Pin IP 67 , mit Anschlusskopf IP 67 , mit Steckverbindung M12-A, 4-Pin IP 67 , mit Kabel, geschirmt
Arbeitstemperaturbereich	-40 … 85 °C
Lagertemperaturbereich	-40 … 85 °C

2019-01-21 DE/2018-03-01 Technische Änderungen vorbehalten

www.baumer.com PBMH Seite 1 von 7

All “things” (nodes) on a page can be created from the Lua side:

```
g = node.new("glyph")
g.char = "a"
g.font = 123
g.width = ...
g.height = ...

-- \hbox{a} in TeX
hlist = node.hpack(g)
```

boxes & glue

Why LuaTEX in the first place?

Very fast!

all features of TEX, but “better” programmable

flexible

contains Harfbuzz

LuaTEX limitations (subjective!)

Error handling

no https:// connections

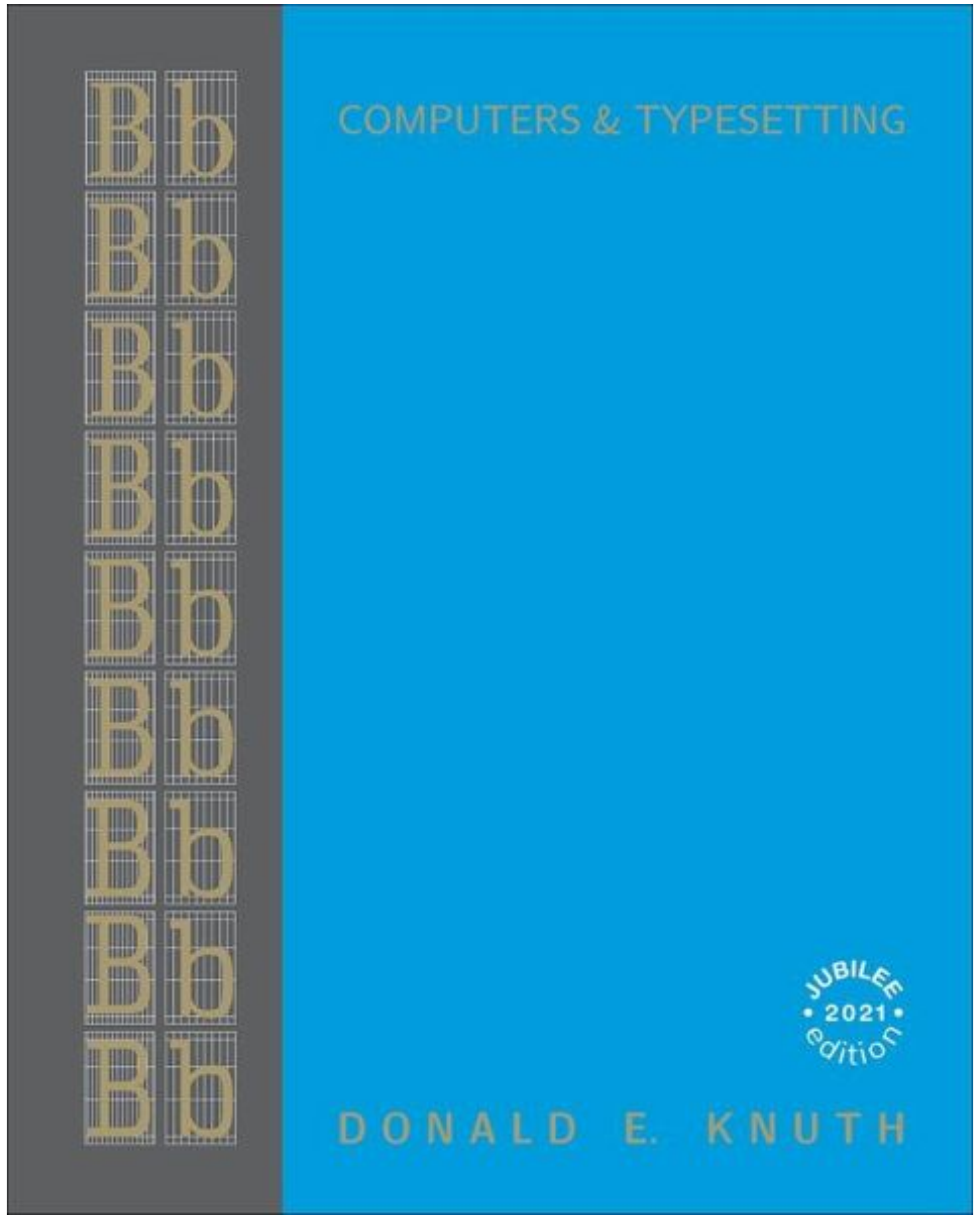
manual memory management

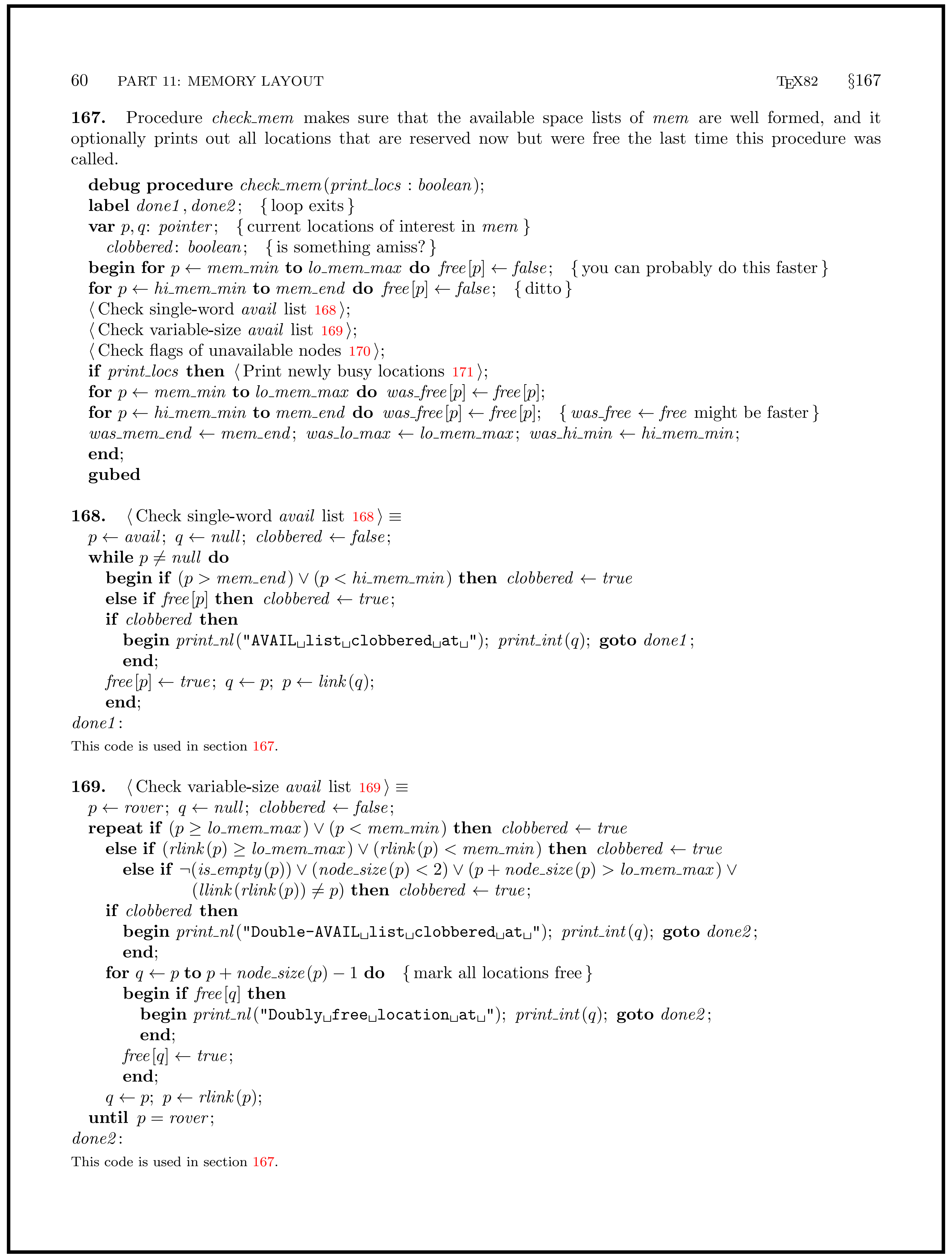
Lua is only suitable for smaller projects

hard to extend

… many little things mark life hard

Idea: Re-implementing TEX




60 PART 11: MEMORY LAYOUT TEX82 §167

167. Procedure *check_mem* makes sure that the available space lists of *mem* are well formed, and it optionally prints out all locations that are reserved now but were free the last time this procedure was called.

```
debug procedure check_mem(print_locs : boolean);
label done1, done2; { loop exits }
var p, q: pointer; { current locations of interest in mem }
  clobbered: boolean; { is something amiss? }
begin for p ← mem_min to lo_mem_max do free[p] ← false; { you can probably do this faster }
for p ← hi_mem_min to mem_end do free[p] ← false; { ditto }
⟨Check single-word avail list 168⟩;
⟨Check variable-size avail list 169⟩;
⟨Check flags of unavailable nodes 170⟩;
if print_locs then ⟨Print newly busy locations 171⟩;
for p ← mem_min to lo_mem_max do was_free[p] ← free[p];
for p ← hi_mem_min to mem_end do was_free[p] ← free[p]; { was_free ← free might be faster }
was_mem_end ← mem_end; was_lo_max ← lo_mem_max; was_hi_min ← hi_mem_min;
end;
gubed
```

168. ⟨Check single-word *avail* list 168⟩ ≡

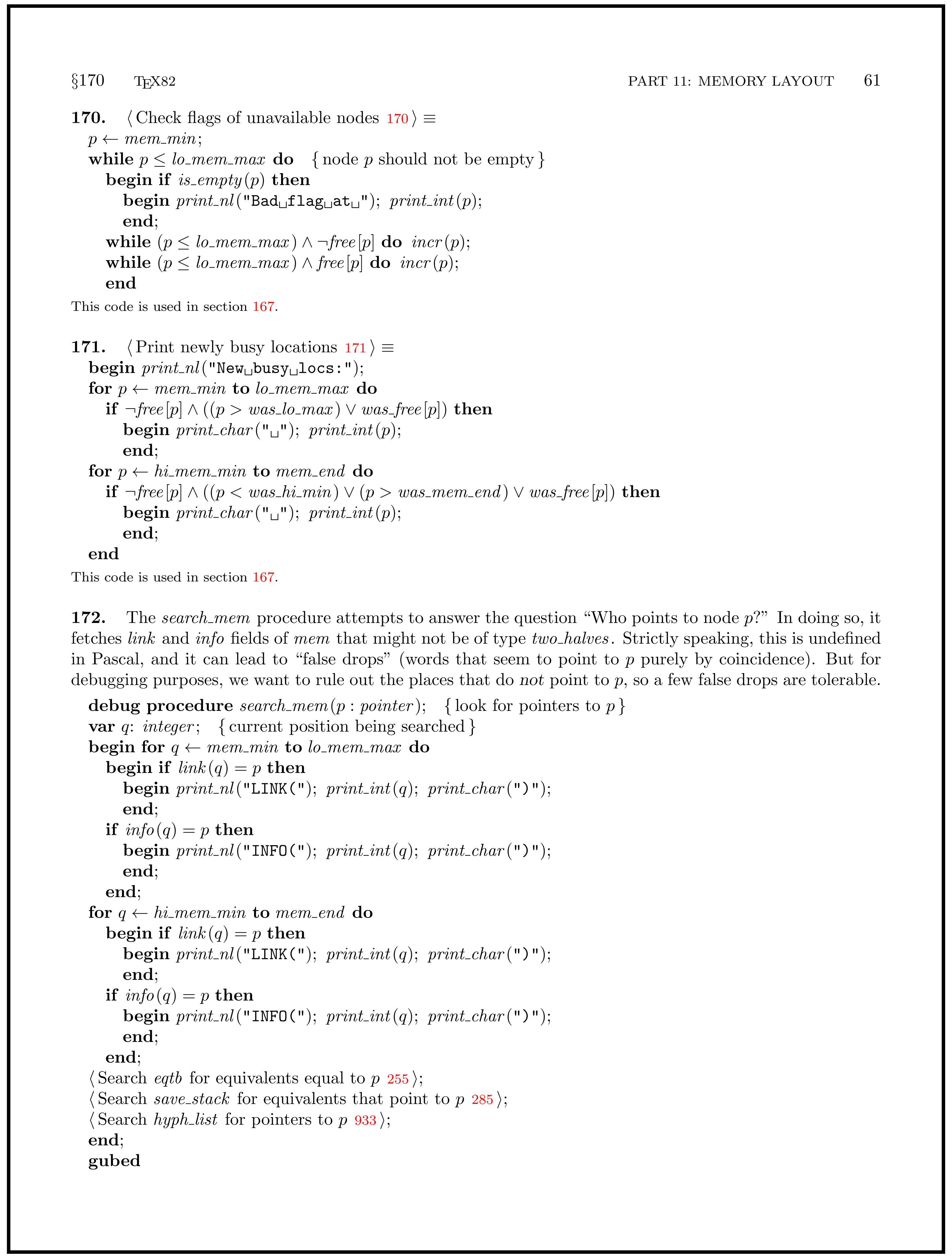
```
p ← avail; q ← null; clobbered ← false;
while p ≠ null do
  begin if (p > mem_end) ∨ (p < hi_mem_min) then clobbered ← true
  else if free[p] then clobbered ← true;
  if clobbered then
    begin print_nl("AVAIL␣list␣clobbered␣at␣"); print_int(q); goto done1;
    end;
  free[p] ← true; q ← p; p ← link(q);
  end;
done1:
```

This code is used in section 167.

169. ⟨Check variable-size *avail* list 169⟩ ≡

```
p ← rover; q ← null; clobbered ← false;
repeat if (p ≥ lo_mem_max) ∨ (p < mem_min) then clobbered ← true
  else if (rlink(p) ≥ lo_mem_max) ∨ (rlink(p) < mem_min) then clobbered ← true
    else if ¬(is_empty(p)) ∨ (node_size(p) < 2) ∨ (p + node_size(p) > lo_mem_max) ∨
          (llink(rlink(p)) ≠ p) then clobbered ← true;
  if clobbered then
    begin print_nl("Double-AVAIL␣list␣clobbered␣at␣"); print_int(q); goto done2;
    end;
  for q ← p to p + node_size(p) − 1 do { mark all locations free }
    begin if free[q] then
      begin print_nl("Doubly␣free␣location␣at␣"); print_int(q); goto done2;
      end;
    free[q] ← true;
    end;
  q ← p; p ← rlink(p);
until p = rover;
done2:
```

This code is used in section 167.


§170 TEX82 PART 11: MEMORY LAYOUT 61

170. ⟨Check flags of unavailable nodes 170⟩ ≡

```
p ← mem_min;
while p ≤ lo_mem_max do { node p should not be empty }
  begin if is_empty(p) then
    begin print_nl("Bad␣flag␣at␣"); print_int(p);
    end;
  while (p ≤ lo_mem_max) ∧ ¬free[p] do incr(p);
  while (p ≤ lo_mem_max) ∧ free[p] do incr(p);
  end
```

This code is used in section 167.

171. ⟨Print newly busy locations 171⟩ ≡

```
begin print_nl("New␣busy␣locs:");
for p ← mem_min to lo_mem_max do
  if ¬free[p] ∧ ((p > was_lo_max) ∨ was_free[p]) then
    begin print_char("␣"); print_int(p);
    end;
for p ← hi_mem_min to mem_end do
  if ¬free[p] ∧ ((p < was_hi_min) ∨ (p > was_mem_end) ∨ was_free[p]) then
    begin print_char("␣"); print_int(p);
    end;
end
```

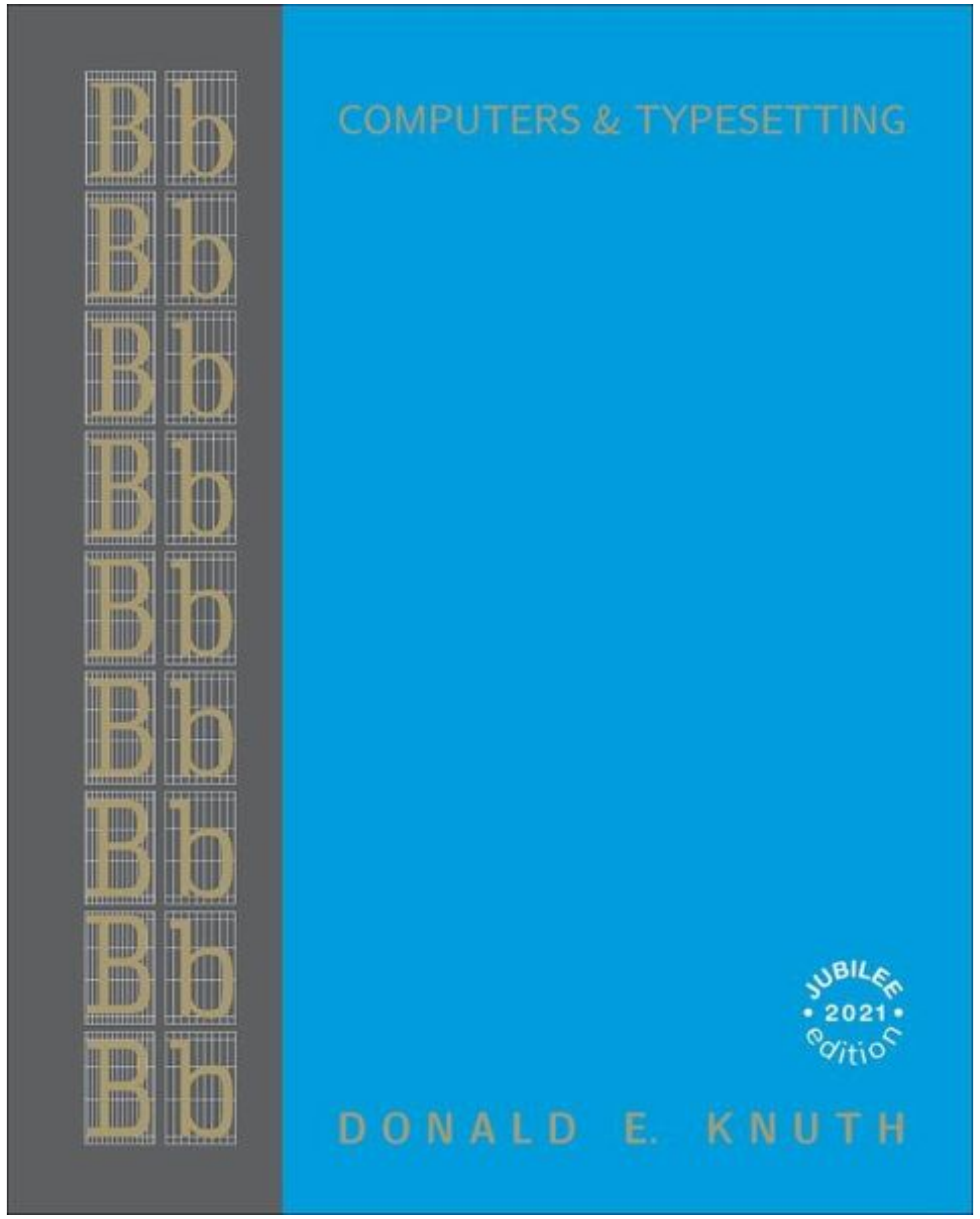
This code is used in section 167.

172. The *search_mem* procedure attempts to answer the question "Who points to node *p*?" In doing so, it fetches *link* and *info* fields of *mem* that might not be of type *two_halves*. Strictly speaking, this is undefined in Pascal, and it can lead to "false drops" (words that seem to point to *p* purely by coincidence). But for debugging purposes, we want to rule out the places that do *not* point to *p*, so a few false drops are tolerable.

```
debug procedure search_mem(p : pointer); { look for pointers to p }
var q: integer; { current position being searched }
begin for q ← mem_min to lo_mem_max do
  begin if link(q) = p then
    begin print_nl("LINK("); print_int(q); print_char(")");
    end;
  if info(q) = p then
    begin print_nl("INFO("); print_int(q); print_char(")");
    end;
  end;
for q ← hi_mem_min to mem_end do
  begin if link(q) = p then
    begin print_nl("LINK("); print_int(q); print_char(")");
    end;
  if info(q) = p then
    begin print_nl("INFO("); print_int(q); print_char(")");
    end;
  end;
⟨Search eqtb for equivalents equal to p 255⟩;
⟨Search save_stack for equivalents that point to p 285⟩;
⟨Search hyph_list for pointers to p 933⟩;
end;
gubed
```

boxes & glue

Idea: Re-implementing TEX



Web (Pascal) code

C-code from LuaTEX

Complete re-implementation*

Compatibility?

```
**function** *badness*($t, s$ : *scaled*): *halfword*;   { compute badness, given $t \ge 0$ }
  **var** $r$: *integer*;   { approximation to $\alpha t/s$, where $\alpha^3 \approx 100 \cdot 2^{18}$ }
  **begin if** $t = 0$ **then** *badness* $\leftarrow 0$
  **else if** $s \le 0$ **then** *badness* $\leftarrow$ *inf_bad*
    **else begin if** $t \le 7230584$ **then** $r \leftarrow (t * 297)$ **div** $s$   { $297^3 = 99.94 \times 2^{18}$ }
      **else if** $s \ge 1663497$ **then** $r \leftarrow t$ **div** ($s$ **div** 297)
        **else** $r \leftarrow t$;
      **if** $r > 1290$ **then** *badness* $\leftarrow$ *inf_bad*   { $1290^3 < 2^{31} < 1291^3$ }
      **else** *badness* $\leftarrow (r * r * r + '400000)$ **div** $'1000000$;
      **end**;   { that was $r^3/2^{18}$, rounded to the nearest integer }
  **end**;
```

$$100\left(\frac{t}{s}\right)^3$$

```
100.0 * math.Pow(t/s, 3)
```

boxes & glue

Algorithms in TEX

Hyphenation

Breaking paragraphs into lines

Math typesetting

Calculation of lenghts (`\hfill`, badness and so on)

Input language

Algorithms: Hyphenation

Start: list of hyphenation patterns

Algorithms: Hyphenation

Start: list of hyphenation patterns

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
autobahn
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
a u t o b a h n
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

a u t o b a h n

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
a u t o b a h n
a2u
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

boxes & glue

Algorithms: Hyphenation

```
a u t o b a h n
a2u
   1t0o
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
  a u t o b a h n
  a2u
     1t0o
 1a0u0t0o
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

boxes & glue

Algorithms: Hyphenation

```
  a u t o b a h n
  a2u
     1t0o
 1a0u0t0o
   2u1t
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
 a u t o b a h n
 a2u
    1t0o
1a0u0t0o
  2u1t
     t2o0b0a
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
  a u t o b a h n
  a2u
     1t0o
 1a0u0t0o
   2u1t
      t2o0b0a
        o1b
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
 a u t o b a h n
 a2u
    1t0o
1a0u0t0o
  2u1t
     t2o0b0a
       o1b
        3b0a0h
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
 a u t o b a h n
 a2u
    1t0o
1a0u0t0o
  2u1t
     t2o0b0a
       o1b
        3b0a0h
            2h0n
```

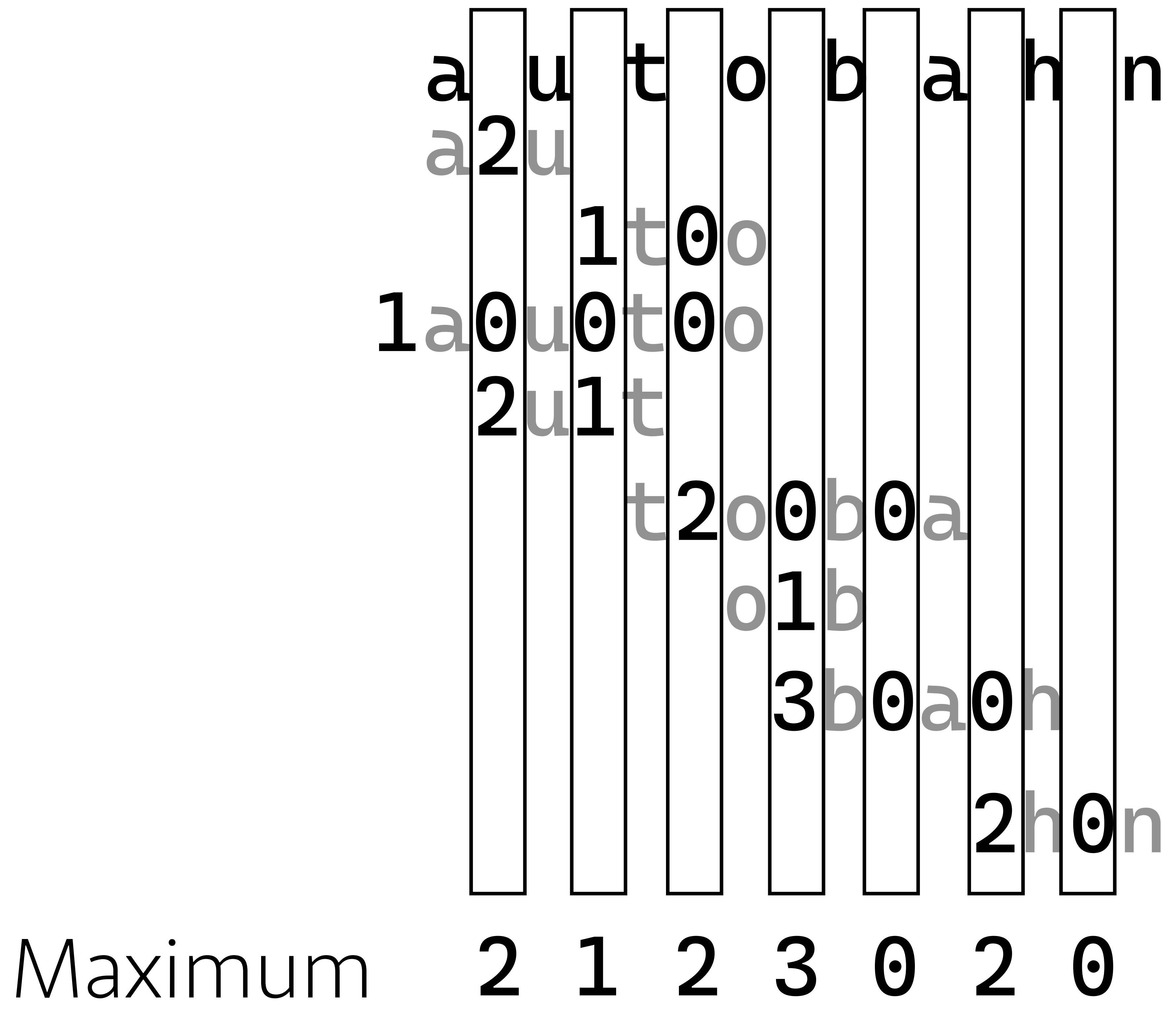
```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation

```
 a u t o b a h n
 a2u
    1t0o
1a0u0t0o
  2u1t
     t2o0b0a
       o1b
        3b0a0h
            2h0n
```

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation



```
     a u t o b a h n
     a2u
         1t0o
    1a0u0t0o
      2u1t
           t2o0b0a
             o1b
              3b0a0h
                    2h0n
```

Maximum 2 1 2 3 0 2 0

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

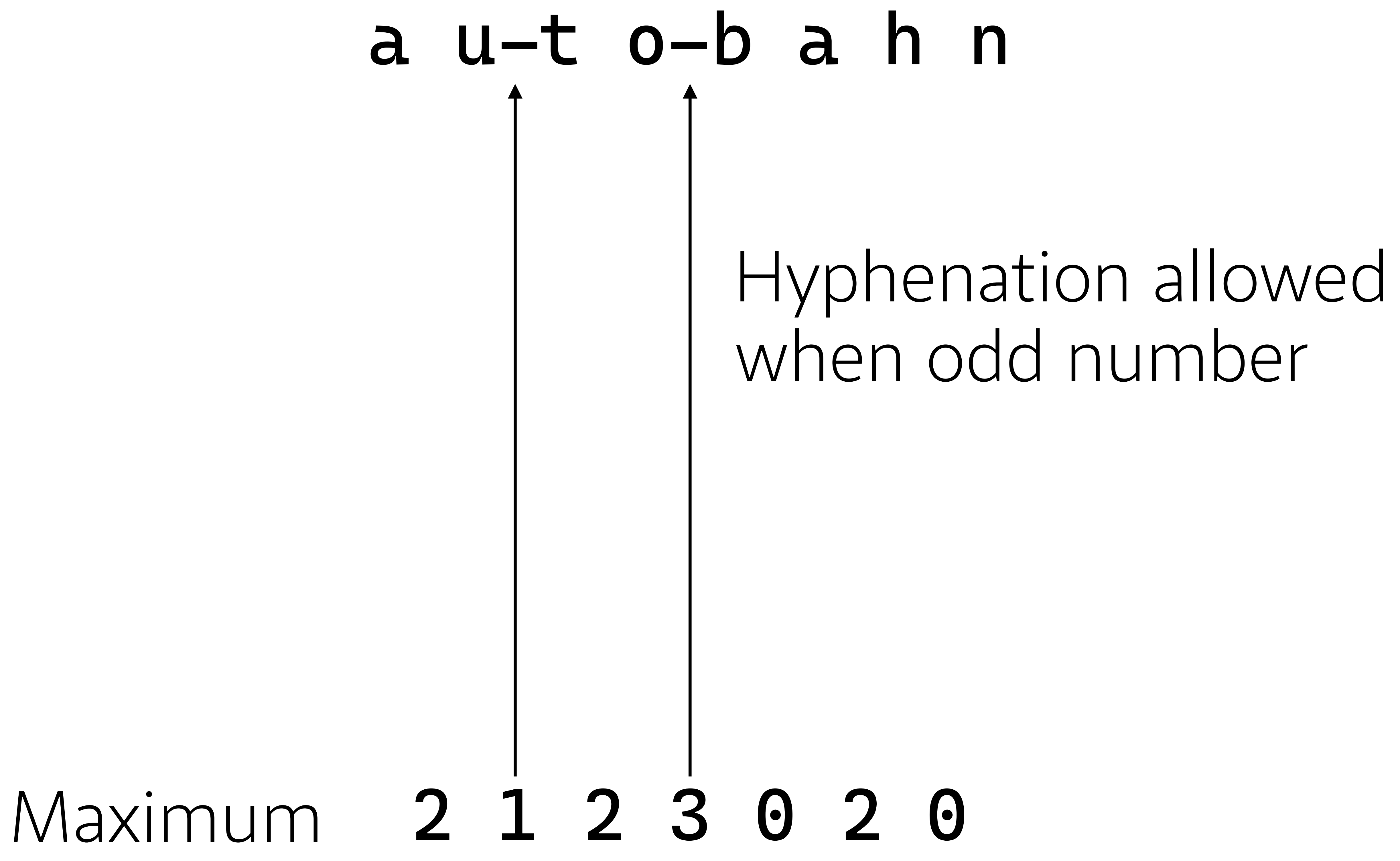
Algorithms: Hyphenation

```
a u t o b a h n
```

Maximum `2 1 2 3 0 2 0`

```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

Algorithms: Hyphenation



```
4anfors
a2u
anf5rau
2anfs
1auto
an3f2u
2u1t
4ang.
1to
1anga
t2oba
o1b
3bah
2anga.
eugin2
fli4ne
2gek.
2hn
```

boxes and glue: design goals

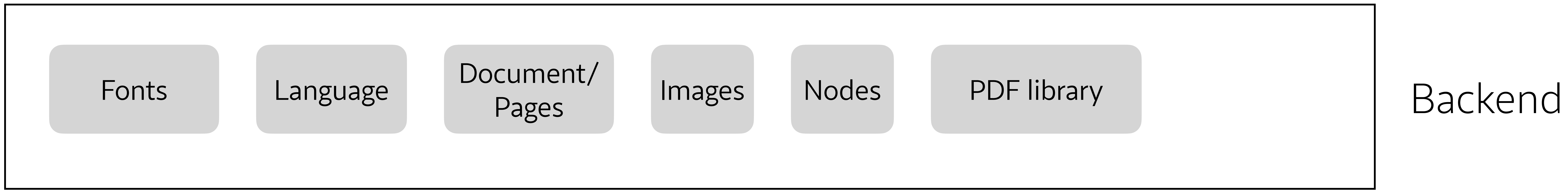
- TeX alike typography and output quality
- Performance
- TeX's data structures
- Arabic et. al. (Unicode, LTR/RTL, Bidi)
- PDF standards

boxes and glue: non design goals

- Compatibility with $\TeX$
- 8-bit fonts, tfm, dvi
- Input language (macros)

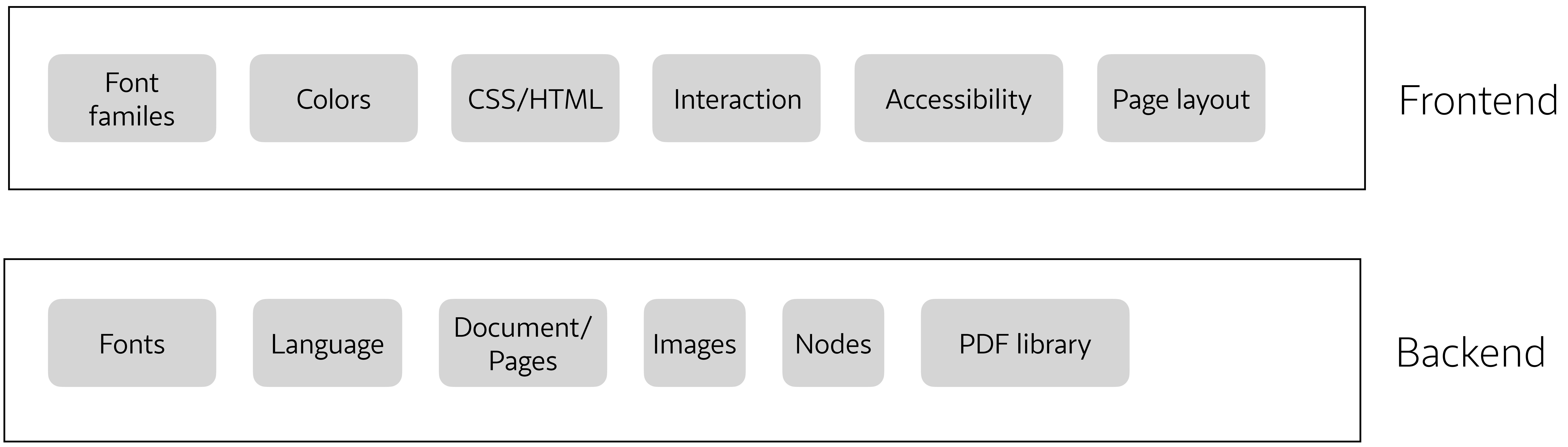
Architecture of boxes and glue

Architecture of boxes and glue

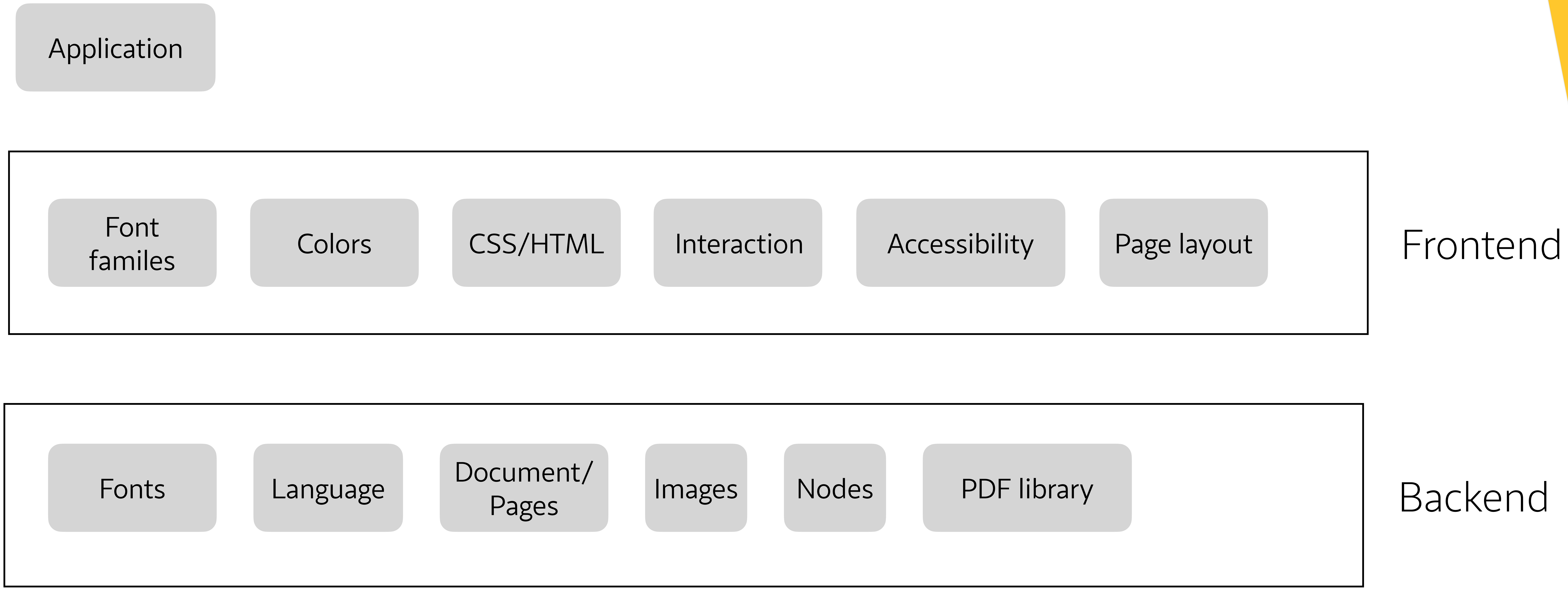


boxes & glue

Architecture of boxes and glue

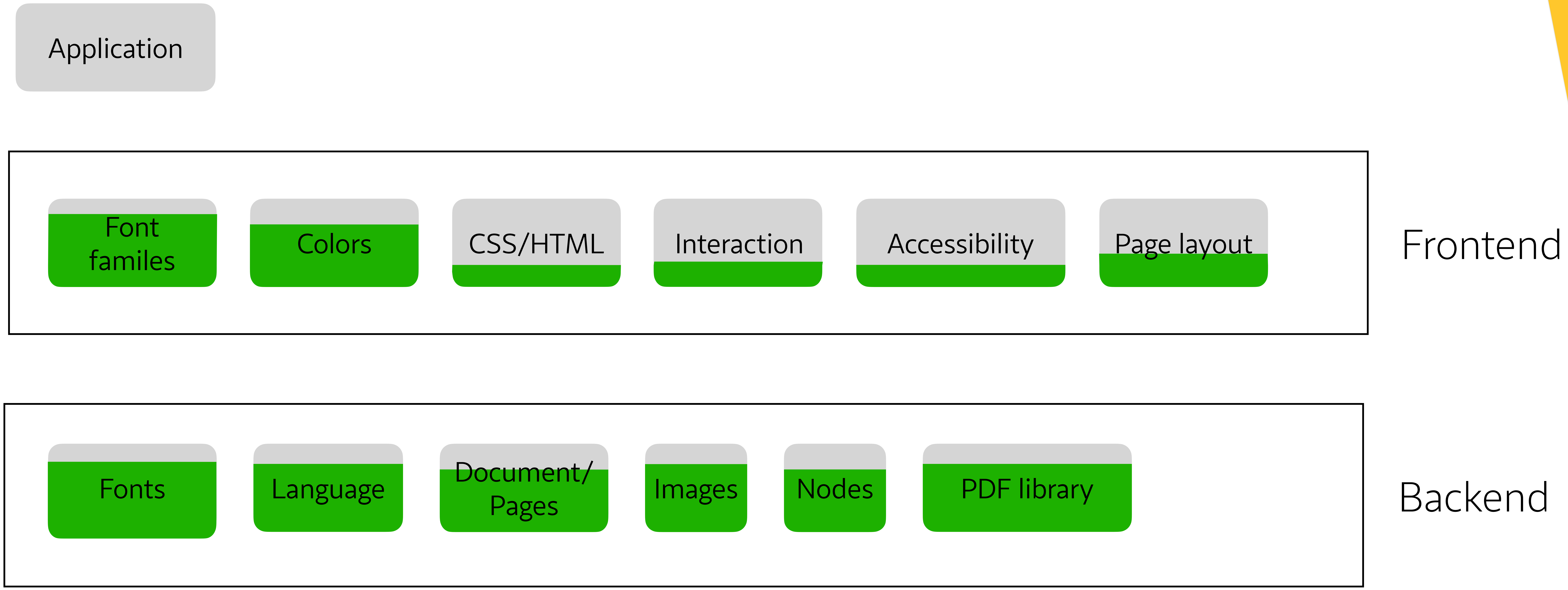


Architecture of boxes and glue



boxes & glue

Architecture of boxes and glue



Example for similarity

In olden times when wishing
still helped one, there lived a
king whose daughters were all
beautiful; and the youngest was
so beautiful that the sun itself,
which has seen so much, was
astonished whenever it shone
in her face. Close by the king's
castle lay a great dark forest,
and under an old lime-tree in the
forest was a well, and when the
day was very warm, the king's
child went out into the forest
and sat down by the side of the
cool fountain; and when she was
bored she took a golden ball, and
threw it up on high and caught
it; and this ball was her favorite
plaything.

LuaTEX

In olden times when wishing
still helped one, there lived a
king whose daughters were all
beautiful; and the youngest was
so beautiful that the sun itself,
which has seen so much, was
astonished whenever it shone
in her face. Close by the king's
castle lay a great dark forest,
and under an old lime-tree in the
forest was a well, and when the
day was very warm, the king's
child went out into the forest
and sat down by the side of the
cool fountain; and when she was
bored she took a golden ball,
and threw it up on high and
caught it; and this ball was her
favorite plaything.

boxes and glue

Example for similarity

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful; and the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain; and when she was bored she took a golden ball, and threw it up on high and caught it; and this ball was her favorite plaything.

bored she took a golden ball,
and threw it up on high and
caught it; and this ball was her
favorite plaything.

LuaTEX

boxes and glue

Todo...

- Input language (or an application)
- Output routine with headers and footers, footnotes
- Math typesetting
- Documentation
- ... and much more

…done

- Fonts, font families
- PDF-output
- TEX-algorithms (except for math)
- Image inclusion
- …

Next steps

- Experiment with the algorithms
- Optimizations for page break and paragraph break
- Parallel tasks

(My) Conclusion

TEX is dead, long live TEX

Homepage	https://boxesandglue.dev
GitHub	https://github.com/speedata/boxesandglue
Twitter	@boxesandglue