The Bourbaki constant 1

Robert M. Solovay

April 15, 2020

1 Introduction

This is a private document [for the eyes of RMS and ARDM only] which extends ARDM's computation of the length of the Bourbaki rendition of "The ineffable name of 1" to the case when the Kuratowski ordered pair is employed. My plan is to write programs in Allegro Common Lisp to compute the relevant numbers.

I program using the style of "literate programming" introduced by Knuth. However the Web and Tangle introduced by Knuth [which have been refined to CWEb and CTangle by Levy] are limited to languages closely linked to C or Pascal. So I prefer to use a more flexible literate programming language which permits fairly arbitrary target languages. Currently, I use Nuweb which is available on the TEX archives in the directory `/web/nuweb`.

One of the nice things about literate programming is that one can write programs in the natural psychological order, but arrange that the output files have the order needed for the target programming language. We will exploit this heavily in what follows.

1.1 Output files

The running of the program we write will be controlled by the variable `p`.

Its meaning is as follows. If `p=0`, we use the original Bourbaki treatment of ordered pairs [in which this is an undefined notion]. If `p=1`, then we use the Kuratowski ordered pair

There will be one output file: `compute.cl`

"compute.cl" 1 ≡
⟨source-file 2a⟩◇

In addition to the formula F, I shall introduce the nomenclature G for the formula $(\exists U)F$, the nomenclature H for the formula $(\exists u)G$ and the nomenclature J for the final term $\tau_Z G$.

So here is the structure of the source file:

⟨source-file 2a⟩ ≡

> ⟨utility functions 3b⟩
> ⟨the ordered pair 3e⟩
> ⟨the null set 7a⟩
> ⟨subset 8c⟩
> ⟨times 8d⟩
> ⟨A 11c⟩
> ⟨B 11a⟩
> ⟨C 7b⟩
> ⟨D 5c⟩
> ⟨E 4a⟩
> ⟨F 3d⟩
> ⟨G 3c⟩
> ⟨H 2c, ... ⟩
> ⟨J 2b⟩
> ◇

Macro referenced in scrap 1.

1.2 Results

1.3 The term J

We will need to have three variables to record the state of the formula H: `H_length`, `H_links`, and `|H_ZZ|`.These record respectively, the length of H, the number of links of H, and the number of ocurrences in H of the variable Z. In the future, we will regard the meaning of analogous variables to be evident from their names.

It is a stupidity of common lisp that internally all case information is lost, so that `z` and `Z` denote the *same* variable. To get around this, we use, e. g., `H_ZZ` to denote the number of occurrences of `Z` in H, and `H_z` to denote the number of occurrences of `z`.

⟨J 2b⟩ ≡

```
(setq J_length (+ 1 H_length))
(setq J_links (+ H_links H_ZZ))
```
◇

Macro referenced in scrap 2a.

1.4 The formula H

⟨H 2c⟩ ≡

```
(setq H_length (* G_length (+ 1 G_u)))
(setq H_ZZ (* G_ZZ (+ 1 G_u)))
```
◇

Macro defined by scraps 2c, 3a.
Macro referenced in scrap 2a.

⟨H 3a⟩ ≡

```
(setq H_links (quantifier_links G_u G_links))
◇
```

Macro defined by scraps 2c, 3a.
Macro referenced in scrap 2a.

1.5 The function quantifier-links

⟨utility functions 3b⟩ ≡

```
(defun quantifier_links (a b) (+ b (* a (+ a b))))

◇
```

Macro referenced in scrap 2a.

1.6 The formula G

⟨G 3c⟩ ≡

```
(setq G_length (* F_length (+ 1 F_UU)))
(setq G_ZZ (* F_ZZ  (+ 1 F_UU)))
(setq G_u (* F_u (+ 1 F_UU)))
(setq G_links (quantifier_links F_UU F_links))
◇
```

Macro referenced in scrap 2a.

1.7 The formula F

⟨F 3d⟩ ≡

```
(setq F_length (+ 16 A_length B_length C_length D_length E_length))
(setq F_ZZ (+ A_ZZ B_ZZ E_ZZ))
(setq F_u A_u)
(setq F_UU (+ A_UU B_UU C_UU D_UU E_UU))
(setq F_links (+ A_links B_links C_links D_links E_links))
◇
```

Macro referenced in scrap 2a.

2 The term for ordered pairs

I've presented these calculations in a letter to ARDM so I'll just record the results here.

⟨the ordered pair 3e⟩ ≡

```
(setq singleton_length 107)
(setq singleton_x 10)
(setq singleton_links 26)
```

```
;;; upair stands, of course, for unordered pair.

(setq upair_length 205)
(setq upair_x 14)
(setq upair_y 14)
(setq upair_links 50)

;;; The values of the corresponding statistics for opair
;;; [the ordered pair] depend on which case we are in.

(setq opair_length (if (= p 1) 4545 3))
(setq opair_x (if (= p 1) 336 1))
(setq opair_y (if (= p 1) 196 1))
(setq opair_links (if (= p 1) 1114 0))
◇
```

Macro referenced in scrap 2a.

2.1 The formula E

The formula E has the form $\forall y\, E1(y, Z, U)$

So:

⟨E 4a⟩ ≡

```
⟨E1 4b⟩
(setq E_length (+ 1 (* (+ 1 E1_y) (+ 1 E1_length))))
(setq E_ZZ (* E1_ZZ (+ 1 E1_y)))
(setq E_UU (* E1_UU (+ 1 E1_y)))
(setq E_links (quantifier_links E1_y E1_links))
◇
```

Macro referenced in scrap 2a.

Note that in the actual source file generated by this nuweb file, the definitions corresponding to `E1` will appear prior to the definitions corresponding to `E`.

$E1(y, Z, U)$ has the form $E2(y, Z) \Rightarrow E3(y, U)$. The various quantities for $E2$ are visibile by inspection.

So:

⟨E1 4b⟩ ≡

```
⟨E3 5a⟩
(setq E1_length (+ 5 E3_length))
(setq E1_links E3_links)
(setq E1_UU E3_UU)
(setq E1_ZZ 1)
(setq E1_y (+ 1 E3_y))
◇
```

Macro referenced in scrap 4a.

$E3(y, U)$ has the form $\exists x\, E4(x, y, U)$. So:

⟨E3 5a⟩ ≡

```
⟨E4 5b⟩
(setq E3_length (* E4_length (+ 1 E4_x)))
(setq E3_links  (quantifier_links E4_x E4_links))
(setq E3_y (* E4_y (+ 1 E4_x)))
(setq E3_UU (* E4_UU (+ 1 E4_x)))
◇
```

Macro referenced in scrap 4b.

The various statistics for $E4$ can be written down by inspection:

⟨E4 5b⟩ ≡

```
(setq E4_length (+ 2 opair_length))
(setq E4_links opair_links)
(setq E4_x opair_x)
(setq E4_y opair_y)
(setq E4_UU 1)
◇
```

Macro referenced in scrap 5a.

2.2 The formula D

This has the form $\forall x D1(x, U)$. So:

⟨D 5c⟩ ≡

```
⟨D1 5d⟩
(setq D_length (+ 1 (* (+ 1 D1_length) (+ 1 D1_x))))
(setq D_links  (quantifier_links D1_x D1_links))
(setq D_UU (* D1_UU (+ 1 D1_x)))
◇
```

Macro referenced in scrap 2a.

$D1(x, U)$ has the form $\forall y D2(x, y, U)$ So:

⟨D1 5d⟩ ≡

```
⟨D2 6a⟩
(setq D1_length (+ 1 (* (+ 1 D2_length) (+ 1 D2_y))))
(setq D1_links (quantifier_links D2_y D2_links))
(setq D1_x (* D2_x (+ 1 D2_y)))
(setq D1_UU (* D2_UU (+ 1 D2_y)))
◇
```

Macro referenced in scrap 5c.

$D2(x, y, U)$ has the form $\forall z D3(x, y, z, U)$ So:

⟨D2 6a⟩ ≡

```
⟨D3 6b⟩
(setq D2_length (+ 1 (* (+ 1 D3_length) (+ 1 D3_z))))
(setq D2_links (quantifier_links D3_z D3_links))
(setq D2_x (* D3_x (+ 1 D3_z)))
(setq D2_UU (* D3_UU (+ 1 D3_z)))
(setq D2_y (* D3_y (+ 1 D3_z)))
◇
```

Macro referenced in scrap 5d.

$D3(x, y, z, U)$ has the form $D4(x, y, z, U) \Rightarrow D5(y, z)$. The various statistics for $D5(y, z)$ are evident by inspection. So:

⟨D3 6b⟩ ≡

```
⟨D4 6c⟩
(setq D3_length (+ 5 D4_length))
(setq D3_links D4_links)
(setq D3_x D4_x)
(setq D3_y (+ 1 D4_y))
(setq D3_z (+ 1 D4_z))
(setq D3_UU D4_UU)
◇
```

Macro referenced in scrap 6a.

It remains to consider $D4(x, y, z, U)$. The "et" contributes 4 to the length.
So:

⟨D4 6c⟩ ≡

```
(setq D4_length (+ 4 (* 2 (+ 2 opair_length))))
(setq D4_links (* 2 opair_links))
(setq D4_x (* 2 opair_x))
(setq D4_y opair_y)
(setq D4_z opair_y)
(setq D4_UU 2)
◇
```

Macro referenced in scrap 6b.

2.3 The null set and its singleton

We agree with the calculations for $\emptyset$. The calculation of the length of $\{\emptyset\}$ is correct, but the links count must be corrected.

⟨the null set 7a⟩ ≡

```
(setq 0_length 12)
(setq 0_links 3)
(setq 1_length 217)
(setq 1_links (+ 26 (* 10 3)))
```
◇

Macro referenced in scrap 2a.

2.4 The formula C

This has the form: $\forall x C1(x, U)$. So:

⟨C 7b⟩ ≡

```
⟨C1 7c⟩
(setq C_length (+ 1 (* (+ 1 C1_length) (+ 1  C1_x))))
(setq C_links (quantifier_links C1_x C1_links))
(setq C_UU (* C1_UU (+ 1 C1_x)))
```
◇

Macro referenced in scrap 2a.

The formula $C1(x, U)$ has the form $C2(x) \Rightarrow C3(x, U)$. So:

⟨C1 7c⟩ ≡

```
⟨C2 7d⟩
⟨C3 8a⟩
(setq C1_length (+ 2 C2_length C3_length))
(setq C1_links (+ C2_links C3_links))
(setq C1_x (+ C2_x C3_x))
(setq C1_UU C3_UU)
```
◇

Macro referenced in scrap 7b.

The statistics of $C2(x)$ can be read off by inspection:

⟨C2 7d⟩ ≡

```
(setq C2_length (+ 2 1_length))
(setq C2_links 1_links)
(setq C2_x 1)
```
◇

Macro referenced in scrap 7c.

The formula $C3(x, U)$ has the form $\exists y C4(x, y, U)$. So:

⟨C3 8a⟩ ≡

```
⟨C4 8b⟩
(setq C3_length (* C4_length (+ 1 C4_y)))
(setq C3_links (quantifier_links C4_y C4_links))
(setq C3_UU (* C4_UU (+ 1 C4_y)))
(setq C3_x (* C4_x (+ 1 C4_y)))
◇
```

Macro referenced in scrap 7c.

Finally, the statistics of $C4(x, y, U)$ can be read off by inspection:

⟨C4 8b⟩ ≡

```
(setq C4_length (+ 2 opair_length))
(setq C4_links opair_links)
(setq C4_x opair_x)
(setq C4_y opair_y)
(setq C4_UU 1)
◇
```

Macro referenced in scrap 8a.

2.5 The formula $U \subset V$

Most of the calculation in the text is correct, but the count of links is off.

⟨subset 8c⟩ ≡

```
(setq subset_length 28)
(setq subset_links 4)
(setq subset_UU 3)
(setq subset_VV 3)
◇
```

Macro referenced in scrap 2a.

2.6 The term $X \times Y$

The term $X \times Y$ has the form $\tau_w T1(X, Y, w)$. So:

⟨times 8d⟩ ≡

```
⟨T1 9a⟩
(setq times_length (+ 1 T1_length))
(setq times_links (+ T1_w T1_links))
(setq times_XX T1_XX)
(setq times_YY T1_YY)
◇
```

Macro referenced in scrap 2a.

The formula $T1(X, Y, w)$ has the form $\forall z T2(X, Y, w, z)$. so:

⟨T1 9a⟩ ≡

```
⟨T2 9b⟩
(setq T1_length (+ 1 (* (+ 1 T2_length) (+ 1 T2_z))))
(setq T1_links (quantifier_links T2_z T2_links))
(setq T1_XX (* T2_XX (+ 1 T2_z)))
(setq T1_YY (* T2_YY (+ 1 T2_z)))
(setq T1_w (* T2_w (+ 1 T2_z)))
◇
```

Macro referenced in scrap 8d.

The formula $T2(X, Y, w, z)$ has the form $T3(w, z) \Leftrightarrow T4(X, Y, z)$. So:

⟨T2 9b⟩ ≡

```
⟨T3 9c⟩
⟨T4 9d⟩
(setq T2_length (+ 8 (* 2 (+ T3_length T4_length))))
(setq T2_links (* 2 (+ T3_links T4_links)))
(setq T2_XX (* 2 T4_XX))
(setq T2_YY (* 2 T4_YY))
(setq T2_w (* 2  T3_w))
(setq T2_z (* 2 (+ T3_z T4_z)))
◇
```

Macro referenced in scrap 9a.

The various statistics for $T3(w, z)$ can be written down by inspection:

⟨T3 9c⟩ ≡

```
(setq T3_length 3)
(setq T3_links 0)
(setq T3_w 1)
(setq T3_z 1)
◇
```

Macro referenced in scrap 9b.

The formula $T4(X, Y, z)$ has the form $\exists x T5(X, Y, z, x)$. So:

⟨T4 9d⟩ ≡

```
⟨T5 10a⟩
(setq T4_length (* T5_length (+ 1 T5_x)))
(setq T4_links (quantifier_links T5_x T5_links))
(setq T4_XX (* T5_XX (+ 1 T5_x)))
(setq T4_YY (* T5_YY (+ 1 T5_x)))
(setq T4_z (* T5_z (+ 1 T5_x)))
◇
```

Macro referenced in scrap 9b.

The formula $T5(X, Y, z, x)$ has the form $\exists y T6(X, Y, z, x, y)$. So:

⟨T5 10a⟩ ≡

```
⟨T6 10b⟩
(setq T5_length (* T6_length (+ 1 T6_y)))
(setq T5_links (quantifier_links T6_y T6_links))
(setq T5_XX (* T6_XX (+ 1 T6_y)))
(setq T5_YY (* T6_YY (+ 1 T6_y)))
(setq T5_z (* T6_z (+ 1 T6_y)))
(setq T5_x (* T6_x (+ 1 T6_y)))
◇
```

Macro referenced in scrap 9d.

The formula $T6$ has the form $T7(x, y, z)\&T8(x, X)\&T9(y, Y)$. The formulas $T8$ and $T9$ are simple enought to handle by inspection. So:

⟨T6 10b⟩ ≡

```
⟨T7 10c⟩
(setq T6_length (+ 8 3 3 T7_length))
(setq T6_links T7_links)
(setq T6_XX 1)
(setq T6_YY 1)
(setq T6_z 1)
(setq T6_x (+ 1 T7_x))
(setq T6_y (+ 1 T7_y))
◇
```

Macro referenced in scrap 10a.

Finally, the statistics for $T7$ are evident.

⟨T7 10c⟩ ≡

```
(setq T7_length (+ 2 opair_length))
(setq T7_links opair_links)
(setq T7_x opair_x)
(setq T7_y opair_y)
◇
```

Macro referenced in scrap 10b.

2.7 The formula B

The formula $B(U, Z)$ has the form $U \subset B2(Z)$ So:

⟨B 11a⟩ ≡

```
⟨B2 11b⟩
(setq B_length (+ subset_length (- subset_UU) (- subset_VV)
                  (* subset_UU 1) (* subset_VV B2_length)))
(setq B_links (+ subset_links (* subset_VV B2_links)))
(setq B_UU (* subset_UU 1))
(setq B_ZZ (* subset_VV B2_ZZ))
◇
```

Macro referenced in scrap 2a.

Finally, we handle $B2(Z)$:

⟨B2 11b⟩ ≡

```
(setq B2_length (+ times_length (- times_XX) (- times_YY)
        (* times_XX 1_length)
        (* times_YY 1)))
(setq B2_links (+ times_links (* times_XX 1_links)))
(setq B2_ZZ (* times_YY 1))
◇
```

Macro referenced in scrap 11a.

2.8 The formula A

This has the form $u = A1(U, Z)$. So:

⟨A 11c⟩ ≡

```
⟨A1 12a⟩
(setq A_length (+ 2 A1_length))
(setq A_links A1_links)
(setq A_u 1)
(setq A_UU A1_UU)
(setq A_ZZ A1_ZZ)
◇
```

Macro referenced in scrap 2a.

The term $A1(U, Z)$ has the form $(A2(U), Z)$. So:

⟨A1 12a⟩ ≡

```
⟨A2 12b⟩
(setq A1_length (+ opair_length (- opair_x) (- opair_y)
                (* opair_x A2_length)
                (* opair_y 1)))
(setq A1_links (+ opair_links
                (* opair_x A2_links)))
(setq A1_UU (* opair_x A2_UU))
(setq A1_ZZ opair_y)
◇
```

Macro referenced in scrap 11c.

The term $A2(U)$ is $(U, \{\emptyset\})$. So:

⟨A2 12b⟩ ≡

```
(setq A2_length (+ opair_length (- opair_x) (- opair_y)
                (* opair_x 1)
                (* opair_y 1_length)))
(setq A2_links (+ opair_links (* opair_y 1_links)))
(setq A2_UU opair_x)

◇
```

Macro referenced in scrap 12a.