There Can Be No Turing-Test--Passing Memorizing Machines

Calculating the critical Turing Test length

Supplement to "There can be no Turing-Test–passing memorizing machines"

Stuart M. Shieber

February 8, 2014

This Mathematica document is intended as a supplement to the paper "There can be no Turing-Test–passing memorizing machines" to appear in *Philosopher's Imprint*. It provides derivations of all of the primary technical results presented in that paper, as well as the development of the graphs. It is not intended a a standalone document; it should be used in conjunction with the associated paper.

Functions of interest

The maximum time of a Turing Test that a memorizing machine can support depends on the radius *r* of its storage, as well as the spatial density of information α and the temporal density β, as well as the number of half-rounds of communication that must be allowed for.

In[612]:=
```
t[r_, α_, β_, k_] := Log2[(α / bit) r^3] bit / (k β)
```

Basic units and conversions

Spatial density of information assuming 1 bit per Planck volume, given in bits per cubic light second. This is a gross overestimate of information density.

In[613]:=
```
α := 4 π / 3 (1 bit / plancklength^3 × (1.616 × 10^35 plancklength / m)^3 × (2.998 × 10^8 m / lightsecond)^3)
```

(Planck length value taken from [Mohr:2012:CRV, page 1587]. Meters to light-seconds is the SI definition of the meter ("the length of the path travelled by light in vacuum during a time interval of 1/299 792 458 of a second". [Taylor:2008:ISU, page 18])

In[614]:=
```
N[α]
```

Out[614]=
$$\frac{4.7633 \times 10^{131}\ \text{bit}}{\text{lightsecond}^3}$$

A slightly more reasonable estimate of spatial information density, assuming a terabit per cubic centimeter.

In[615]:= $\alpha 1 := \frac{4\pi}{3}\left(\frac{10^{12}\,\text{bit}}{(.01\,\text{m})^3}\times\left(\frac{3\times 10^8\,\text{m}}{\text{lightsecond}}\right)^3\right)$

In[616]:= `N[α1]`

Out[616]= $\frac{1.13097\times 10^{44}\,\text{bit}}{\text{lightsecond}^3}$

The temporal density of spoken information, based on entropy estimates of written English and standard speaking rates.

In[617]:= $\beta := \frac{5\,\text{bit}}{\text{word}}\times\frac{200\,\text{word}}{\text{minute}}\times\frac{\text{minute}}{60\,\text{second}}$

(Entropy of English from [Brown:1992:EUB] is 1.75 bits per character. This is an upper bound, so use a conservative estimate of 1 bit per character, 5 characters per word.)

In[618]:= `N[β]`

Out[618]= $\frac{16.6667\,\text{bit}}{\text{second}}$

Trivial conversion of time and space units.

In[619]:= $\gamma := \frac{1\,\text{second}}{\text{lightsecond}}$

An estimate of the size of the universe (its radius) in light seconds, based on its age.

In[620]:= $\text{runiv} := \frac{13.798\times 10^9}{2}\,\text{lightyear}\times\frac{31\,557\,600\,\text{lightsecond}}{\text{lightyear}}$

(Age of the universe from [2013arXiv1303.5062P], page 36. Astronomy uses the Julian year, defined as 365.25 days of 86,400 SI seconds. [http://www.iau.org/science/publications/proceedings_rules/units/])

In[621]:= `N[runiv]`

Out[621]= $2.17716\times 10^{17}\,\text{lightsecond}$

Replicating and extending the “communication-free” estimates

Replicating the “less than a minute” estimate of Shieber (2007)

In[622]:= `N[t[runiv, α, β, 1]]`

Out[622]= 36.6126 second

If the universe were ten times older, the critical length doesn’t increase much.

In[623]:= `N[t[10 runiv, α, β, 1]]`

Out[623]= 37.2106 second

Reducing this estimate based on a single round of communication. The process could be repeated to

get successively shorter estimates but convergence is extremely rapid. We get convergence to four significant digits in just three rounds.

```
In[624]:= N[t[36.509834144709295 lightsecond, α, β, 1]]
Out[624]= 27.1797 second

In[625]:= N[t[% / γ, α, β, 1]]
Out[625]= 27.1031 second

In[626]:= N[t[% / γ, α, β, 1]]
Out[626]= 27.1023 second
```

Solving for the fixed point

The function `f` from Appendix B determines if the constraint of Equation (1) is solvable.

```
In[627]:= f[r_, β_, k_] := (r^3 2^(-k β r γ / bit) bit)

In[628]:= f[r lightsecond, β, 2] / (bit lightsecond^3)
Out[628]= 2^(-100 r/3) r^3

In[629]:= α^-1 (bit / lightsecond^3)
Out[629]= 2.09939 × 10^-132

In[630]:= Off[NSolve::ifun]

In[631]:= solns = NSolve[f[r, β, 2] / (bit lightsecond^3) == α^-1 bit / lightsecond^3, r]
Out[631]= {{r → (-6.40227 × 10^-45 - 1.10891 × 10^-44 i) lightsecond},
  {r → (-6.40227 × 10^-45 + 1.10891 × 10^-44 i) lightsecond},
  {r → 1.28045 × 10^-44 lightsecond}, {r → 13.4603 lightsecond}}
```

We ignore the two imaginary solutions. The other two solutions provide the lower and upper bounds on the range for which communication is feasible.

```
In[632]:= lower := r /. solns[[3]]

In[633]:= upper := r /. solns[[4]]

In[634]:= lower
Out[634]= 1.28045 × 10^-44 lightsecond

In[635]:= upper
Out[635]= 13.4603 lightsecond
```

Verifying the solution

In[636]:= `N[t[upper, α, β, 2]]`

Out[636]= 13.4603 second

A machine of this radius can support a Turing Test of the following length:

In[637]:= `N[t[upper, α, β, 1]]`

Out[637]= 26.9206 second

What about requiring two roundtrips?

In[638]:= `NSolve[f[r, β, 4] / (bit lightsecond^3) == α^-1 bit / lightsecond^3, r]`

Out[638]= $\{\{r \to (-6.40227 \times 10^{-45} - 1.10891 \times 10^{-44}\, i)\ \text{lightsecond}\},$
$\{r \to (-6.40227 \times 10^{-45} + 1.10891 \times 10^{-44}\, i)\ \text{lightsecond}\},$
$\{r \to 1.28045 \times 10^{-44}\ \text{lightsecond}\}, \{r \to 6.68471\ \text{lightsecond}\}\}$

In[639]:= `N[t[r /. %[[4]], α, β, 1]]`

Out[639]= 26.7388 second

One hundred roundtrips?

In[640]:= `NSolve[f[r, β, 200] / (bit lightsecond^3) == α^-1 bit / lightsecond^3, r]`

Out[640]= $\{\{r \to (-6.40227 \times 10^{-45} - 1.10891 \times 10^{-44}\, i)\ \text{lightsecond}\},$
$\{r \to (-6.40227 \times 10^{-45} + 1.10891 \times 10^{-44}\, i)\ \text{lightsecond}\},$
$\{r \to 1.28045 \times 10^{-44}\ \text{lightsecond}\}, \{r \to 0.128564\ \text{lightsecond}\}\}$

In[641]:= `N[t[r /. %[[4]], α, β, 1]]`

Out[641]= 25.7128 second

Half a roundtrip? (This verifies the iteratively determined fixed point above.)

In[642]:= `NSolve[f[r, β, 1] / (bit lightsecond^3) == α^-1 bit / lightsecond^3, r]`

Out[642]= $\{\{r \to (-6.40227 \times 10^{-45} - 1.10891 \times 10^{-44}\, i)\ \text{lightsecond}\},$
$\{r \to (-6.40227 \times 10^{-45} + 1.10891 \times 10^{-44}\, i)\ \text{lightsecond}\},$
$\{r \to 1.28045 \times 10^{-44}\ \text{lightsecond}\}, \{r \to 27.1023\ \text{lightsecond}\}\}$

Finding the maximum of f

Here we replicate the derivation in Appendix B of the maximum value of f to verify that the maximum is above the needed threshold.

In[643]:= `dimfree[x_] := x /. {lightsecond → 1, second → 1, bit → 1}`

In[644]:= `f[r_] := dimfree[f[r, β1, k1]]`

In[645]:= `f'[r]`

Out[645]= $3 \times 2^{-k1\, r\, \beta 1}\ r^2 - 2^{-k1\, r\, \beta 1}\ k1\ r^3\ \beta 1\ \text{Log}[2]$

In[646]:= `Solve[f'[r] == 0, r]`

Out[646]= $\left\{\{r \to 0\}, \{r \to 0\}, \left\{r \to \frac{3}{k1\, \beta 1\, \text{Log}[2]}\right\}\right\}$

In[647]:= `f[ r /. %[[3]]]`

Out[647]= $\frac{27 \times 2^{-\frac{3}{\text{Log}[2]}}}{k1^3\, \beta 1^3\, \text{Log}[2]^3}$

In[648]:= `rmax := 3/(2 β Log[2])`

In[649]:= `fmax := f[rmax, β, 2]`

In[650]:= `ScientificForm[N[dimfree[fmax]]]`

Out[650]//ScientificForm= 1.08985×10^{-4}

In[651]:= `ScientificForm[N[dimfree[α^-1]]]`

Out[651]//ScientificForm= 2.09939×10^{-132}

Alternative scenarios

What if we require that each round of communication happen in near real time, say, within a second?

In[652]:= `N[t[1 lightsecond, α, β, 1]]`

Out[652]= 26.2455 second

Suppose instead we imagine the device being the size of the earth (.02 light seconds in radius)

In[653]:= `N[t[.02 lightsecond, α, β, 1]]`

Out[653]= 25.2296 second

What about 1 mile in radius?

In[654]:= `N[t[5.36819375 × 10^-6 lightsecond, α, β, 1]]`

Out[654]= 23.0942 second

1 mile with reasonable storage densities?

In[655]:= `N[t[5.36819375 × 10^-6 lightsecond, α1, β, 1]]`

Out[655]= 5.62926 second

Plots

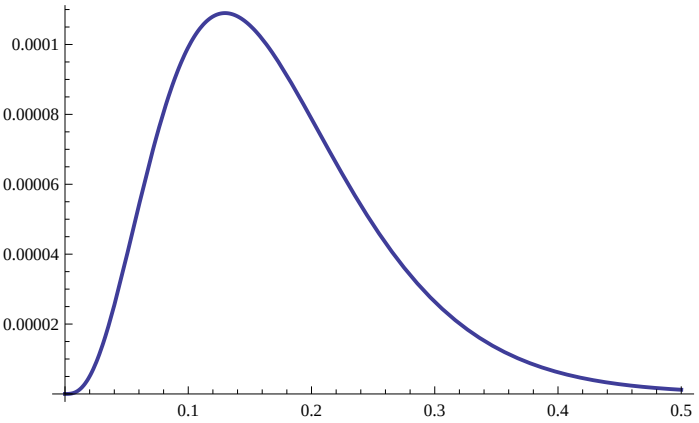
In[656]:=
```
f[r_] := dimfree[f[r, β, 2]]
```

In[657]:=
```
f[r]
```

Out[657]=
$$2^{-100\,r/3}\,r^3$$

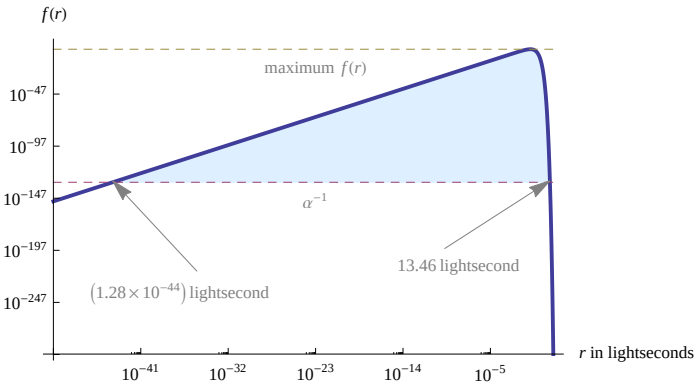
In[658]:=
```
Plot[{f[r]}, {r, 0, .5}, PlotStyle → {Thick}]
```

Out[658]=



In[659]:=
```
LogLogPlot[{f[r], 1/α (bit/lightsecond^3), N[dimfree[fmax]]}, {r, 10^-50, 30},
 PlotStyle → {Thick, Dashed, Dashed},
 AxesLabel → {"r in lightseconds", "f(r)"},
 Filling → {2 → {{1}, {LightBlue, White}}},
 Epilog → {
   {Gray, Text["maximum f(r)", {Log[10^-23], dimfree[fmax]}, {0, 2}],
    Text["α^-1", {Log[10^-23], Log[dimfree[1/α]]}, {0, 1.75}],
    Arrow[{{Log[dimfree[lower]] + 20, -500}, {Log[dimfree[lower]], -300}}],
    Text[NumberForm[lower, 4], {Log[dimfree[lower]], -550}, {-.75, 0}],
    Arrow[{{Log[dimfree[upper]] - 25, -450}, {Log[dimfree[upper]], -300}}],
    Text[NumberForm[upper, 4], {Log[dimfree[upper]], -450}, {1.5, 1.5}]}}]
```

Out[659]=

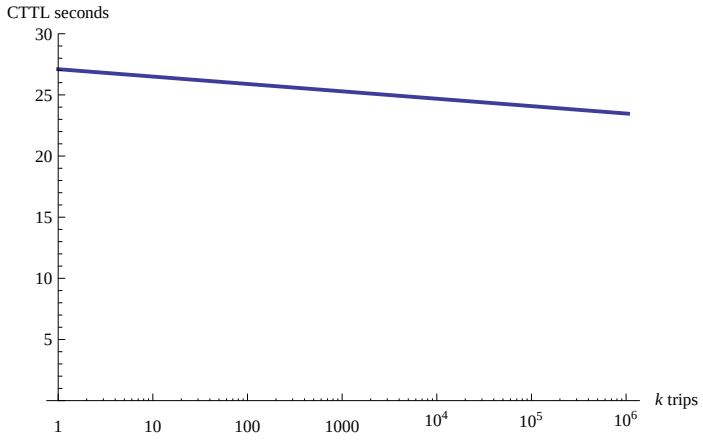


We generate a table of the critical Turing Test length for exponentially increasing values of *k*. The values plot as a straight line on a log-linear scale, showing the extremely slow reduction in CTTL as *k* increases.

In[660]:=
```
ktab := Table[{2^k, dimfree[
    t[r /. NSolve[f[r, β, 2^k] / (bit lightsecond^3) == α^-1 bit / lightsecond^3, r][[4]],
     α, β, 1]]}, {k, 0, 20}]
```

In[661]:=
```
ListLogLinearPlot[ktab,
 Joined → True,
 PlotStyle → {Thick},
 PlotRange → {0, 30},
 AxesLabel → {"k trips", "CTTL seconds"}]
```

Out[661]=



Information density based on surface area

As noted in Footnote 13, the estimates above, based on 1 bit per Planck volume, are extremely conservative. They ignore work in quantum gravity that places limits on the information capacity of a region based on its surface area rather than its volume. (See the footnote for pertinent references.) We can recalculate based on this better estimate of storage capacity limits. The result, as expected, is lower by a factor of 2/3, since volume grows as the cube of the radius, but surface area only as the square.

In[662]:=
$$\alpha 2 := 4\pi \left(\frac{1\ \text{bit}}{\text{plancklength}^2} \times \left(\frac{1.616 \times 10^{35}\ \text{plancklength}}{\text{m}}\right)^2 \times \left(\frac{2.998 \times 10^8\ \text{m}}{\text{lightsecond}}\right)^2\right)$$

In[663]:=
$$\text{t2}[r\_,\ \alpha\_,\ \beta\_,\ k\_] := \frac{\text{Log2}\left[(\alpha\ /\ \text{bit})\ r^2\right]\ \text{bit}}{k\ \beta}$$

In[664]:=
```
N[t2[runiv, α2, β, 1]]
```

Out[664]=
```
24.5448 second
```

In[665]:=
```
N[t2[% lightsecond / second, α2, β, 1]]
```

Out[665]=
```
18.1875 second
```

In[666]:=
```
N[t2[% lightsecond / second, α2, β, 1]]
```

Out[666]=
```
18.1356 second
```

In[667]:=
$$\text{f2}[r\_,\ \beta\_,\ k\_] := \left(r^2\ 2^{-k\,\beta\,r\,\gamma\ /\ \text{bit}}\ \text{bit}\right)$$

In[668]:=
```
f2[r lightsecond, β, 2]
```

Out[668]=
$$2^{-100\,r/3}\ \text{bit}\ \text{lightsecond}^2\ r^2$$

In[669]:=
```
α^-1
```

Out[669]=
$$\frac{2.09939\times 10^{-132}\ \text{lightsecond}^3}{\text{bit}}$$

In[670]:=
```
NSolve[f2[r , β, 2] / (bit lightsecond^2) == α2^-1 bit / lightsecond^2, r]
```

Out[670]=
$$\{\{r\to -5.82267\times 10^{-45}\ \text{lightsecond}\},\ \{r\to 5.82267\times 10^{-45}\ \text{lightsecond}\},\ \{r\to 9.00697\ \text{lightsecond}\}\}$$

In[671]:=
```
N[t2[r /. %[[3]], α2, β, 1]]
```

Out[671]=
$$18.0139\ \text{second}$$