Sizes of Up-to-$n$ Halting Testers

Antti Valmari
Tampere University of Technology
FINLAND

1 Motivation
2 Up-to-$n$ (3-way) Deciders
3 Self-Delimiting Representations
4 Upper Bound
5 Earlier Upper Bound Results
6 Lower Bounds
7 Earlier Lower Bound Results
1 Motivation

• Standard proof of **undecidability of halting testing** is
  – hard for my students: looks like a magician’s trick to them
  – common target of objections in net and other discussions
    (well, nothing changes the minds of some people . . . )

⇒ I have wanted a proof that would raise less objections, would “feel better”
⇒ Got interested in the “Busy Beaver” proof (discussed later)

• Triggered by yet another crazy net discussion, this June I tried to write the Busy Beaver proof in a very clear and convincing form

• Accidentally got an easy but interesting result that I had not known before

• No one else seemed to have seen it either

⇒ So here it comes
  – here formulated using Turing machines
  – programming language version (not here) intended for software people
  – the theorems of the two versions have surprising differences!
2 Up-to-\(n\) (3-way) Deciders 1/2

- **Decision problem** \(\varphi := \{0, 1\}^* \rightarrow \{\text{“yes”}, \text{“no”}\}\)

- **Decider for** \(\varphi := \) universal Turing machine program that computes \(\varphi\)

- **Up-to-\(n\) \((n \in \mathbb{N})\) decider for** \(\varphi := \) universal Turing machine program that
  - if \(|\text{input}| \leq n\), then replies \(\varphi(\text{input})\)
  - if \(|\text{input}| > n\), then may do anything: reply wrong, fail to terminate, . . .

- **Up-to-\(n\) 3-way decider for** \(\varphi := \) universal Turing machine program that
  - if \(|\text{input}| \leq n\), then replies \(\varphi(\text{input})\)
  - if \(|\text{input}| > n\), then replies “too big”

- We study families \((D_n)_{n \in \mathbb{N}}\) of up-to-\(n\) (3-way) deciders

- For every \(\varphi\), such a family exists
  - look up the answer in a pre-determined array of \(2^0 + 2^1 + \ldots + 2^n\) bits
  \(\Rightarrow\) its size is \(2^{n+1} + O(n)\) (Section 3 explains \(O(n)\))

- \(\varphi\) is decidable if and only if it has an up-to-\(n\) decider of size \(O(1)\)
2 Up-to-\(n\) (3-way) Deciders 2/2

Grey part not in the paper

- Let \(\mu_n\) be the above-mentioned bit string of answers for \(|\text{input}| \leq n\)
  - \(|\mu_n| = 2^{n+1} - 1\)

- \(|\text{a smallest up-to-}\(n\) 3-way decider} \mid \cong \) the s.d.-Kolmogorov complexity of \(\mu_n\)
  - s.d. := self-delimiting (Section 3 discusses)

\[\geq: \text{run } D_n \text{ for all strings } \varepsilon, 0, 1, 00, \ldots \text{ until it starts saying “too big”}\]

\[\leq: \text{construct } \mu_n, \text{ check “too big” against } |\mu_n|, \text{ pick “yes”/“no” from } \mu_n\]

- Let
  - \(e-i := \) empty input = \(0^\omega\)
  - \(\lg := \) base-2 logarithm

- previous slide: \(O(1) \leq |D_n| \leq 2^{n+1} + O(n)\)

- Our main result: if \(\varphi\) is e-i halting testing, then
  - the size of the smallest up-to-\(n\) decider is between \(n - \lg n - 2 \lg \lg n - O(1)\) and \(n + O(1)\)
  - the size of the smallest up-to-\(n\) 3-way decider is between \(n \pm O(1)\)

\(\Rightarrow\) for \(\mu_n = \) e-i halting testing answers, s.d.-Kolmogorov\((\mu_n) = \lg |\mu_n| \pm O(1)\)
3 Self-Delimiting Representations 1/2

- The program must know where it ends and the input begins
  ⇒ it is assumed that programs are self-delimiting
  - no proper prefix of a program is a program

- We will need self-delimiting representations for arbitrary $\beta \in \{0, 1\}^*$

- $\ell(\beta) :=$ the size of the representation of $\beta$

- **Theorem** No self-delimiting representation system for all finite bit strings satisfies $\ell(\beta) = |\beta| + \lg |\beta| + O(1)$.
  - if such a system exists, then there is a $c$ such that when $|\beta| \geq 1$, then $\ell(\beta) \leq |\beta| + \lg |\beta| + c$
  - by Kraft’s inequality, for any $m \in \mathbb{N}$,

\[
1 \geq \sum_{1 \leq |\beta| \leq m} 2^{-\ell(\beta)} = \sum_{n=1}^{m} \sum_{|\beta|=n} 2^{-\ell(\beta)} \geq \sum_{n=1}^{m} \sum_{|\beta|=n} 2^{-(n+\lg n+c)}
\]

\[
= \sum_{n=1}^{m} 2^n 2^{-(n+\lg n+c)} = 2^{-c} \sum_{n=1}^{m} \frac{1}{n}
\]
3 Self-Delimiting Representations 2/2

- **Theorem** No self-delimiting representation system for all non-negative integers satisfies \( \ell(n) = \lg n + \lg \lg n + O(1) \).
  
  - when \( \beta \in \{0, 1\}^* \) and \( n > 0 \), let \( 1\beta \leftrightarrow n \), then previous theorem

- Actually, for any \( k \in \mathbb{N} \), no representation system satisfies \( \ell(n) = \lg n + \lg \lg n + \ldots + (\lg)^k n + O(1) \)

\[
\sum_{i=2^{n-1}+1}^{2^n} \frac{1}{i \lg i \cdots (\lg)^{k+1} i} \geq \frac{2^{n-1}}{2^n \lg 2^n \cdots (\lg)^{k+1} 2^n} = \frac{1}{2^n \lg n \cdots (\lg)^{k+1} n}
\]

- So a non-negative integer \( n \) needs more than that many bits!

- A self-delimiting representation system with \( \ell(\beta) = |\beta| + 2[\lg(|\beta| + 2)] \)

\[
|\beta| = 1i_n \cdots i_2i_1i_0 - 2 \quad \beta
\]

\[
\begin{array}{cccccccc}
i_n & 0 & \cdots & i_2 & 0 & i_1 & 0 & i_0 & 1 & \cdots
\end{array}
\]

- Practical programming languages have \( \ell(\beta) = c|\beta| + O(1) \) for some \( c > 1 \)
4 Upper Bound

- If $n$ is so small that no $Q$ with $|Q| \leq n$ e-i halts, then $H_n$ is trivial to design
  - if $|Q| \leq n$ then reply “no” else reply “too big”

⇒ From now on we assume that some $Q$ with $|Q| \leq n$ e-i halts

- Let $P_n$ be a slowest e-i halting program of size at most $n$ bits
  - exists, because there are $< 2^{n+1}$ programs of size at most $n$ bits

- $H_n$ and its input $Q$ use the tape like this

  \[
  \begin{array}{cccc}
  \text{main prog} & P_n & 0^{n-|P_n|} & Q \\
  0 \cdots & 0 & 1 & 0 \cdots \\
  \end{array}
  \]

  - $P_n$ is self-delimiting, because it is a program
  ⇒ main program gets $n$ by finding the end of $P_n$ and then a 1
  ⇒ can check if $Q$ is too big
  - then main program e-i executes $Q$ and $P_n$ one step at a time
  - $Q$ terminates first: “yes”, $P_n$ terminates first: “no”
  - main program does not depend on $n$

⇒ An $n + O(1)$ family of up-to-$n$ 3-way e-i halting testers exists
5 Earlier Upper Bound Results

- Knowing $n$ first bits of G. Chaitin’s $\Omega$ facilitates up-to-$n$ e-i halting testing
  - $\Omega := \sum_{P \text{ e-i halts}} 2^{-|P|}$
  - also his programs are self-delimiting, so $0 < \Omega < 1$
  - simulate all programs (also those $> n$ bits) maintaining
    a lower approximation of $\Omega$ until it matches $\Omega_{1:n}$

- $\Omega_{1:n}$ is not self-delimiting
  $\Rightarrow$ only the bound $n + \lg n + 2 \lg \lg n + O(1)$ obtained

- It is widely known that knowing (the running time of) a slowest
  e-i halting program of size $\leq n$ facilitates up-to-$n$ e-i halting testing
  - from this, proof of $n + O(1)$ for up-to-$n$ e-i halting testers is immediate
  - I do not know if anyone has carefully (self-delimiting!) formulated it

- Extending the proof to 3-way testers seems new
  - 3-way is important for the lower bounds, this observation seems new
  - 3-way result yields self-delimiting Kolmogorov complexity result
6 Lower Bounds

- Let $H_n$ be any family of up-to-$n$ 3-way e-i halting testers

  ```
  print 1
  for $\beta := \varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots$ do
    if $\beta$ is a program then
      $r := H_n(\beta)$
      if $r = \text{"too big"}$ then halt
      if $r = \text{"yes"}$ then e-i simulate $\beta$ and print its result
  ```

- The above program
  - is of size $O(1) + |H_n|$  
  - e-i halts
  - catenates 1 and the outputs of all e-i halting programs $\leq n$ bits

  $\Rightarrow$ must be of size $> n$

  $\Rightarrow |H_n| > n - O(1)$

- If $H_n$ is not 3-way, the program is modified to test $|\beta| > n$ and halt then  
  - the program needs a representation of $n$

  $\Rightarrow \lg n + 2\lg \lg n + O(1)$ additional bits

  $\Rightarrow |H_n| > n - \lg n - 2\lg \lg n - O(1)$
7 Earlier Lower Bound Results

- For non-3-way testers, the proof is a variant of the Busy Beaver proof of non-existence of halting testers
  - Busy Beaver := \( n \)-state TM that prints as many 1’s as possible and halts
  - an \( O(\log n) \) bit program prints something that a \( < n \) bit program cannot

- G. Chaitin proved in a similar way that to produce \( \Omega_{1:n} \), a program of size \( n - O(1) \) is needed
  - however, \( \Omega_{1:n} \) is affected by e-i halting programs \( > n \) bits
  \( \Rightarrow \) it is not obvious how \( H_n \) would yield \( \Omega_{1:n} \)
  \( \Rightarrow \) no obvious lower bound for \( |H_n| \)

\( \Rightarrow \) A gap in my (and others’) knowledge on the self-delimiting Kolmogorov complexity of \( \Omega_{1:n} \)
  - between \( n - O(1) \) and \( n + \lg n + 2\lg\lg n + O(1) \)

- Although the lower bound results in this talk are simple, it seems that they have not been formulated precisely before