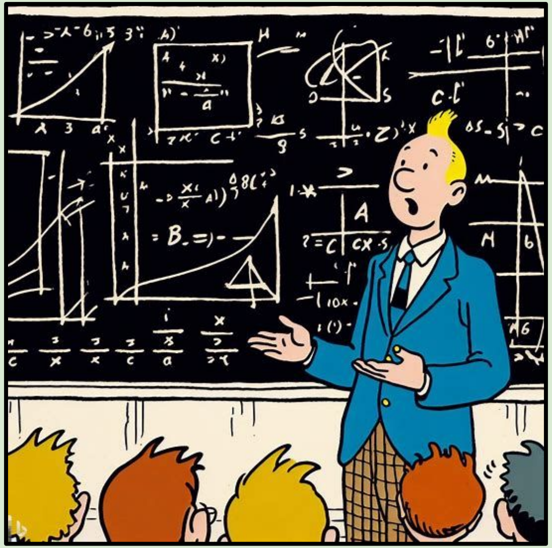
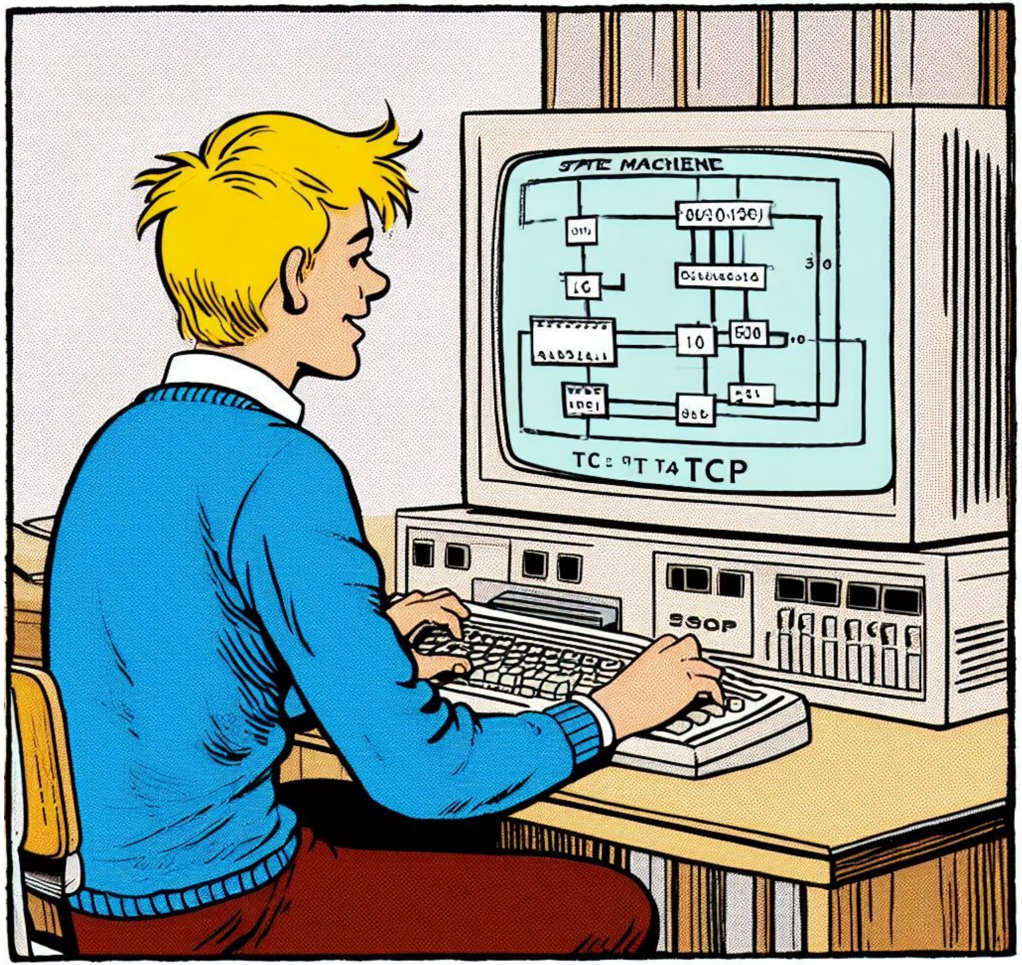
A Case Study in Analytic Protocol Analysis in ACL2

ACL2 Workshop 2023



👋**Max von Hippel**, Pete Manolios, Ken McMillan, Cristina Nita-Rotaru, & Lenore Zuck

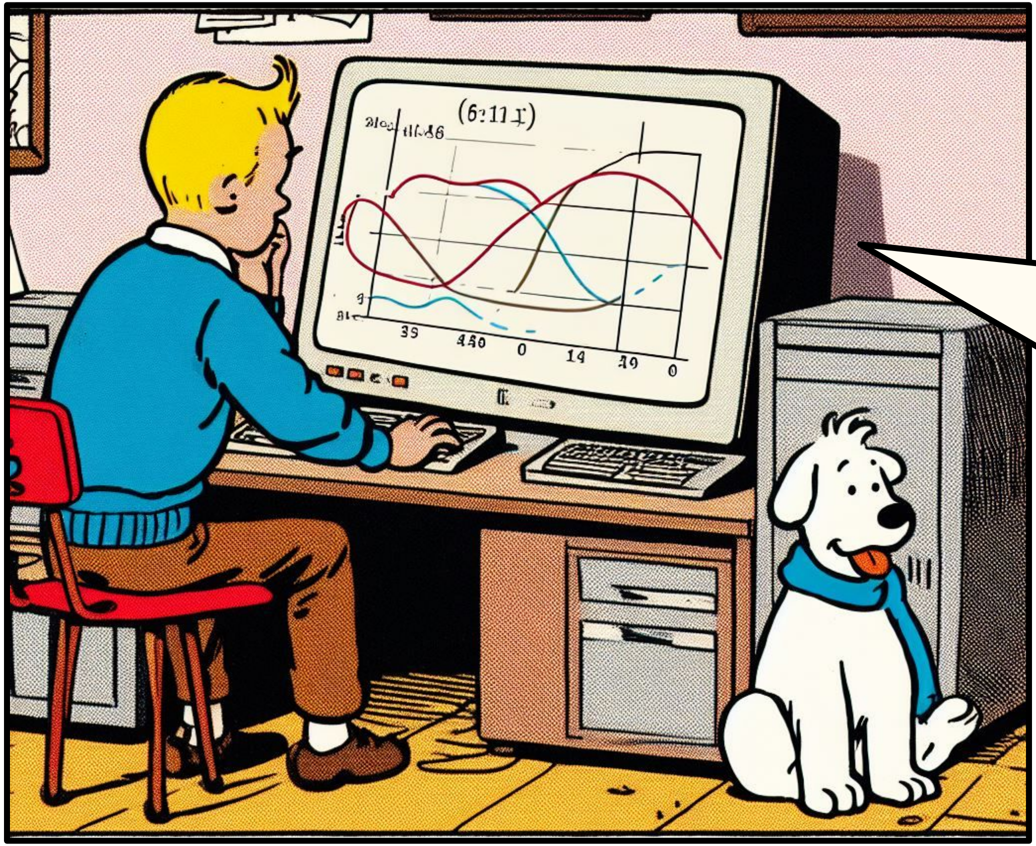


In transport protocols, we have a *sender* and a *receiver*.

RTT = time between transmission & receipt of confirmation of delivery.

RTO = time sender will wait without a new ACK, before retransmitting.

RTO is computed recursively over RTTs.

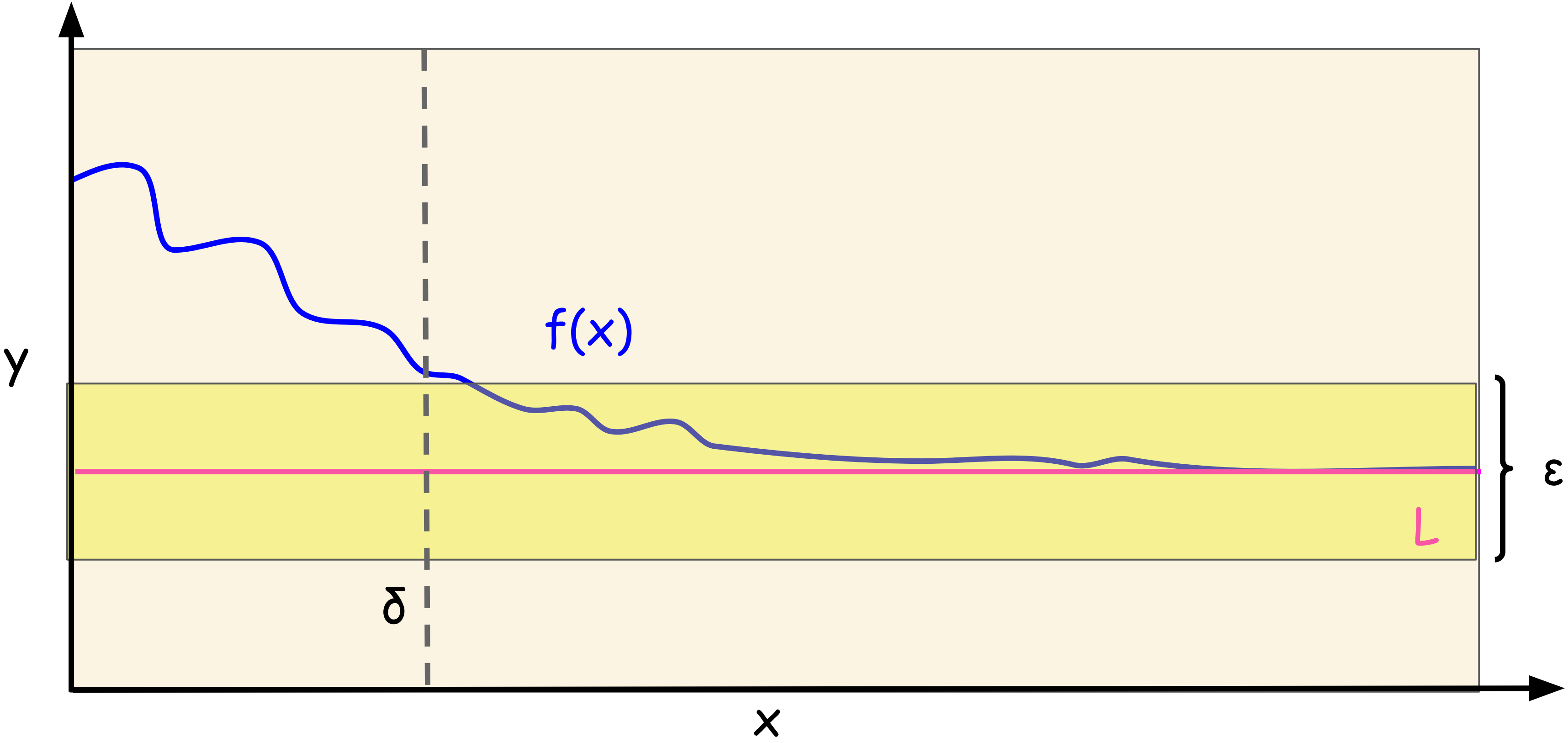


Like lots of other normal well-adjusted people, I spent most of Spring 2023 studying the RTO.

If the RTTs are bounded, can I prove asymptotic bounds on the vars used to compute the RTO?

Unexpected hurdle: Showing that if $0 \leq a < 1$, then $\text{Lim}_{n\text{->}\infty} a^n=0$

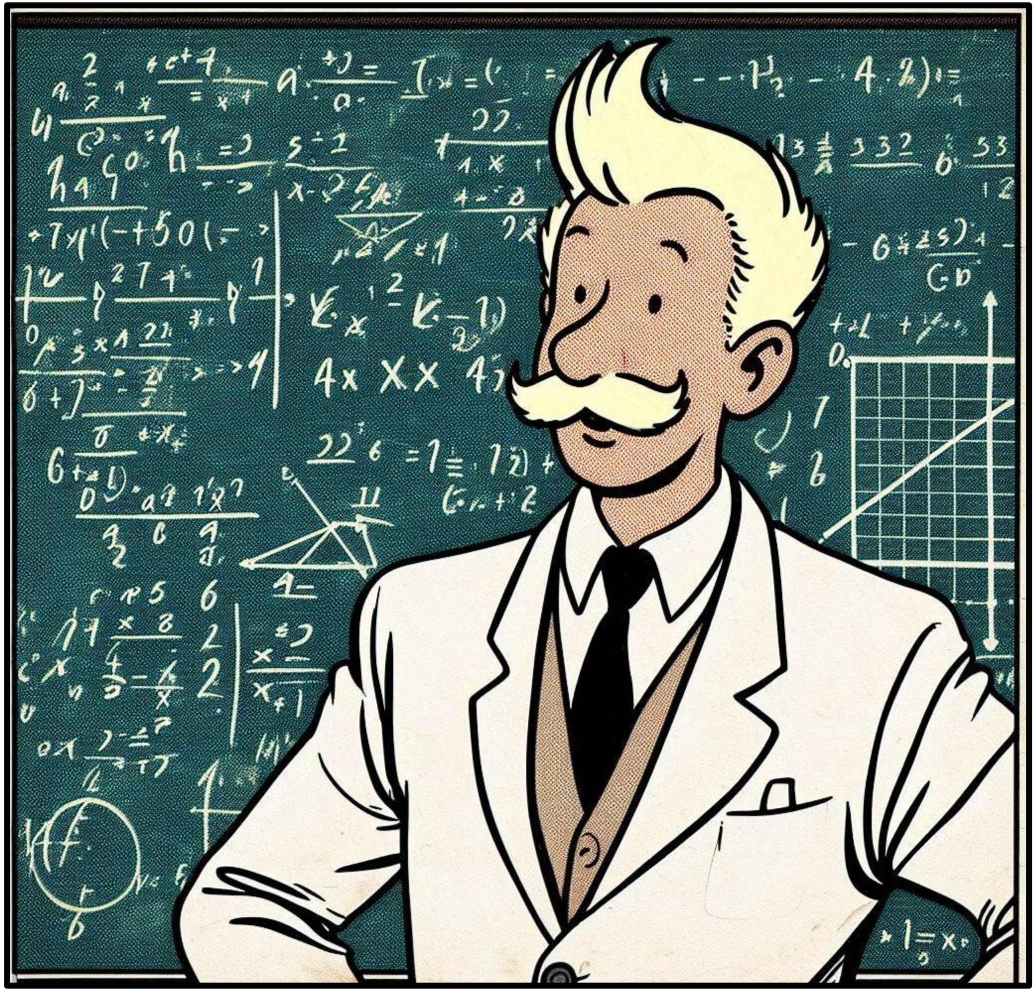
Def: $\lim_{x \to \infty} f(x) = L \Leftrightarrow (\ \forall\ \varepsilon > 0 :: \exists\ \delta > 0 :: x > \delta \Rightarrow |f(x) - L| < \varepsilon)$



Def: $\lim_{n \to \infty} f(x) = L \Leftrightarrow (\forall\ \varepsilon > 0 :: \exists\ \delta > 0 :: \delta < x \Rightarrow |f(x) - L| < \varepsilon)$

```
;; Replace *L* with the actual limit value
(defun-sk limit (x ε)
  (declare (xargs :guard (^ (posratp x) (posratp ε)) :verify-guards t))
  ;; For all e > 0, there exists some δ > 0 such that …
  (exists (δ) (^ (posratp δ)
      ;; … if δ < x, then |f(x) - *L*| < ε.
      (=> (< δ x) (< (abs (- (f x) *L*)) ε)))))

;; To actually prove it, we need to provide “delta”.
(property limit-holds (x e :posrat)
  (limit x e) :instructions …) ;; Use (limit-suff (δ (delta e)))
```



TODAY'S TALK:

- The "Obvious Proof", in ACL2(r), with a little help from Ruben
- Ken's "Ceiling Proof"
- Pete's "Binomial Proof"
- Putting it all in context
- Some thoughts about real numbers in the ACL2 ecosystem.

The "Obvious Proof" that $0 \leq a < 1 \Rightarrow (\forall \varepsilon > 0 :: \exists \delta > 0 :: \delta > n \Rightarrow a^n < \varepsilon)$

Let $0 < \varepsilon < 1$. Set $\delta = \log_a(\varepsilon)$

$$\begin{aligned} n > \delta \quad & \Leftrightarrow n > \log_a(\varepsilon) && \{ \text{def. } \delta \} \\ & \Leftrightarrow a^n < a^{\log_a(\varepsilon)} && \{ \text{as } a < 1 \} \\ & \Leftrightarrow a^n < \varepsilon && \{ x^{\log_x(y)} = y \} \end{aligned}$$

QED

The “Obvious Proof” in ACL2(r)

Let $\varepsilon > 0$. Set $\delta = \ln(\varepsilon)/\ln(a)$. Then …

$n > \delta \quad \Leftrightarrow \quad n > \ln(\varepsilon)/\ln(a)$ { by def. of δ }

$\Leftrightarrow n \ln(a) < \ln(\varepsilon)$ { note, $\ln(a) < 0$ }

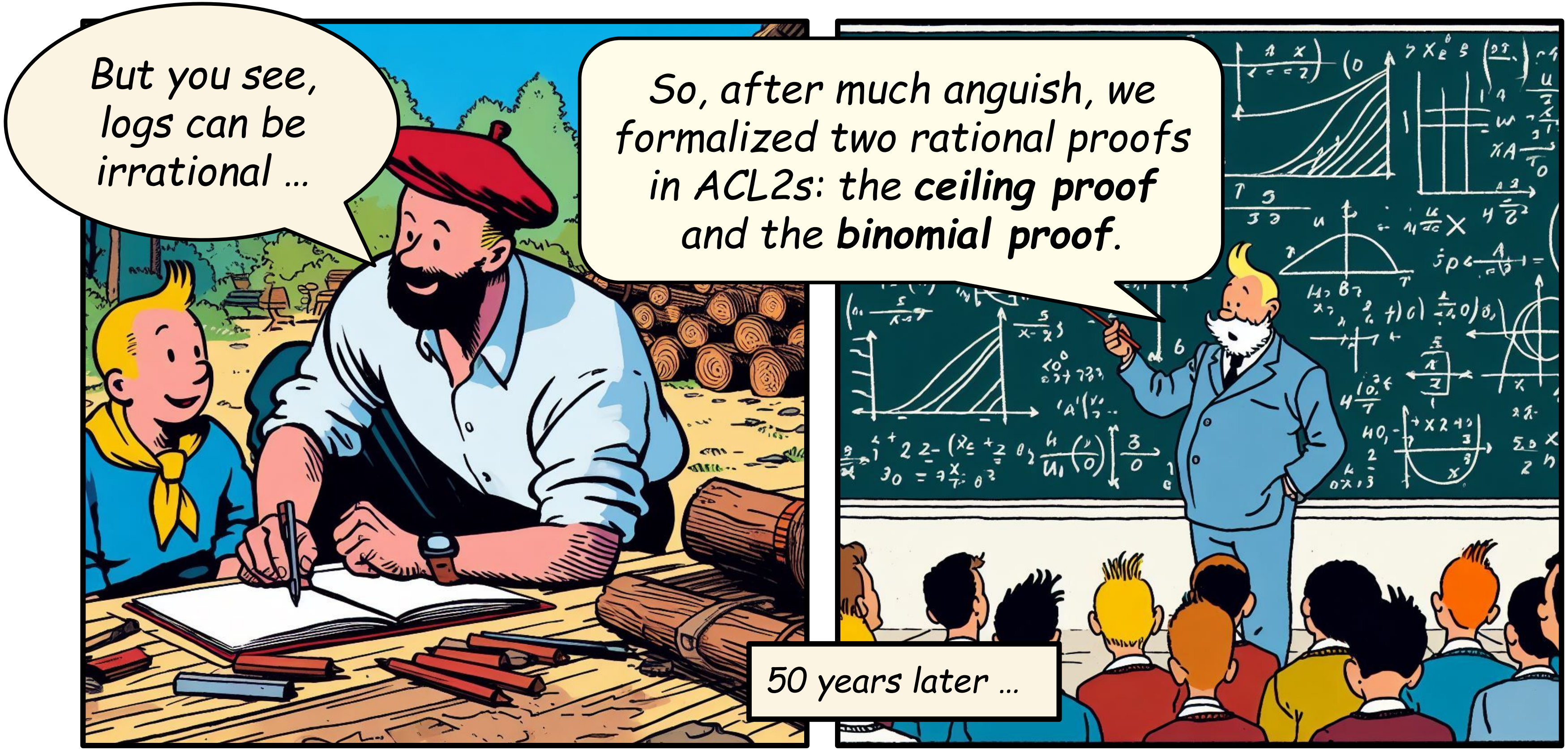
$\Leftrightarrow e^{n \ln(a)} < e^{\ln(\varepsilon)}$ { monotonicity of exp }

$\Leftrightarrow e^{n \ln(a)} < \varepsilon$ { $e^{\ln(x)}=x$ }

$\Leftrightarrow e^{\ln(a^n)} < \varepsilon$ { $n \ln(a) = \ln(a^n)$ }

$\Leftrightarrow a^n < \varepsilon$, QED { $e^{\ln(x)}=x$ }

The “Obvious Proof” doesn’t work in ACL2 …


But you see, logs can be irrational …
So, after much anguish, we formalized two rational proofs in ACL2s: the **ceiling proof** and the **binomial proof**.
50 years later …

The “Ceiling Proof” in ACL2

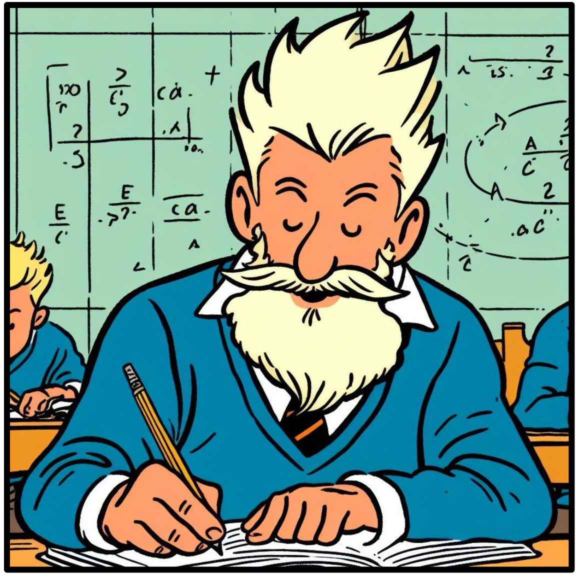
Let $\varepsilon > 0$. Let $k = \lceil a/(1-a) \rceil$.

Let $f_a(n) = ka^k/n$.

Lem A: $\lceil x/mn \rceil = \lceil \lceil x/m \rceil / n \rceil$.

Lem B: $a \leq k/(1+k)$.

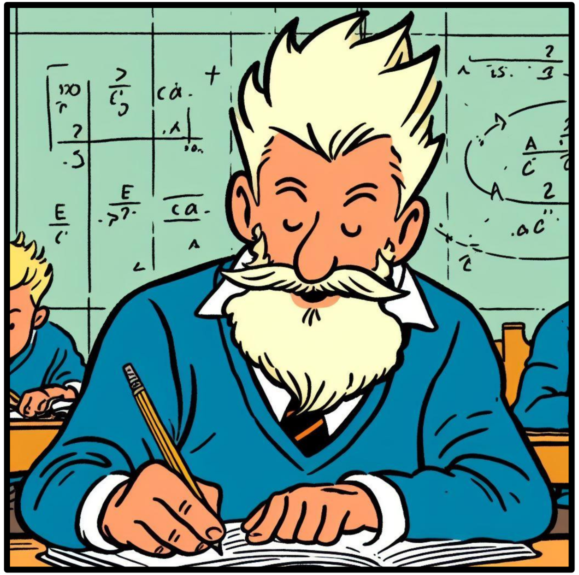
Lem C: For all $n \geq k$, $a^n \leq f_a(n)$.



Thm: Let $d = \lceil ka^k/\varepsilon \rceil$ and let $\delta = \max(k, d)$. $\delta < n \Rightarrow a^n < \varepsilon$.

The “Ceiling Proof” in ACL2

Let $\varepsilon > 0$. Let $k = \lceil a/(1-a) \rceil$.
Let $f_a(n) = ka^k/n$.
Lem A: $\lceil x/mn \rceil = \lceil \lceil x/m \rceil / n \rceil$.
Lem B: $a \le k/(1+k)$.
Lem C: For all $n \ge k$, $a^n \le f_a(n)$.



Thm: Let $d = \lceil ka^k/\varepsilon \rceil$ and let $\delta = \max(k, d)$. $\delta < n \Rightarrow a^n < \varepsilon$.

If $\delta \le n$ then $\lceil ka^k/\varepsilon \rceil < n$, thus $ka^k/\varepsilon < n$. But this implies $ka^k/n < \varepsilon$, and therefore, by Lem C, $a^n \le \varepsilon$. Repeat with, say, $\varepsilon/100$, and you’re done.

The “Binomial Proof” in ACL2

Let $\varepsilon = x/y > 0$, $a = p/q$, and $b = p/(p+1)$.

Lem 1: $a \leq b$.

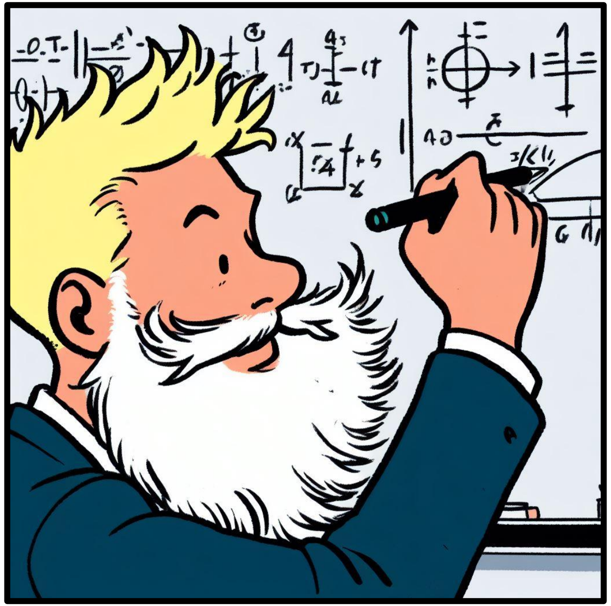
Lem 2: $b^p = p^p/(p+1)^p$ { def. b }

$= p^p/(\ldots + pp^{p-1} + p^p)$ { bin. thm }

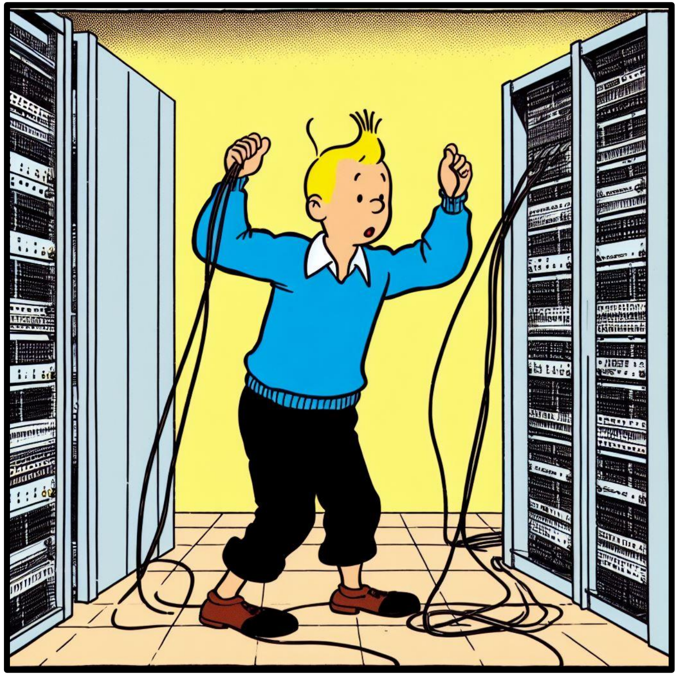
$\leq 1/2$

Lem 3: $1/2^y \leq \varepsilon$

Let $\delta = py$. Then $n > \delta \Rightarrow a^n < a^{py} \leq b^{py} \leq \frac{1}{2}^y \leq \varepsilon$. QED



Putting it all in Context



RTT = time between when a sender sends a packet, and when it first receives an ACK for that packet.

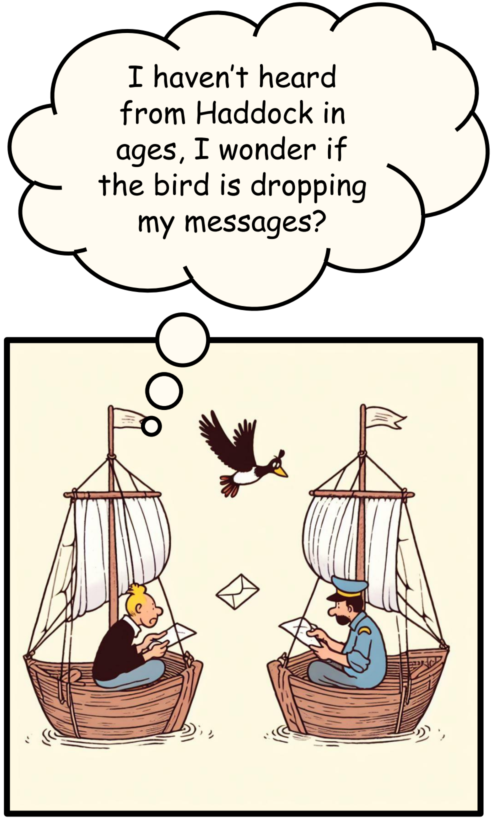
RTT samples S_0, S_1, … measured with Karn's Alg.

RTTs are used to compute the Retransmission TimeOut value.

If > rto time passes without any new ACKs, the sender "backs off" and retransmits unACKed packets.

What happens when samples S_i are bounded?

Putting it all in Context


I haven't heard from Haddock in ages, I wonder if the bird is dropping my messages?

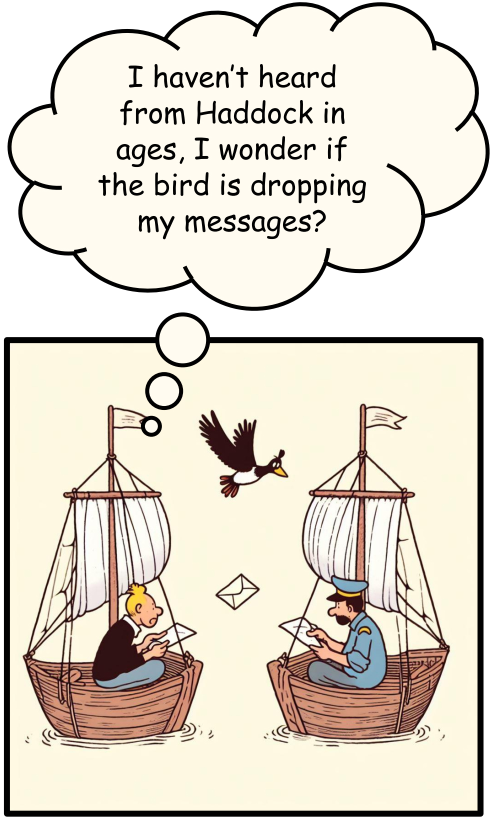
rto is parameterized by $a, b \in [0, 1)$ and $G > 0$.

$$srtt_i = (1-a)srtt_{i-1} + a\, S_i$$

$$rttvar_i = (1-b)rttvar_{i-1} + b\, |srtt_{i-1} - S_i\,|$$

$$rto_i = srtt_i + \max(G, 4 * rttvar_i).$$

Putting it all in Context


I haven't heard from Haddock in ages, I wonder if the bird is dropping my messages?

rto is parameterized by a, $b \in [0, 1)$ and $G > 0$.

$$srtt_i = (1-a)srtt_{i-1} + a\ S_i$$

$$rttvar_i = (1-b)rttvar_{i-1} + b\ |srtt_{i-1} - S_i\ |$$

$$rto_i = srtt_i + \max(G, 4 * rttvar_i).$$

When RTT samples are bounded, srtt is of the form:

$$srtt_i = (1-a)srtt_{i-1} + a\ \text{(some bounds)}$$

When difference between srtt and samples is bounded, rttvar is of the form:

$$rttvar_{i-1} = (1-b)rttvar_{i-1} + b\ \text{(some bounds)}$$

Common Pattern in SRTT and RTTVar Bound Limits

$$X_i = (1-C)X_{i-1} + CD$$

$$= (1-C)^2X_{i-2} + (1-C)CD + CD$$

$$= \dots$$

$$= (1-C)^iX_0 + \Sigma_{j=0}{}^{i-1}(1-C)^jCD$$

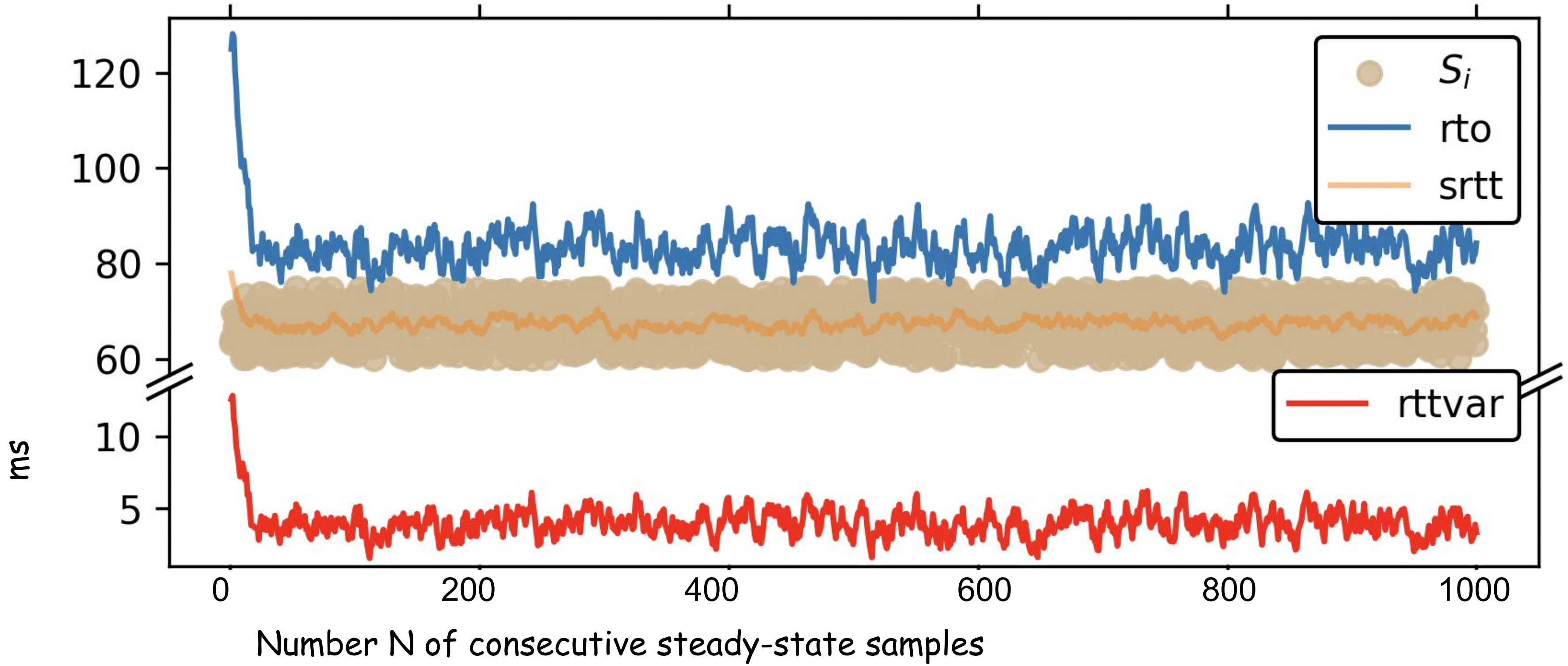
$$= (1-C)^iX_0 + [\ C(1-C)^{i-1} - (1-C)^{i-1} + 1\ ]\ D$$

Common Pattern in SRTT and RTTVar Bound Limits

$$
\begin{aligned}
\lim_{i\to\infty} X_i &= (1-C)X_{i-1} + CD \\
&= (1-C)^2 X_{i-2} + (1-C)CD + CD \\
&= \ldots \\
&= (1-C)^i X_0 + \sum_{j=0}^{i-1} (1-C)^j CD \\
&= \underbrace{(1-C)^i}_{0} X_0 + [\ C\underbrace{(1-C)^{i-1}}_{0} - \underbrace{(1-C)^{i-1}}_{0} + 1\ ]\ D \rightarrow D
\end{aligned}
$$

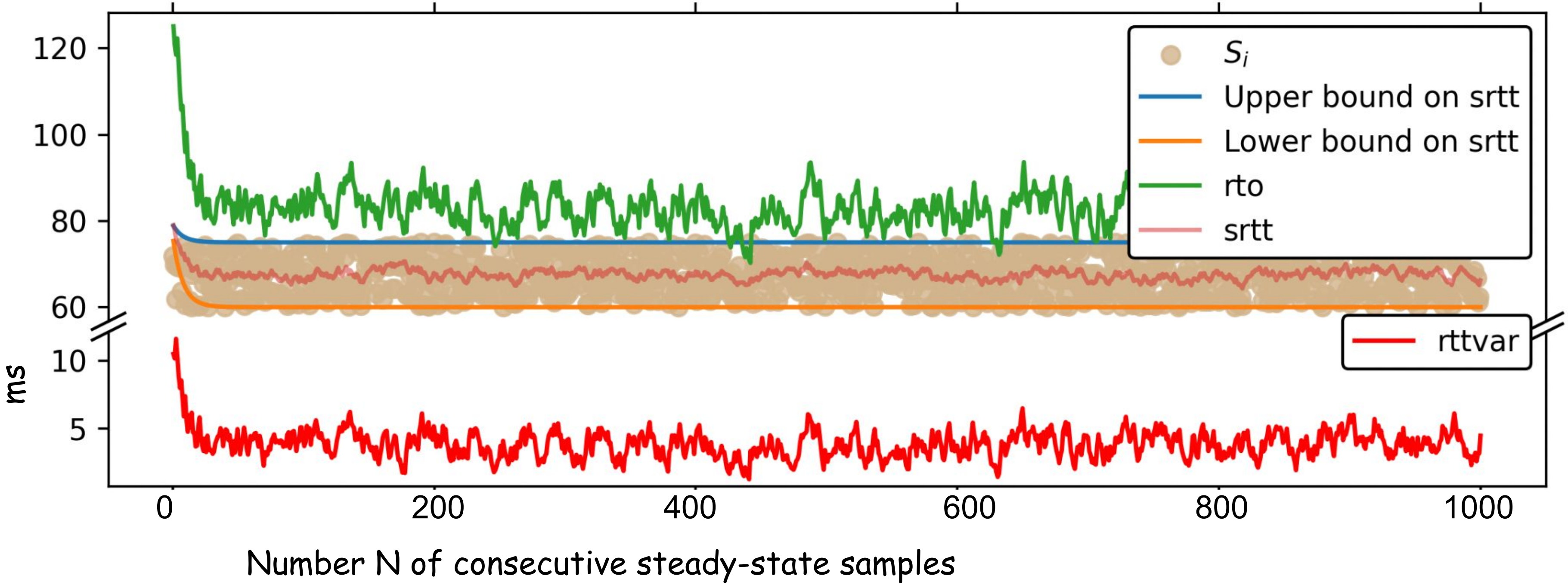
Putting it all in Context

Suppose the samples are bounded.



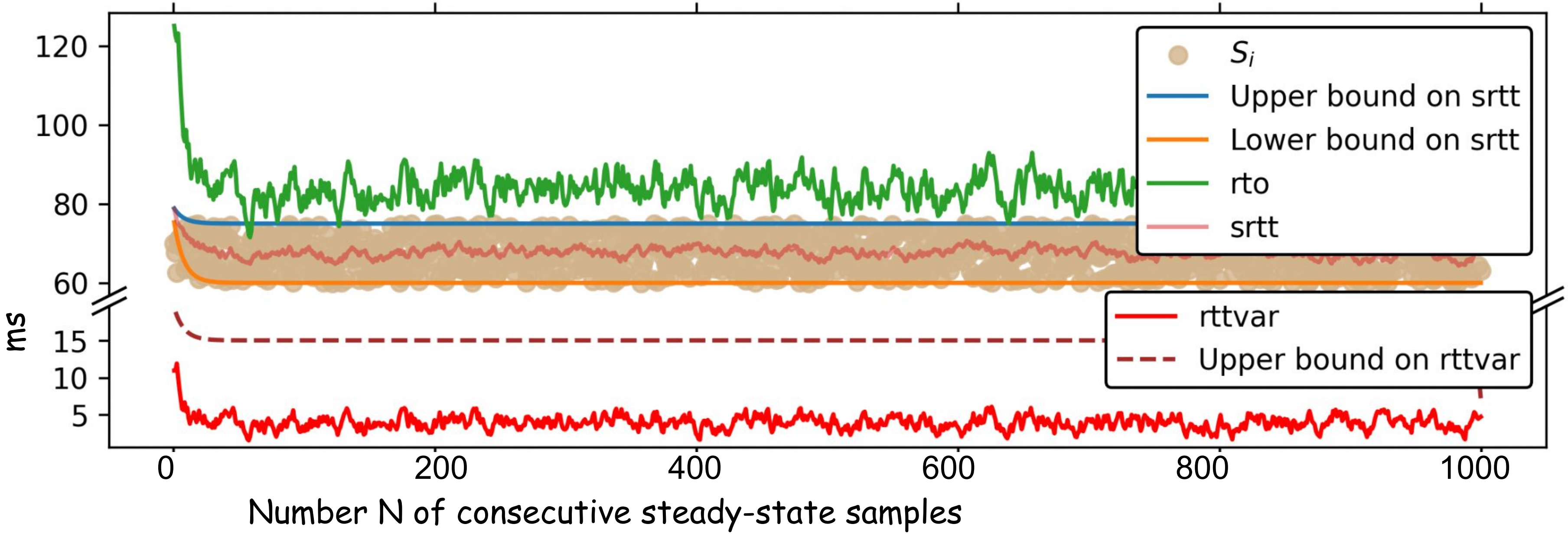
Putting it all in Context

Suppose the samples are bounded. Then the srtt is bounded as well, and its bounds converge to the sample bounds.

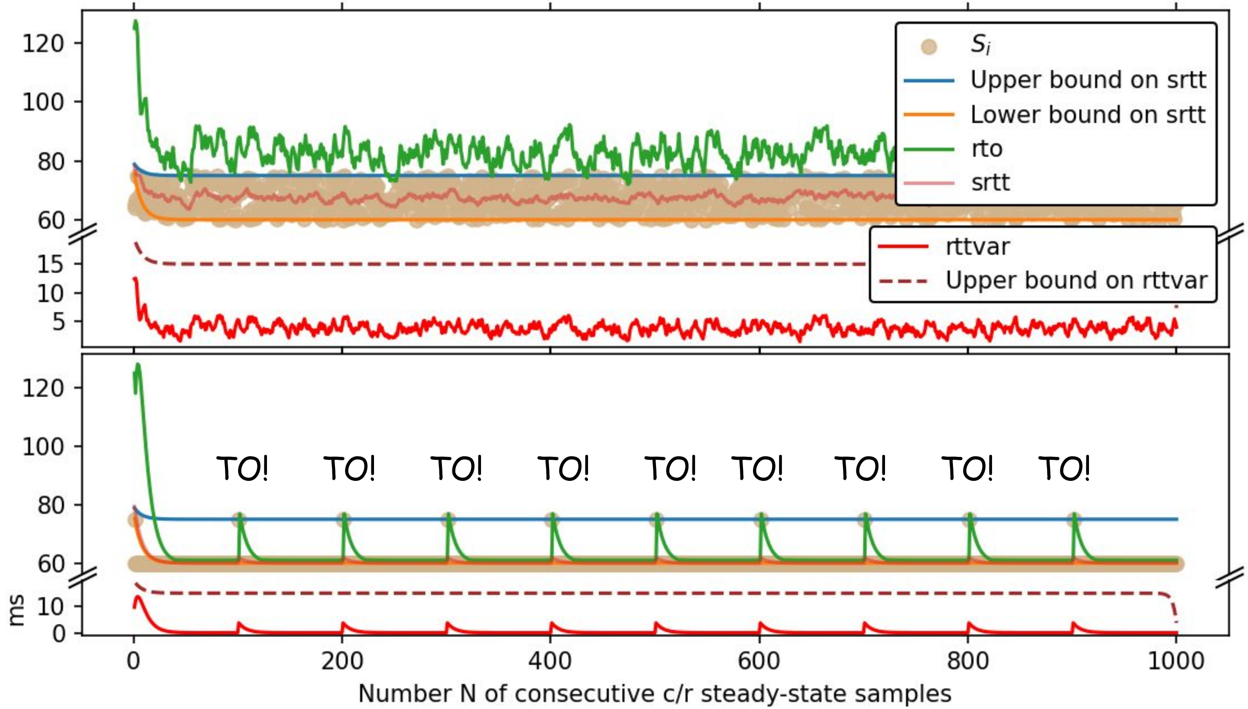


Putting it all in Context

Suppose the samples are bounded. Then the srtt is bounded as well, and its bounds converge to the sample bounds. Plus, the rttvar's upper bound converges to the diameter of the sample bounds.



Nevertheless, infinitely many timeouts could occur!



$S_i \in U(60, 75)$

S_i = 75 if i % 100 = 0 else 60

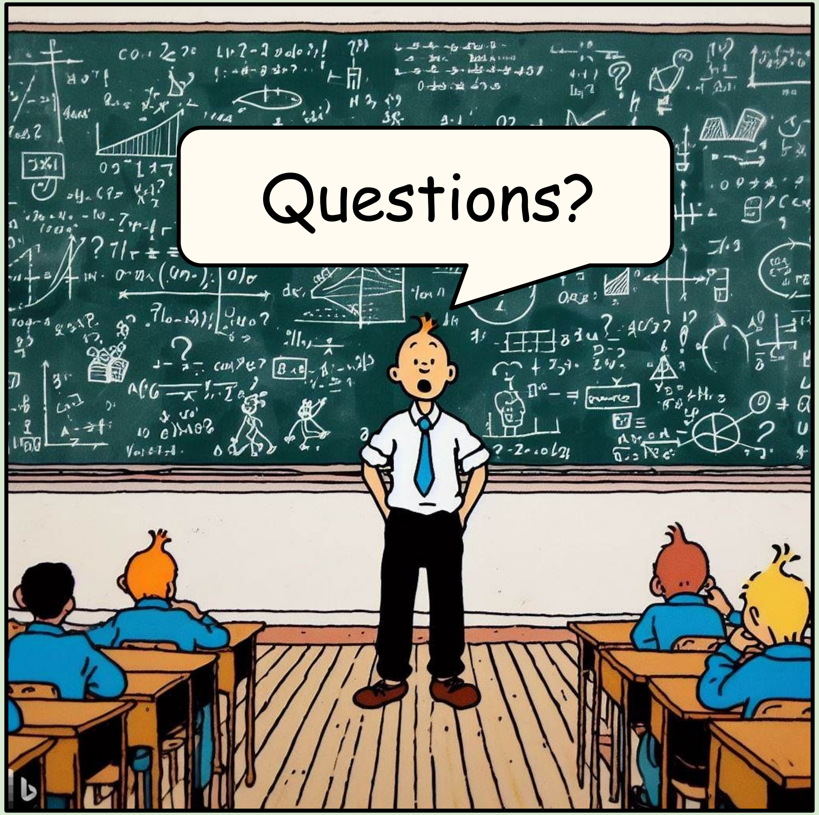
Some thoughts about real numbers in the ACL2 ecosystem.

- ACL2r could benefit from more arithmetic machinery, e.g., arbitrary exponent, arbitrary-base logarithm.

- Although some lemmas we proved already existed in the ACL2 books, searchability could be improved.
  - E.g. would be really interesting to have an LLM-based search tool …

- Easiest proof to write & understand in ACL2 was binomial argument.

- Would be really useful to have some kind of “bridge” between ACL2r and ACL2, but not sure how this could be done (if at all …)
  - (Maybe something building on Grant Passmore’s work?)


Questions?