A Limit Without a Computable Clock

Completion Tails, Halting, and Agent Ensembles

J. R. Landers

Abstract

An unfinished computation may be slow, or it may never finish. We express this distinction through its remaining completion probability. If finite-time completion probabilities are computable and the eventual completion probability is $\theta < 1$, the minimal confidence schedule has exactly the Turing degree of θ. Computable deadlines and universal halting are two cases of this theorem; every computably enumerable degree occurs between them. In the universal case, conditional confidence tends to the known limit one, but its minimal deadline dominates every computable function. Adaptive agent ensembles provide a concrete computable case when their exploration policy supplies an effective tail bound. The arc is to identify the remaining opportunity, classify the information needed to bound it, and use that bound to decide whether to continue.

1 Formalism: unfinished means late or never

Fix a completion criterion and a continuation policy. A run finishes the first time it meets the criterion, and then stops. Let $\tau \in (0, \infty]$ be that completion time; $\tau = \infty$ means the criterion is never met. Randomness may come from choosing a program, drawing model outputs, or executing a randomized procedure. For every finite $t \geq 0$, define

$$S(t) = \mathbb{P}(\tau > t), \qquad M(t) = \mathbb{P}(t < \tau < \infty), \qquad \pi = \mathbb{P}(\tau = \infty). \tag{1}$$

Here $S(t)$ is the unfinished fraction, while $M(t)$ is the part of that fraction that will eventually finish. Splitting the unfinished runs into two disjoint events gives

$$S(t) = \pi + M(t), \qquad \boxed{p_\infty(t) := \mathbb{P}(\tau = \infty \mid \tau > t) = \frac{\pi}{S(t)} = 1 - \frac{M(t)}{S(t)}.} \tag{2}$$

Conditional probabilities are understood only when $S(t) > 0$. Equation (2) holds in discrete or continuous time. It says that a deadline is justified by a bound on the remaining completion mass, rather than by elapsed time alone.

For a chosen error tolerance $\varepsilon \in (0, 1)$, a time T_ε is a valid noncompletion deadline if

$$M(T_\varepsilon) \leq \varepsilon S(T_\varepsilon) \quad \Longrightarrow \quad p_\infty(T_\varepsilon) \geq 1 - \varepsilon. \tag{3}$$

This follows by direct substitution into (2). An upper bound on M can replace its exact value. The probability concerns the population of runs still unfinished at that time; conditioning on an entire observed trajectory can require a different model. A deadline supplies evidence under the fixed policy. Changing the policy changes the future completion probability.

There is a differential form. Assume the finite-time distribution is absolutely continuous and has a nonnegative, locally integrable hazard $h(t)$, where

$$\mathbb{P}(t < \tau \leq t + dt \mid \tau > t) = h(t)\, dt + o(dt).$$

Thus $h(t)$ is the instantaneous completion rate among unfinished runs. Since $\tau > 0$ almost surely, surviving one more interval gives

$$S(t + dt) = S(t)(1 - h(t)dt) + o(dt) \quad \Longrightarrow \quad S'(t) = -h(t)S(t), \quad S(0) = 1. \tag{4}$$

Solving this scalar ODE and taking $t \to \infty$ yields

$$S(t) = e^{-\int_0^t h(s)\,ds}, \qquad \pi = e^{-\int_0^\infty h(s)\,ds}. \tag{5}$$

We use $e^{-\infty} = 0$. Define the remaining integrated hazard

$$R(t) = \int_t^\infty h(s)\,ds. \qquad \text{Then} \qquad \boxed{p_\infty(t) = e^{-R(t)}.} \tag{6}$$

The last identity follows from $p_\infty(t) = \pi/S(t)$. Consequently, a justified bound $R(t) \le \overline{R}(t)$ gives the certificate

$$\overline{R}(T_\varepsilon) \le -\log(1-\varepsilon) \quad \Longrightarrow \quad p_\infty(T_\varepsilon) \ge 1-\varepsilon. \tag{7}$$

These are standard survival identities; distributions with $\pi > 0$ are called defective, or cure, distributions [1]. Their relevance here is computational: can the remaining tail actually be bounded? A hazard tending to zero is insufficient. For example, $h(t) = 1/(1+t)$ still has infinite integrated hazard, so completion occurs almost surely. Finite total integrated hazard, by contrast, gives $\pi > 0$.

The degree of a deadline. Return to integer time. Suppose the rational probabilities $H_t = \mathbb{P}(\tau \le t)$ are uniformly computable and $H_t \uparrow \theta < 1$. The unfinished fraction is $1 - H_t$, and (2) gives

$$p_\infty(t) = \frac{1-\theta}{1-H_t} \uparrow 1, \qquad m(k) = \min\{t \in \mathbb{N}_0 : p_\infty(t) \ge 1 - 2^{-k}\}, \quad k \ge 1. \tag{8}$$

Here $\mathbb{N}_0$ denotes the nonnegative integers. Every value $m(k)$ exists. A valid schedule T meets confidence $1 - 2^{-k}$ at time $T(k)$ for every k.

Theorem 1 (Deadline-degree equivalence)**.** *For each fixed completion system above, T is valid exactly when $T(k) \ge m(k)$ for every k. Furthermore,*

$$\boxed{m \equiv_T \theta, \qquad \theta \le_T T \text{ for every valid } T.} \tag{9}$$

The relation $X \le_T Y$ means that X can be computed with oracle access to Y; $\equiv_T$ means both directions. A real oracle supplies certified approximations to any requested accuracy. Its Turing degree records its information content under this equivalence.

Proof. Monotonicity of p_∞ proves validity iff $T \ge m$. Every valid T satisfies

$$\theta - H_{T(k)} \le 2^{-k}(1 - H_{T(k)}) \le 2^{-k}, \qquad H_{T(k)} \le \theta \le H_{T(k)} + 2^{-k}. \tag{10}$$

Thus T computes θ, in particular $\theta \le_T m$. Conversely, the defining inequality for $m(k)$ is

$$\theta \le H_t + 2^{-k}(1 - H_t).$$

If θ is irrational, the right side is rational and unequal to θ, so increasingly precise oracle approximations decide each comparison. Searching $t = 0, 1, \ldots$ returns $m(k)$. If θ is rational, its exact value can instead be built into an algorithm, making every comparison decidable directly. This proves $m \le_T \theta$ in both cases. The rational case is nonuniform: an arbitrary approximation oracle need not reveal that its value is rational or supply the exact value. □

Corollary 2 (Computable deadlines)**.** *Under the theorem's assumptions,*

$$\boxed{\textit{a computable confidence schedule exists} \quad \Longleftrightarrow \quad \theta \textit{ is computable.}} \tag{11}$$

Proof. A computable valid T computes θ by (10); computable θ makes the minimal schedule m computable by the theorem. □

This is a degree formulation of the standard computable-modulus principle. It states the information needed for a deadline. Finding checkable structural conditions that provide that information remains a separate task.

2 The halting problem: a limit without a computable deadline

For a fixed deterministic program and input, halting is a definite property. A probability requires a distribution. Consider an optimal universal prefix-free machine U fed independent fair bits. No halting program is a proper prefix of another, so their corresponding input events are disjoint. The eventual halting probability is

$$\Omega = \mathbb{P}(\tau < \infty) = \sum_{w:\, U(w) \text{ halts}} 2^{-|w|}, \qquad 0 < \Omega < 1, \tag{12}$$

where w is a binary program and $|w|$ its length. Optimal universality permits simulation of other prefix-free machines with constant overhead in program length. Chaitin's classical theorem makes Ω algorithmically random and hence uncomputable [2].

Use a machine convention with at most one input bit read and one output bit written per step. Simulating all 2^t bit prefixes for t steps computes the rational probability H_t exactly, and $H_t \uparrow \Omega$. The general theorem applies with $\theta = \Omega$:

$$p_\infty(t) = \frac{1 - \Omega}{1 - H_t} \uparrow 1, \qquad m_U \equiv_T \Omega. \tag{13}$$

The subscript distinguishes this system's minimal schedule. Its confidence limit is known; the obstruction is its rate, or *modulus of convergence*.

Let $B_U(\ell)$ be the longest halting runtime among U-programs of length at most ℓ, with value zero if there are none. This is a program-length Busy Beaver function [3].

Corollary 3 (Universal deadlines)**.** *Every valid schedule for U computes Ω and decides halting. The minimal schedule has exactly the halting problem's Turing degree, and*

$$\boxed{m_U(k) \geq B_U(k-1).} \tag{14}$$

Consequently $m_U(k)$ eventually exceeds every total computable function of k.

Proof. If a halting program of length at most $k-1$ remained unfinished at $m_U(k)$, it would contribute at least $2^{-(k-1)} > 2^{-k}$ to $\Omega - H_{m_U(k)}$, contradicting (10). This proves the bound. A valid $T \geq m_U$ therefore decides whether $U(w)$ halts by simulating it for $T(|w|+1)$ steps. Conversely, a halting oracle decides whether some H_t exceeds any given rational, hence computes Ω and then m_U.

For any total computable function f, a self-delimiting description of k and a fixed routine can print $f(k)+1$ bits and halt. Optimality supplies a U-program for this output of length $O(\log k) + c_f$, where c_f depends only on f. Its runtime exceeds $f(k)$ by the output convention. For all sufficiently large k its length is at most $k-1$, so $B_U(k-1) > f(k)$. □

The degrees between the extremes. A real is *left-c.e.* if it is the limit of a computable increasing rational sequence. A set is computably enumerable (c.e.) if an algorithm can list its elements. A c.e. Turing degree is one containing such a set.

Corollary 4 (The deadline-degree spectrum)**.** *Among systems with uniformly computable rational $H_t \uparrow \theta < 1$, the degrees of minimal schedules are exactly the c.e. Turing degrees. Every such degree already occurs for a prefix-free machine.*

Proof. Each θ is left-c.e. Its lower rational cut $\{r \in \mathbb{Q} : r < \theta\}$ is a c.e. set, since $r < \theta$ exactly when $r < H_t$ for some t. For irrational θ this cut and θ compute one another; rational θ has computable degree. Theorem 1 therefore restricts the spectrum to c.e. degrees.

Conversely, for any c.e. set $W \subseteq \mathbb{N}_0$, put

$$\theta_W = \sum_{j \in W} 4^{-(j+1)} \leq \frac{1}{3} < 1. \tag{15}$$

Enumeration makes θ_W left-c.e. Its base-four digits are the membership bits of W. Knowing those bits approximates the sum with a geometric tail bound. Conversely, subtract the known earlier digits and multiply by 4^{j+1}: the residual lies in $[0, 1/3]$ if $j \notin W$ and in $[1, 4/3]$ if $j \in W$. The gap lets sufficiently accurate approximations determine each bit. Thus $\theta_W \equiv_T W$.

For each new $j \in W$, request a halting code of length $2j + 2$. Their total Kraft weight is $\theta_W \leq 1/3$, so the Kraft–Chaitin construction [4] supplies a prefix-free machine with exactly these halting codes and halting probability θ_W. Its computable finite-time probabilities fall under Theorem 1, giving a minimal deadline of degree W. □

Computable deadlines are the degree-zero case; universal Ω gives the halting degree, conventionally written $\mathbf{0}'$. The intermediate c.e. degrees are genuine possibilities. A nonminimal valid schedule can encode extra information, so the equivalence concerns the minimal schedule. The ingredients are classical effective convergence and prefix-free coding; the deadline formulation shows how their information requirements match.

Calude and Stay [5] obtain effective probabilistic deadlines by postulating a computable runtime distribution. Ranking and repulsing supermartingales provide structural guarantees for specified probabilistic program classes [6]. The remaining research question is which checkable assumptions make the relevant completion probability, or a sufficient tail bound, computable.

3 An application to adaptive agent ensembles

Fix a question Q and one frozen language model. Its n agents are instances of that model, distinguished by sampling and feedback. Conditional on the current history $\mathcal{F}_k$, their round-k proposals can be written

$$A_{1,k}, \ldots, A_{n,k} \overset{\text{iid}}{\sim} D_Q(\cdot \mid \mathcal{F}_k), \qquad A'_k = T_k(A_{1,k}, \ldots, A_{n,k}), \tag{16}$$

where T_k aggregates the proposals. Repeated sampling underlies self-consistency [7]; adaptation changes the next distribution using earlier outputs. Define completion as the first aggregate answer to pass a fixed acceptance test. The stopping time τ refers to that test, whose soundness determines what acceptance guarantees.

A policy with an explicit tail. To obtain a tractable aggregation rule, let the aggregator check every proposal and accept the first that passes. We now use continuous proposal times rather than synchronous rounds.

Before acceptance, let each agent propose at the times of an independent Poisson process of rate $r > 0$. Suppose all agents use

$$D_t = (1 - e^{-\beta t}) D_{\text{stalled},t} + e^{-\beta t} D_{\text{explore}}, \qquad \beta > 0. \tag{17}$$

The stalled component never passes the test. Each exploratory proposal passes and is recognized with fixed probability $q \in (0, 1]$. The aggregator checks each proposal and accepts the first that passes. Draws and proposal clocks are independent; feedback may alter the stalled component while its acceptance probability remains zero. This is an explicit model of a policy increasingly favoring familiar proposals.

In an interval of length dt, there is one proposal with probability $nr\,dt + o(dt)$ and more than one with probability $o(dt)$. A proposal is exploratory and acceptable with probability $qe^{-\beta t}$. Therefore the completion hazard is derived from the policy:

$$h(t) = nrqe^{-\beta t} = ae^{-\beta t}, \qquad a := nrq. \tag{18}$$

The rates a and β have units of inverse time. Substitution into the formalism of Section 1 gives

$$\begin{aligned} S'(t) &= -ae^{-\beta t}S(t), \qquad S(0) = 1, \\ S(t) &= \exp\left[-\frac{a}{\beta}(1 - e^{-\beta t})\right], \qquad \pi = e^{-a/\beta} > 0, \\ R(t) &= \frac{a}{\beta}e^{-\beta t}, \qquad p_\infty(t) = \exp\left[-\frac{a}{\beta}e^{-\beta t}\right]. \end{aligned} \tag{19}$$

Here $R(t)$ is also the expected number of future acceptable opportunities in the Poisson proposal stream if it were continued without stopping. Exploration supplies a finite total of such opportunities, which explains the positive probability of never accepting an answer.

Proposition 5 (An effective deadline for fading exploration)**.** *Under this policy, for any desired confidence $p \in (0, 1)$, the deadline*

$$\boxed{T_p = \max\left\{0, \frac{1}{\beta}\log\left(\frac{nrq}{\beta(-\log p)}\right)\right\}} \tag{20}$$

satisfies $\mathbb{P}(\tau = \infty \mid \tau > t) \geq p$ for every $t \geq T_p$.

Proof. By (19), the desired inequality is equivalent to

$$\frac{nrq}{\beta}e^{-\beta t} \leq -\log p, \qquad t \geq \frac{1}{\beta}\log\left(\frac{nrq}{\beta(-\log p)}\right).$$

Restricting time to $t \geq 0$ gives (20). □

For example, take $n = 4$, $r = 1$ per unit time, $q = 1/4$, and $\beta = 1$ per unit time. Then $a/\beta = 1$, $\pi = e^{-1} \approx 0.368$, and

$$T_{0.95} \approx 2.970, \qquad \mathbb{P}(\tau = \infty \mid \tau > 3) = e^{-e^{-3}} \approx 0.9514.$$

An unfinished run after three time units is therefore over 95% likely never to pass the test under this policy. Resetting exploration changes that assessment.

The exploration threshold. Replace $e^{-\beta t}$ by any deterministic exploration probability $e(t) \in [0, 1]$. Under the same proposal assumptions, with fixed $n, r, q > 0$,

$$h(t) = nrqe(t), \qquad \boxed{\pi > 0 \iff \int_0^\infty e(t)\,dt < \infty.} \tag{21}$$

This follows directly from (5). Exponential decay gives finite total exploration; $e(t) = 1/(1 + t)$ gives infinite total exploration and eventual acceptance almost surely.

The same coverage principle appears in reinforcement learning: GLIE means *greedy in the limit with infinite exploration* [8]. Vanishing exploration must still supply infinitely many visits to the relevant state-action pairs. In a multi-state system, a non-summable exploration schedule alone need not ensure that coverage. Simulated annealing adds a barrier-depth dependence [9]: in an idealized barrier model, an escape hazard proportional to $\exp[-\Delta/\Theta(t)]$, with barrier height $\Delta > 0$ and temperature $\Theta(t) = c/\log(1+t)$, $c > 0$, is proportional to $(1+t)^{-\Delta/c}$. In normalized time its integral diverges exactly when $c \geq \Delta$. This illustrates why cooling slowly enough preserves escape opportunities; the full annealing theorem also accounts for the landscape geometry.

The universal case and the designed agent policy illustrate the halting-degree and computable-deadline cases, respectively. In this explicit Poisson model, computable parameters n, r, q, β give computable finite-time probabilities and eventual completion probability $1 - e^{-nrq/\beta}$. Its explicit tail bound supplies deadlines directly, without the exact-rational hypothesis of Theorem 1. For a real synthesizing LLM, a justified future-hazard bound, including parameter uncertainty, remains to be established. The task is to identify structural assumptions that provide that information. A known confidence limit becomes a useful stopping rule only when its rate can be controlled.

References

[1] A. D. Tsodikov, J. G. Ibrahim, and A. Y. Yakovlev. Estimating cure rates from survival data: An alternative to two-component mixture models. *JASA* 98(464):1063–1078, 2003.

[2] G. J. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM* 22(3):329–340, 1975.

[3] S. Aaronson. The Busy Beaver frontier. *SIGACT News* 51(3):31–55, 2020.

[4] C. S. Calude. A characterization of c.e. random reals. *Theoretical Computer Science* 271:3–14, 2002.

[5] C. S. Calude and M. A. Stay. Most programs stop quickly or never halt. *Advances in Applied Mathematics* 40(3):295–308, 2008. arXiv:cs/0610153.

[6] K. Chatterjee, P. Novotný, and Đ. Žikelić. Stochastic invariants for probabilistic termination. *POPL*, 2017. arXiv:1611.01063.

[7] X. Wang et al. Self-consistency improves chain of thought reasoning in language models. *ICLR*, 2023. arXiv:2203.11171.

[8] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning* 38(3):287–308, 2000.

[9] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research* 13(2):311–329, 1988.