Making Solomonoff Induction Effective

You Can Learn What You Can Bound

Jörg Zimmermann and Armin B. Cremers

Institute of Computer Science
University of Bonn, Germany

The General Prediction Problem

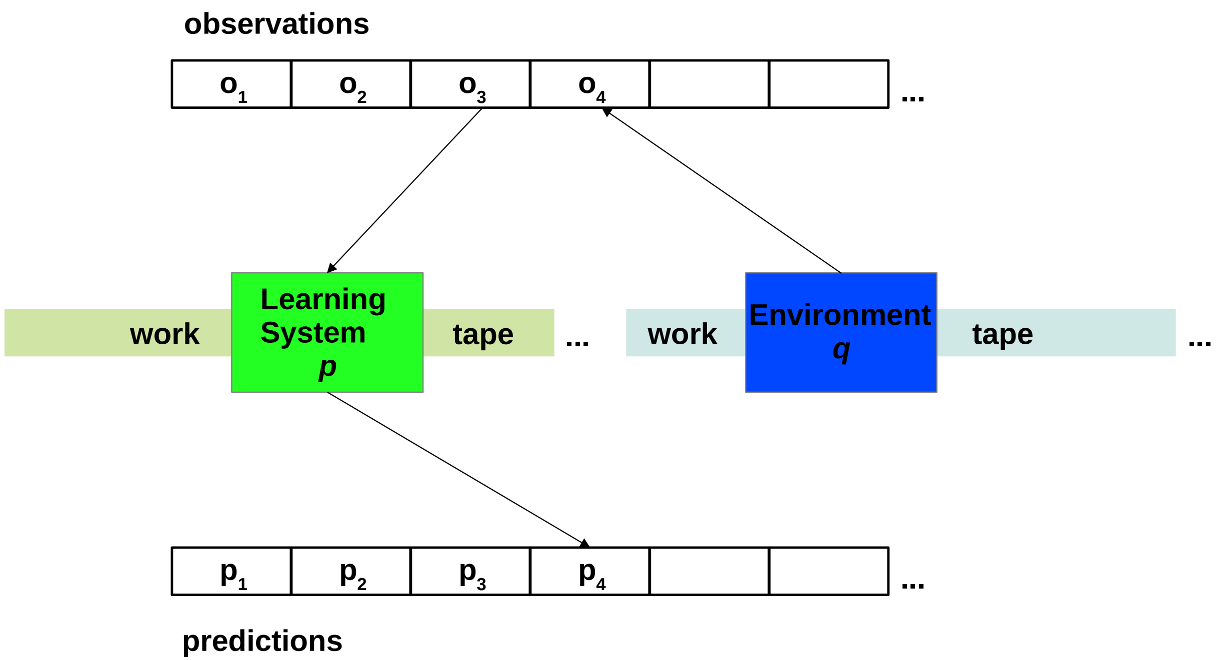
Given a finite sequence of bits, e.g.:

0010010000111111011010101000100010000101

Question: What is the next bit?

Asynchronous Learning Framework (ALF)

A learning system observing and predicting an environment:



Solomonoff Induction

- Bayesian learning in program space.
- Prior $\sim 2^{-|p|}$, $|p| =$ length of program p in bits.
- But posterior distribution on program space is not computable!
  (the programs stopping to produce output cause trouble).

Key Points of our Approach

- Learning driven by a combined search in program and proof space.
- Reduction of learnability to provability and set existence axioms.

  Axiom systems of reverse mathematics and large cardinal axioms can be used to show that proof-theoretic strength translates into learning strength.
- Introduction of a new learning framework, the *Synchronous Learning Framework (SLF)*, which couples the time scales of the learning system and the environment.

Probabilistic Learning Systems

$$\Lambda : \{0,1\}^* \times \{0,1\} \to [0,1]_{\mathbf{Q}}$$

with $\Lambda(x,0) + \Lambda(x,1) = 1$ for all $x \in \{0,1\}^*$.

Λ is an *effective* probabilistic learning system if Λ is a total recursive function.

Learnability

Learning as learning in the limit:

Eventually the learning system will become near certain about the true continuation of the observed bit sequence.

Definition: An infinite bit sequence s is learnable in the limit by the probabilistic learning system Λ, if for all $\epsilon > 0$ there is an n_0 so that for all $n \geq n_0$ and all $k \geq 1$:

$$\Lambda^{(k)}(s_{1:n}, s_{n+1:n+k}) > 1 - \epsilon.$$

$\Lambda^{(k)}$: extending prediction horizon to k bits by feeding Λ with its own predictions.

Σ-driven Learning Systems

- Turning an Axiom System Σ into a Learning System Λ:

$$\Sigma \quad \longrightarrow \quad \Lambda(\Sigma)$$

- A Σ-driven learning system is a learning system using the background theory Σ in order to derive totality proofs for recursive functions.
- These provably recursive functions are used to build a guard function, which schedules the learning process and guarantees its effectiveness.

Generator Time Function

The *generator time function* of a program p is defined as:

$$G_p : \mathbf{N} \to \mathbf{N} \cup \{\infty\}$$

$G_p(n) = \#$transitions executed by p to generate the first n bits.

Observation Equivalence

$s_p =$ the bit sequence generated by program p.

Then the *observation class* $[s]$ of a bit sequence s is defined as:

$$p \in [s] \quad \text{iff} \quad s = s_p.$$

Generator-Predictor Theorem

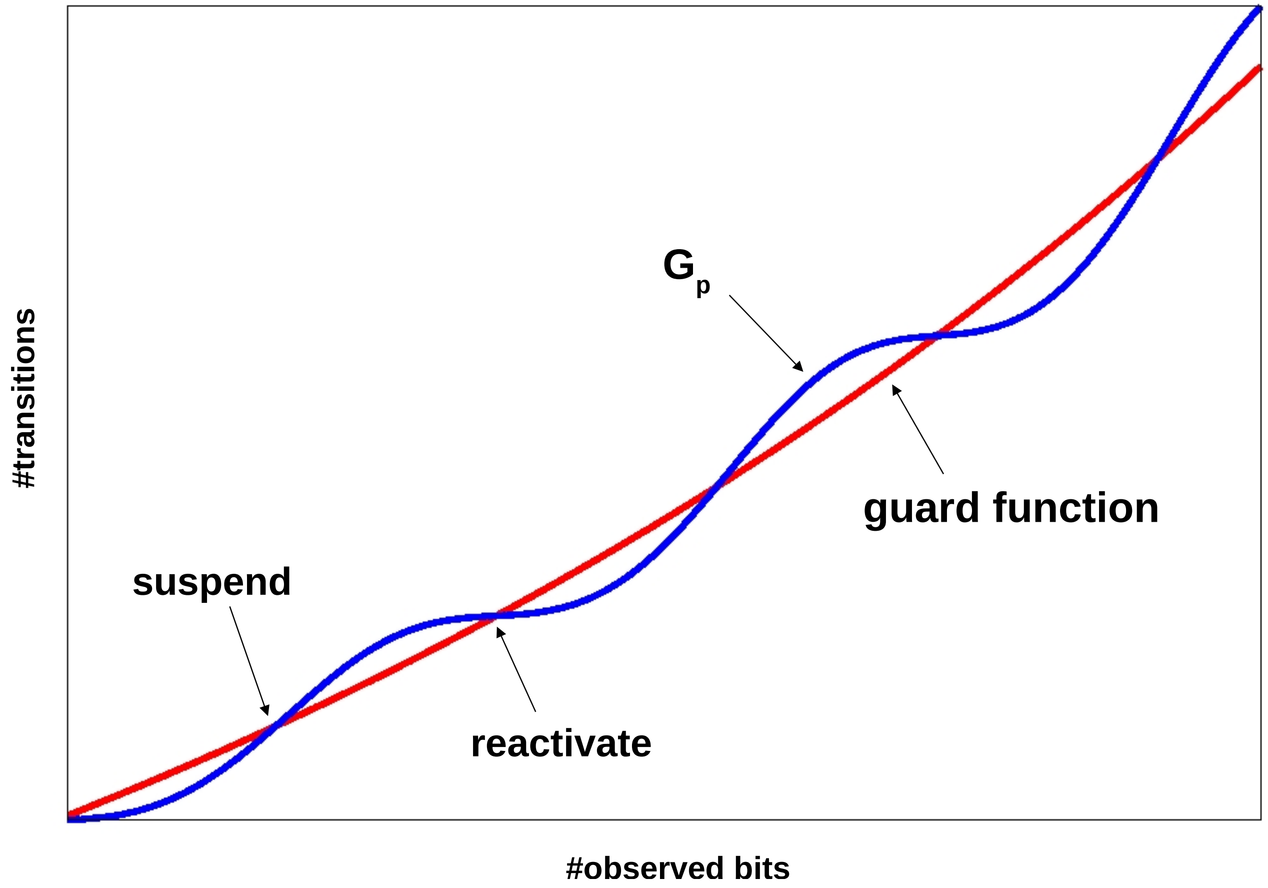
The infinite bit sequence s is learnable by $\Lambda(\Sigma)$, if:

$$\exists\, p \in [s], f \text{ recursive function :} \quad \Sigma \vdash \phi_{tot}(f) \;\text{ and }\; f \geq_d G_p.$$

$\phi_{tot}(f) \;=\; f$ is a total recursive function.

$f \geq_d g \;=\; f$ dominates g (i.e., $\exists n_0\ \forall n \geq n_0 : f(n) \geq g(n)$).

The hard case: infinite number of switches



Σ-driven probabilistic learning system

Idea: retroactive change of prior:

$$\sim 2^{-(|p|+switch(p,n))} \qquad \text{(Solomonoff prior} \sim 2^{-|p|}\text{)}$$

$\Longrightarrow$ Dynamic Bayesian Inference, i.e., construction of model space and prior probabilities is interleaved with the inference process.

Conclusions 1

- The generator-predictor theorem establishes a natural perspective on the effective core of Solomonoff induction.
- This shifts the questions related to learnability to questions related to provability, and therefore into the realm of the foundations of mathematics.

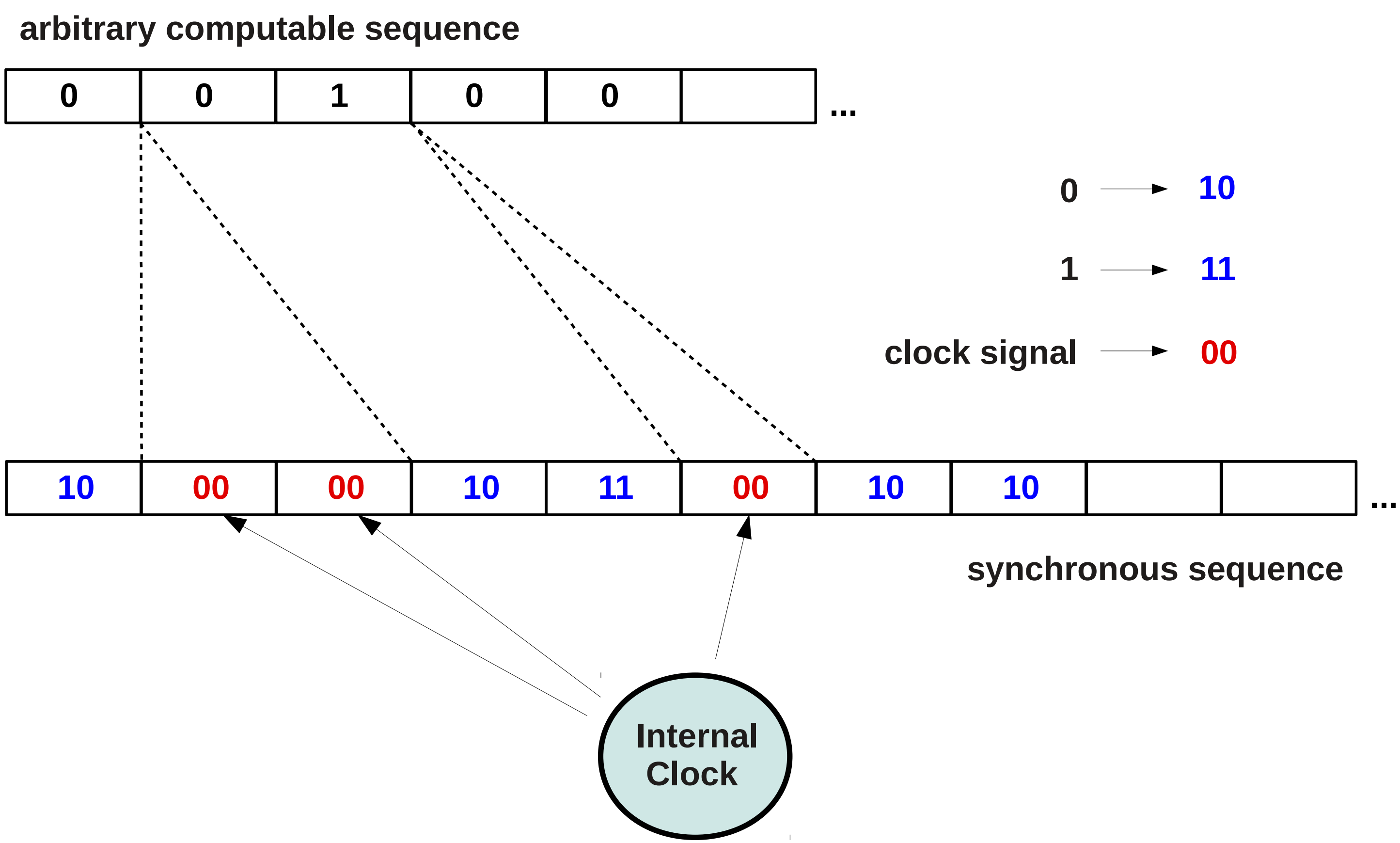
Synchronous Learning Framework (SLF)

Observation: in real world learning situations, the generator and the learner are not suspended while the other one is busy.

s is synchronous $\quad :\Longleftrightarrow \quad \limsup_{n\to\infty} \frac{G_p(n)}{n} < \infty$ for at least one $p \in [s]$.

$\Longrightarrow$ the time scales of the learning system and the environment are coupled.

Clockification



Synchronous Learning Framework

- Clockification transforms every computable bit sequence into a synchronous one.
- All synchronous bit sequences are learnable by $\Lambda(\Sigma)$, if $\Sigma \vdash$ "n^2 is a total recursive function".
- Thus in the SLF all effectively generated bit sequences can be effectively learned.

Final Conclusion

If the learning system is enhanced by an internal clock:

Effective universal induction is possible!

Hence future research can focus on efficient universal induction.