# Constructive Algorithms

- A **algorithm** is **constructive** if its initial states consist of free constructors.

# Constructive Algorithms

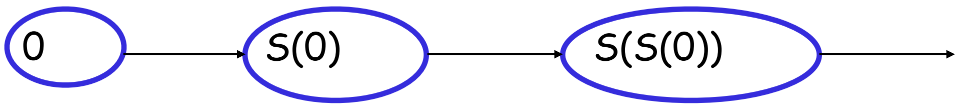
- *A* **algorithm** is **constructive** if its initial states consist of free constructors, plus operations (inductively) constructed in the same way.

# Effective Base Structure

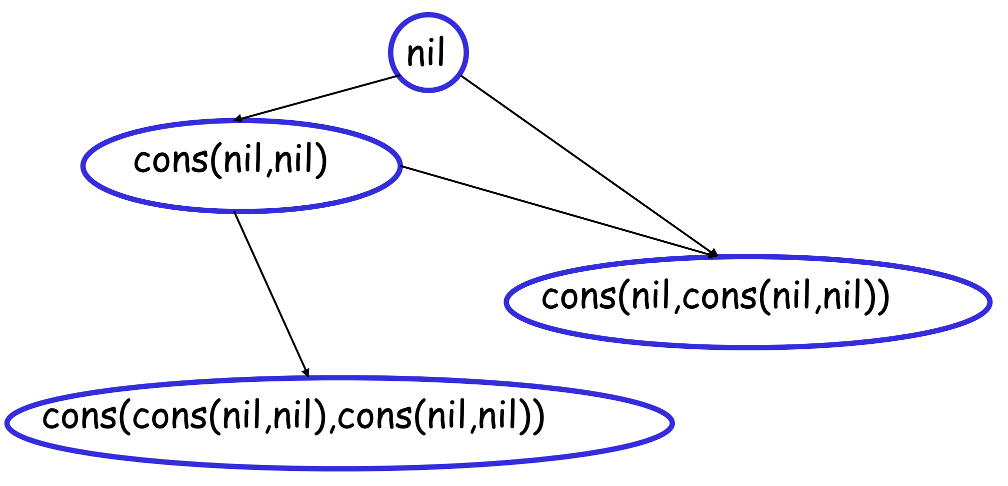
- Every element is the value of a unique term.
- Isomorphic to the free term (Herbrand) algebra.
  - **If one wants more than one name, then one must program them**

# Examples

- $A = \{ \mathbf{N}; 0, S \}$



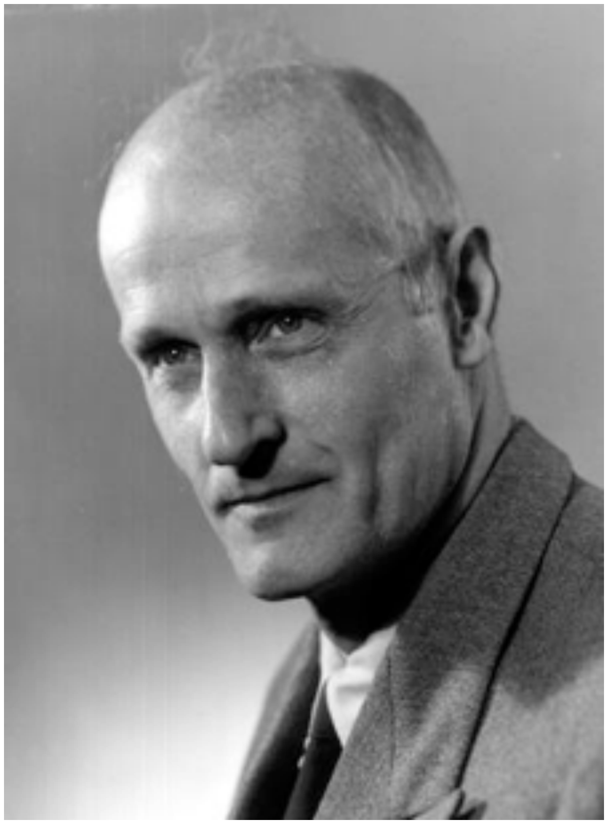

- B = { binary trees; nil, cons }



# Turing's Thesis

Turing Machines
capture
mechanical
human
computation

# Kleene (1936)



So Turing's and Church's theses are equivalent. We shall usually refer to them both as Church's thesis, or in connection with that one of its... versions which deals with "Turing machines" as the Church-Turing thesis.

# Church-Turing Thesis

A function is intuitively computable if and only if it is computable by a Turing machine, or equivalently if it is specified by a recursive function.

# Computability Thesis

A function is intuitively computable if and only if it is computable by a Turing machine, or equivalently if it is specified by a recursive function.

# Thesis I

Every effectively calculable function (effectively decidable predicate) is general recursive.

# Thesis I$^{\dagger}$

Every partial function which is effectively calculable (in the sense that there is an algorithm by which its value can be calculated for every n-tuple belonging to its range of definition) is potentially partial recursive.

# Thesis I*†

Every partial function which is effectively calculable relative to some initial functions is partial recursive relative to those functions.

# Post-Turing Thesis

A set B is effectively reducible to another set C iff B is Turing reducible to C by a Turing oracle machine ($B \leq_T C$).

# ASM Postulates

I. An algorithm determines a computational sequence.

II. Elements of the sequence can be arbitrary (first-order) structures.

III. Transitions are governed by some finite description.

# I. Sequential Time

- An algorithm is a state-transition system.
- Its transitions are partial functions.

# II. Abstract State

- States are (first-order) structures.
- All states share the same (finite) vocabulary.
- Transitions preserve the domain (base set) of states.
- States (and initial and terminal states) are closed under isomorphism.
- Transitions commute with isomorphisms.

# III. Bounded Exploration

- Transitions are determined by a fixed finite set of terms, such that states that agree on the values of these terms, also agree on all state changes.

# But...

The initial state may contain any abstract operation, including uncomputable oracles

# Martin Davis

If non-computable inputs are permitted, then non-computable outputs are attainable.

# No Magic

THE WITCH OF ENDOR

# IV. Initial State

- Shared initial operations are all recursive.

- Inputs are numbers.

- Rest is undefined.

# ASM* Postulates

I. An algorithm determines a computational sequence.

II. Elements of the sequence can be arbitrary (first-order) structures.

III. Transitions are governed by some finite description.

IV. Only basic arithmetic operations are given initially.

# Procedures

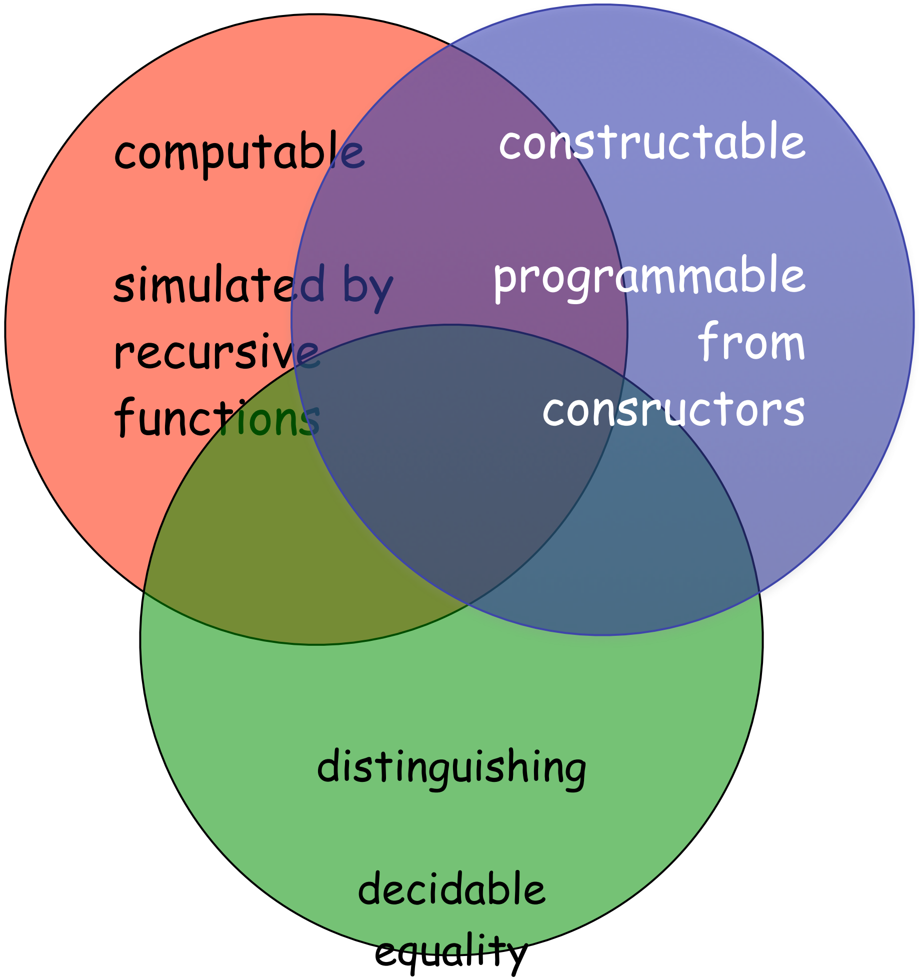
- A procedure is a sequential algorithm, in which the initial states of all runs are the same, except for inputs.
- Terminal states, if any, contain the output value(s) at some fixed location.

# Church-Turing Theorem

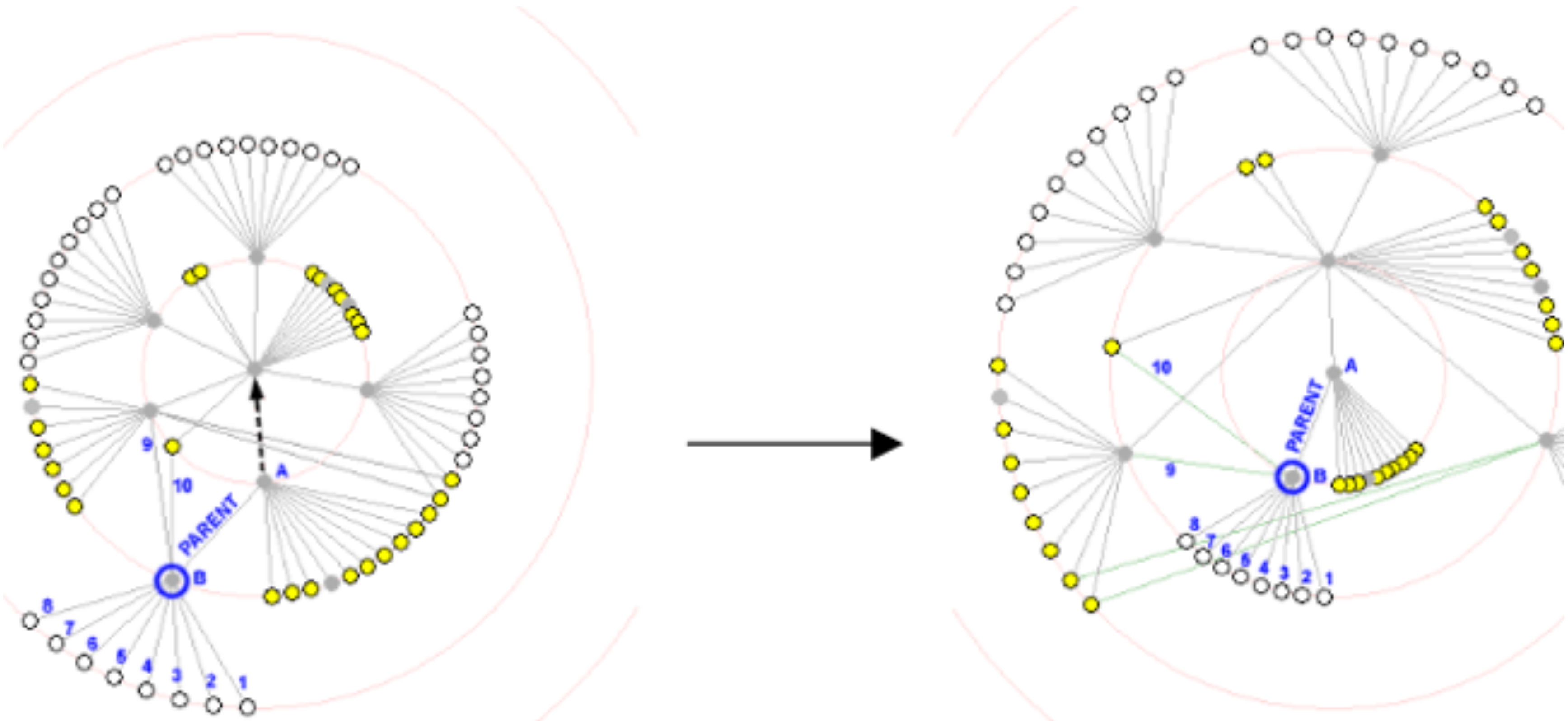
Any numeric function computed by a transition system that satisfies the above postulates is partial recursive.

# BUT

There is more to algorithms than numbers!


computable
simulated by recursive functions
constructable
programmable from consructors
distinguishing
decidable equality

# Graph Domain



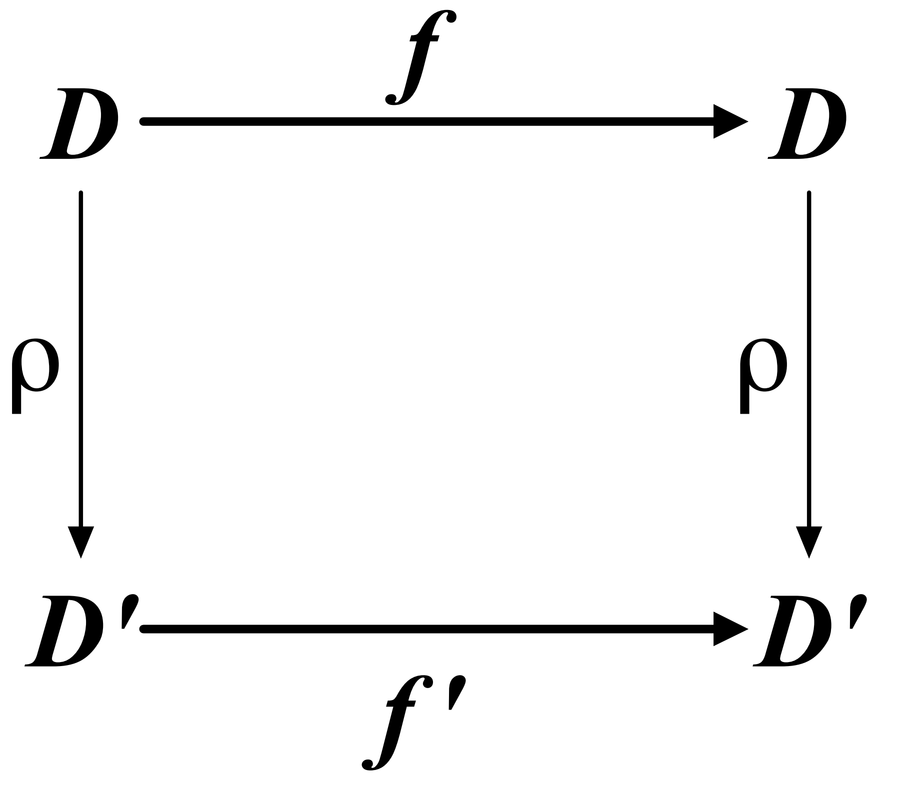
# What else is effective?

- Consider

$$f(n) = \min i.\ p(g^{n+i}(c))$$

where p,g,c work over graphs, say.

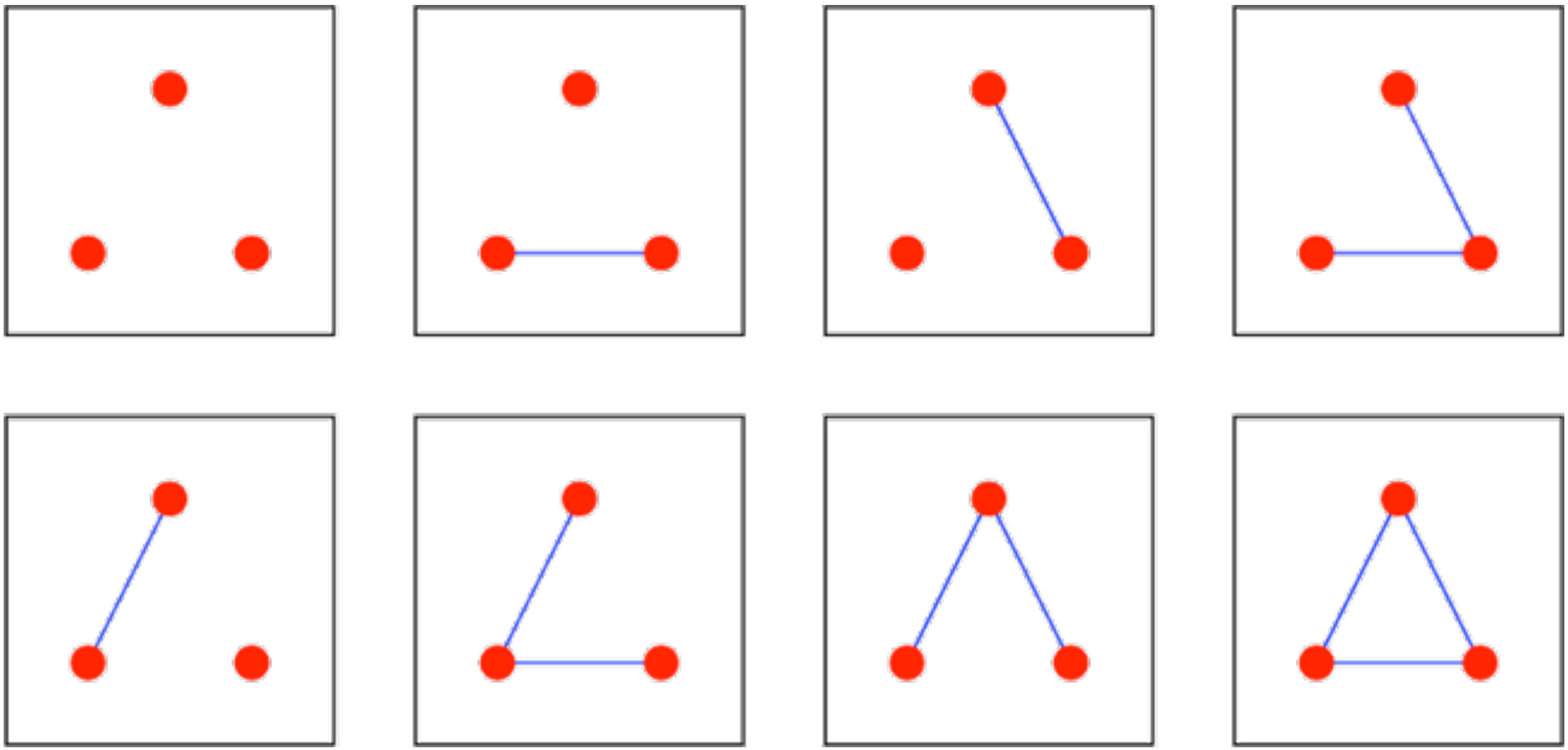
# Simulations



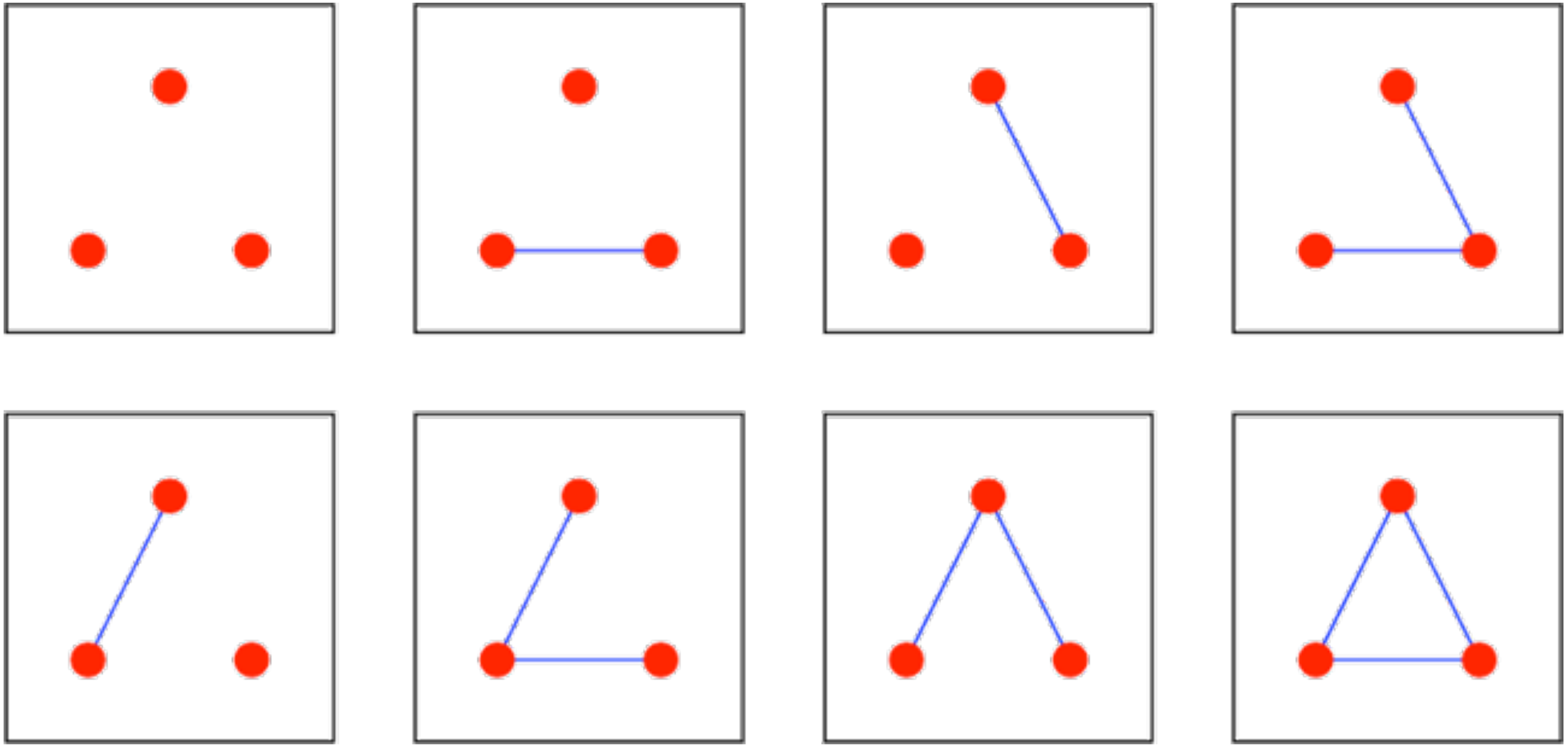

- ρ is an injection/bijection
- ρ y = ⊥ iff y = ⊥

# Computable Algebras

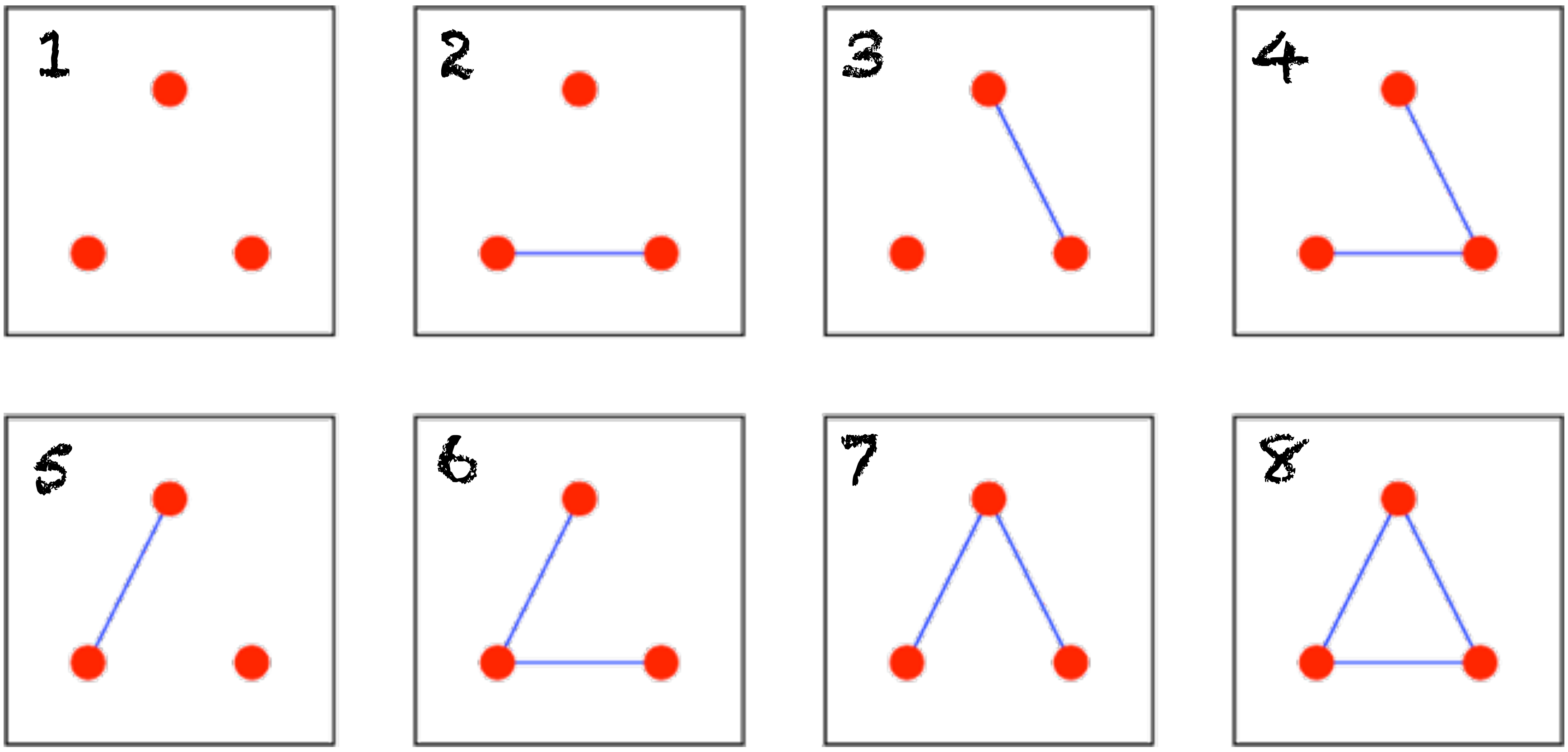
- An **algebra** *A* with domain is **computable** if it is simulated by the recursive functions.



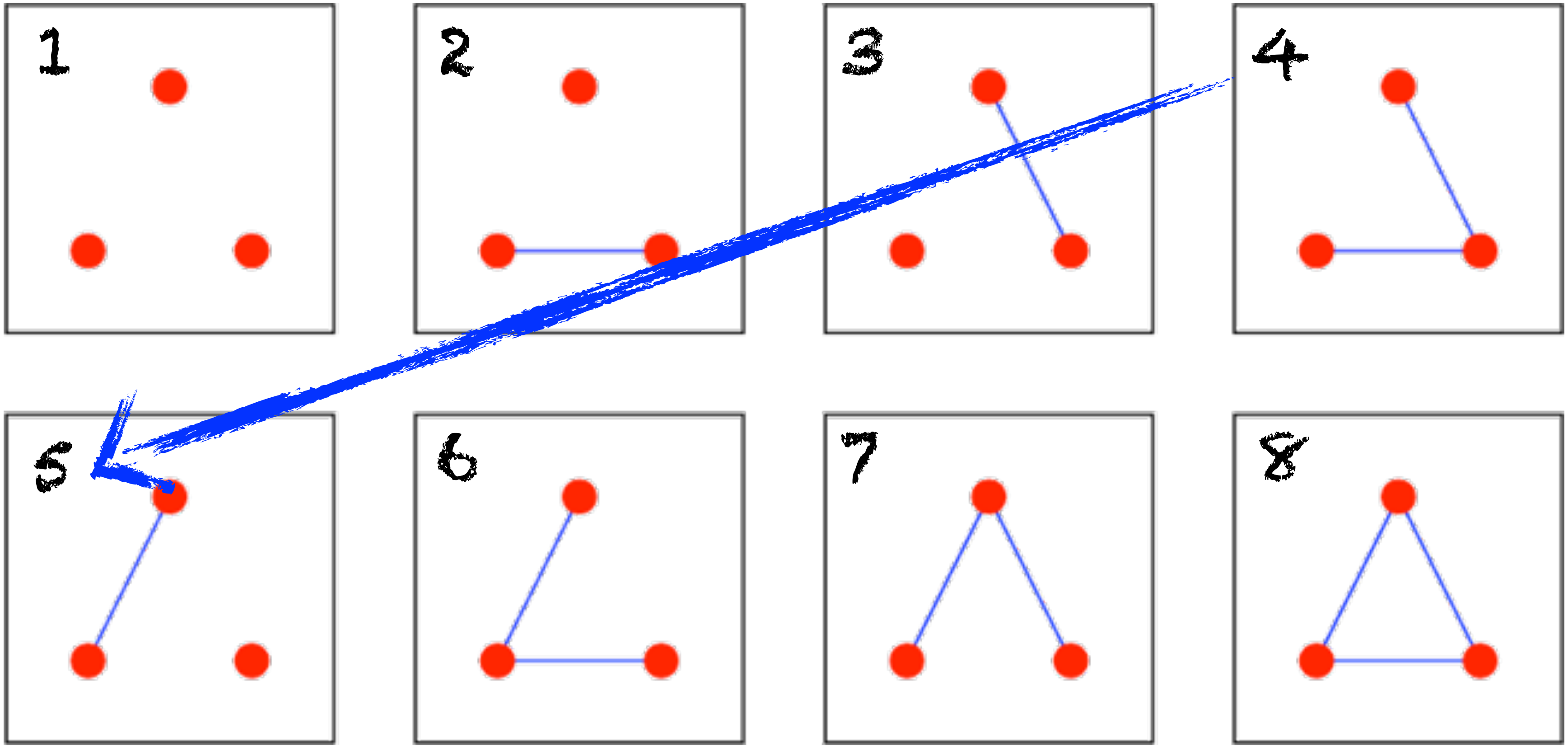
# Tracking Functions



# Tracking Functions



# Tracking Functions



# Distinguishing States

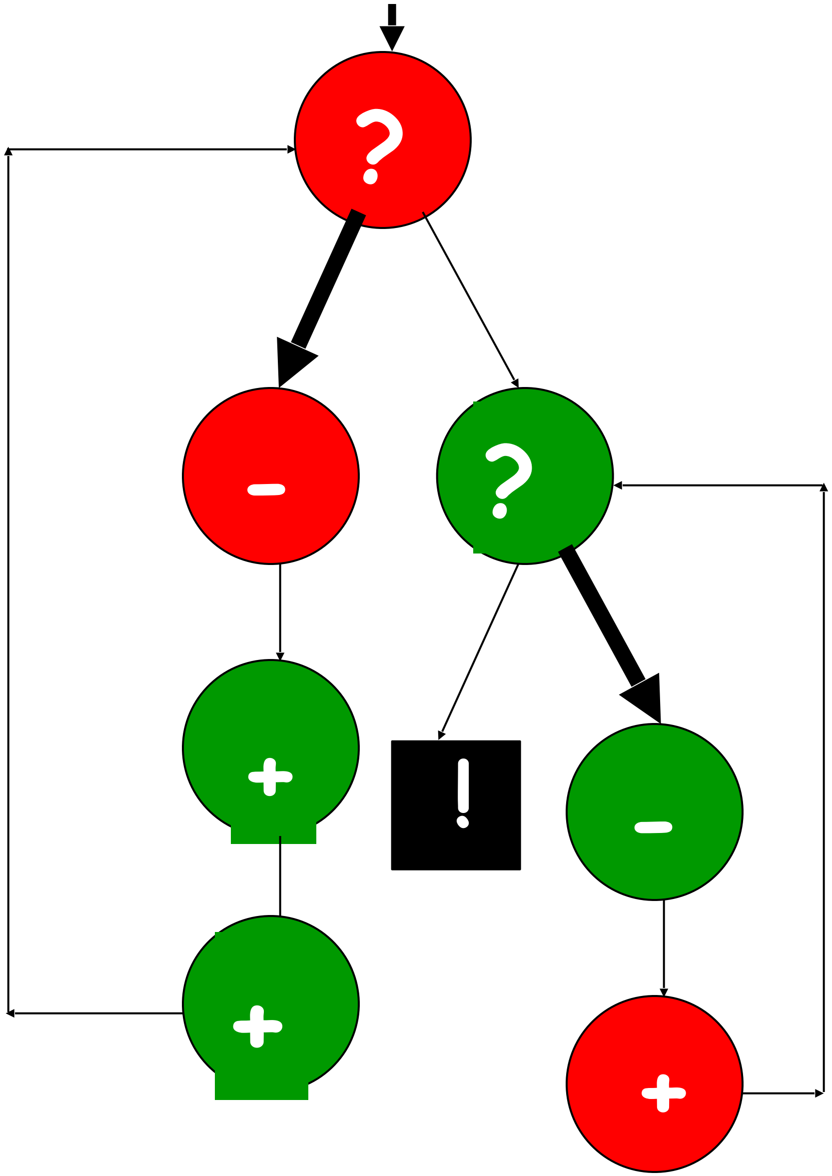
- *A* **state** (X) is **distinguishing** if its induced equivalence relation (X |= s=t) is Turing-decidable.

# ASMT $\Rightarrow$ CT

If an ASM's initial state has only basic arithmetic ($N$;0,1,+,x,=,<), then it computes a partial recursive function.

A numerical function is effective iff it is (partial) recursive.

# My favorite computer



# Church's Theorem

A numerical function is effective

iff

it is (partial) recursive.

# Church's Theorem

Let f be a function from naturals to naturals..

The following are equivalent:

- f is recursive.
- An arithmetical algorithm computes f.
- An arithmetical ASM computes f.

# Distinguishing Models

- A **model** is **distinguishing** if the equivalence induced by every finite set of initial states (across different algorithms) is decidable.

# Computable Models

- A **model** is **computable** if all its states are.

# Constructive Models

- A **model** is **constructive** if all its algorithms are, via the same constructors.

# The 3 Notions are 1

- A model of computation is computable iff it is constructive iff it is distinguishing.

# Facts

Turing Machines simulate (up to isomorphism) all effective models (under some representation).

And under no representation can an effective model do more.

Up to isomorphism, there is only one maximal effective model.

# Ian Parberry

- The Extended Church-Turing Thesis states ... that time on all "reasonable" machine models is related by a polynomial.

# Complexity

- Q. What is a single step?
  - A. Application of constructor.
- Q. How to measure an input?
  - A. Minimal number of constructor operations required to access it.

# Classical computational thesis

- Any classical effective algorithm may be simulated by a random-access machine for only constant multiplicand price (provide that RAM may manipulate word of logarithmic size in one time unit)