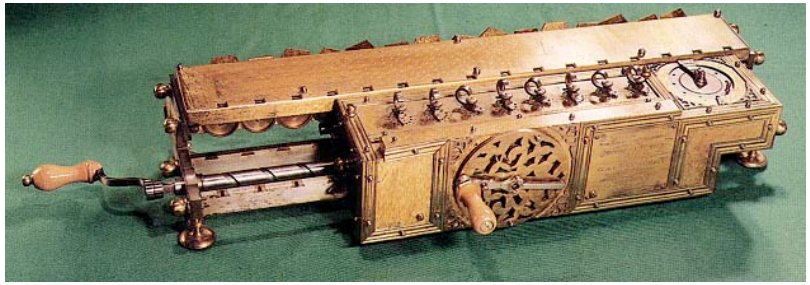
Euclid (c. -300)

Euclid's GCD algorithm appeared in his *Elements*.

Formulated geometrically: Find common measure for 2 lines.

Used repeated subtraction of the shorter segment from the longer.

Gottfried Wilhelm Leibniz (1666)



The only way to rectify our reasonings is to make them as tangible as those of the Mathematician, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: **Let us calculate**, without further ado, in order to see who is right.

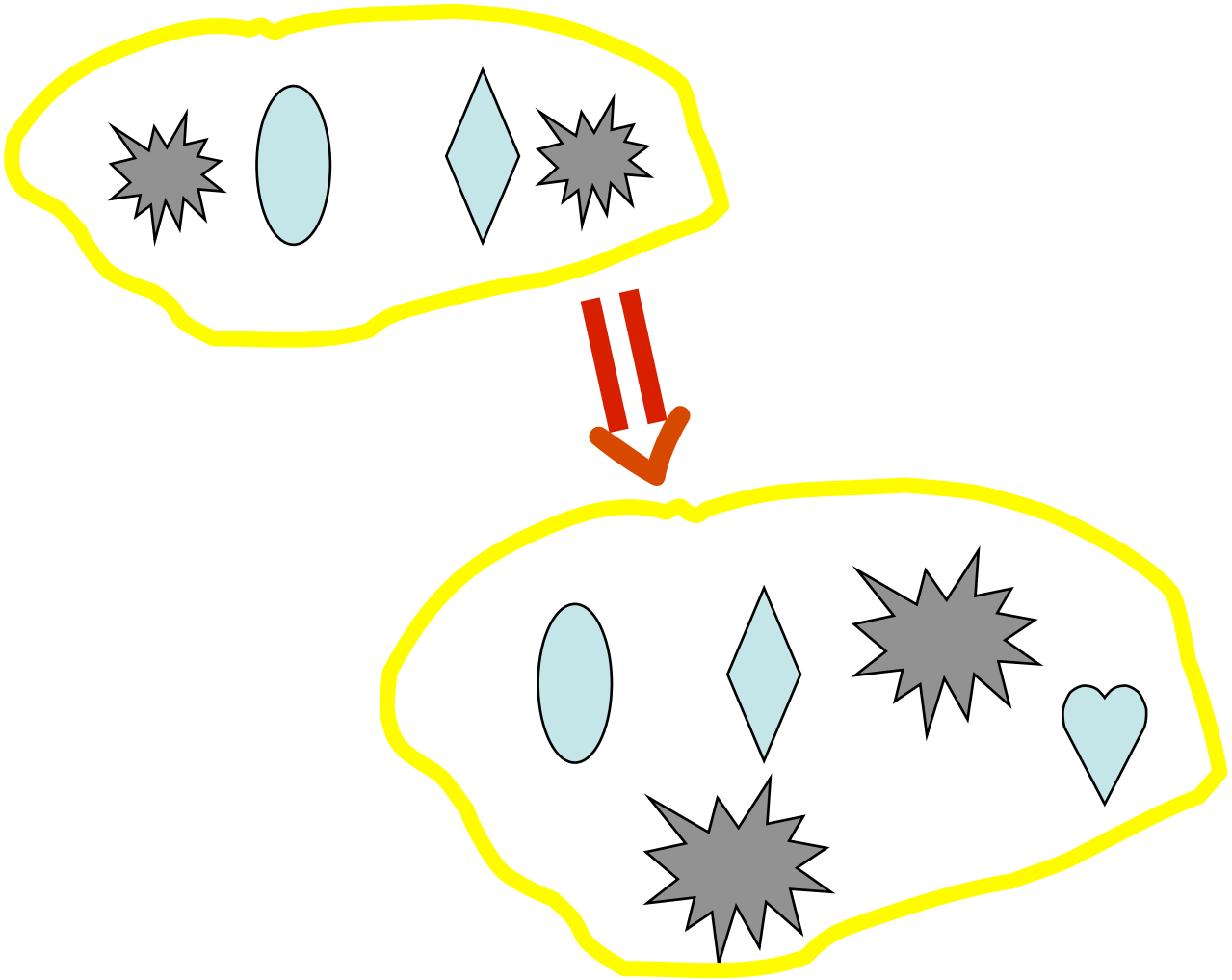
David Hilbert (c. 1900)

#2. Provide an effective method to determine if a formula is valid.

Hilbert & Ackermann (1928)

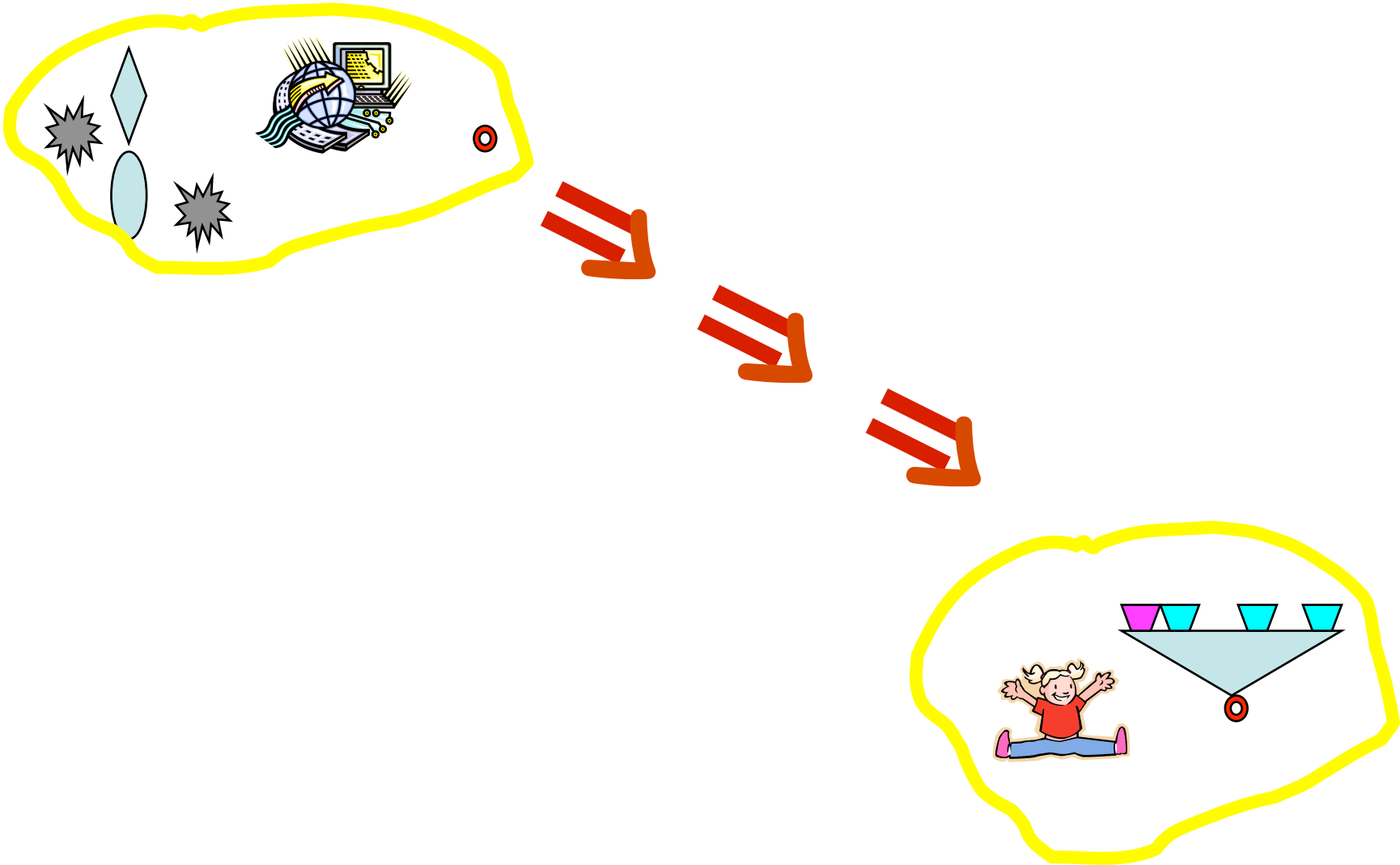
- The Entscheidungsproblem is solved when we know a procedure that allows for any given logical expression to decide by finitely many operations its validity or satisfiability.
- The Entscheidungsproblem must be considered the main problem of mathematical logic.
- The solution of the Entscheidungsproblem is of fundamental significance for the theory of all domains whose propositions could be developed on the basis of a finite number of axioms.

Expansion

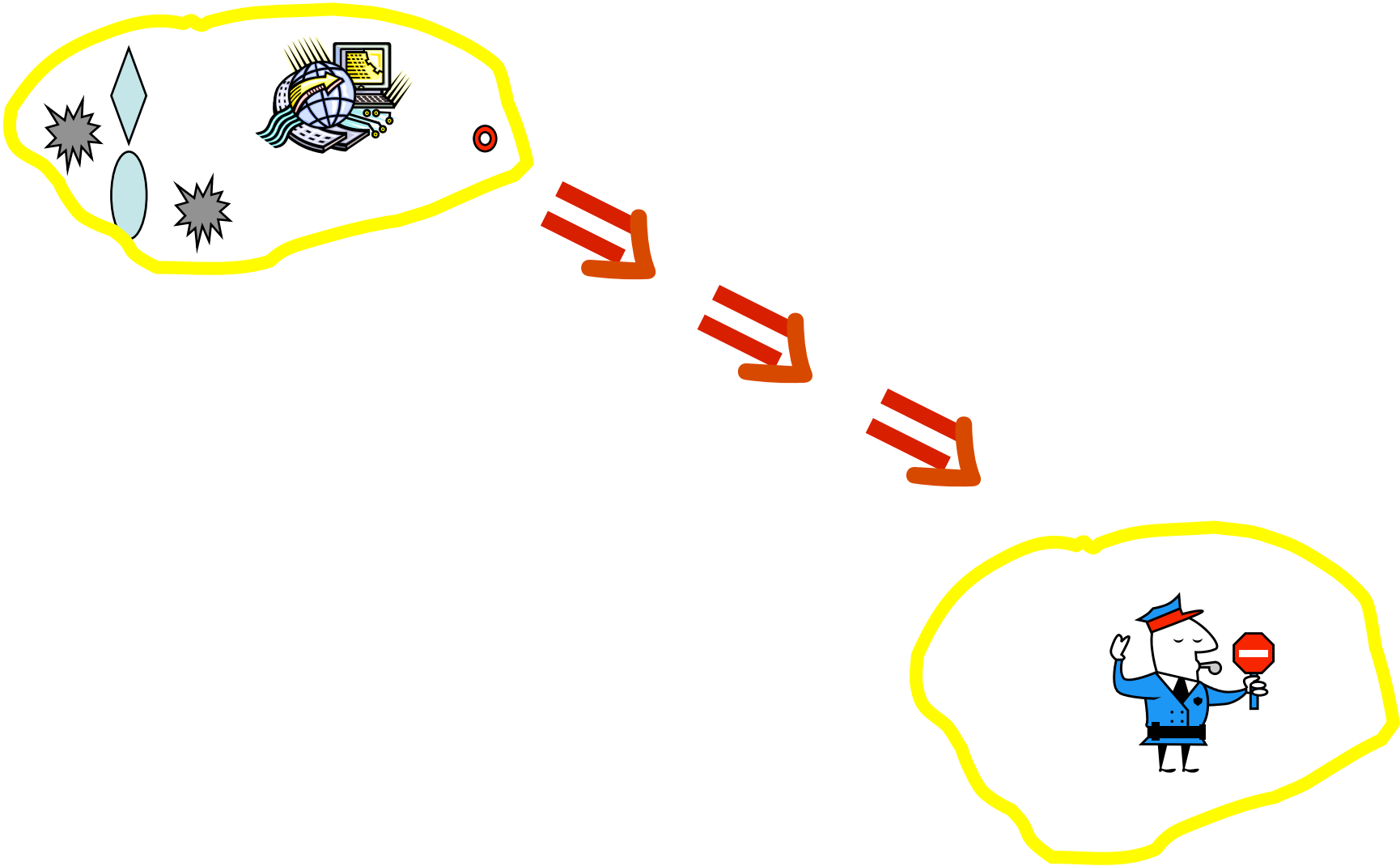


State Transition

Computation



Computation



Jules Henri Poincaré

We might imagine a machine where we should put in axioms at one end and take out theorems at the other, like that legendary machine in Chicago where pigs go in alive and come out transformed into hams and sausages.

What is an algorithm?

Looking for a canonical notion of algorithm.

Looking for a canonical notion of an effective algorithm.

Looking for a canonical notion of an effective numeric algorithm.

What is an algorithm?

Looking for a canonical notion of algorithm.

Looking for a canonical notion of an effective algorithm.

Looking for a canonical notion of an effective numeric algorithm.

Don Knuth (1966)

Algorithms are concepts which have existence apart from any programming language… I believe algorithms were present long before Turing et al. formulated them, just as the concept of the number "two" was in existence long before the writers of first grade textbooks and other mathematical logicians gave it a certain precise definition.

Church's Thesis

Church's Thesis (1936)

Recursive functions "capture" effective computability.

Richard Montague (1960)

Discussion of Church's thesis has suffered for lack of a precise general framework within which it could be conducted.

Recursive Functions

- Partial functions on natural numbers.
- Initial functions 0 and successor.
- Other functions defined via equations:

$x + 0 = x$

$x + y' = (x + y)'$

$0 - y = 0$

$x' - y = (x - y)'$

[Hilbert, Ackermann, Peter, Herbrand, Gödel]

Kurt Gödel (1930s)

$$f(n) = \begin{cases} 1 & n=0 \\ n \times f(n-1) & \text{otherwise} \end{cases}$$

Alonzo Church

[We] propose a definition of effective calculability which is thought to correspond satisfactorily to [a] somewhat vague intuitive notion....

We now define ... the notion of *effective calculable* function of the positive integers by identifying it with the notion of a recursive function.

Why Believe?

- Experience: programming languages compute only recursive functions.
  - But history is full of delayed discoveries.
- Model equivalence: Church, Turing, Post, Markov, Kolmogorov.
  - But what excludes a systematic error?
- Turing's analysis. This is by far the strongest argument that convinced even skeptical Gödel.

Alan Turing (1936)

A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine.

Turing's Thesis

Turing Machines
capture
mechanical
human
computation.

Alan Turing (1936)

A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine.

The theorem that all effectively calculable sequences are computable and its converse are proved....

Turing (1948)

Logical Computing Machines can do anything that could be described as "rule of thumb" or "purely mechanical". This is sufficiently well established that it is now agreed amongst logicians that "calculable by means of an LCM" is the correct rendering of such phrases.

Turing's Axioms

- Sequential symbol manipulation
- Finite internal states
- Linear external memory
- Finite symbol space
- Finite observability and local action
- Deterministic

Finite Internal States

- If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused

Linear External Memory

- The two-dimensional character of paper is no essential of computation

Finite Alphabet

- If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrary small extent

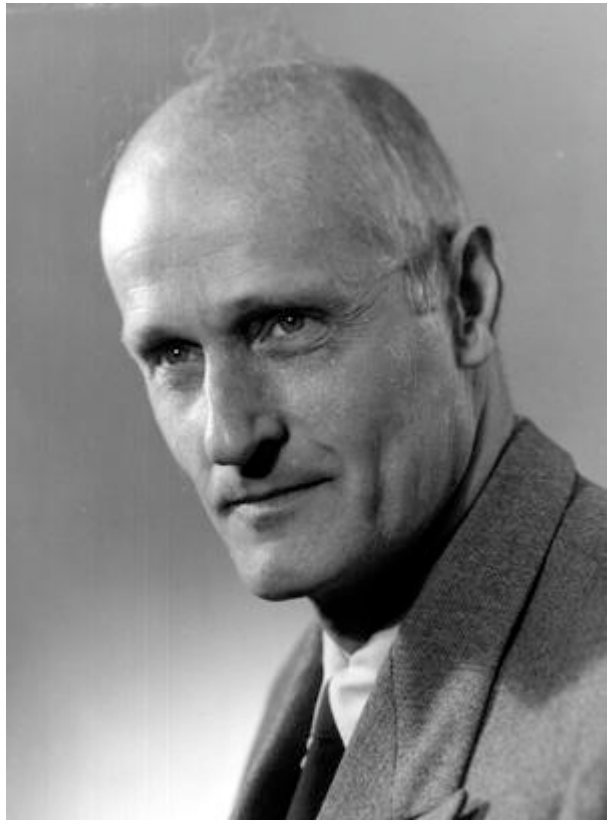
Finiteness and Localness

- [Operations] so elementary that it is not easy to imagine them further divided
  - Can only observe a finite number of symbols at each step

Deterministic

- The behavior at any moment is determined by the symbols which he is observing and his "state of mind" at the moment

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.*

The Turing Tarpit

- Thue systems
  - Post systems
- Lambda calculi
- Partial recursion
- Turing machines
- Markov normal algorithms
- Minsky counter machines
- Type 0 languages
- Kolmogorov-Uspenskii machines
- Neuring machines
- Wang machines
- Random access machines
- Quantum computers
- Billiard ball computers
- Least fixpoints
- Fortran, Algol, Lisp, C, Pascal, Logo, Ada, Java, ...

Bernard Moret

The remarkable result about these varied models is that all of them define exactly the same class of computable functions: whatever one model can compute, all the others can too!

Turing (1948)

Logical Computing Machines can do anything that could be described as "rule of thumb" or "purely mechanical". This is sufficiently well established that it is now agreed amongst logicians that "calculable by means of an LCM" is the correct rendering of such phrases.

Godel

- It was only by Turing's work that it became completely clear, that my proof is applicable to *every* formal system containing arithmetic.

- I think the reader has a right to be informed about the present state of affairs.

Church to Kleene (1935)

[Gödel thought] that it might be possible ... to state a set of axioms which would embody the generally accepted properties of [effective calculability], and to do something on that basis.

Laszlo Kalmár (1959)

There are pre-mathematical concepts which must remain [so]....
Among these belong ...
such concepts as that of effective calculability ...
the extension of which cannot cease to change during the development of mathematics.

Joe Shoenfield (1991)

It may seem that it is impossible to give a proof of Church's Thesis. However, this is not necessarily the case. We can write down some axioms about computable functions which most people would agree are evidently true. It might be possible to prove Church's Thesis from such axioms.

However, despite strenuous efforts, no one has succeeded in doing this (although some in teresting partial results have been obtained).

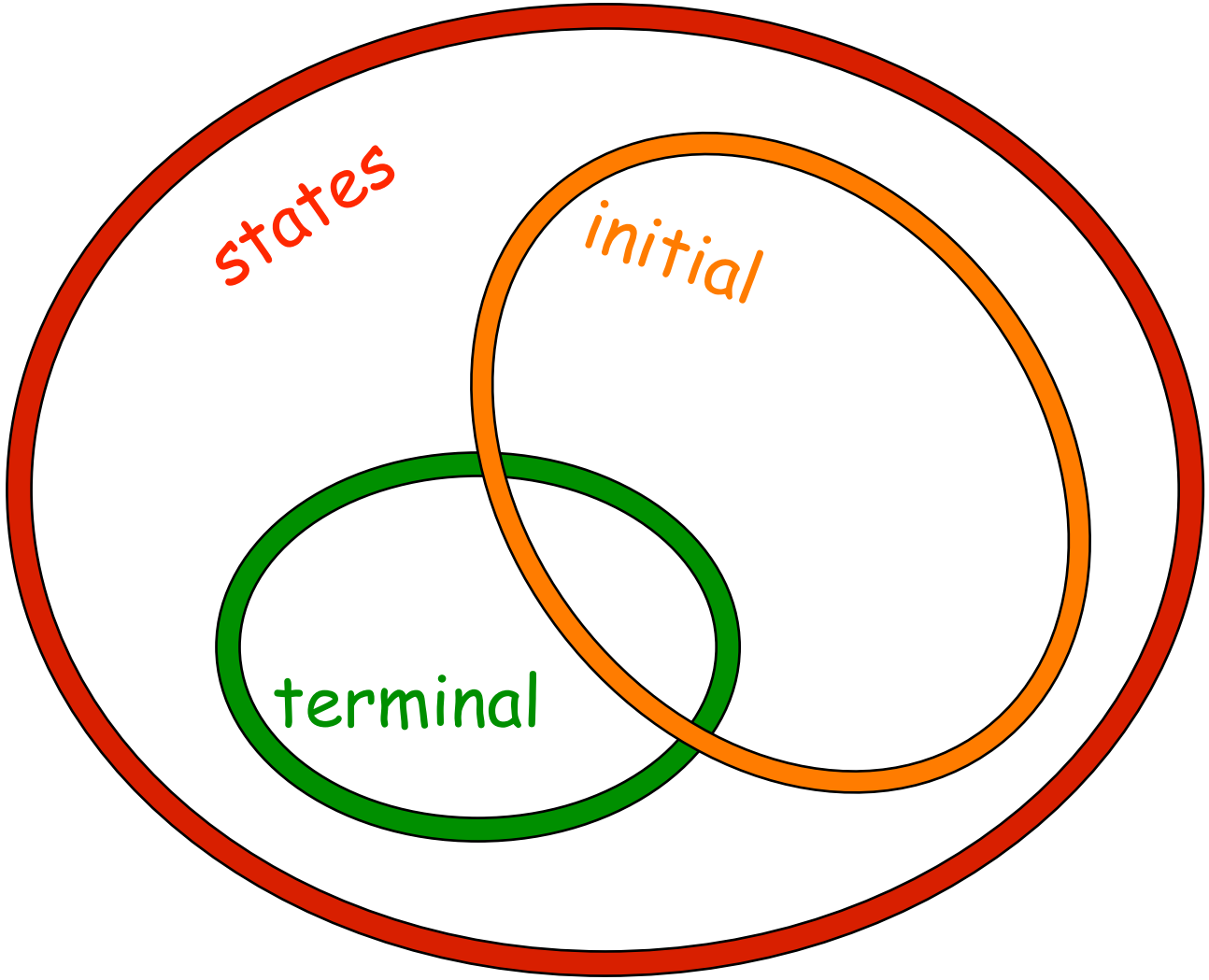
Axiomatics

Donald Knuth (1968)

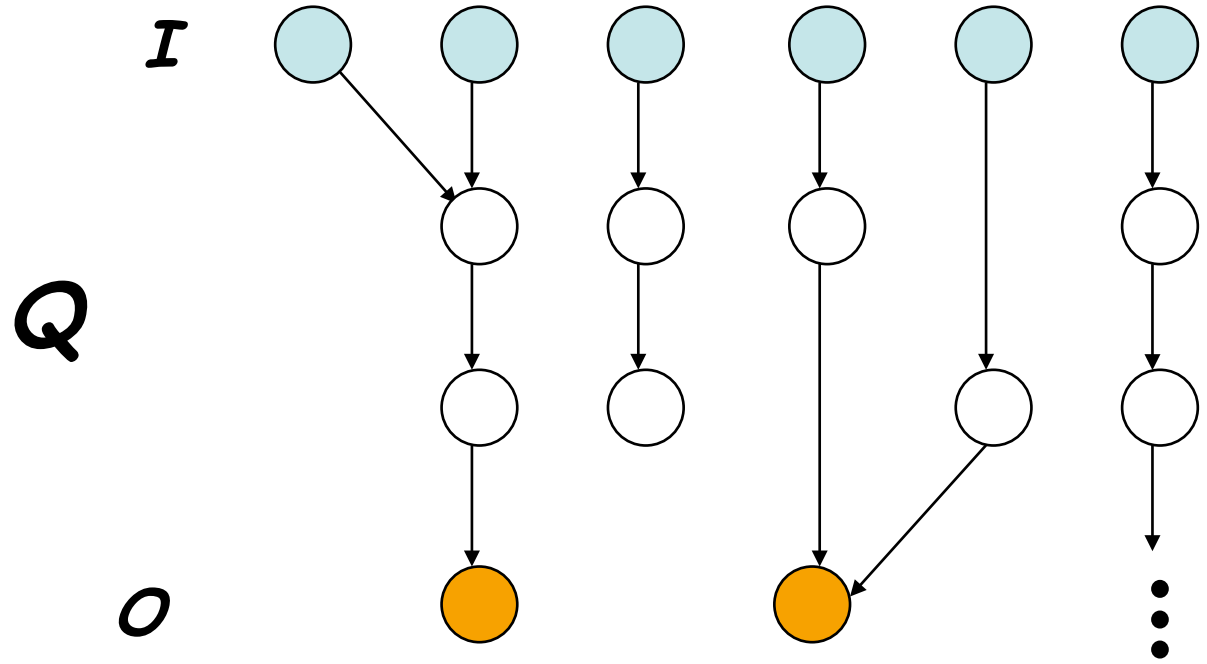
A *computational method* consists of:

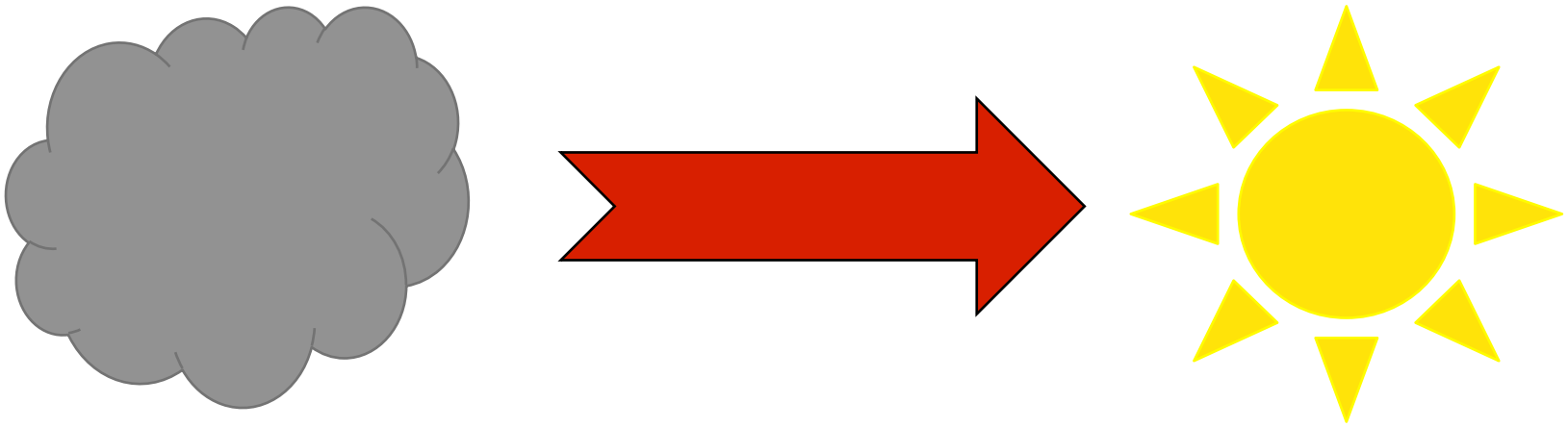
1. States Q
2. Input $I \subseteq Q$
3. Output $O \subseteq Q$
4. Transitions $\tau : Q \to Q$

In this way we can divorce abstract algorithms from particular programs that represent them.


states
initial
terminal

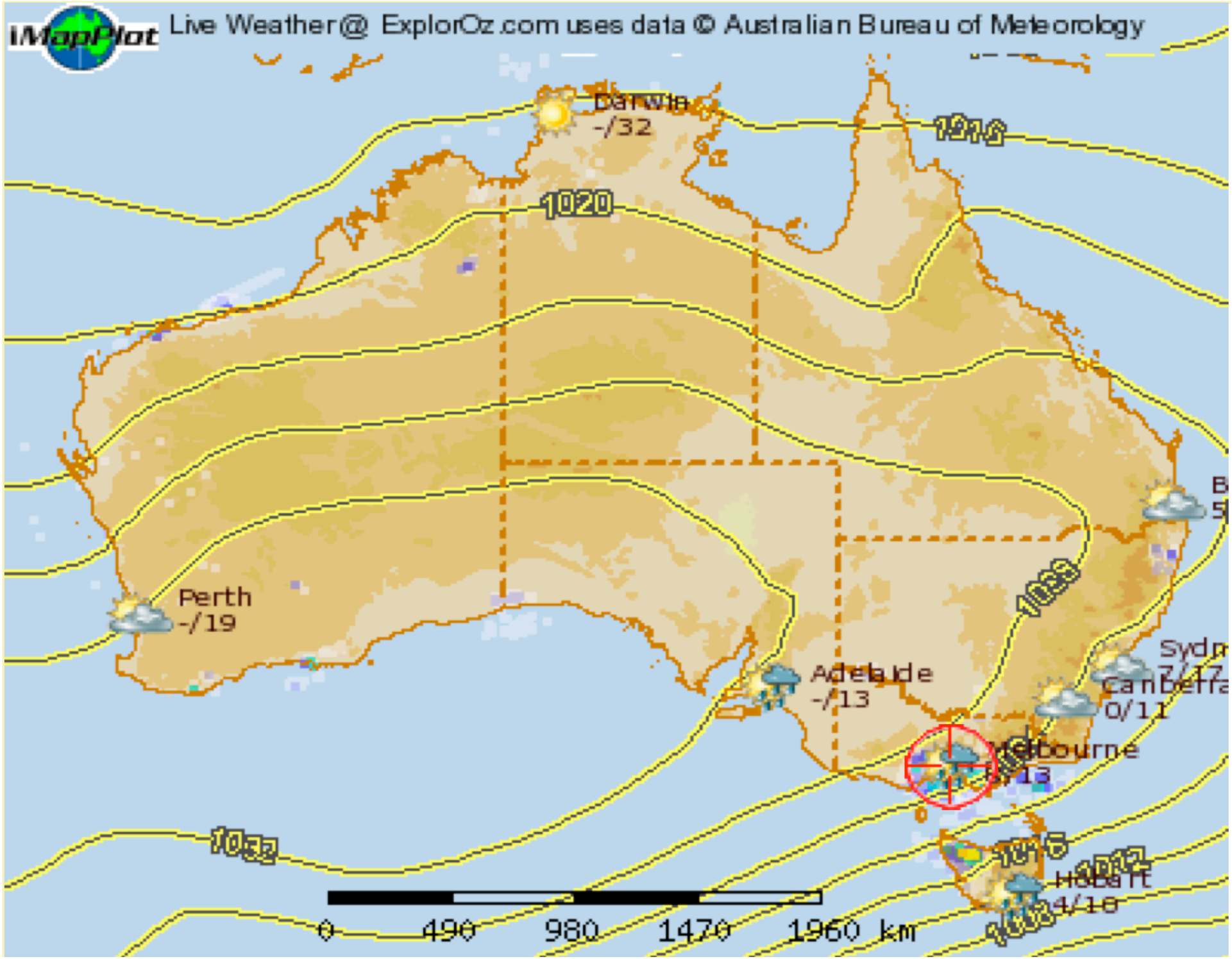
Algorithm = Transition System

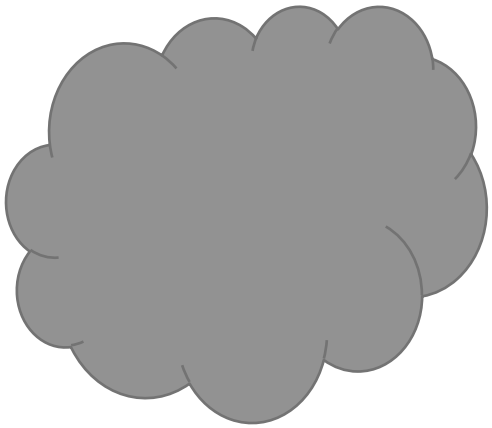




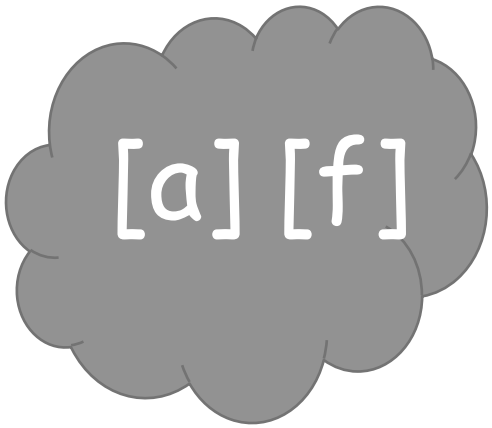
I. Sequential Time

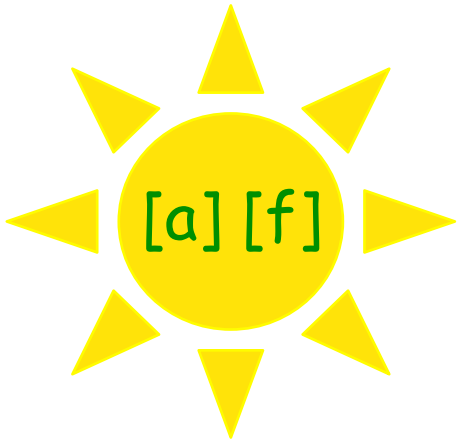
- An algorithm is a state-transition system.
- Its transitions are partial functions.

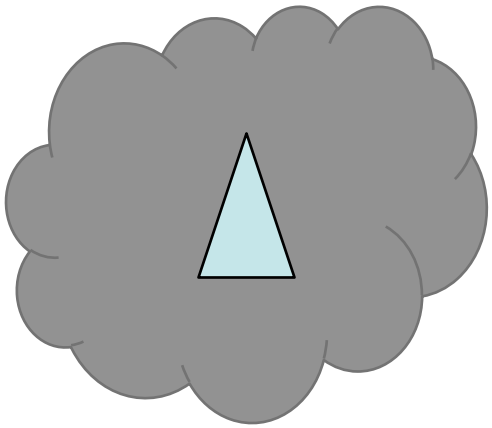

iMapPlot
Live Weather@ ExplorOz.com uses data © Australian Bureau of Meteorology
Darwin
-/32
1016
1020
Perth
-/19
Adelaide
-/13
1028
Canberra
0/11
Melbourne
1032
Hobart
4/10
0
490
980
1470
1960 km

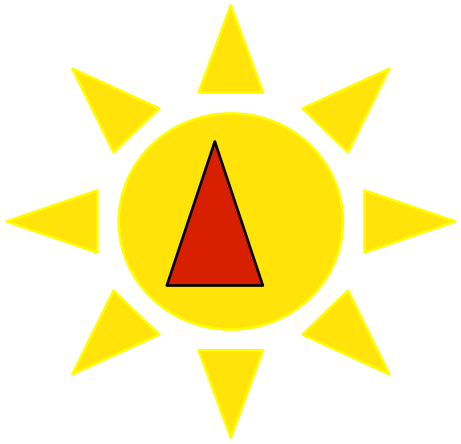


State encapsulates all relevant data!


[a] [f]

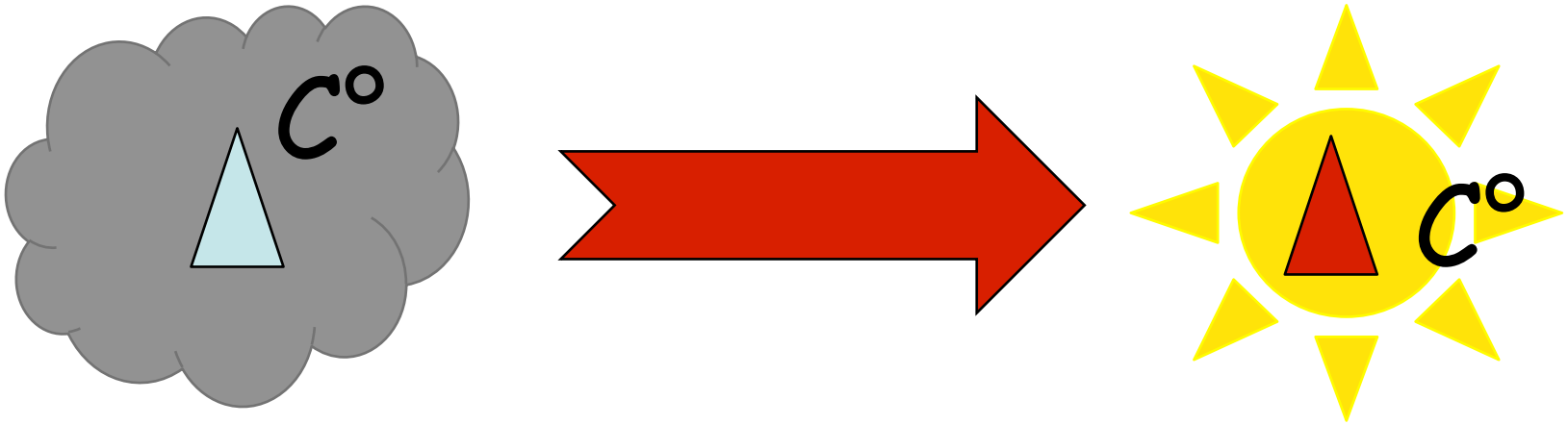

[a] [f]





Kleene

An algorithm in our sense must be fully and finitely described before any particular question to which it is applied is selected. When the question has been selected, all steps must then be predetermined and performable without any exercise of ingenuity or mathematical invention by the person doing the computing.


C°
C°

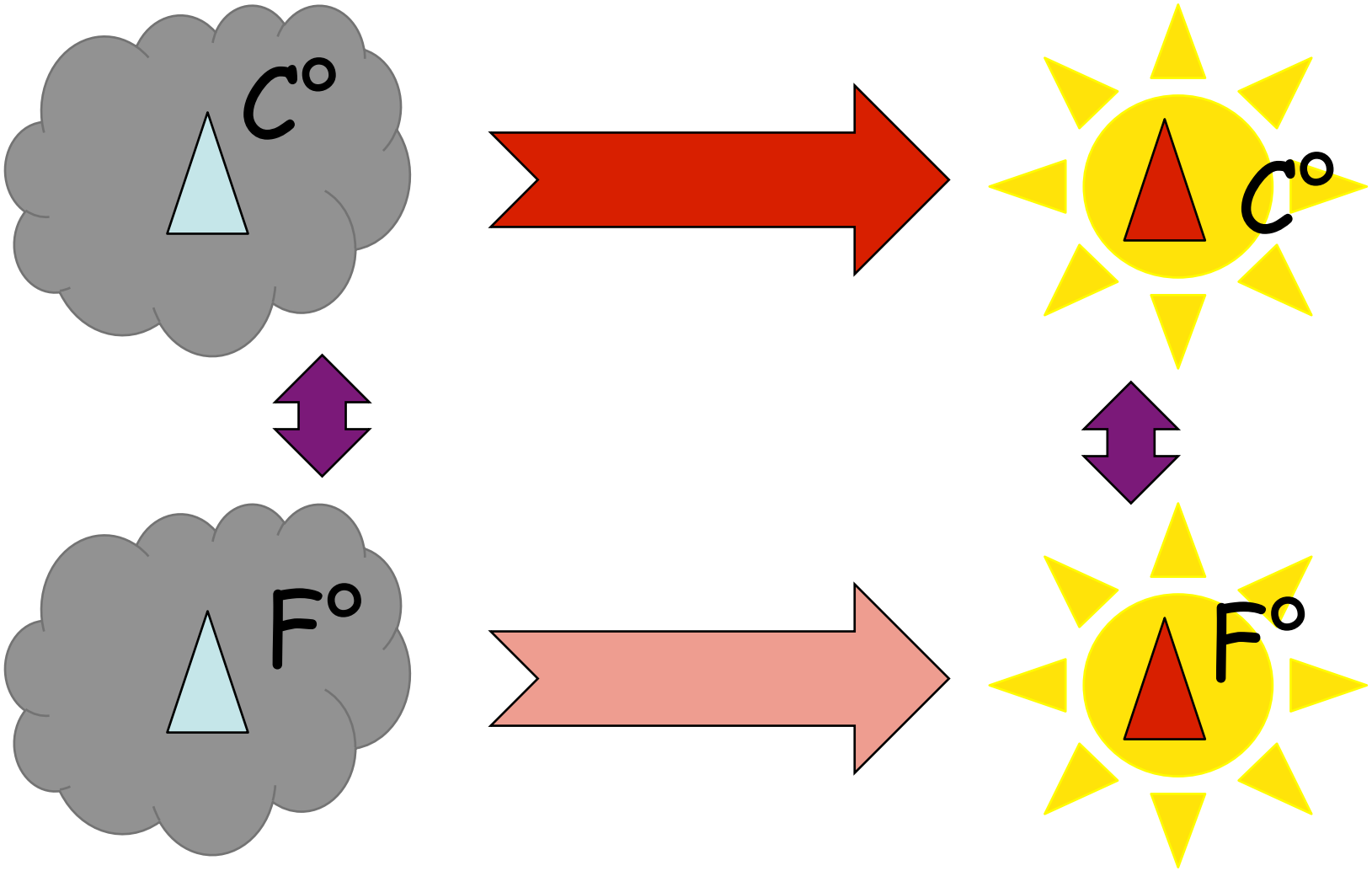
II. Abstract State

- States are (first-order) structures.
- All states share the same (finite) vocabulary.
- Transitions preserve the domain (base set) of states.

Joe Shoenfield

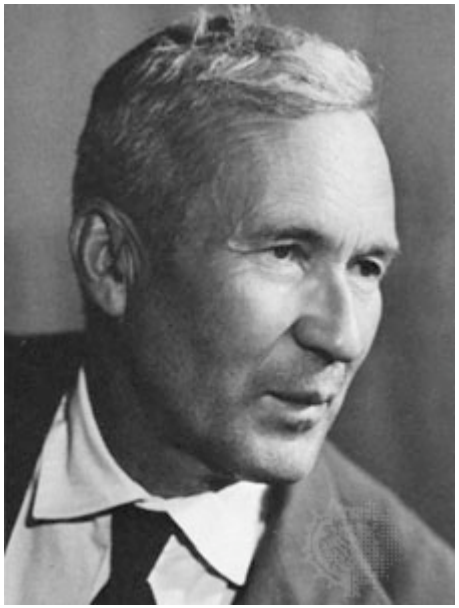
A method must be *mechanical*…

Methods which involve chance procedures are excluded;… methods which involve magic are excluded;… methods which require insight are excluded.


C°
C°
F°
F°

II. Abstract State

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



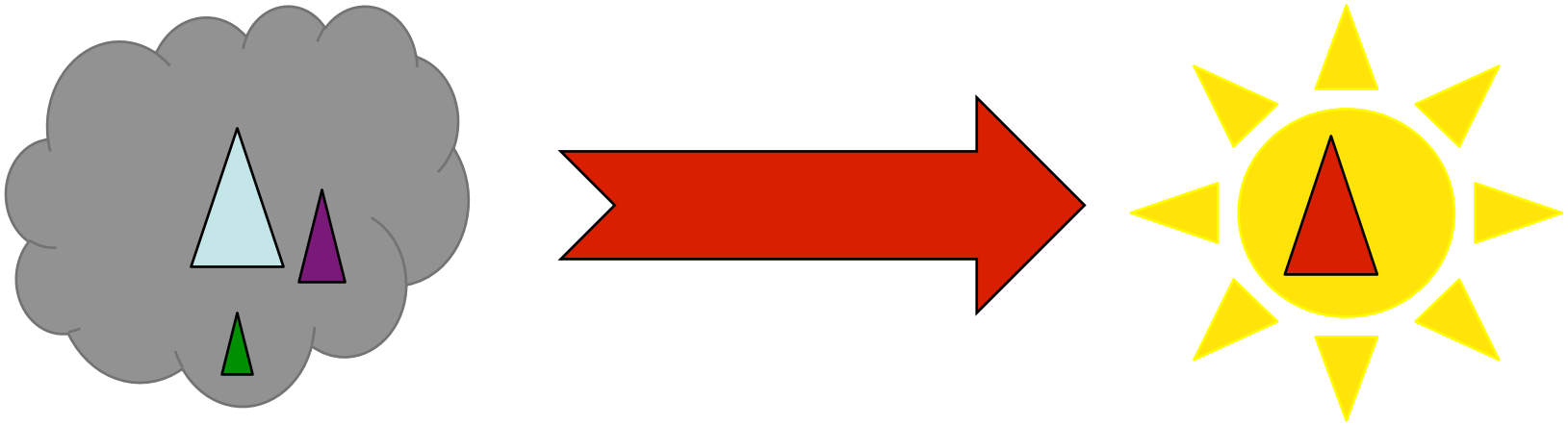
Kolmogorov

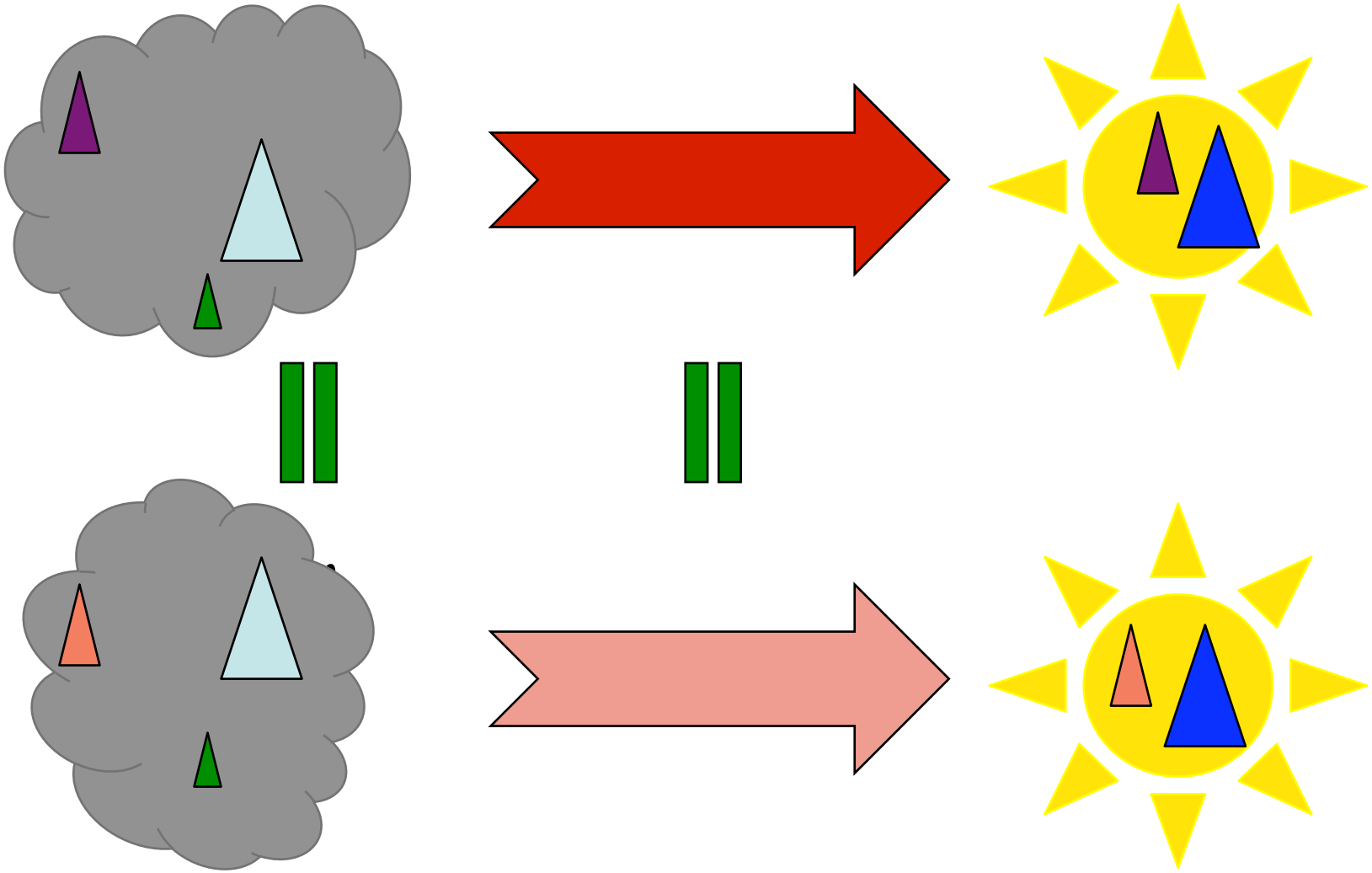
The mathematical notion of Algorithm has to preserve two properties:

1. The computational operations are carried out in discrete steps, where every step only uses a bounded part of the results of all preceding operations.
2. The unboundedness of memory is only quantitative: i.e., we allow an unbounded number of elements to be accumulated, but they are drawn from a finite set of types, and the relations that connect them have limited complexity…

Kolmogorov

It seems plausible to us that an arbitrary algorithmic process satisfies our definition of algorithms. We would like to emphasize that we are talking not about a reduction of an arbitrary algorithm to an algorithm in the sense of our definition, but that every algorithm essentially satisfies the proposed definition.





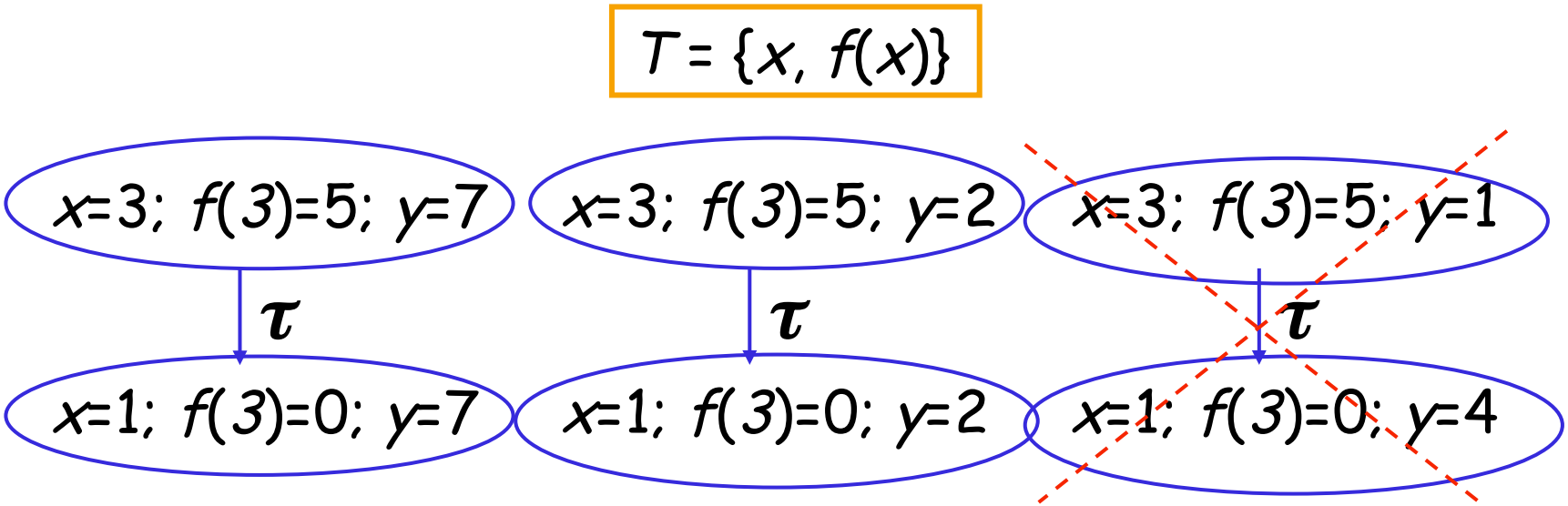
III. Bounded Exploration

- Transitions are determined by a fixed finite "glossary" of "critical" terms.

- That is, there exists some finite set of (ground) terms over the vocabulary of the states, such that states that agree on the values of these glossary terms, also agree on all state changes.

Bounded Exploration

- Whenever X, Y coincide on T, $\tau(x)-x = \tau(y)-y$



ASM Theorem

Abstract state machines emulate any program satisfying these postulates.

Ada Lovelace

Many persons who are not conversant with mathematical studies imagine that because the business of [the Analytical Engine] is to give its results in numerical notation, the nature of its processes must consequently be arithmetical and numerical rather than algebraical and analytical. This is an error. The engine can arrange and combine its numerical quantities exactly as if they were letters or any other general symbols; and in fact it might bring out its results in algebraical notation were provisions made accordingly.

The Analytical Engine does not occupy common ground with mere "calculating machines." …
In enabling mechanism to combine together *general* symbols in successions of unlimited variety and extent, a uniting link is established between the operations of matter and the abstract mental processes of the *most abstract* branch of mathematical science.

A.A.L.