P systems with hybrid sets

$$w : \Sigma \to \mathbb{N}\ \mathbb{Z}$$

Omar Belingheri, Antonio E. Porreca, Claudio Zandron
Università degli Studi di Milano-Bicocca, Italy

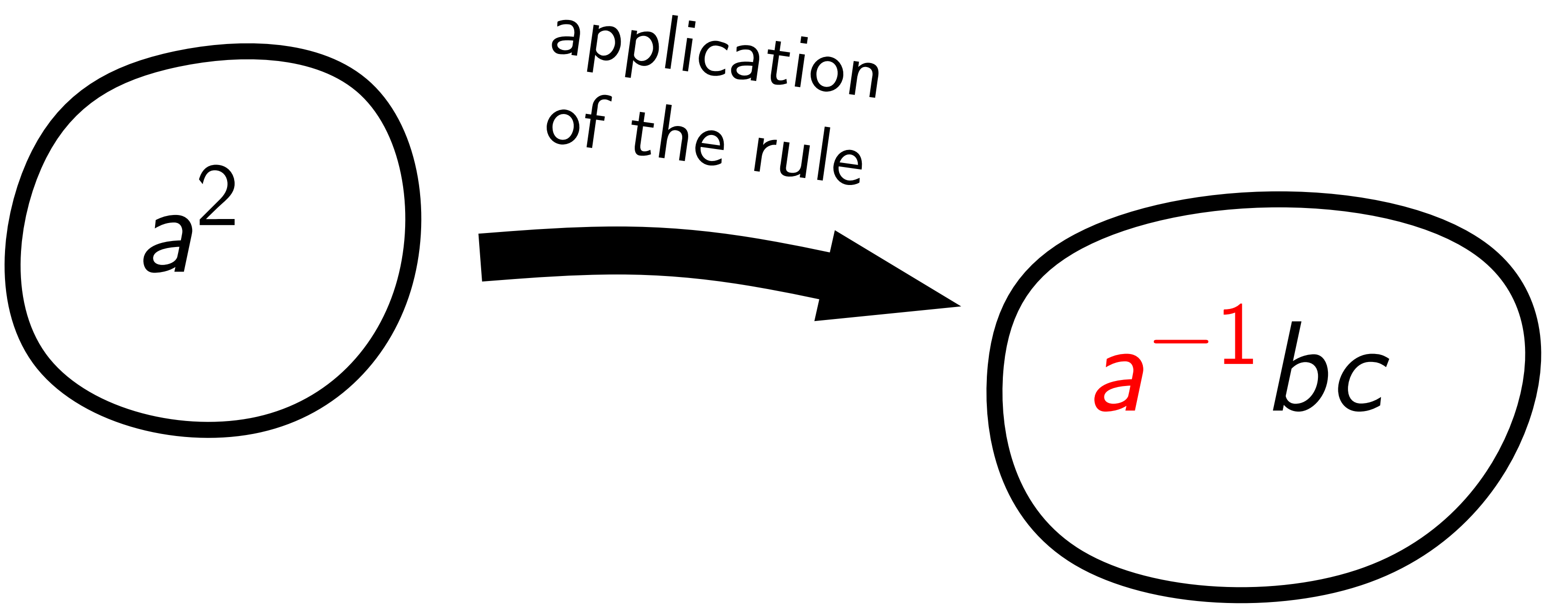
Inspiration & other works

Gheorghe Păun, Some quick research topics, *Proceedings of the Thirteenth Brainstorming Week on Membrane Computing*

Rudi Freund, Sergiu Ivanov, Sergey Verlan, P systems with generalized multisets over totally ordered abelian groups, *Proceedings of the 16th International Conference on Membrane Computing (CMC16)*

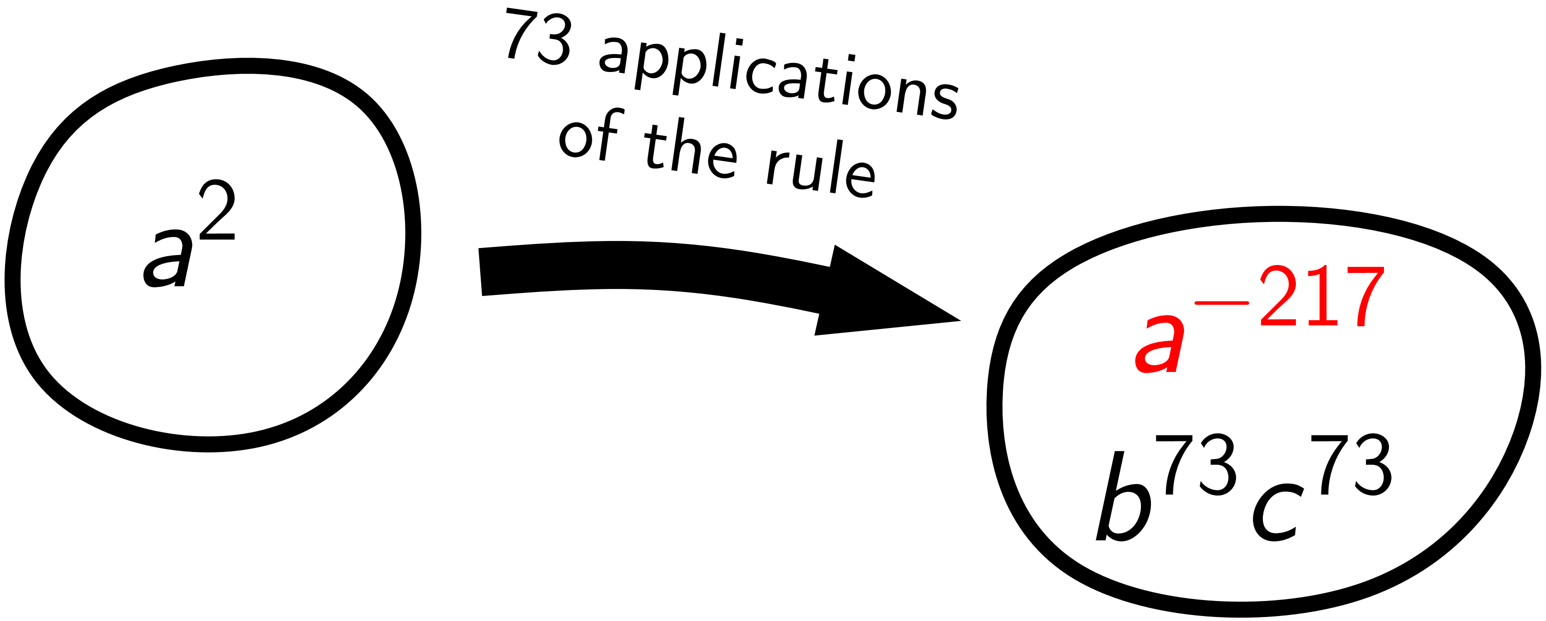
Objects with negative multiplicity

$$a^3 \rightarrow bc$$



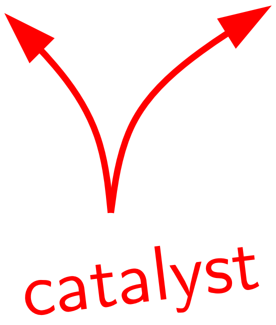
When do we stop?

$$a^3 \to bc$$



Proposal: have (moving) catalyst objects

$$u\,{\color{red}k} \to v\,{\color{red}k}$$



Catalysts obey mass conservation
and cannot have negative multiplicity

General rule form

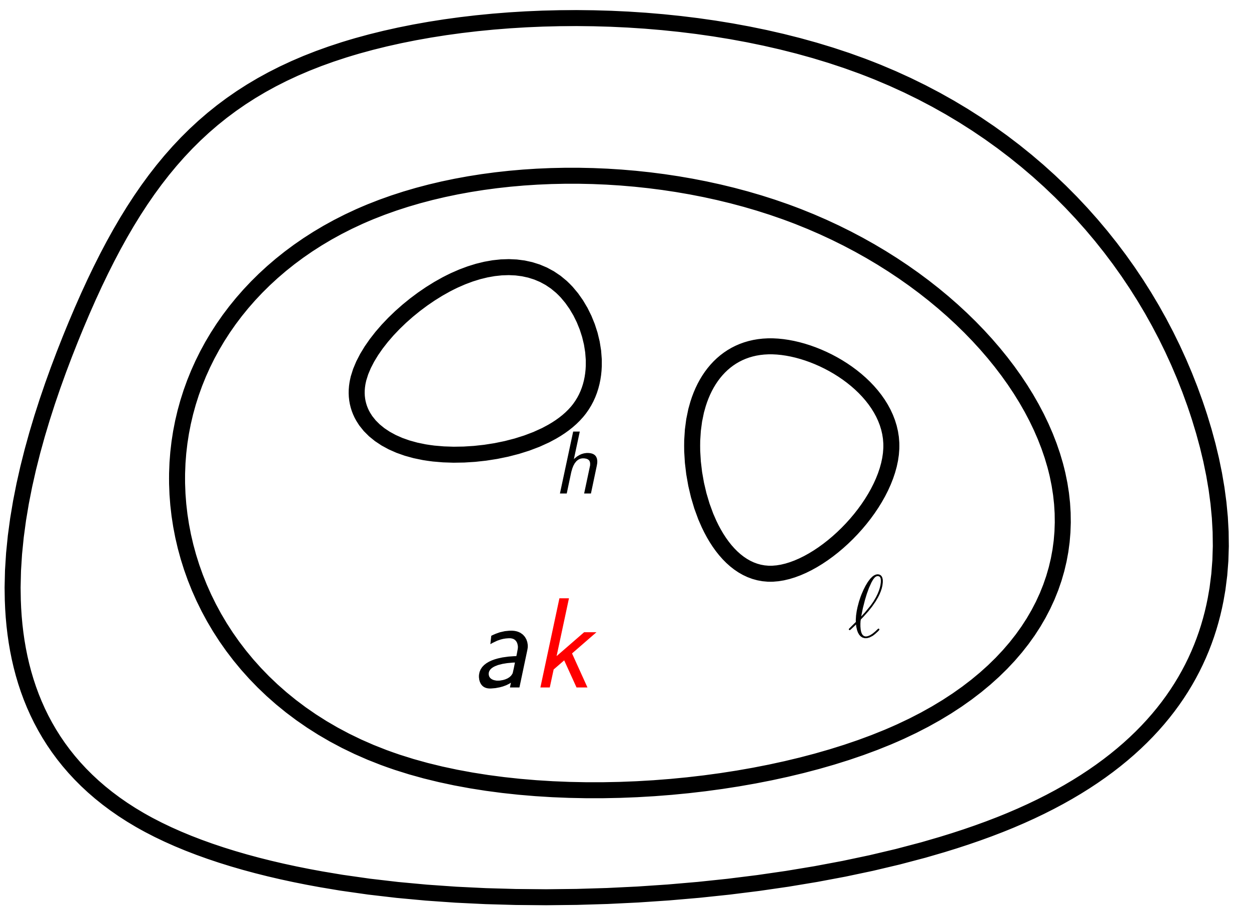
optional dissolution

$$a \cdots z\, k \rightarrow a'_{t_1} \cdots z'_{t_n}\, k_t\, \delta$$

targets
(in, out, label, . . .)

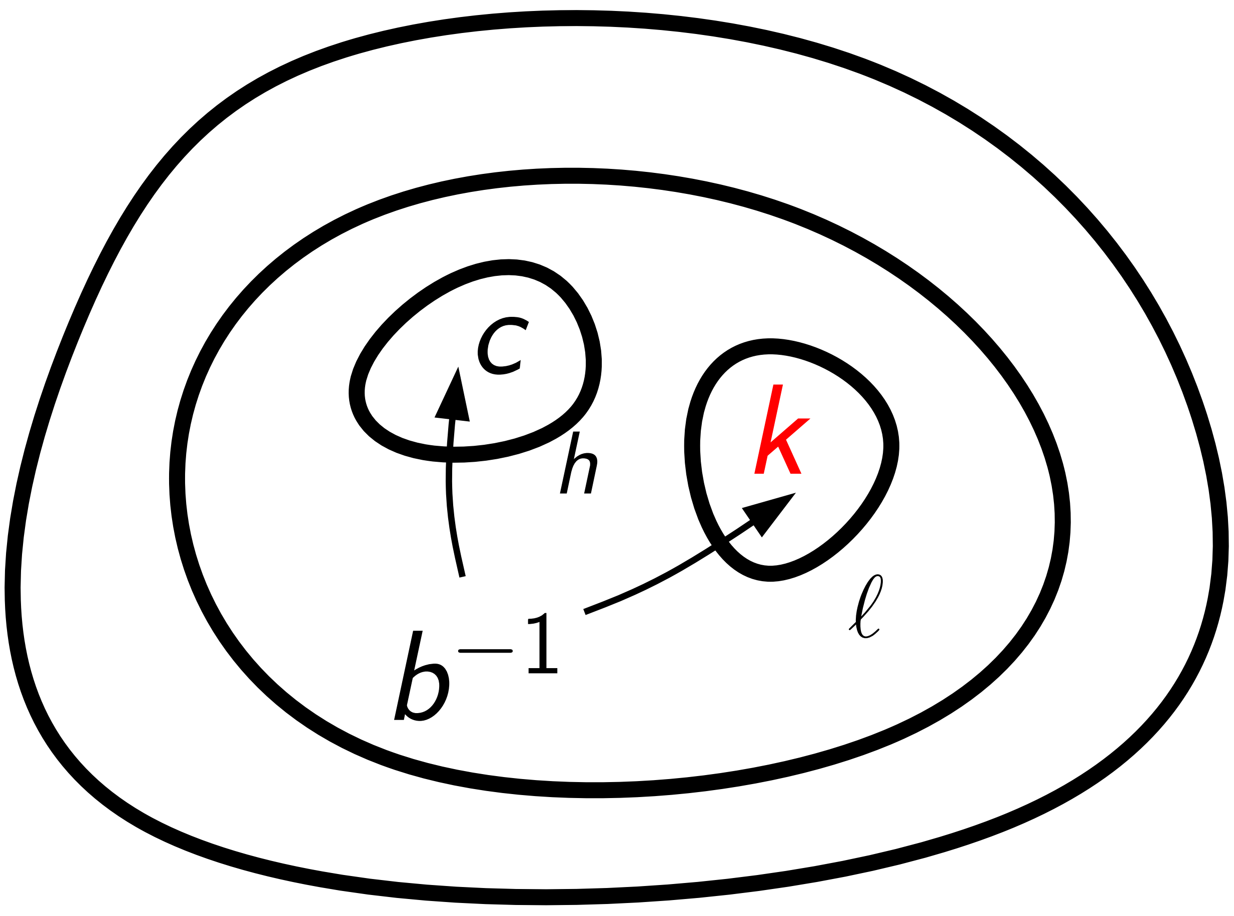
Example

$$ab\mathbf{k} \to c_h d_k \mathbf{k}_\ell$$



Example

$$ab\mathbf{k} \to c_h \mathbf{k}_\ell$$



Consequence: rules become context-free

$$u\,k \to v\,k$$

$$\Downarrow$$

$$k \to vu^{-1}k$$

Register machines

$$\ell_1 : \mathsf{add}(r), \ell_2, \ell_3$$

$$\ell_1 : \mathsf{sub}(r), \ell_2, \ell_3$$

$$\ell_1 : \mathsf{halt}$$

Consequence of using hybrid sets: no zero test

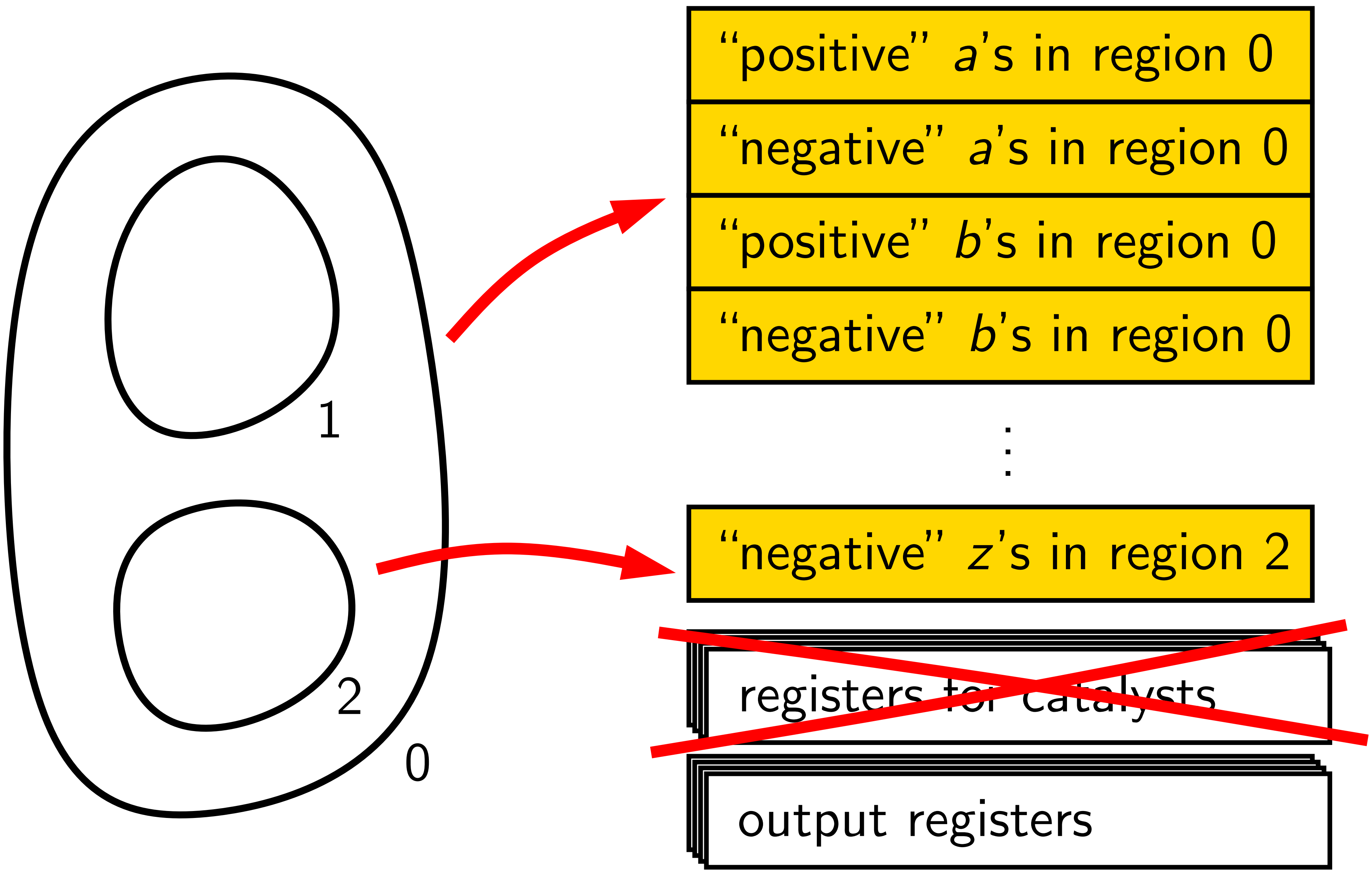
$$\ell_1 : \mathsf{add}(r), \ell_2, \ell_3$$

$$\ell_1 : \mathsf{sub}(r), \ell_2, \mathsf{abort}$$

$$\ell_1 : \mathsf{halt}$$

A subset of the registers must be null at the end of legitimate computations

Simulation by partially blind machines



Simulation algorithm

just look at the catalysts

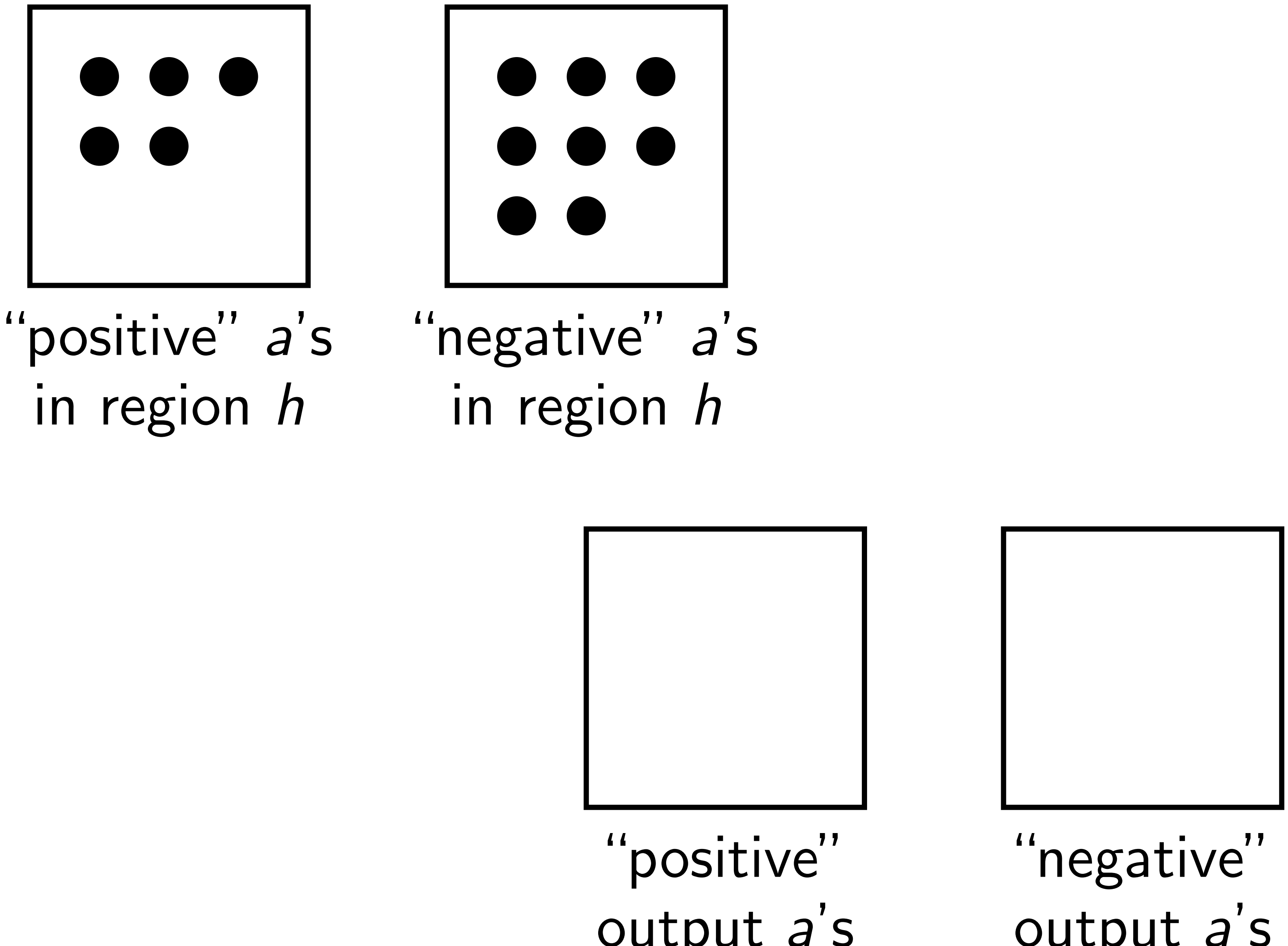
nondeterministically choose a multiset of rules to apply
for each chosen rule $uk \to vk$ with targets **do**
  add u to the corresponding "negative" registers
  add v to the corresponding "positive" registers
jump to the code for the new configuration of catalysts

code for each cfg of catalysts

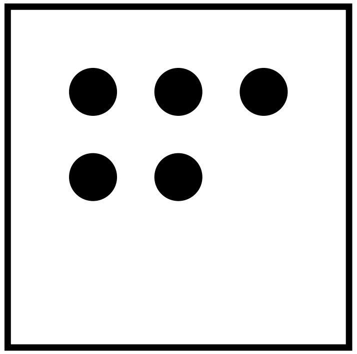
for each output symbol a **do**
  nondeterministically guess if $\#a$ is negative
  if we guessed negative **then**
    compute Δa in the negative output register
  else
    compute Δa in the positive output register

except for halting ones

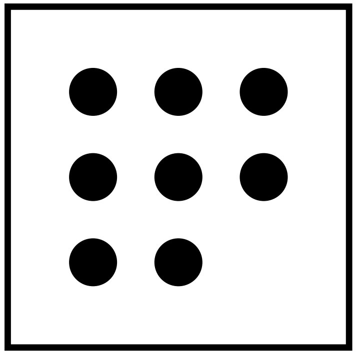
Computing Δa



Computing Δa

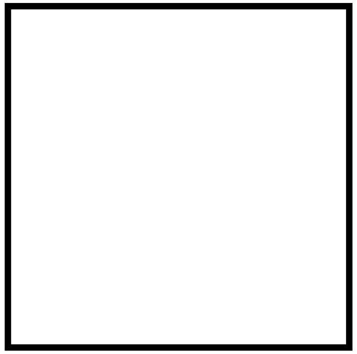


“positive” a’s
in region h

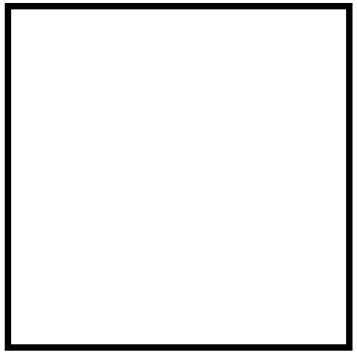


“negative” a’s
in region h

nondeterministic
guess:
$\#a < 0$

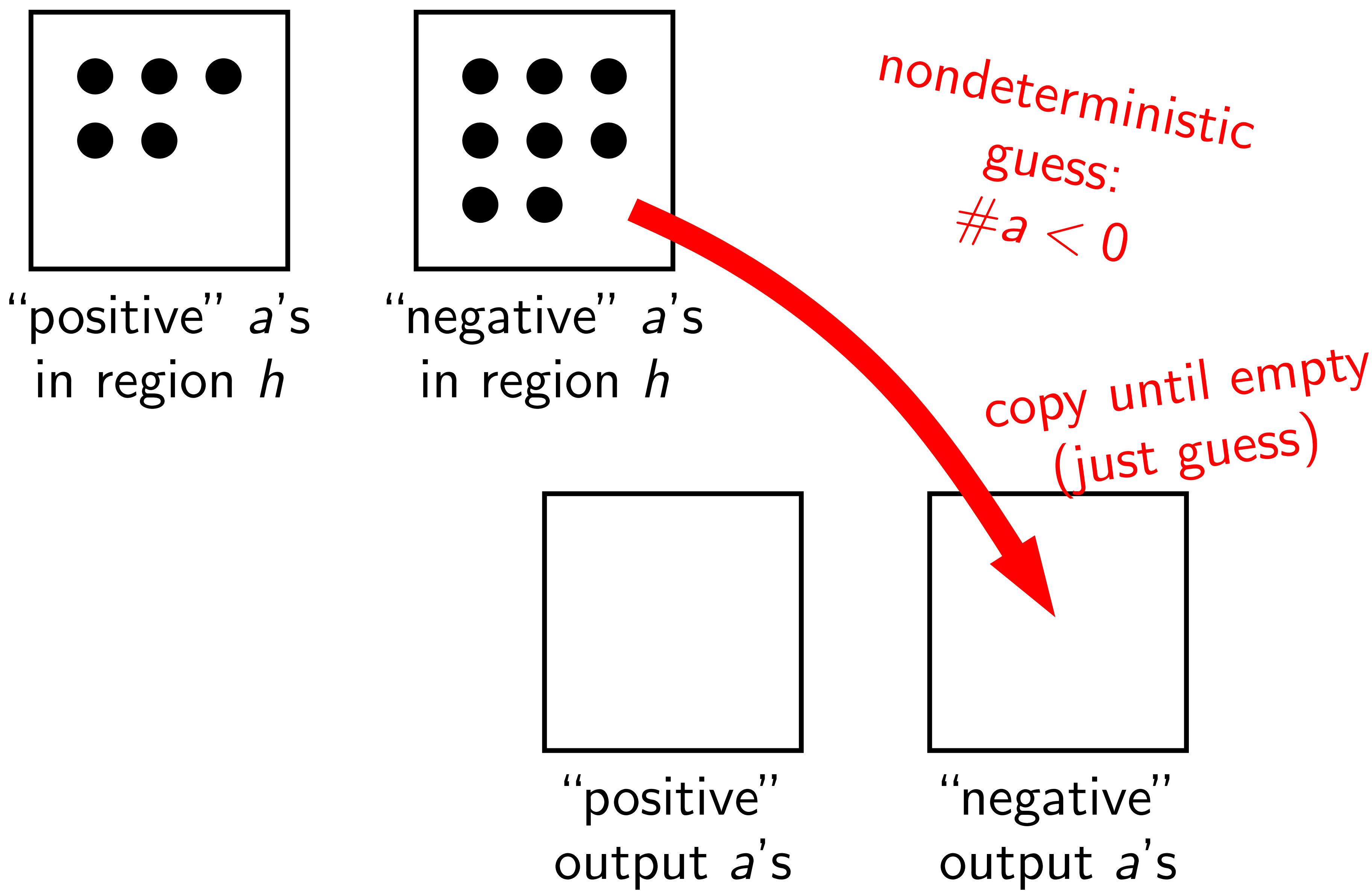


“positive”
output a’s

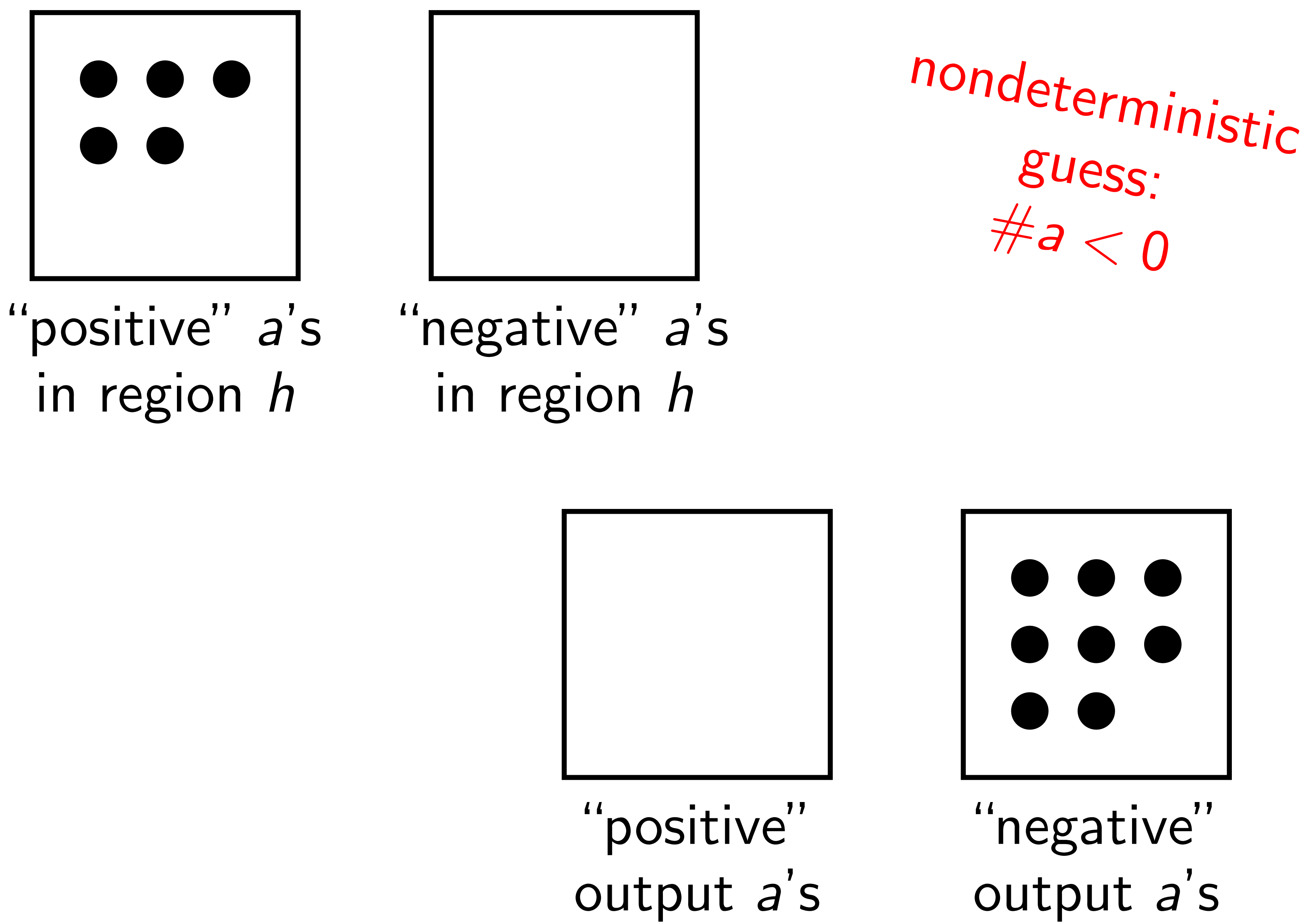


“negative”
output a’s

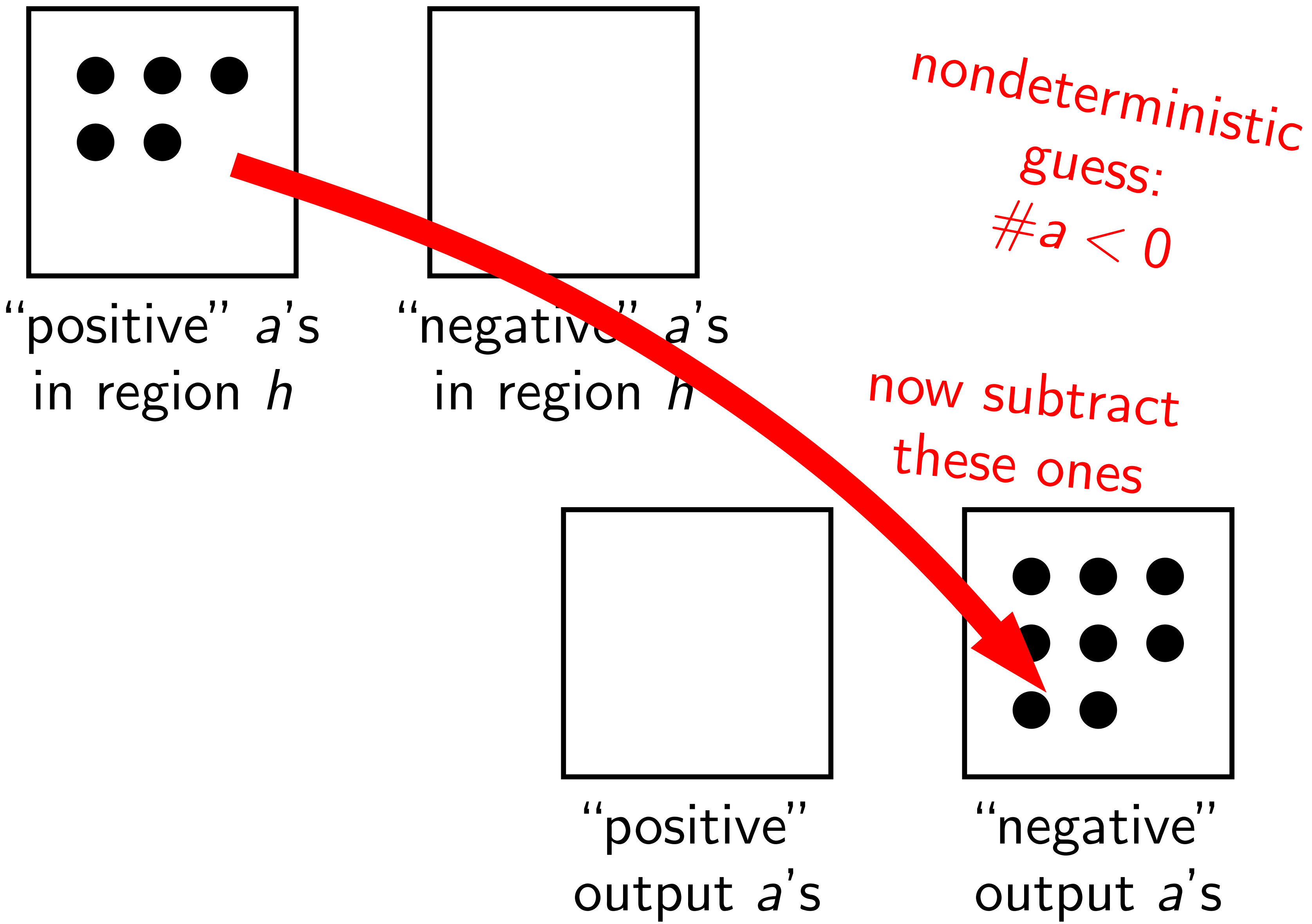
Computing Δa



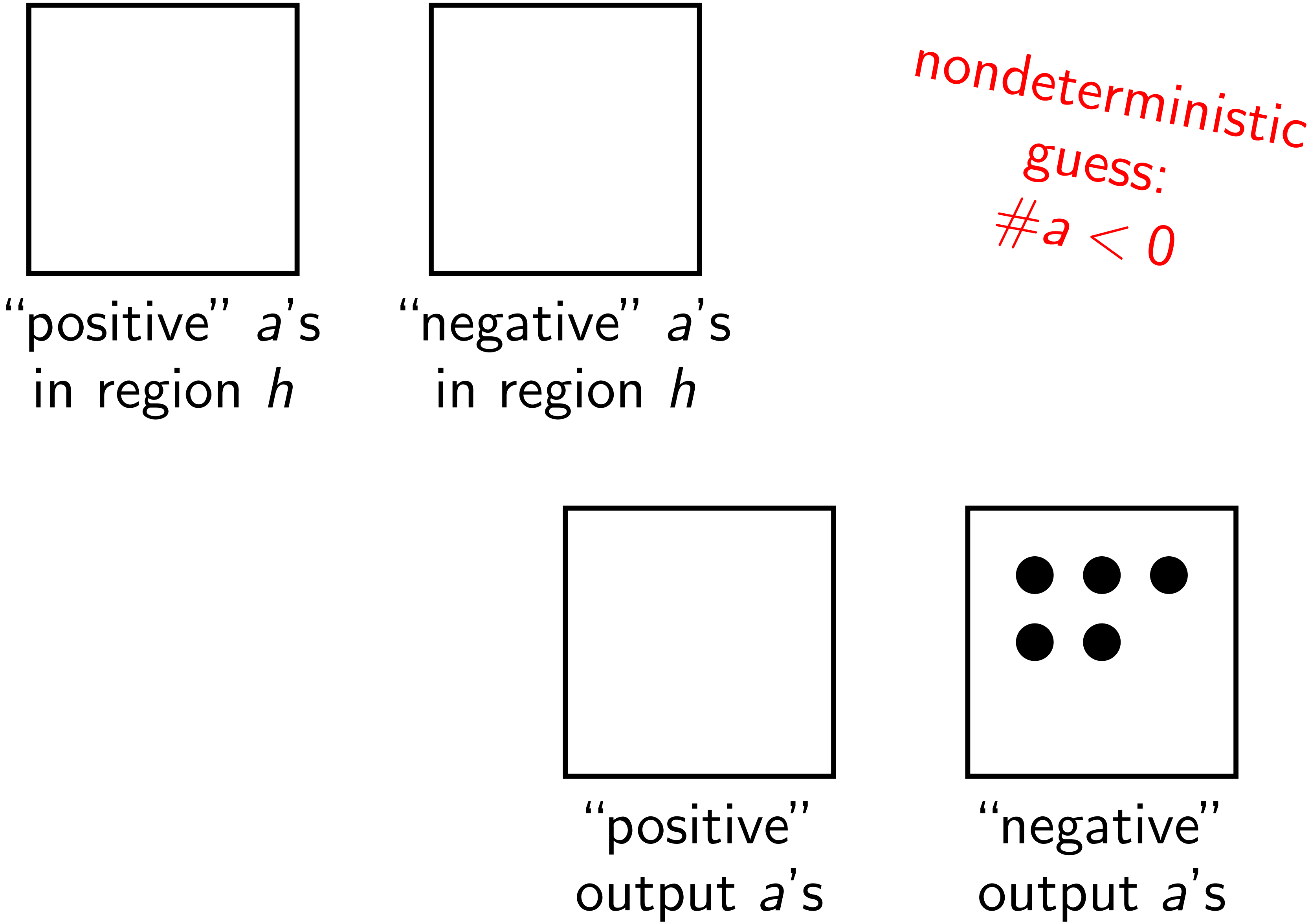
Computing Δa



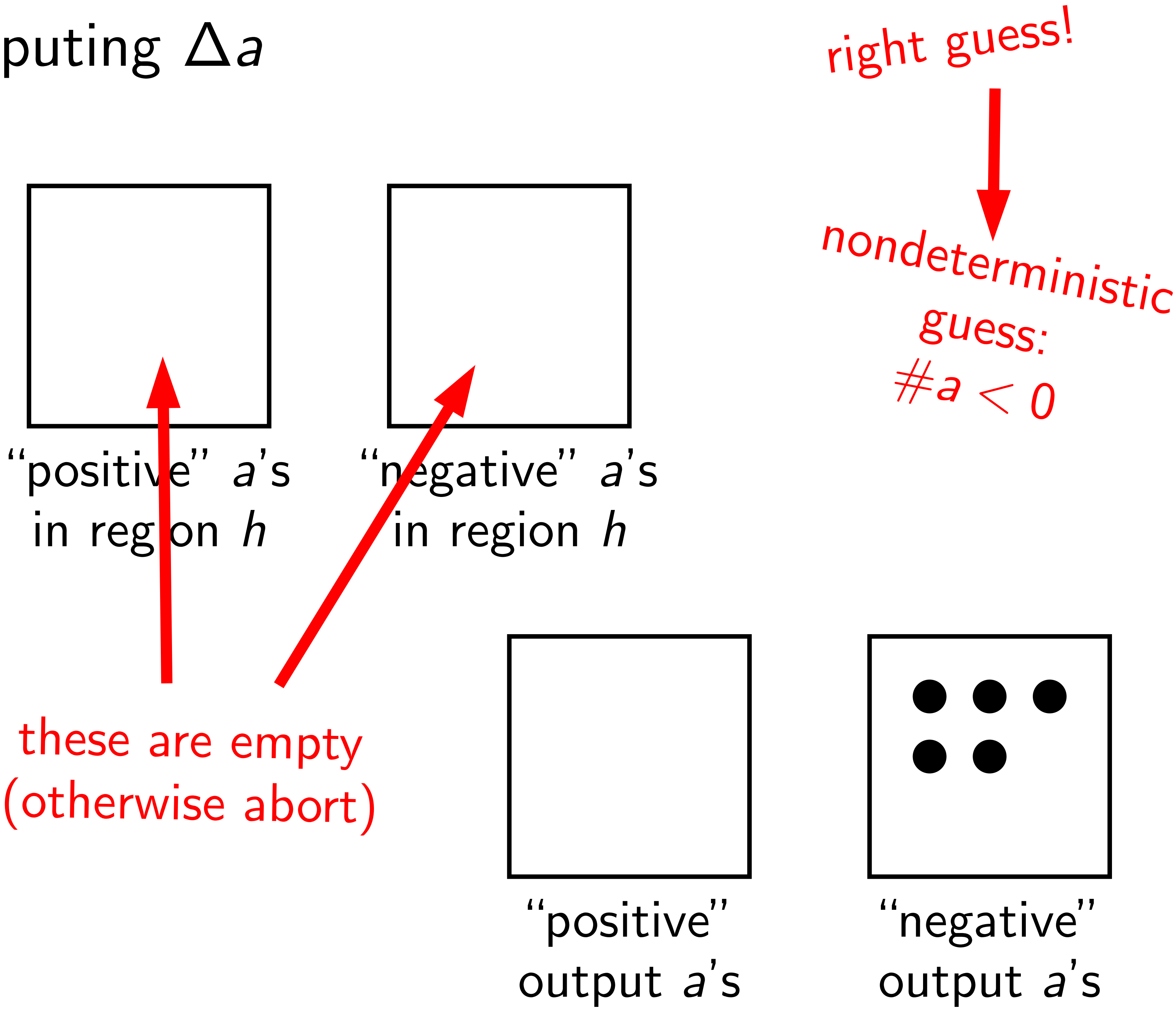
Computing Δa



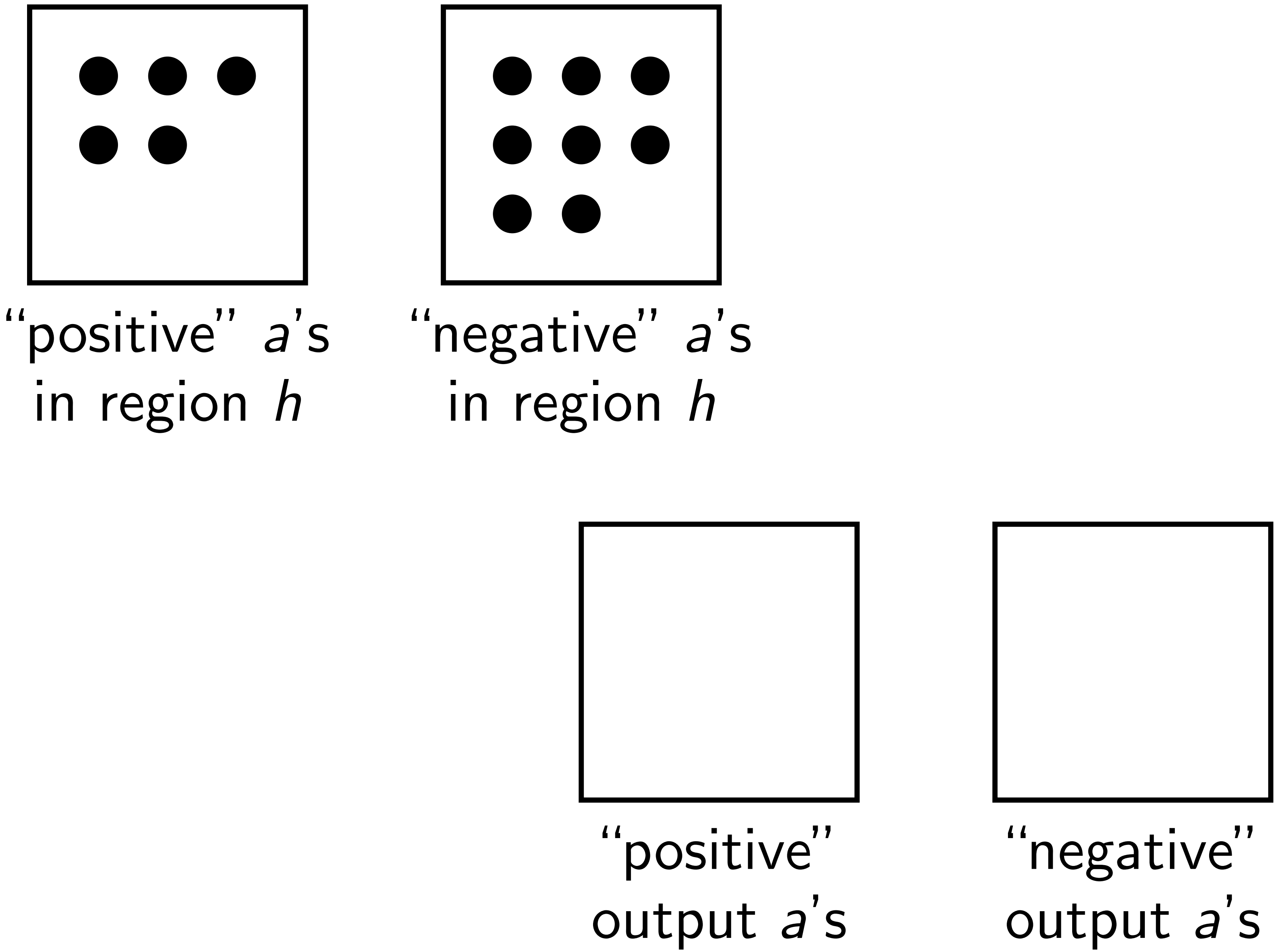
Computing Δa



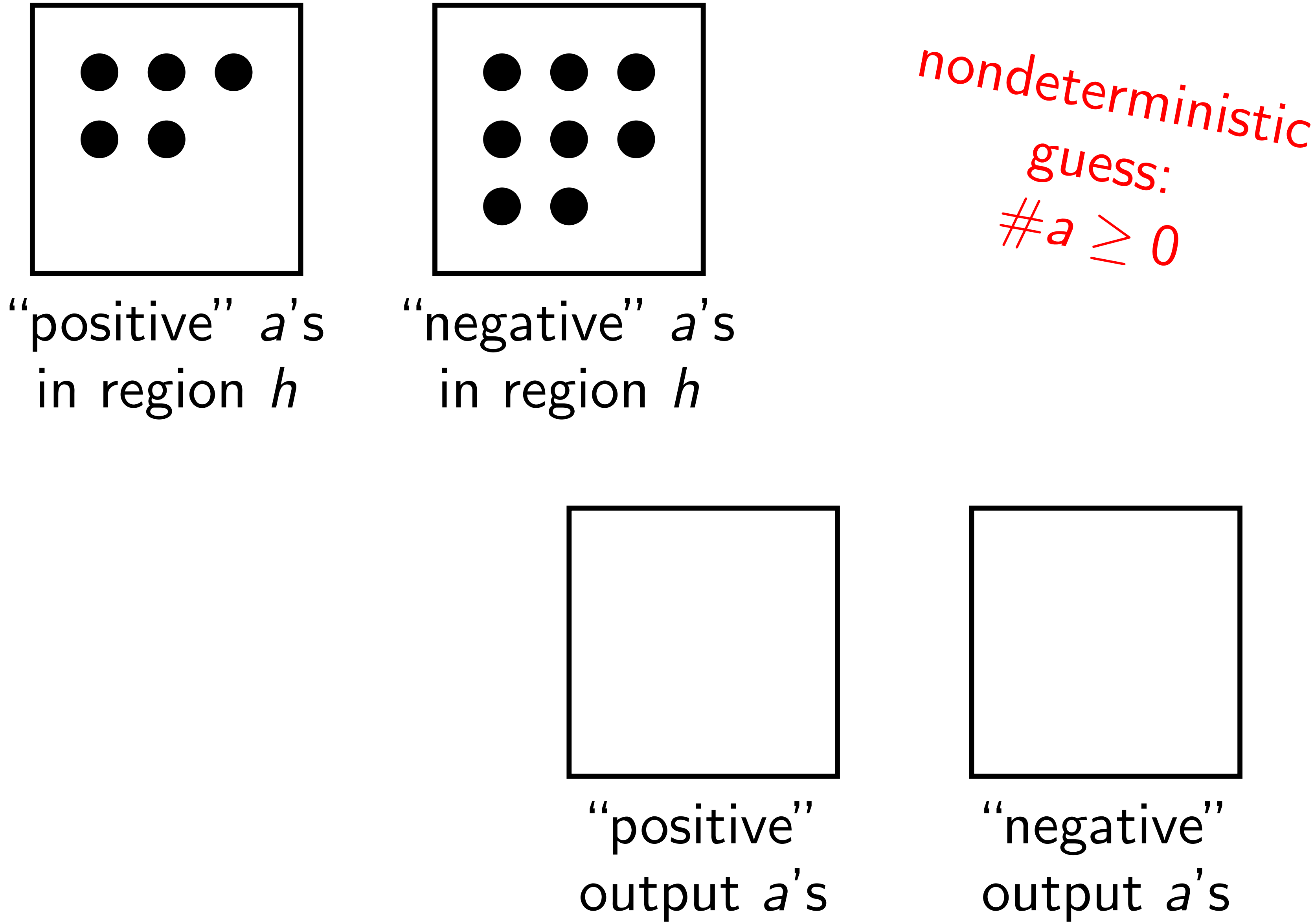
Computing Δa



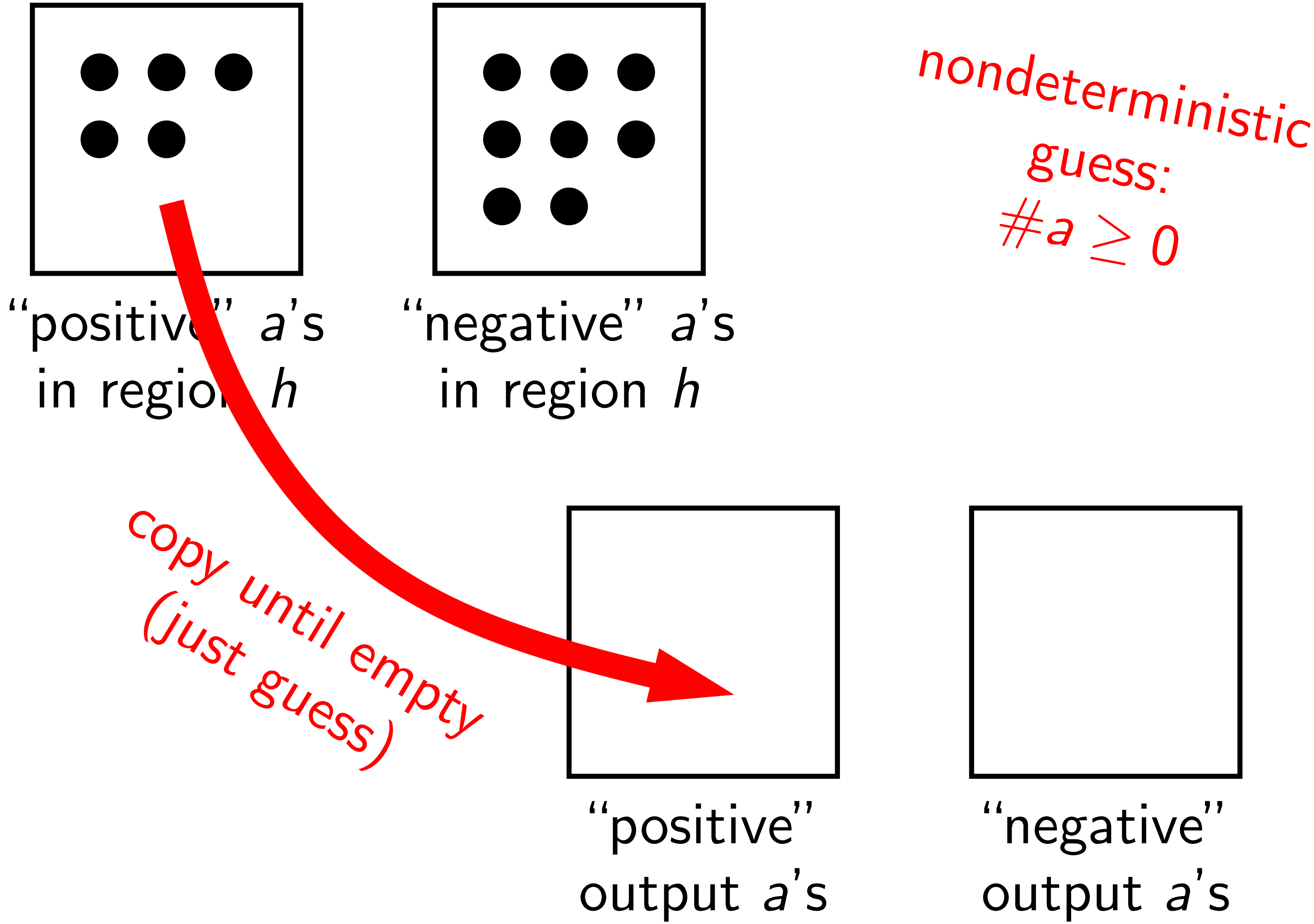
Computing Δa



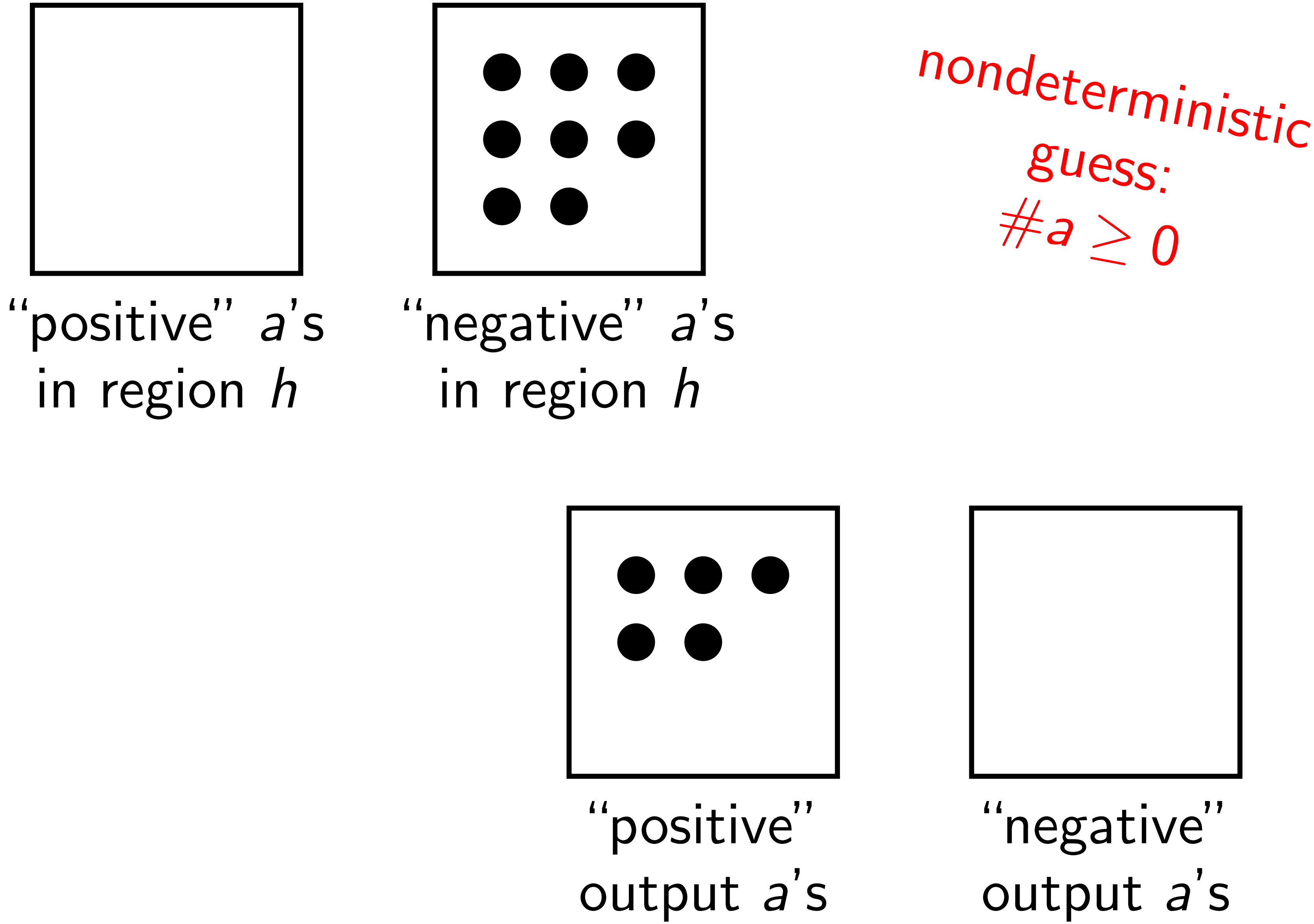
Computing Δa



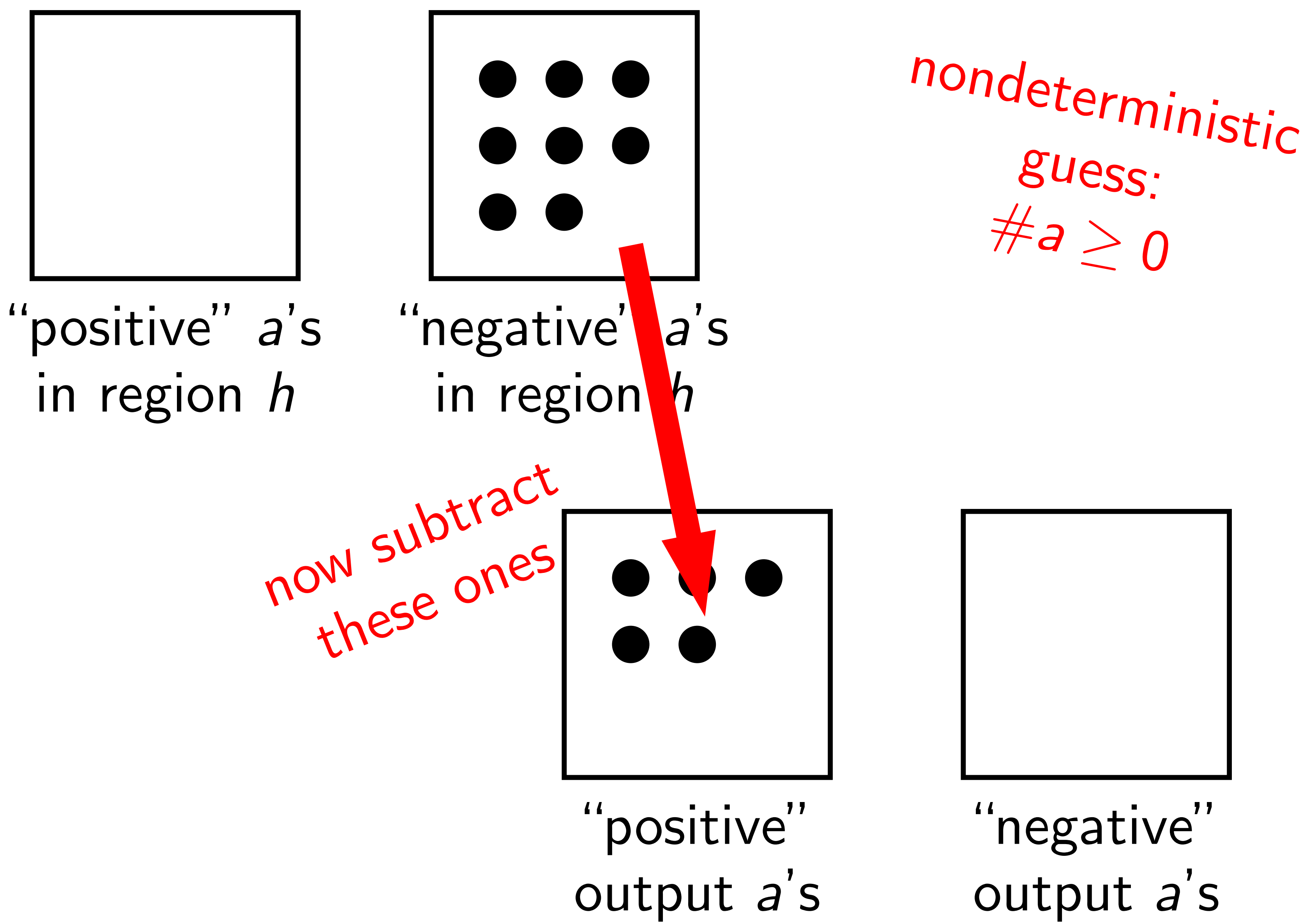
Computing Δa



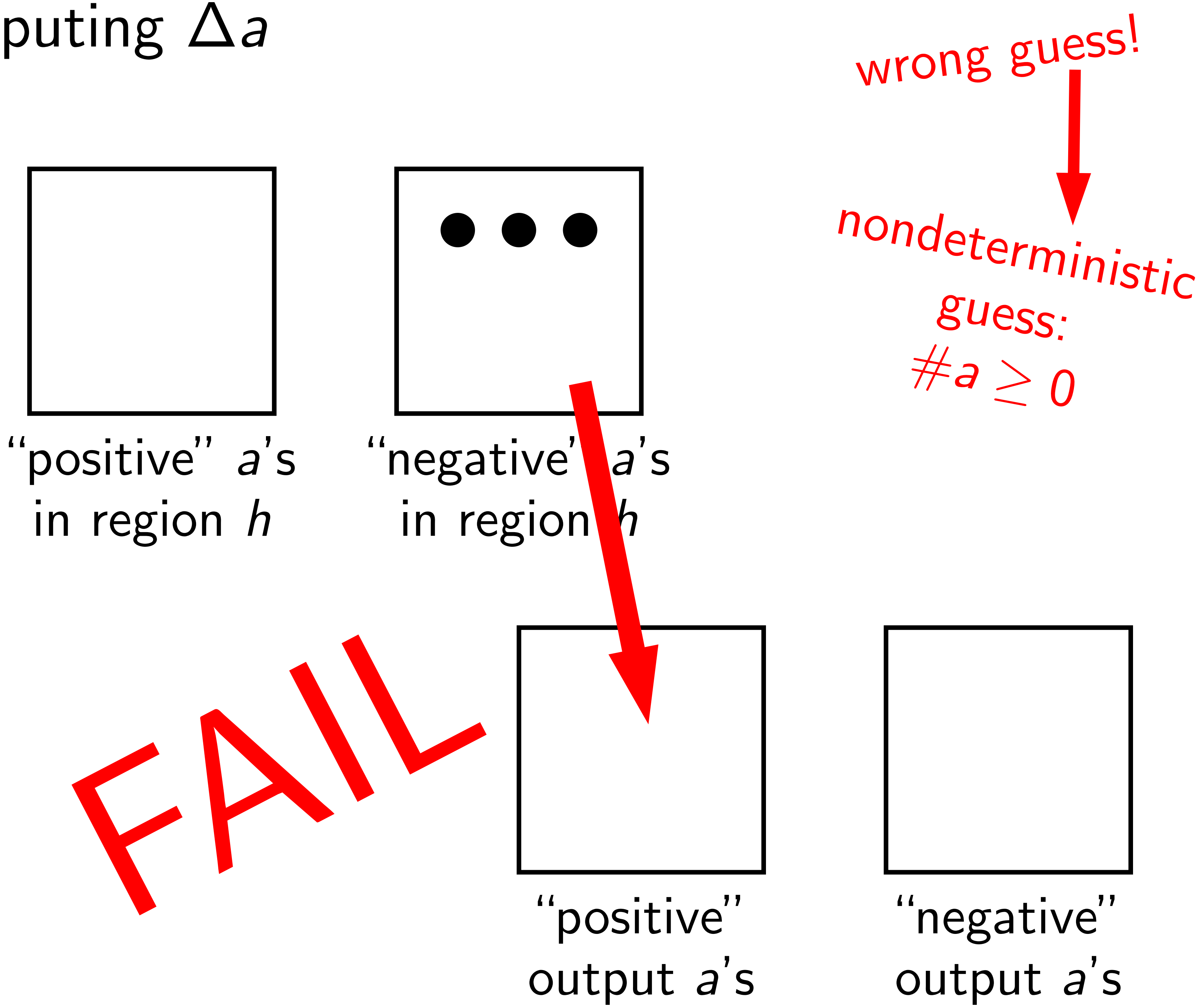
Computing Δa



Computing Δa



Computing Δa



Summary

Theorem. These kinds of P system can be simulated by a partially blind register machine, and so they are not universal □

Improvement (see papers at CMC17 in Milano)

Theorem. These kinds of P system can be simulated by a blind register machine, and so they are even less universal □

Thanks to
Rudi Freund &
Sergiu Ivanov!

Thanks for your attention!