A Universal Reversible Turing Machine

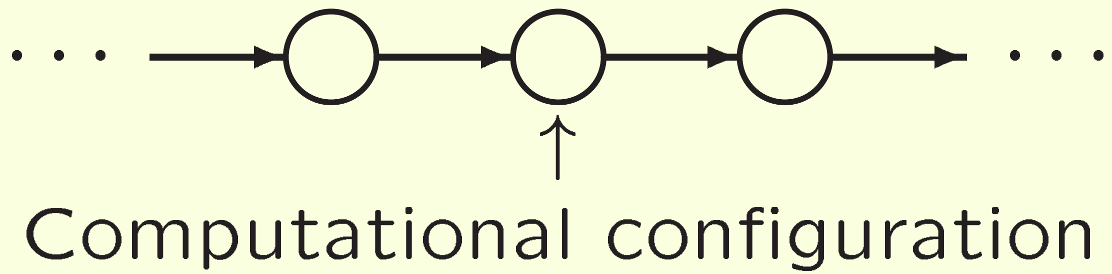
Kenichi Morita, and Yoshikazu Yamaguchi
Hiroshima University

Contents

1. What Is Reversible Computing?

Reversible Computing

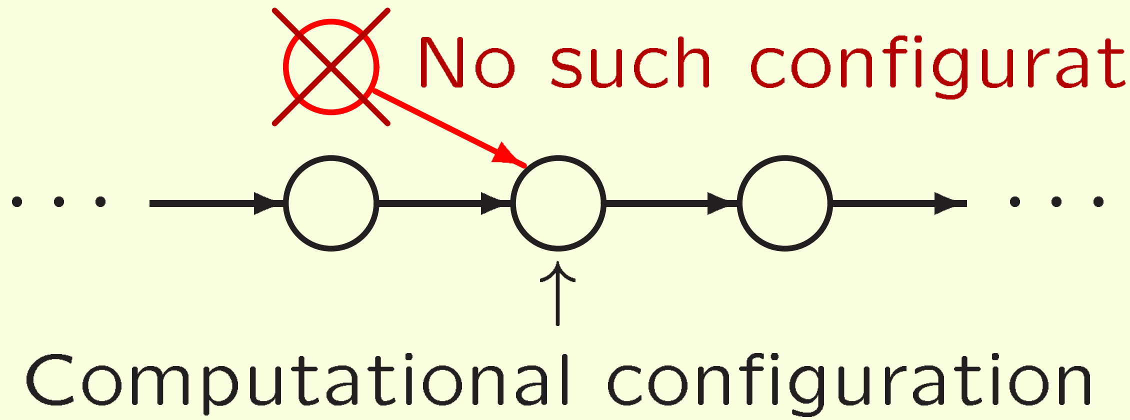
- Roughly speaking, it is a "backward deterministic" computing; i.e., every computational configuration has at most one predecessor.



- Though its definition is rather simple, it reflects physical reversibility well.

Reversible Computing

- Roughly speaking, it is a "backward deterministic" computing; i.e., every computational configuration has at most one predecessor.



- Though its definition is rather simple, it reflects physical reversibility well.

Why Reversible Computing?

- Reversibility is one of the fundamental microscopic physical laws of Nature.
- It is important to investigate how computation can be carried out in a reversible system.
- Because the size of future computing devices will surely become nano-scale ones.
- It relates to energy consumption in computing, and to quantum computing.

Several Models of Reversible Computing

- Reversible Turing Machines (RTM)
- Reversible Logic Elements (RLE)
- Reversible Cellular Automata (RCA)
- Reversible Physical Models

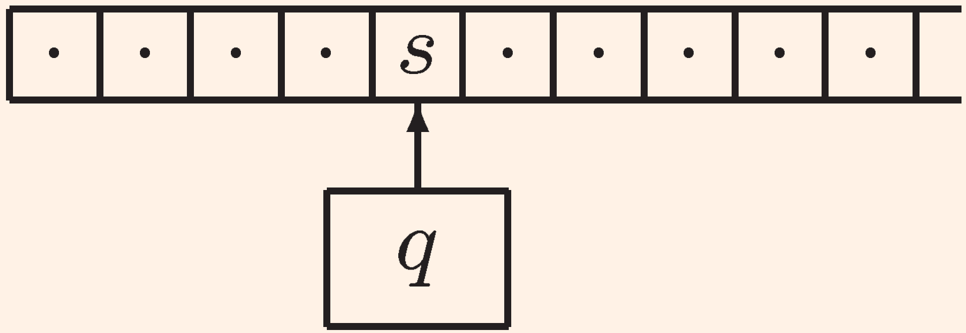
- These models are closely related each other.
- Reversible computers work in a very different fashion from classical computers!

2. Reversible Turing Machines, and Their Relation to Other Reversible Models

2.1 Reversible Turing Machine (RTM)
2.2 Reversible Logic Element (RLE)
2.3 Billiard Ball Model (BBM)
— A Reversible Physical Model —
2.4 Reversible Cellular Automaton (RCA)

2.1 Reversible Turing Machine (RTM)

A “backward deterministic” TM.



Precise definitions for RTMs will be given later.

Universality of RTMs

Theorem [Bennett, 1973]
For any one-tape (irreversible) TM, there is a reversible TM that simulates the former and leaves no garbage information when it halts.

Note: Converting a classical (irreversible) universal TM by this method, we have a universal RTM. But, its size will become very large.

2.2 Reversible Logic Element

A logic element whose function is described by a one-to-one mapping.

(1) Reversible logic elements without memory (i.e., reversible logic gates):

- Toffoli gate [Toffoli, 1980]
- Fredkin gate [Fredkin and Toffoli, 1982]
- etc.

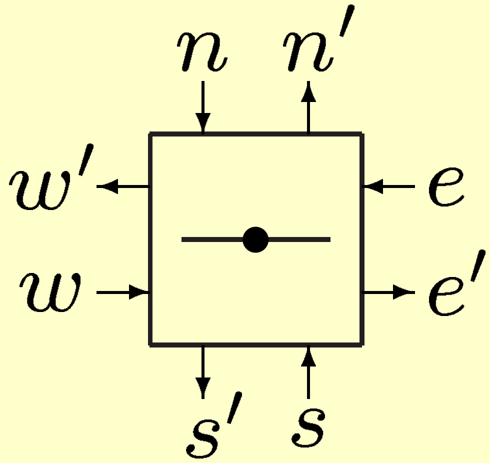
(2) Reversible logic elements with memory:

- Rotary element (RE) [Morita, MCU 2001]
- etc.

(2) Reversible logic elements with memory

Rotary element (RE):

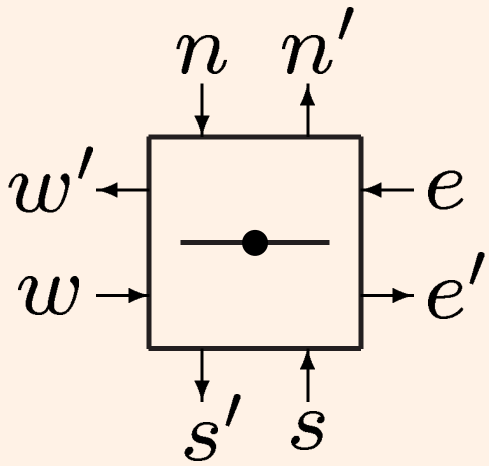
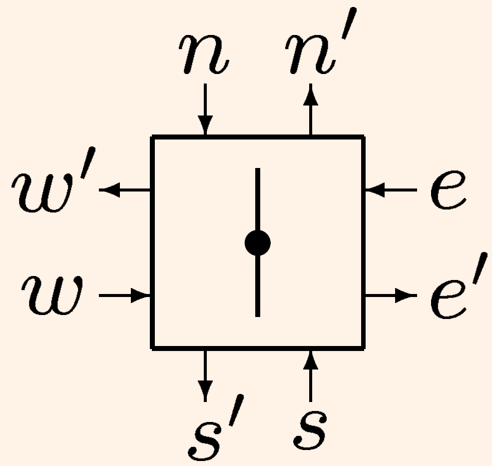
A 2-state 4-input-line 4-output-line element.



(Remark)

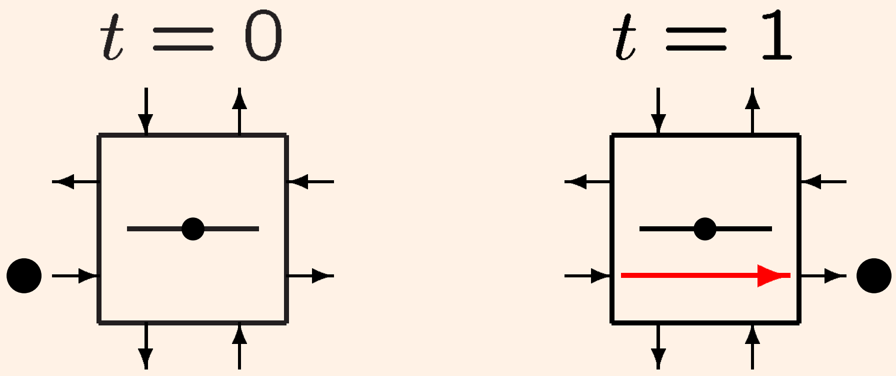
We assume signal "1" is given at most one input line.

Two States of an RE

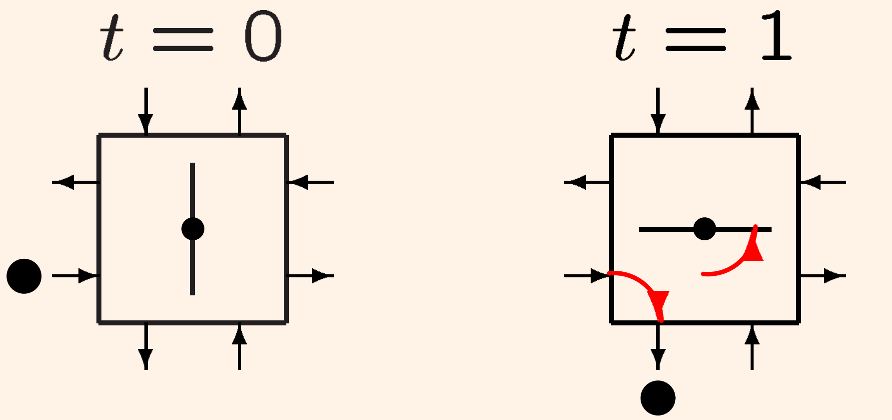
H-state	V-state
	

Operations of an RE

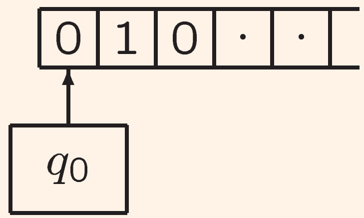
- Parallel case:



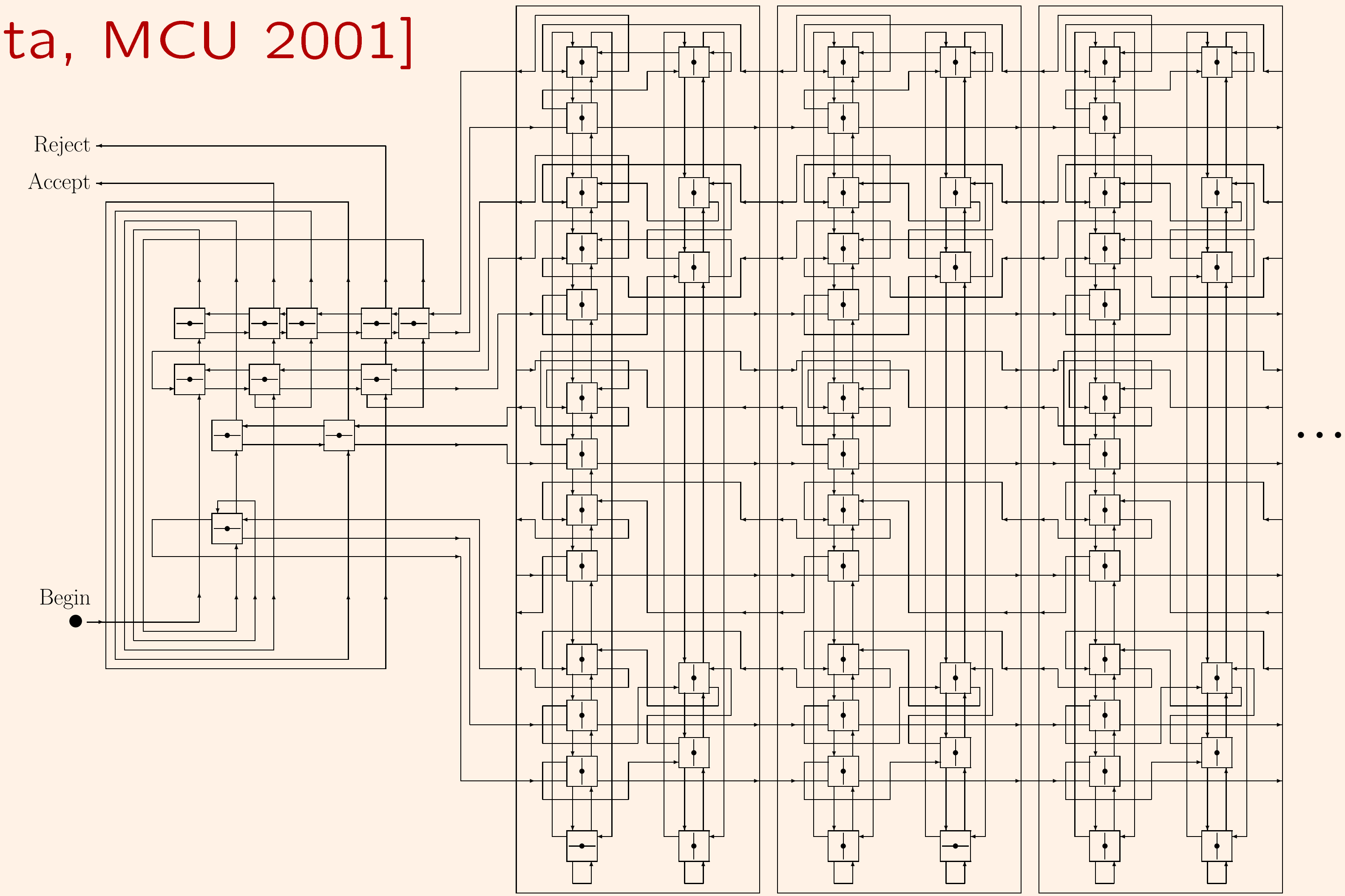
- Orthogonal case:



Any Reversible Turing Machine Can Be Composed Only of REs

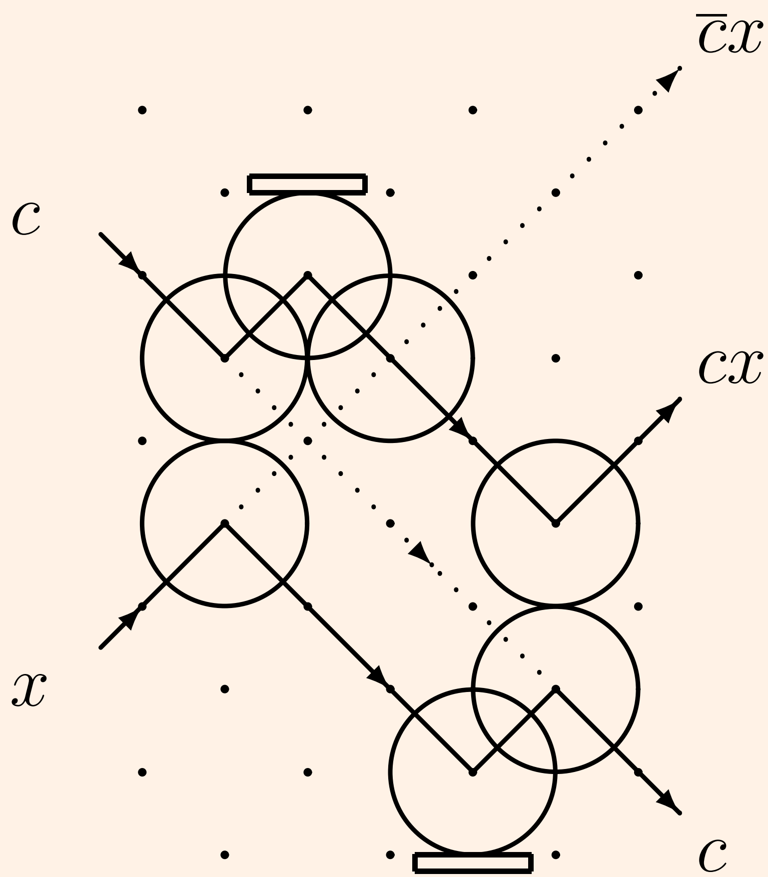


[Morita, MCU 2001]

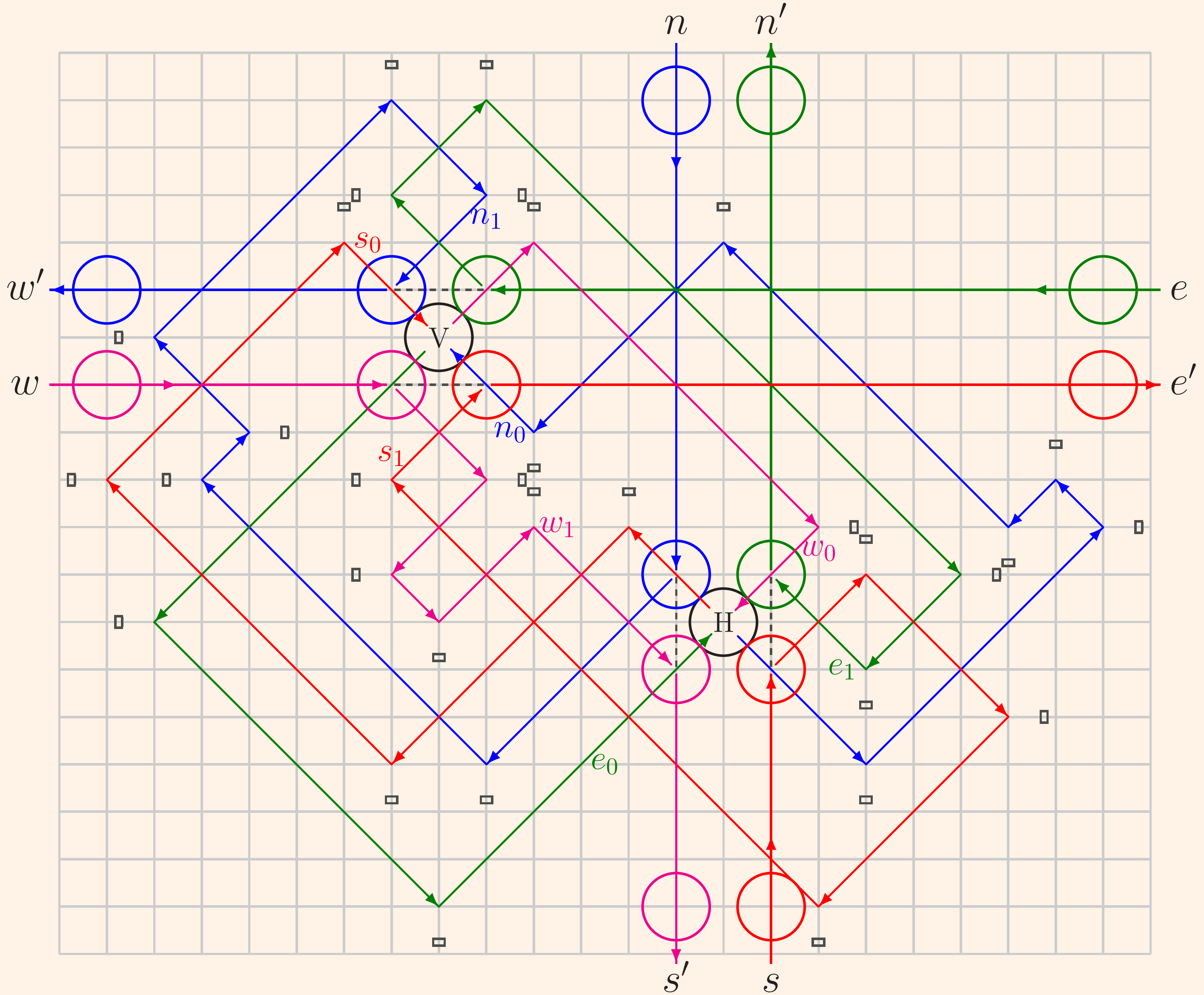


2.3 Billiard Ball Model (BBM)
– A Reversible Physical Model –

[Fredkin and Toffoli, 1982]

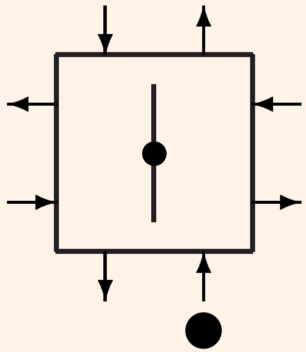


Realization of an RE by BBM [Morita, 2006]

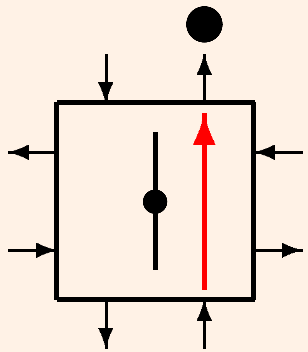


Parallel Case

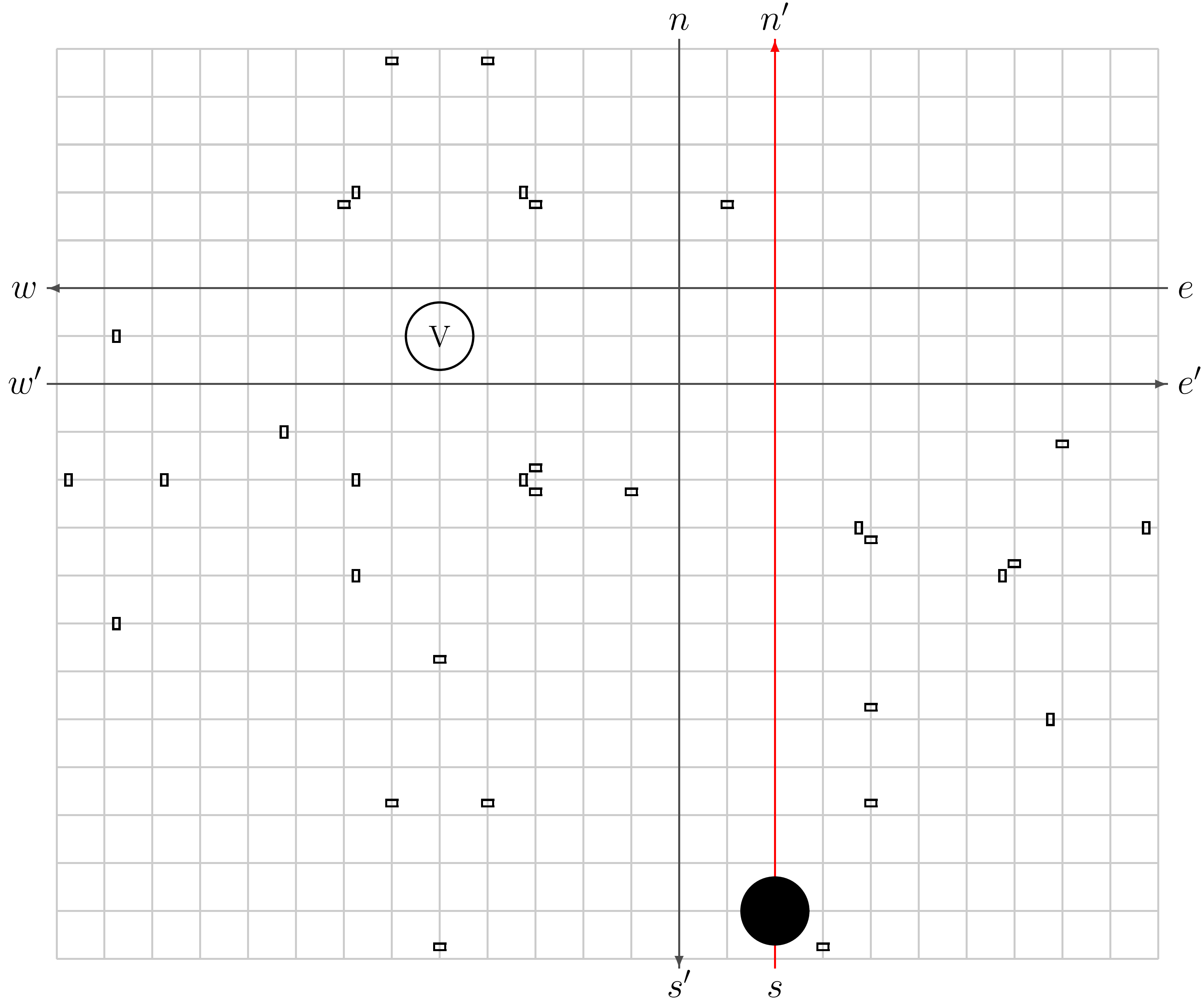
$t = 0$



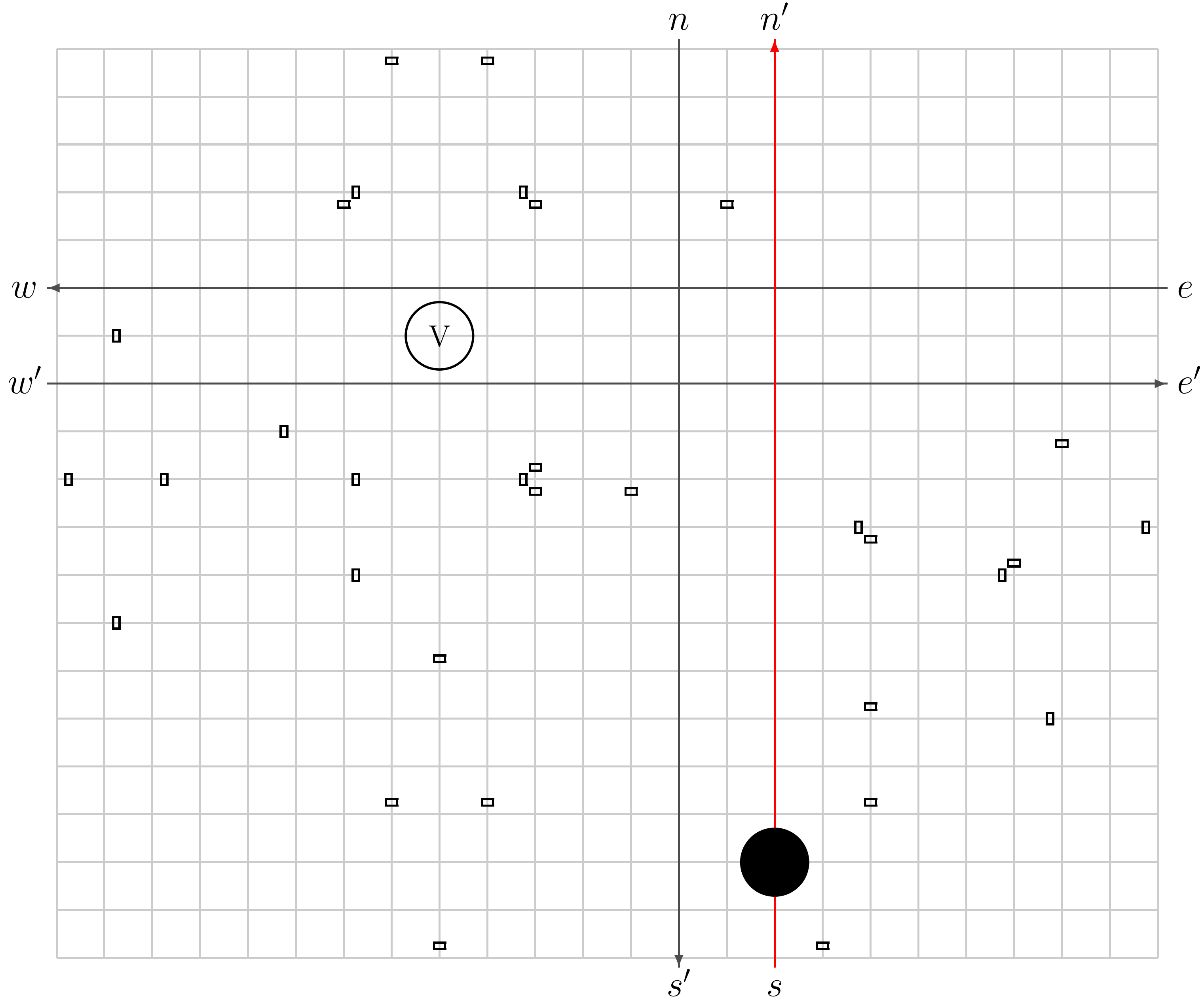
$t = 1$



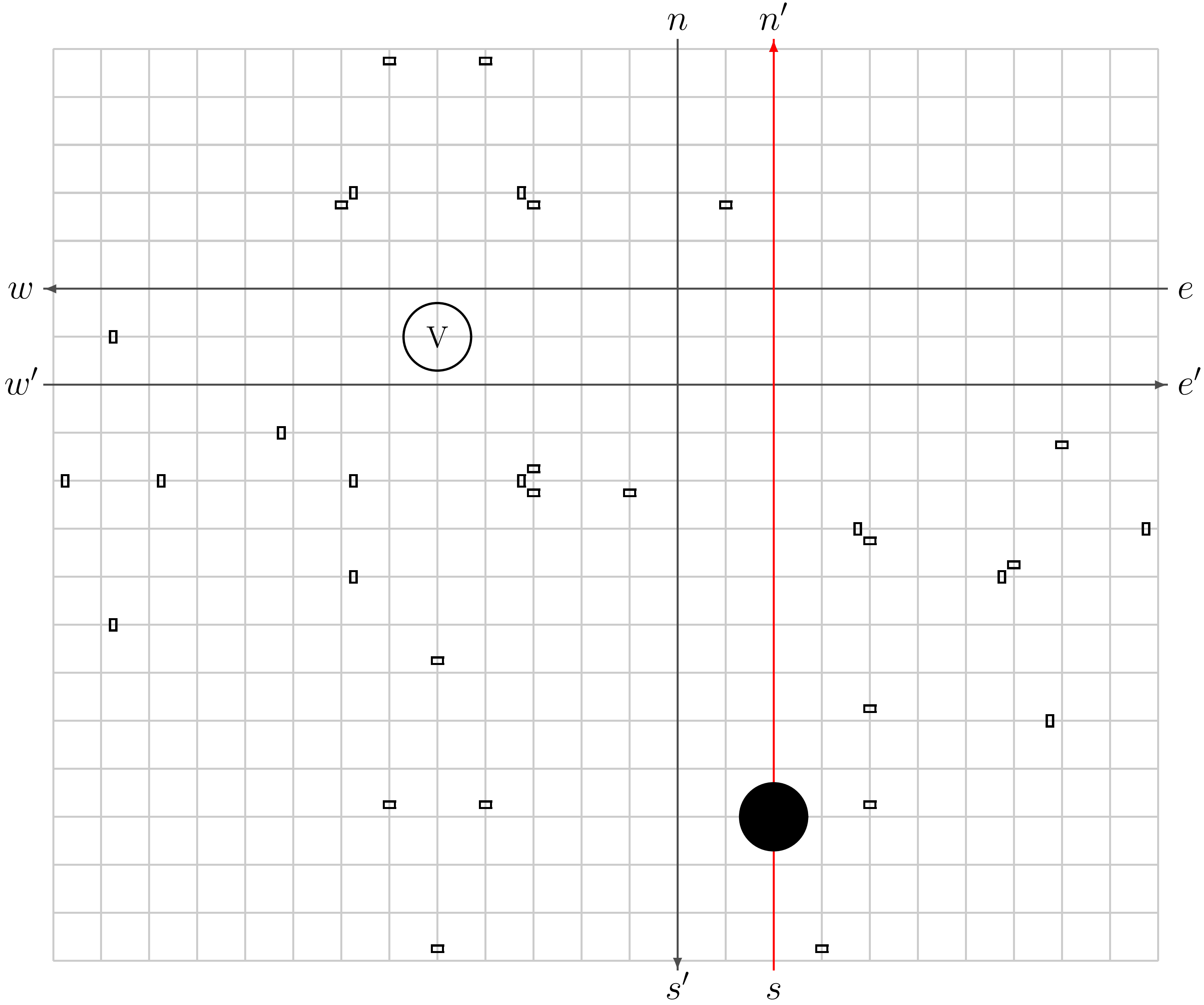
Movements of Balls (State: V, Input: s)



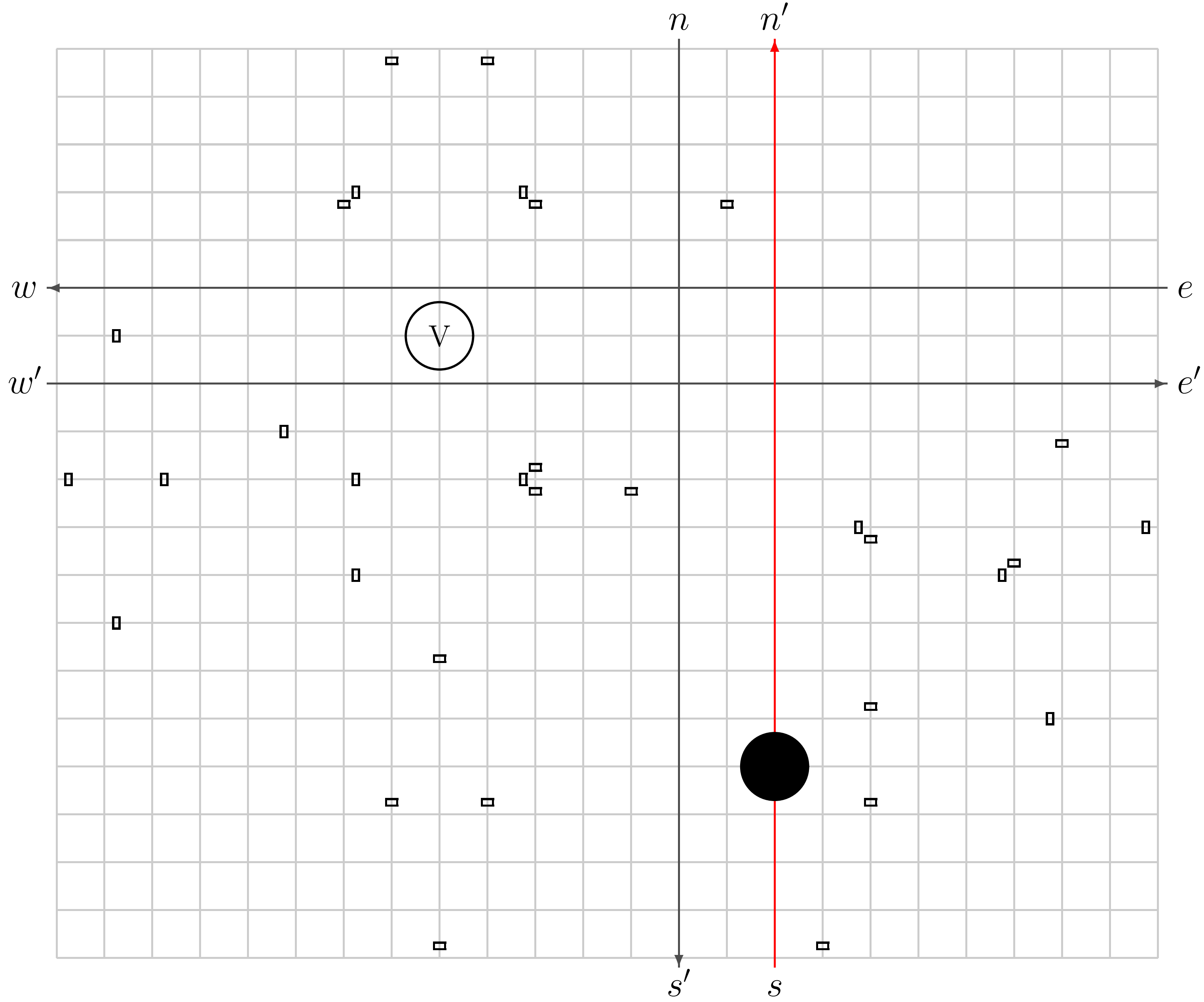
Movements of Balls (State: V, Input: s)



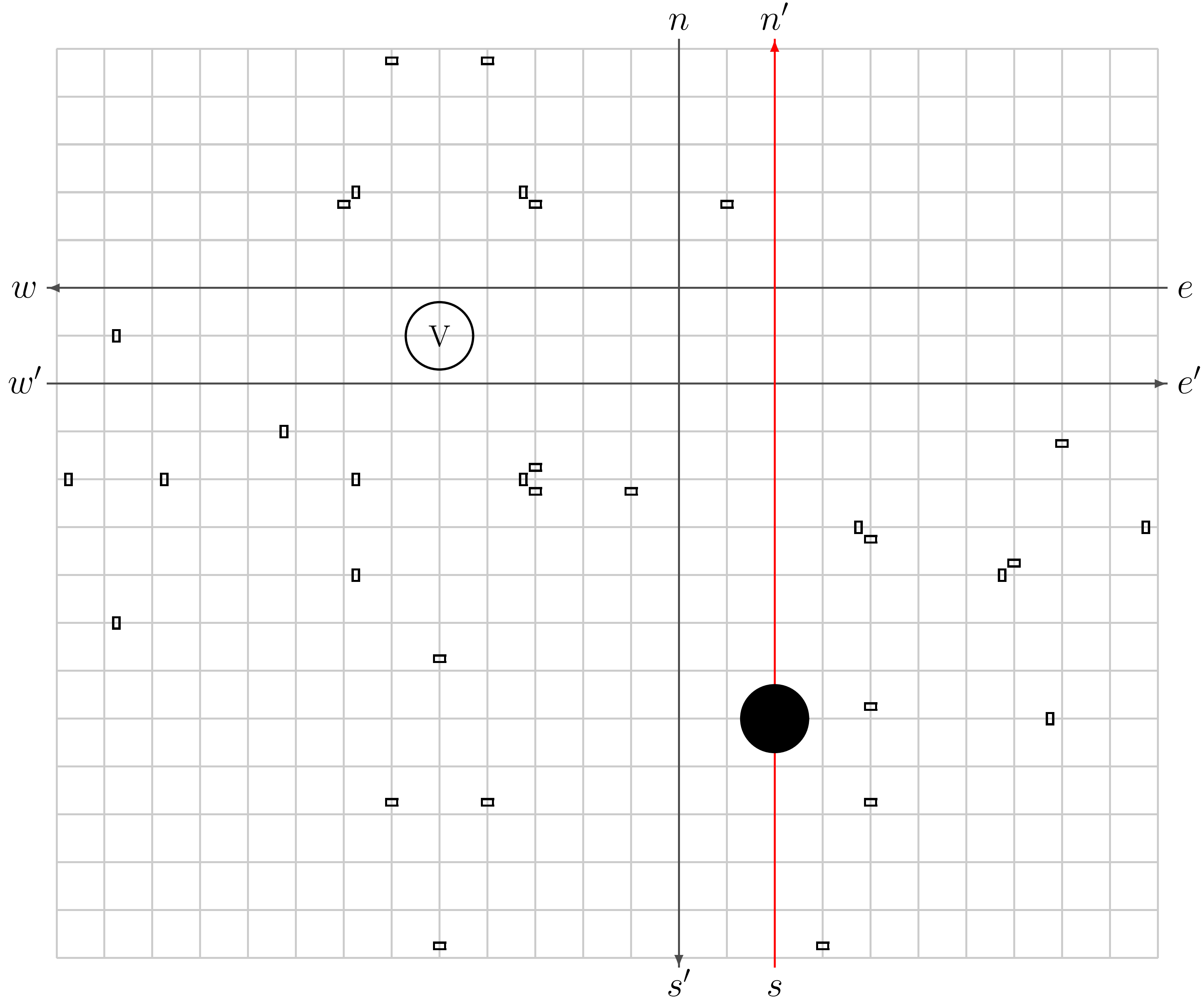
Movements of Balls (State: V, Input: s)



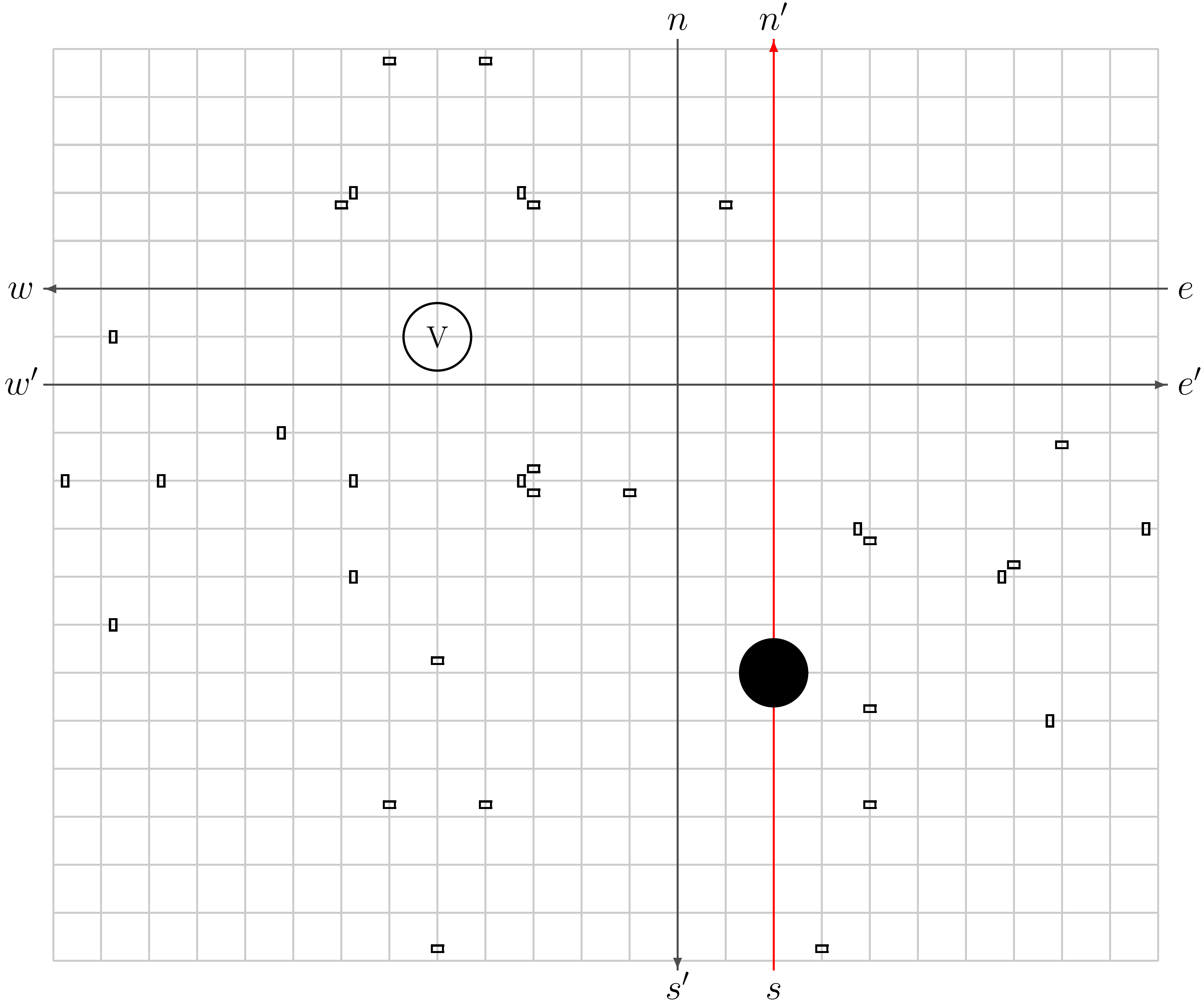
Movements of Balls (State: V, Input: s)



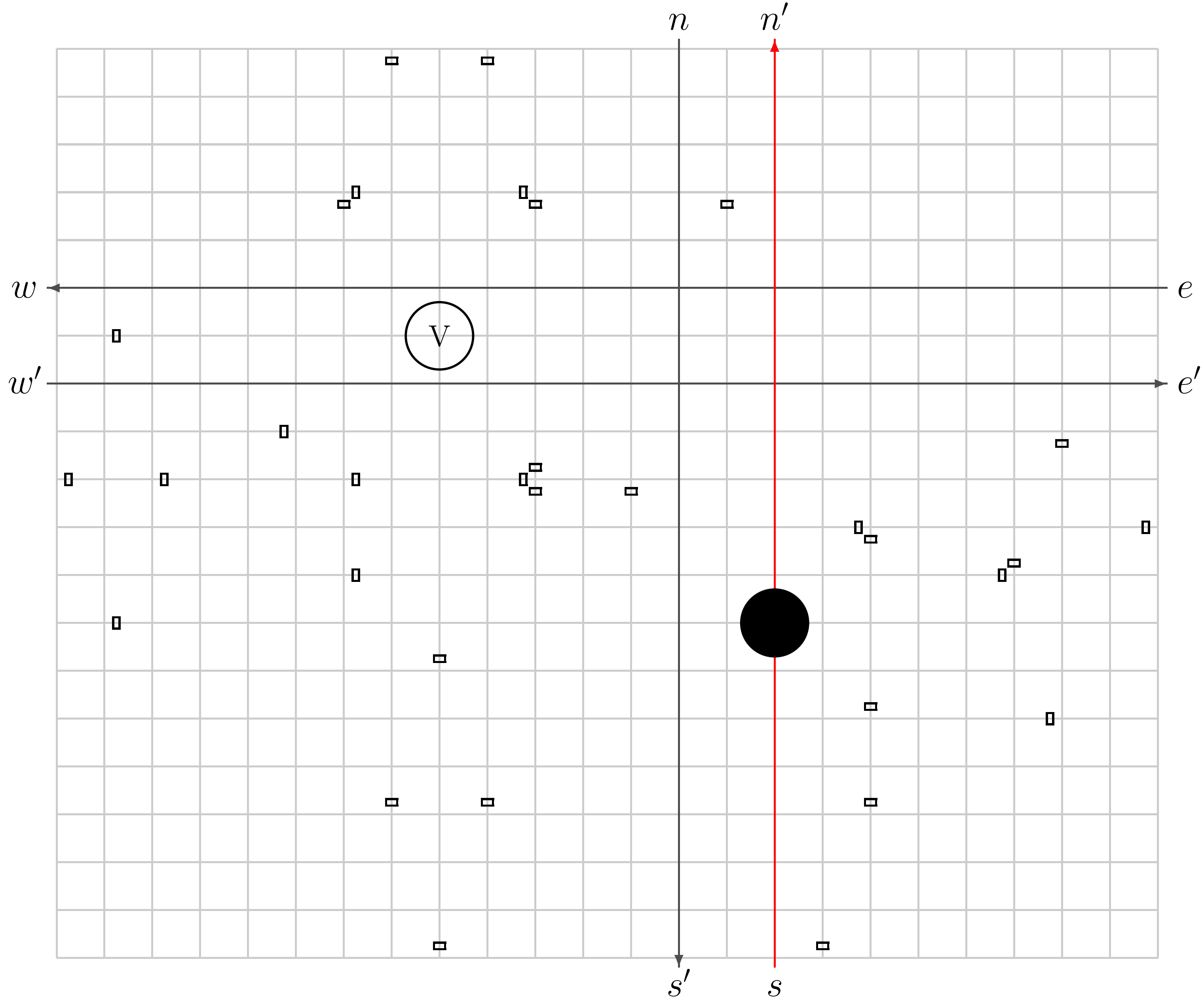
Movements of Balls (State: V, Input: s)



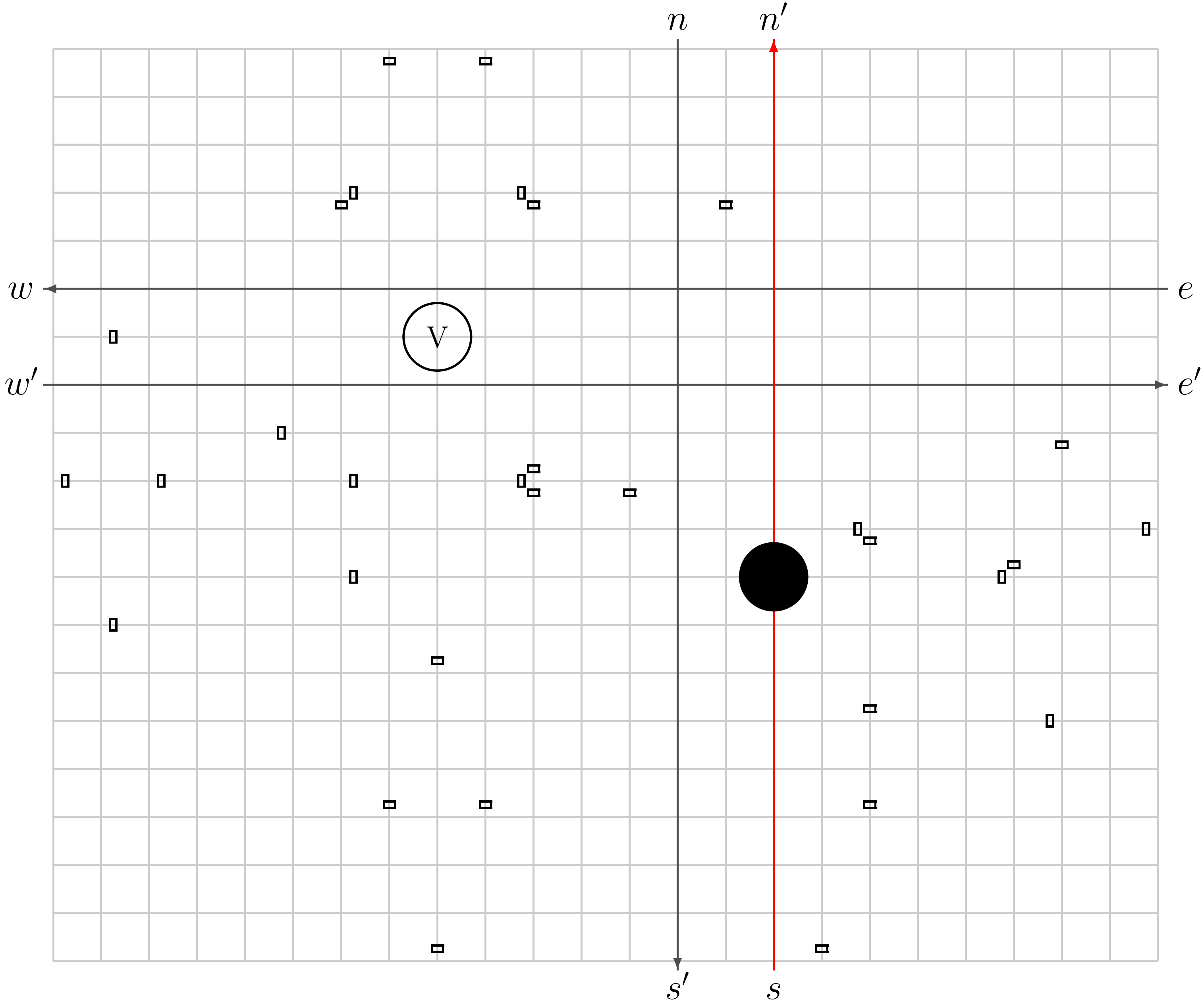
Movements of Balls (State: V, Input: s)



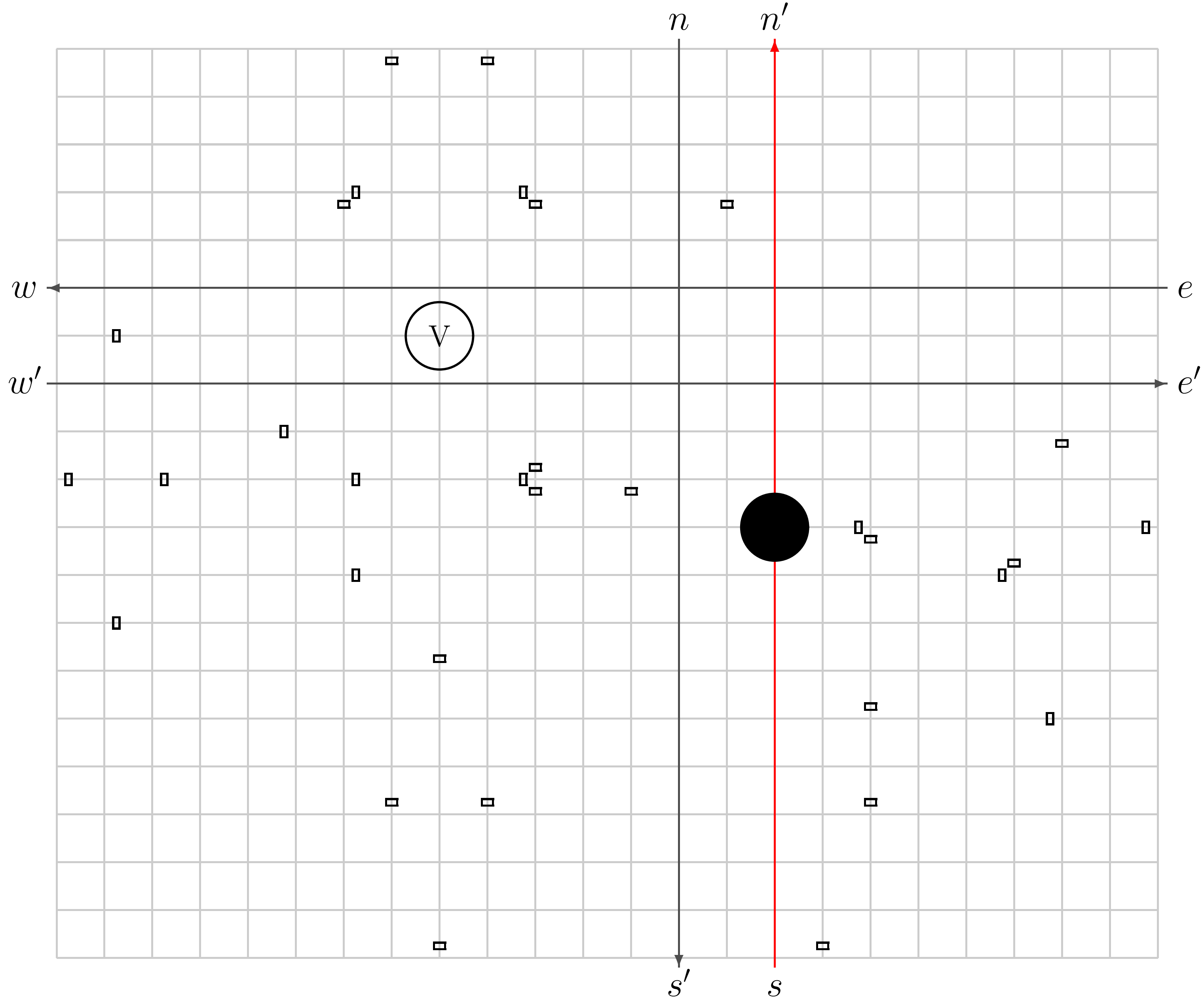
Movements of Balls (State: V, Input: s)



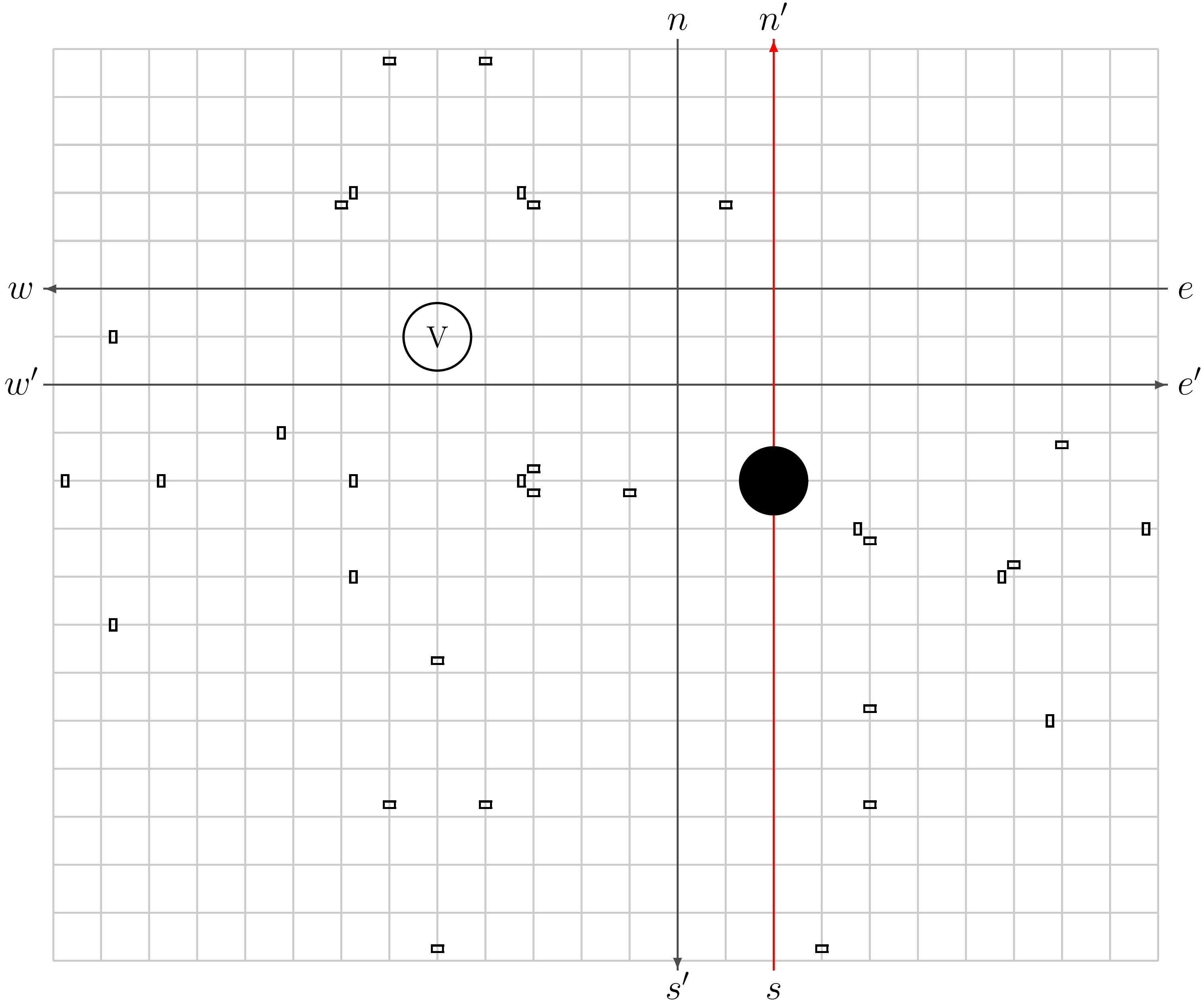
Movements of Balls (State: V, Input: s)



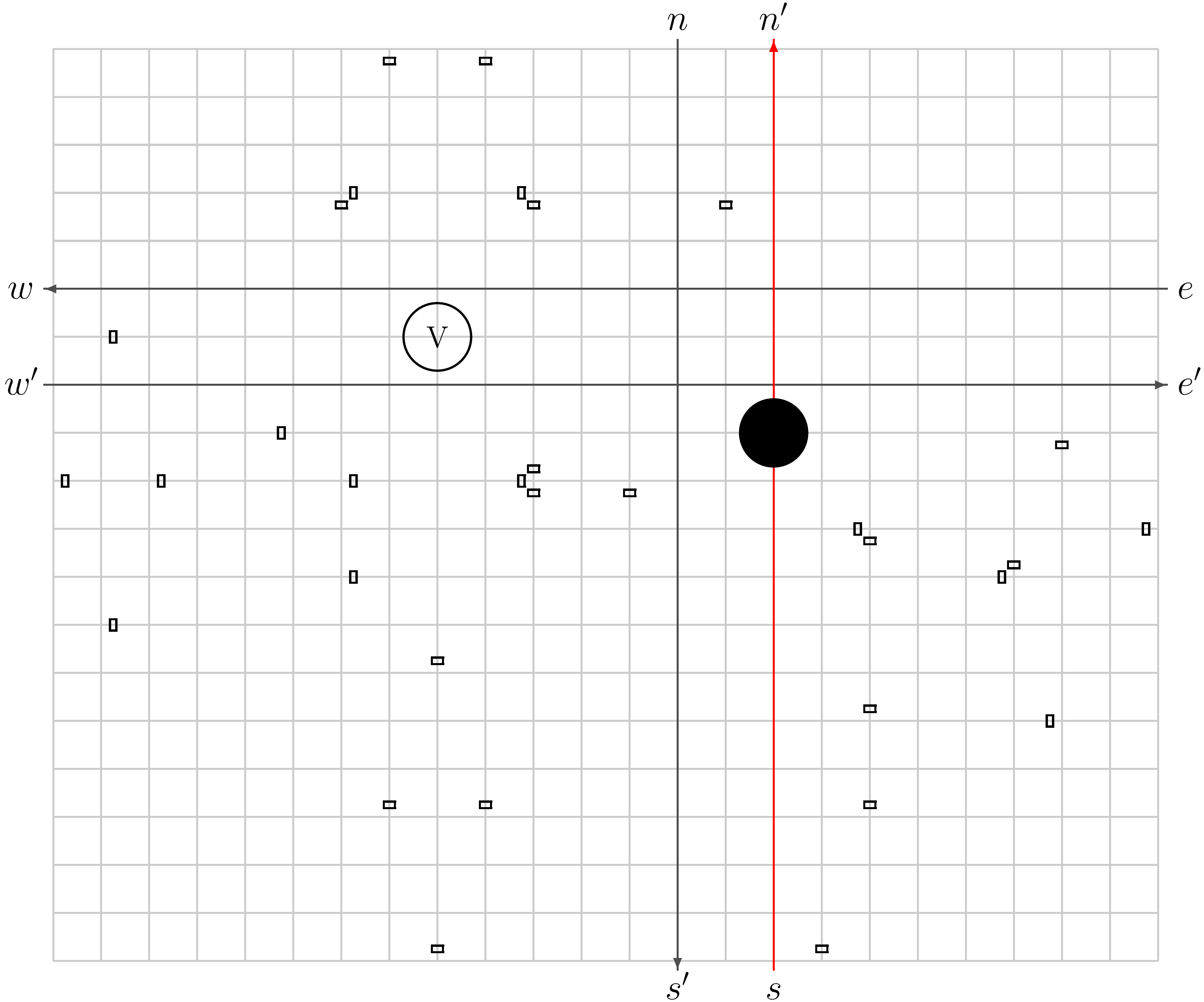
Movements of Balls (State: V, Input: s)



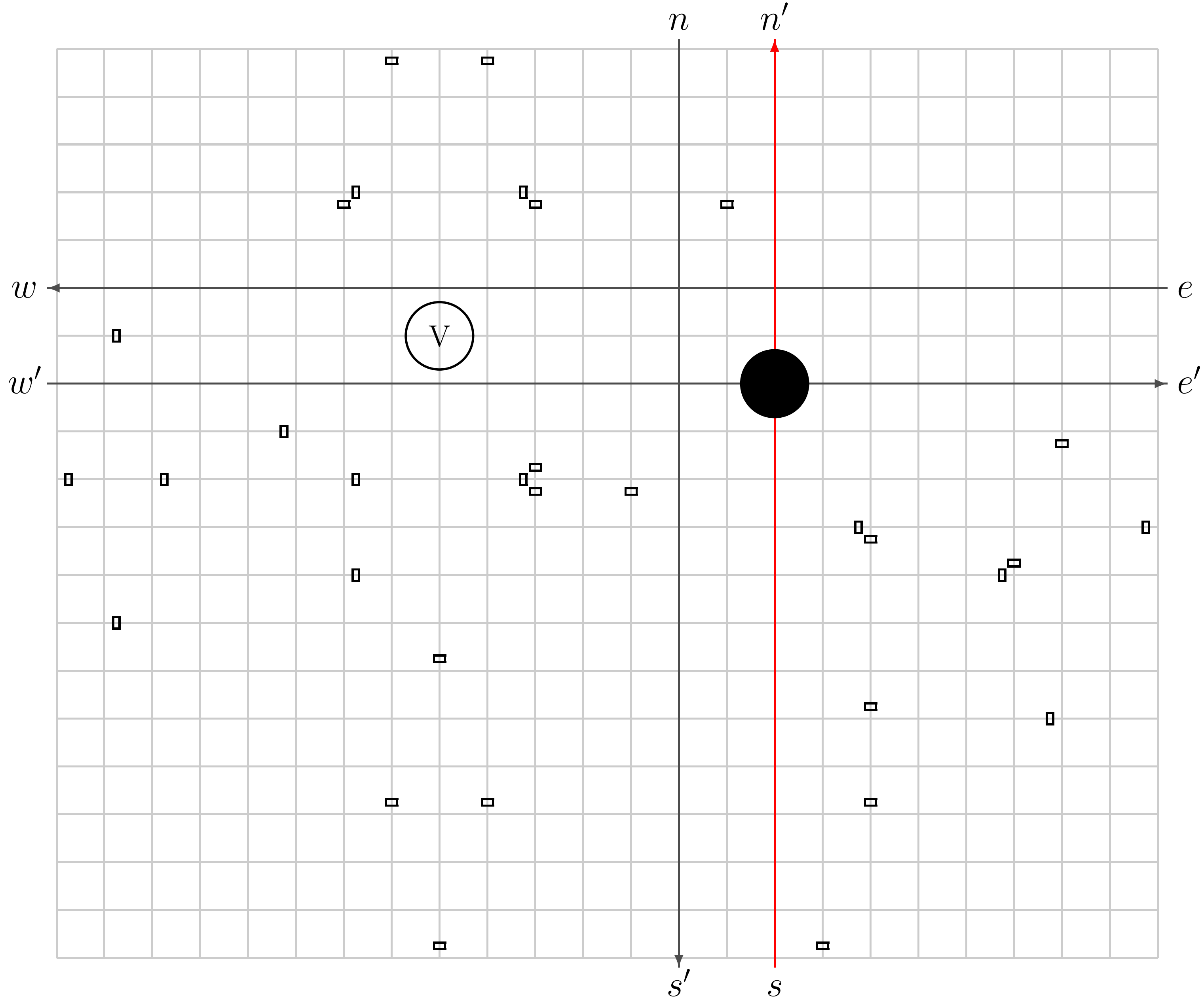
Movements of Balls (State: V, Input: s)



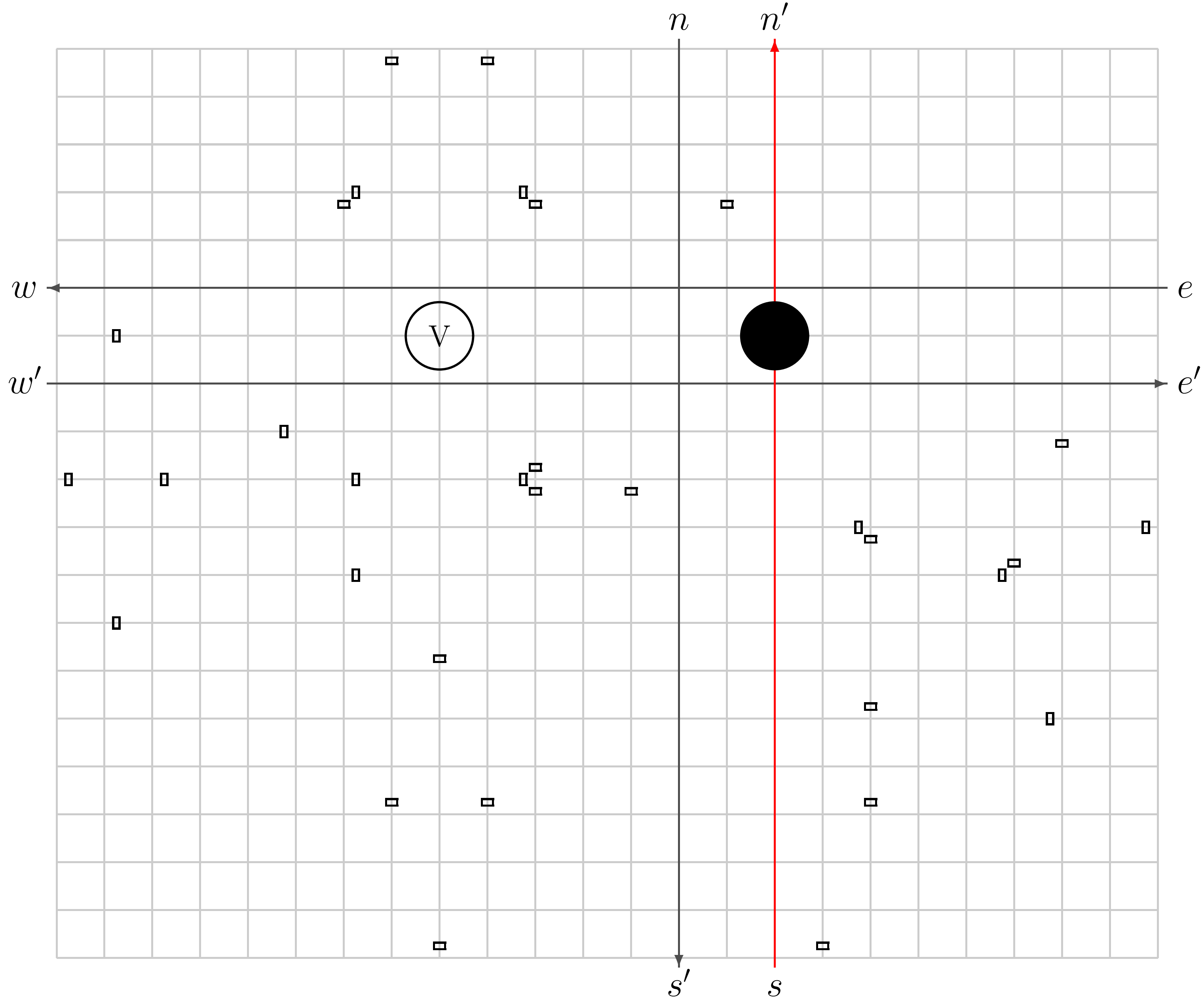
Movements of Balls (State: V, Input: s)



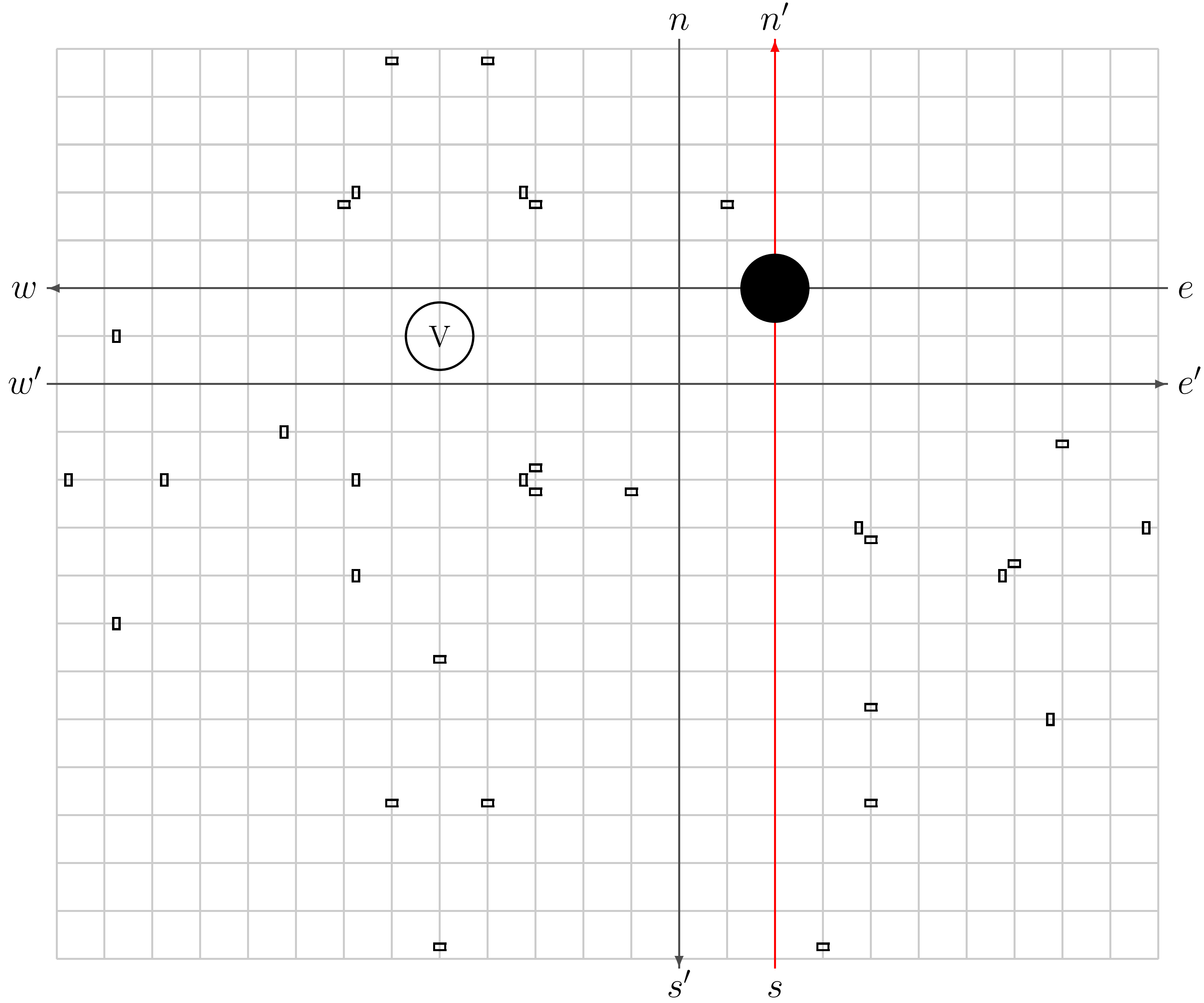
Movements of Balls (State: V, Input: s)



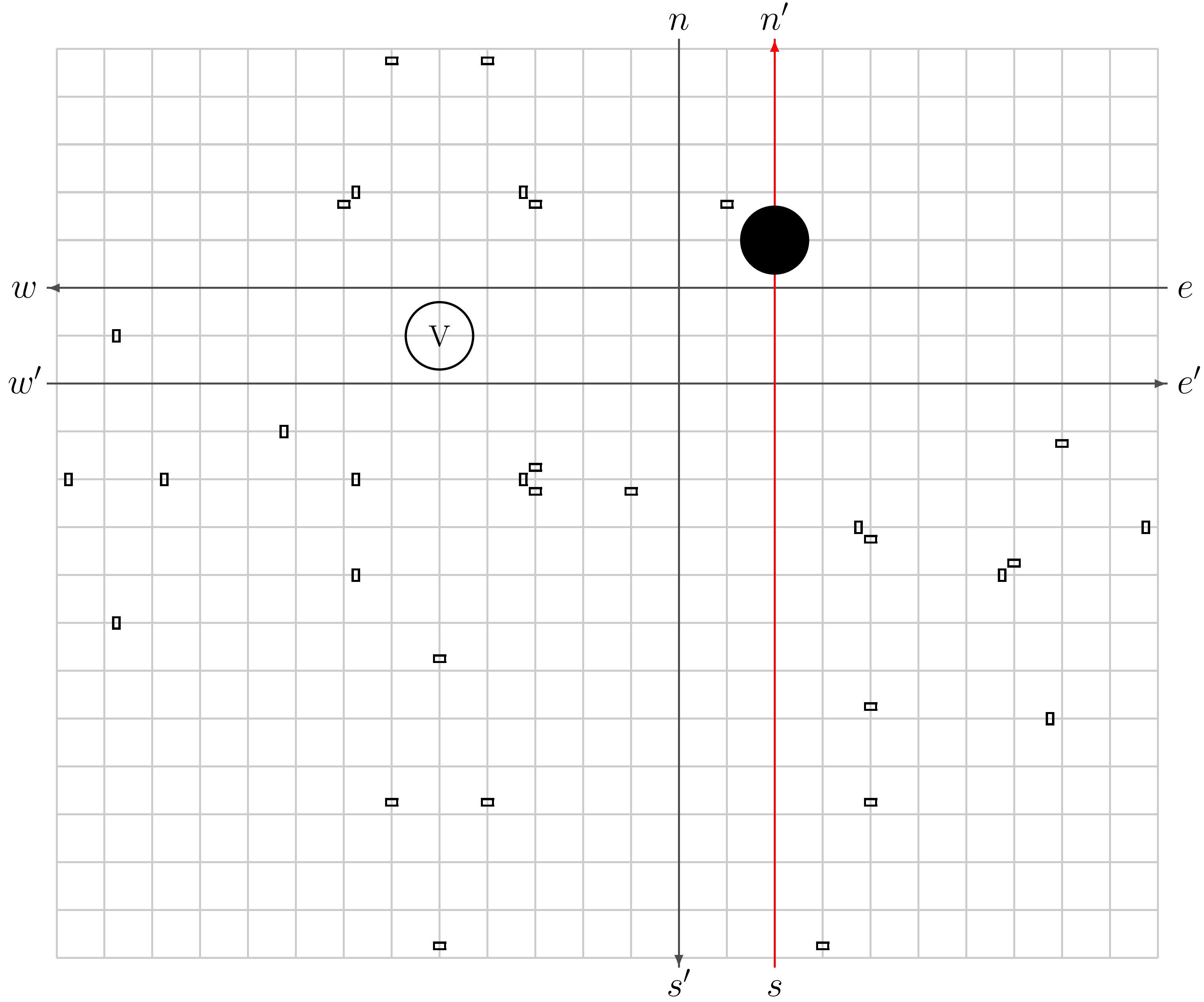
Movements of Balls (State: V, Input: s)



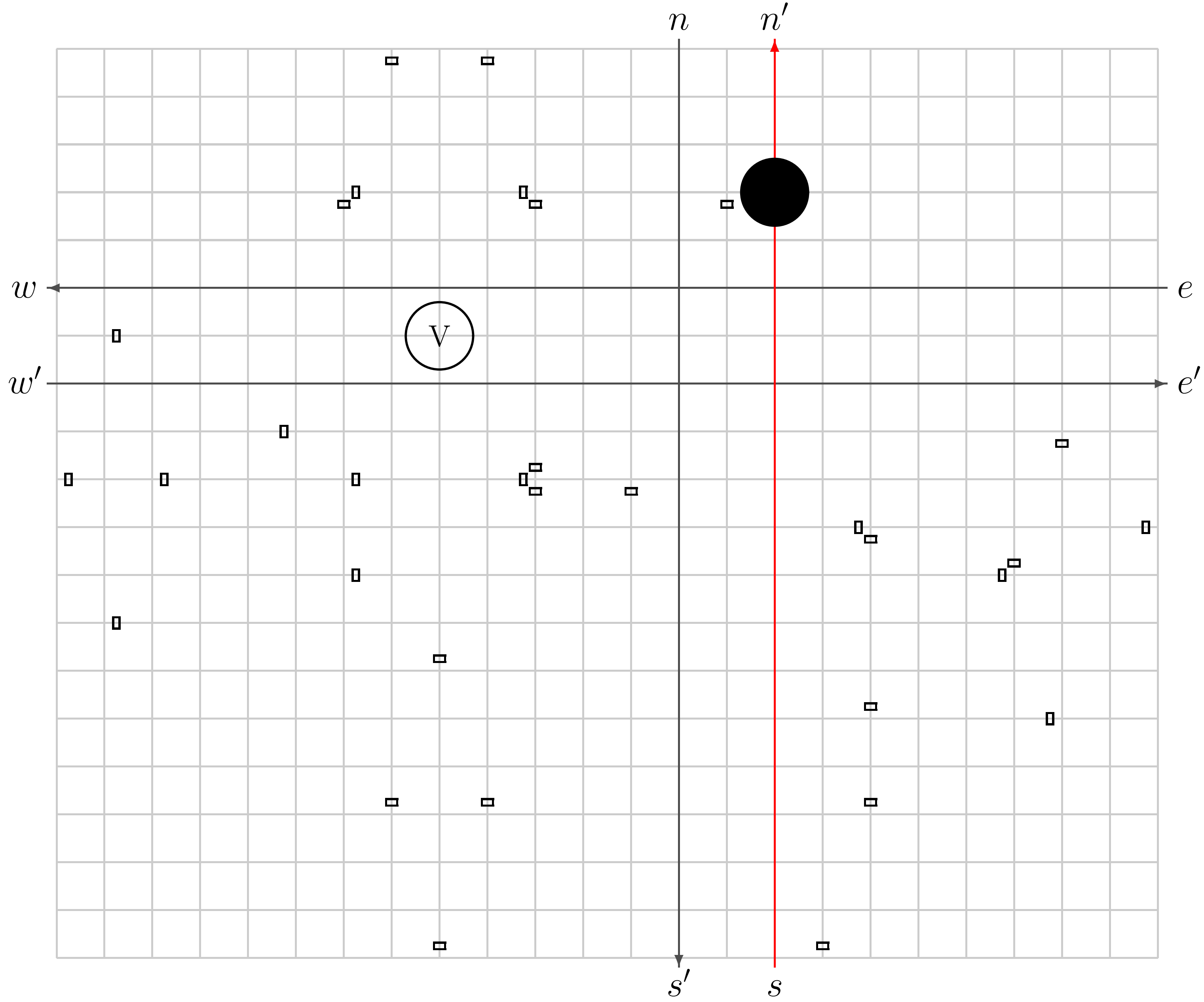
Movements of Balls (State: V, Input: s)



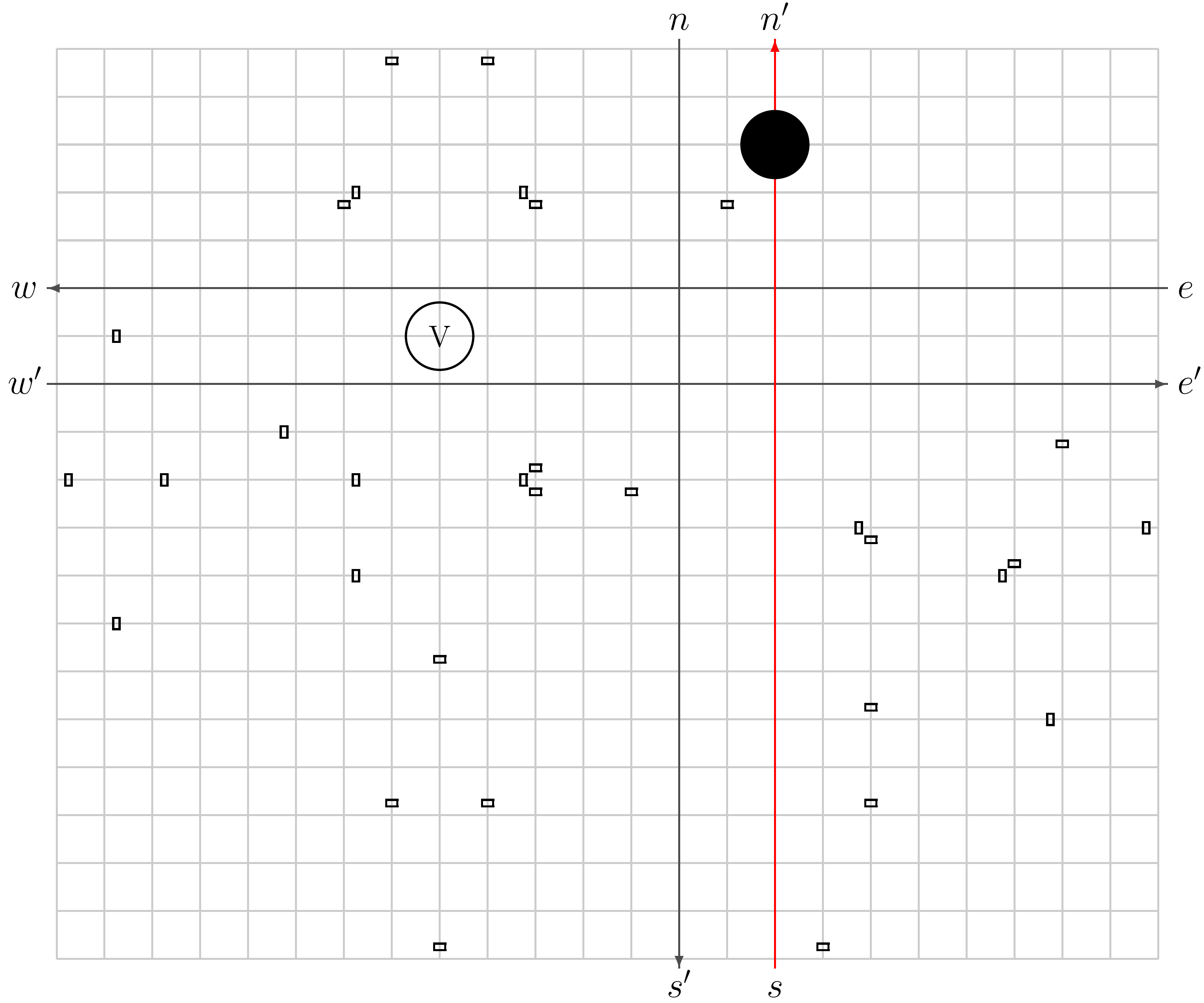
Movements of Balls (State: V, Input: s)



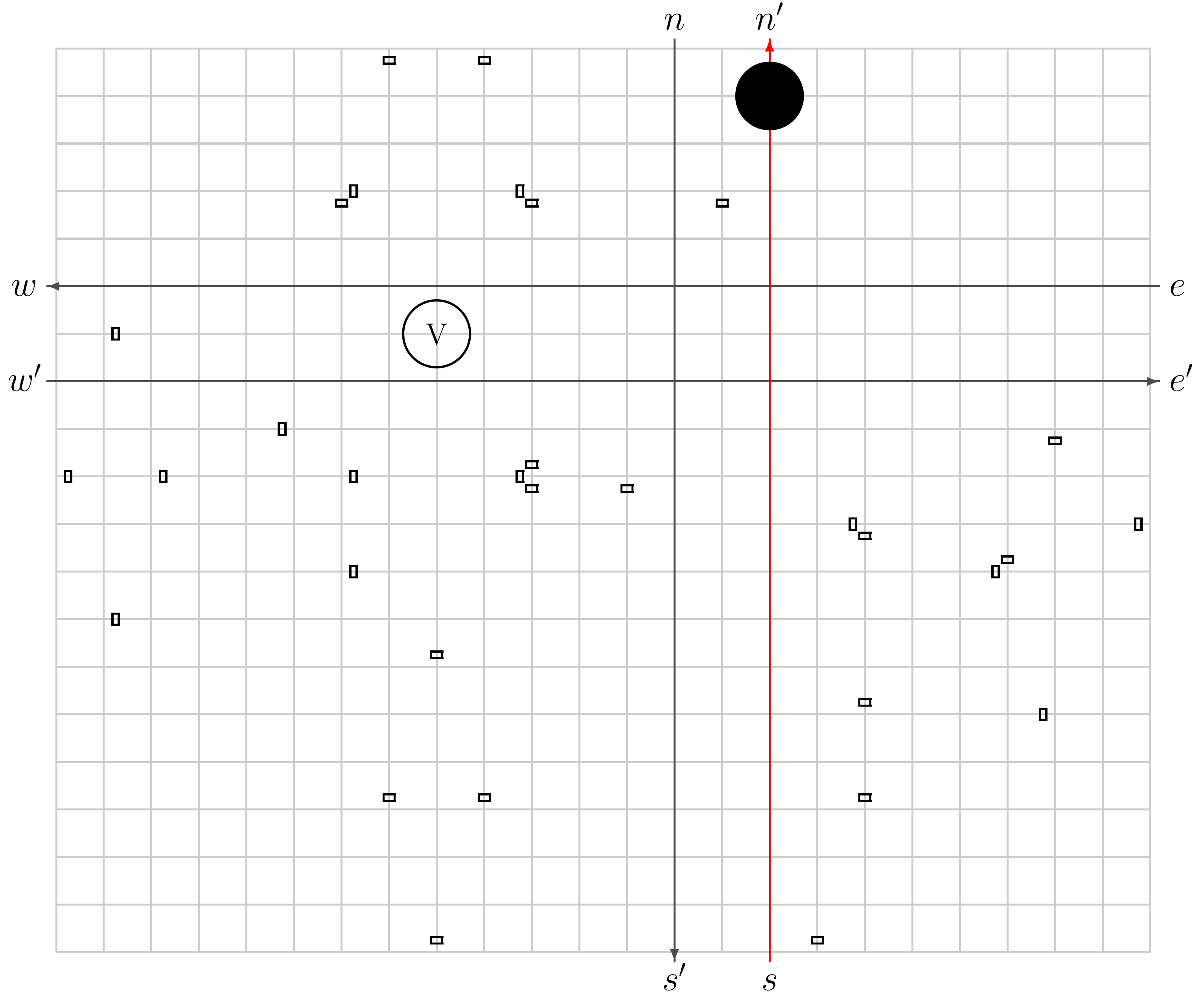
Movements of Balls (State: V, Input: s)



Movements of Balls (State: V, Input: s)

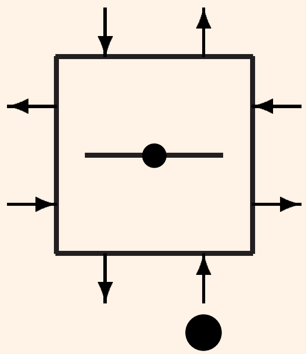


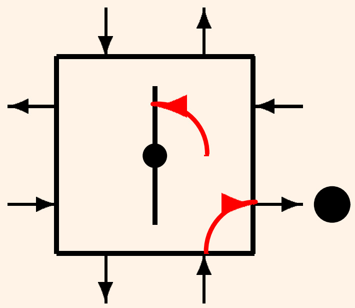
Movements of Balls (State: V, Input: s)



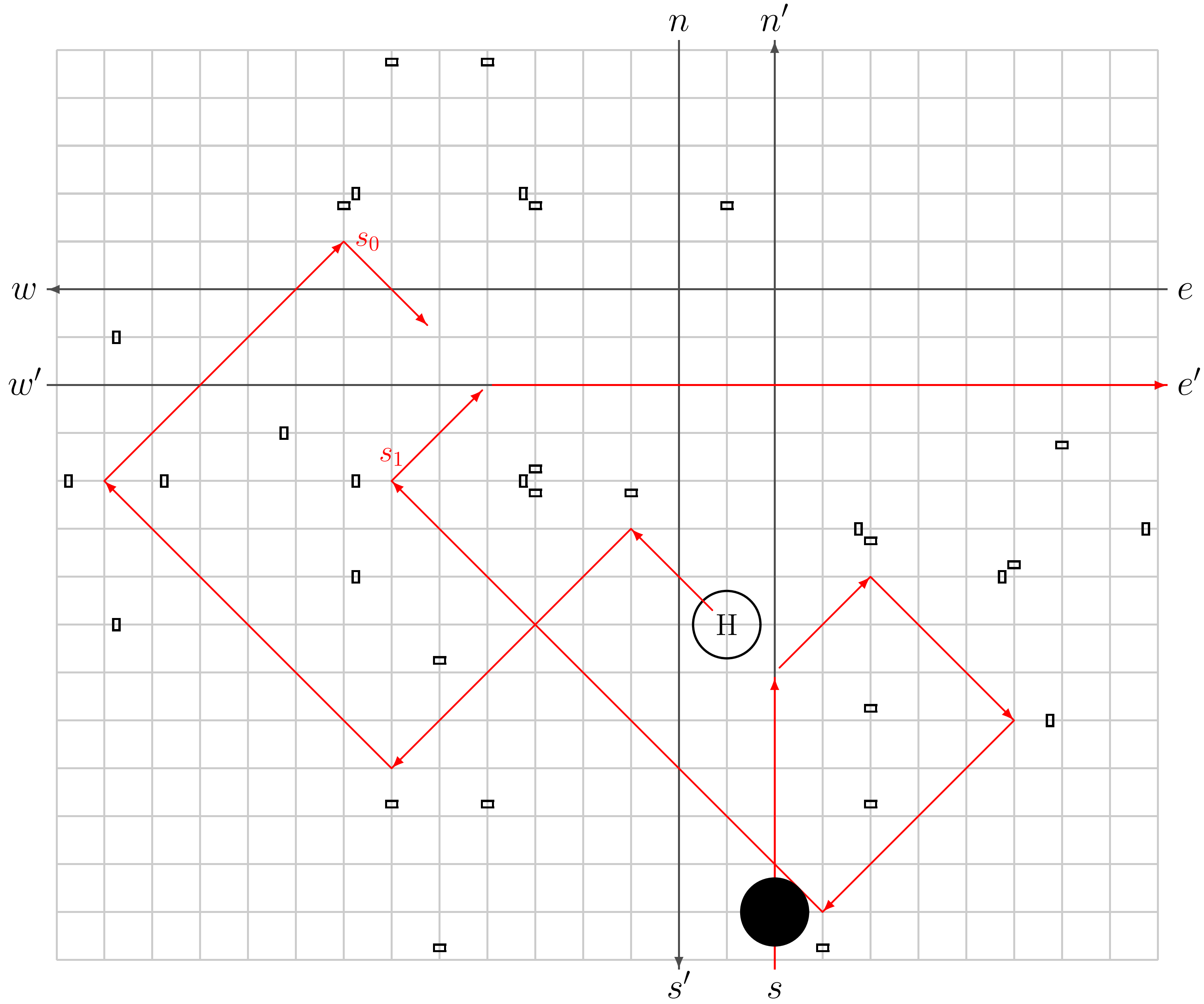
Orthogonal Case

$t = 0$ $t = 1$

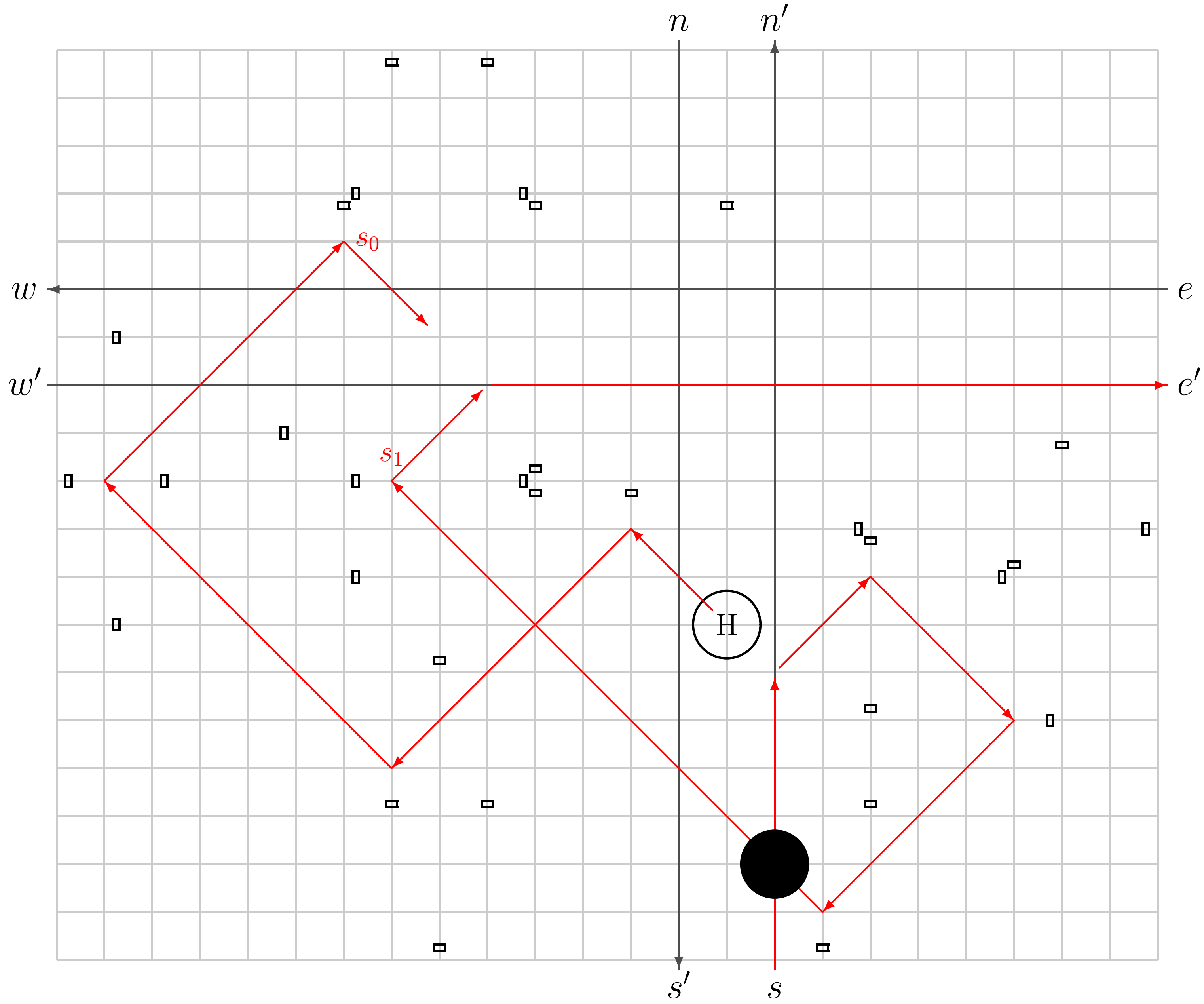




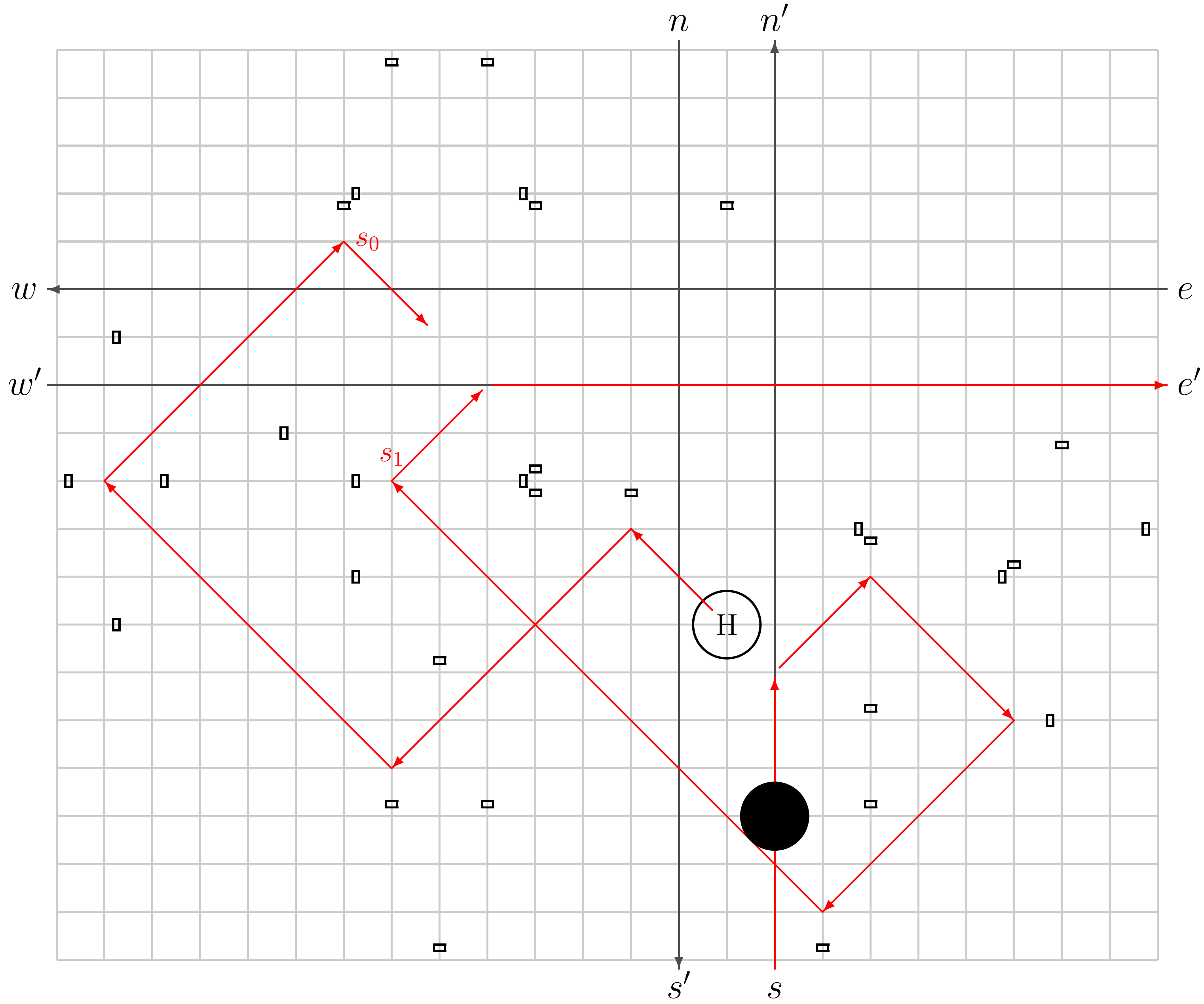
Movements of Balls (State: H, Input: s)



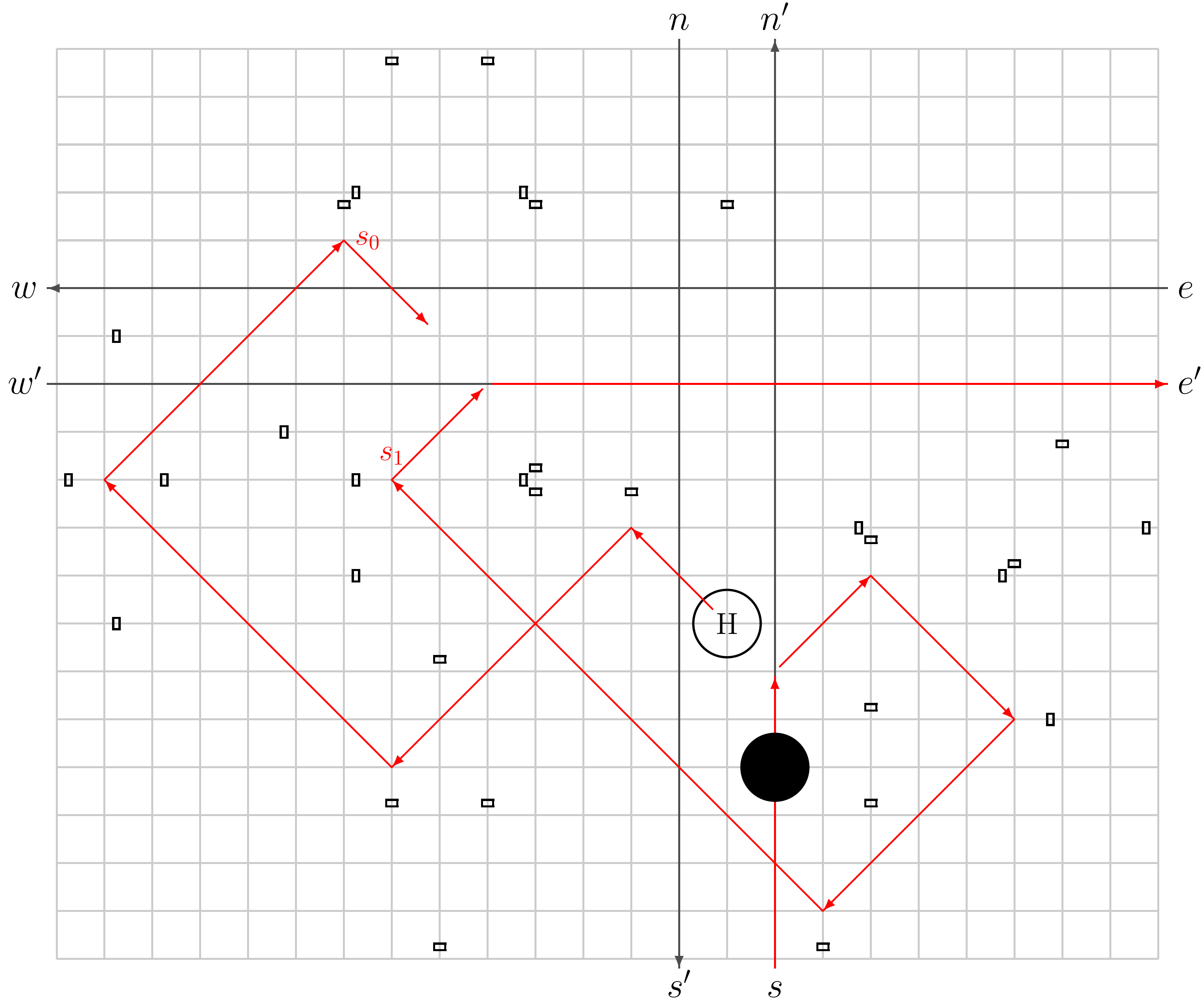
Movements of Balls (State: H, Input: s)



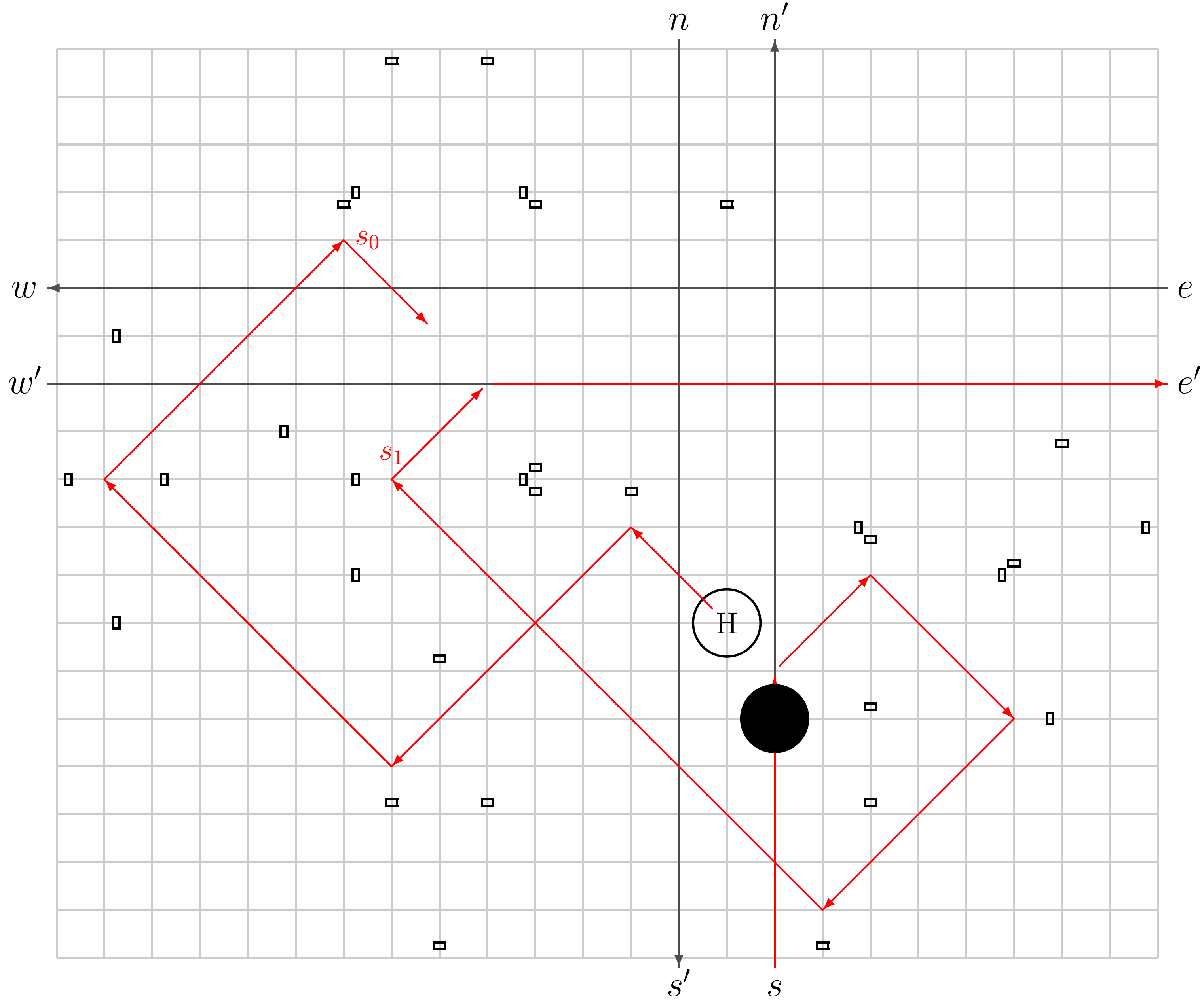
Movements of Balls (State: H, Input: s)



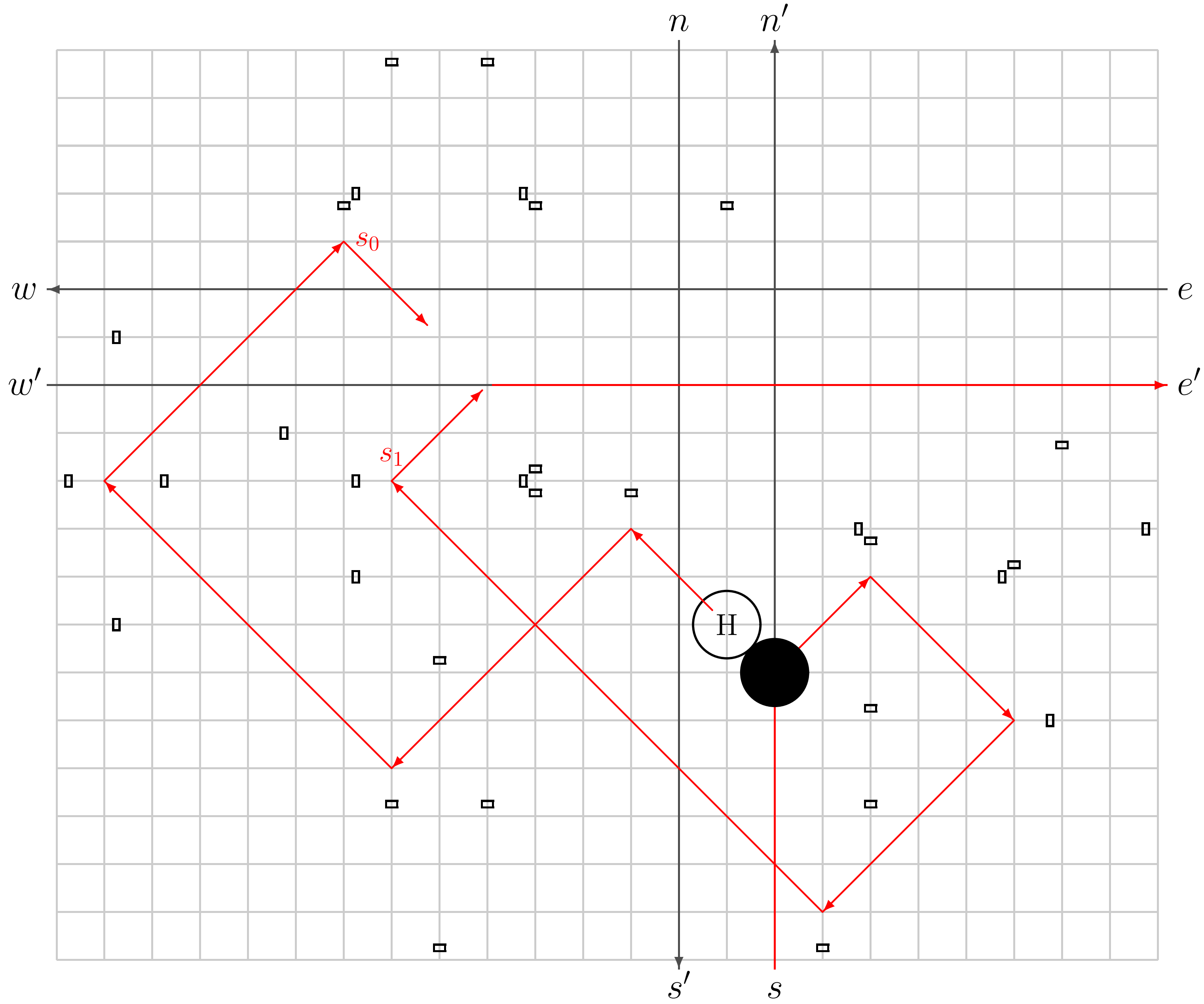
Movements of Balls (State: H, Input: s)



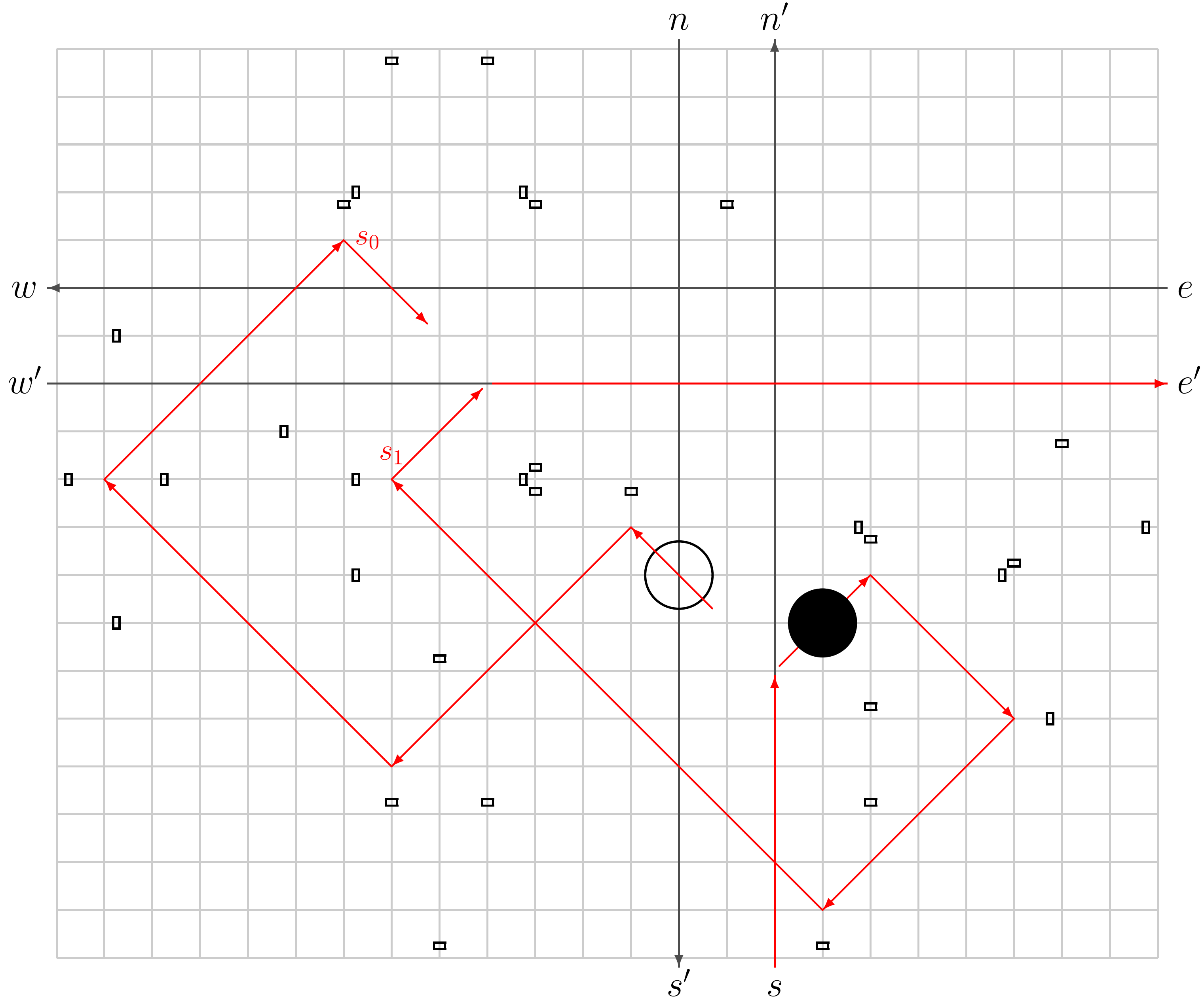
Movements of Balls (State: H, Input: s)



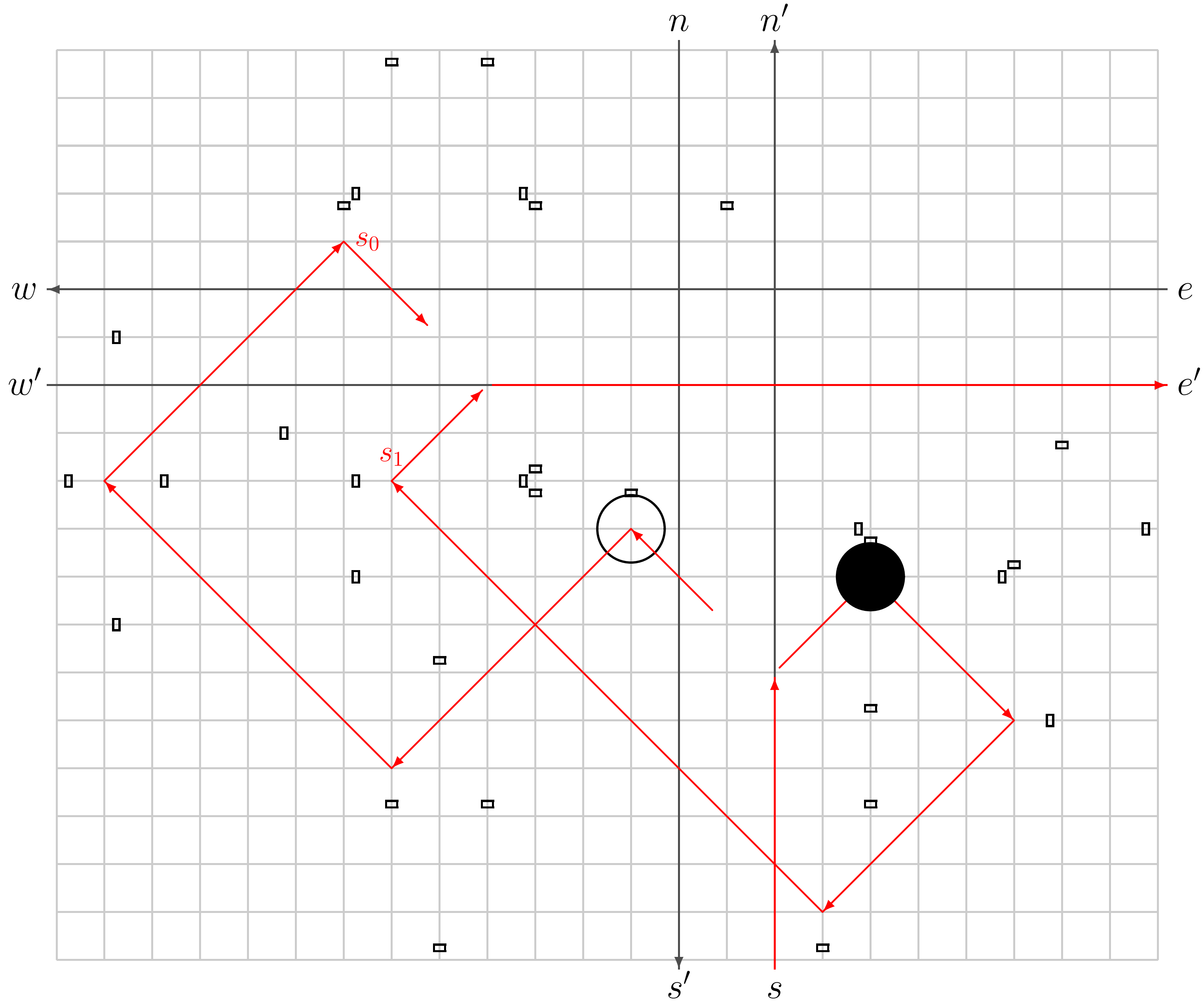
Movements of Balls (State: H, Input: s)



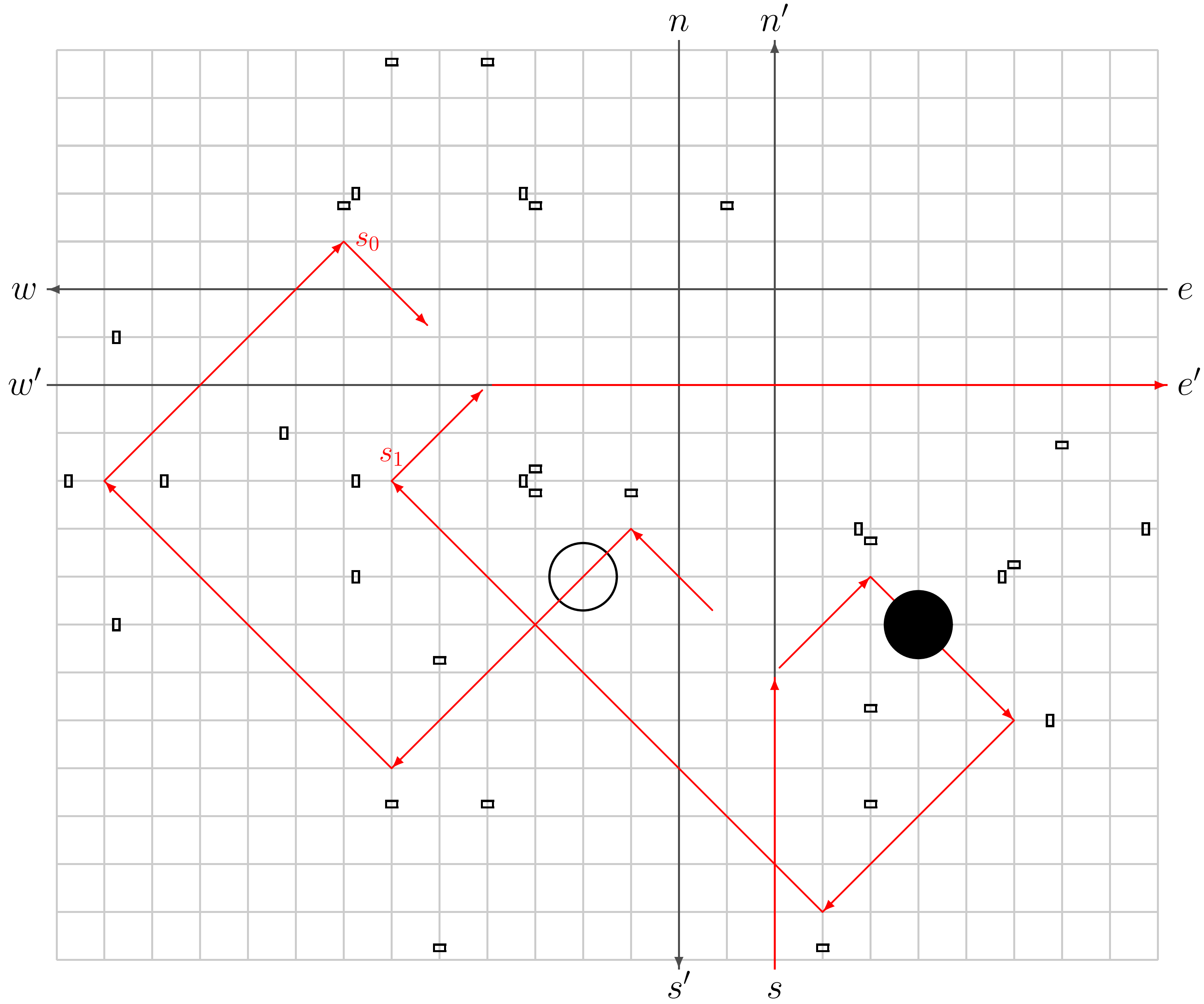
Movements of Balls (State: H, Input: s)



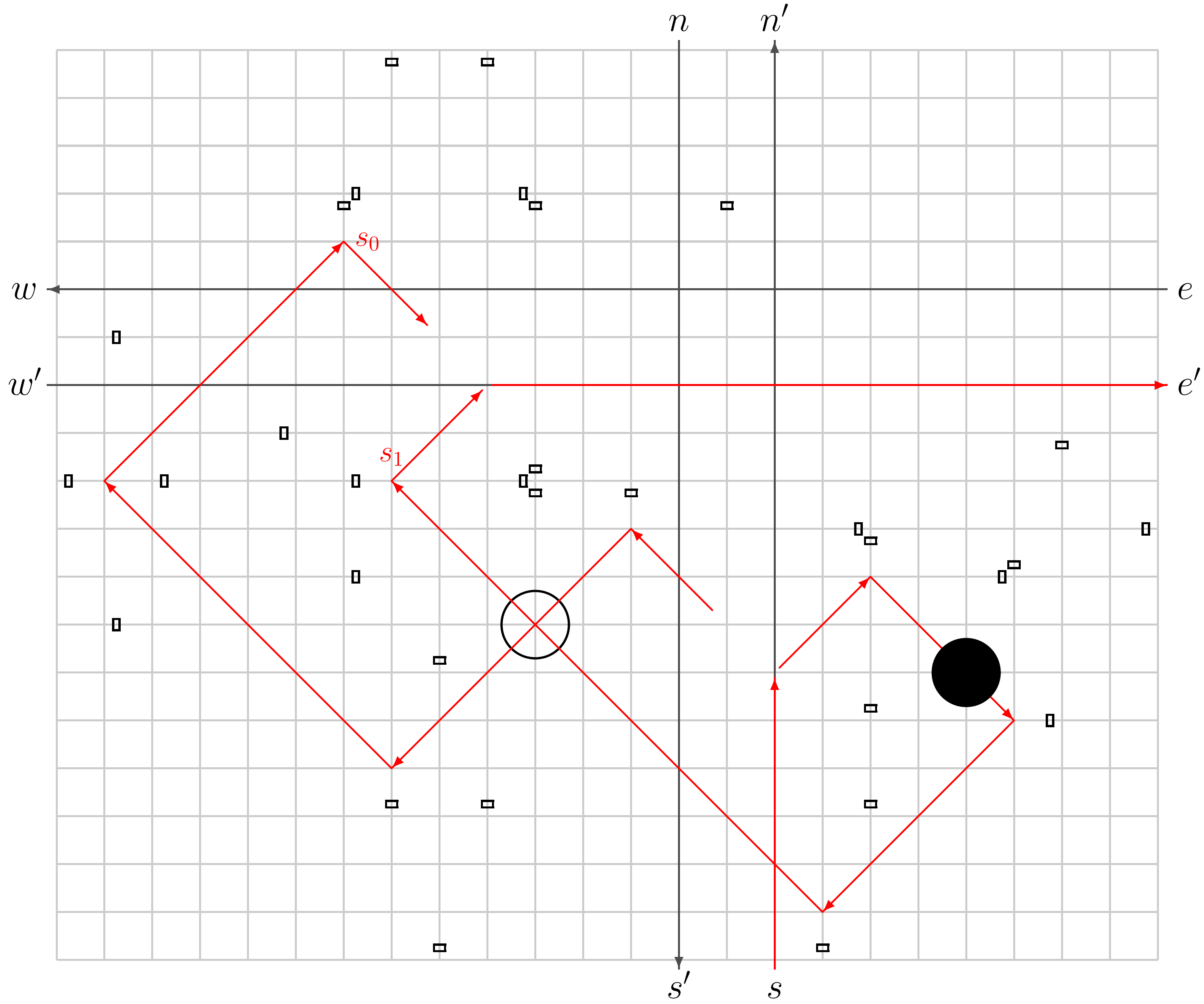
Movements of Balls (State: H, Input: s)



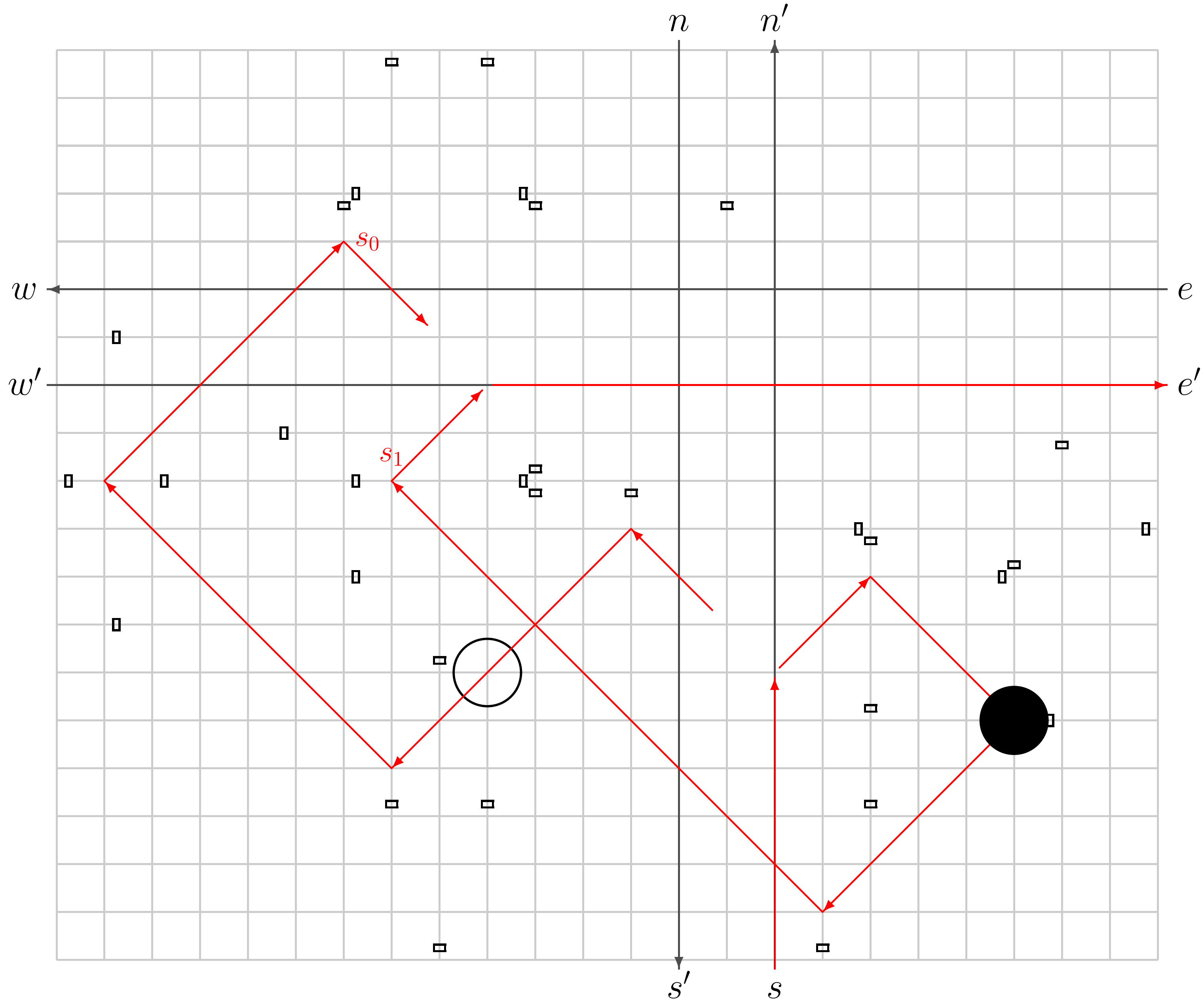
Movements of Balls (State: H, Input: s)



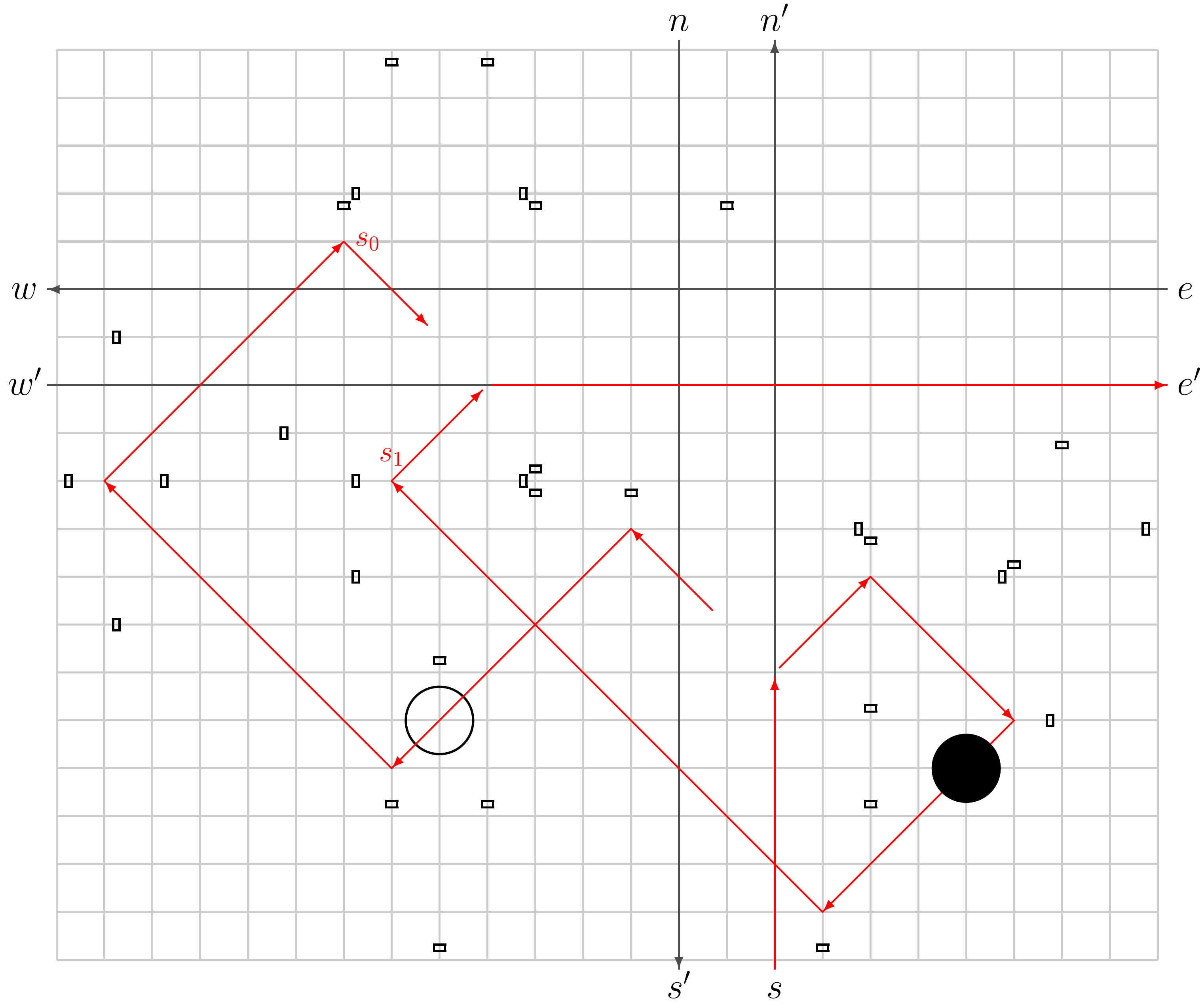
Movements of Balls (State: H, Input: s)



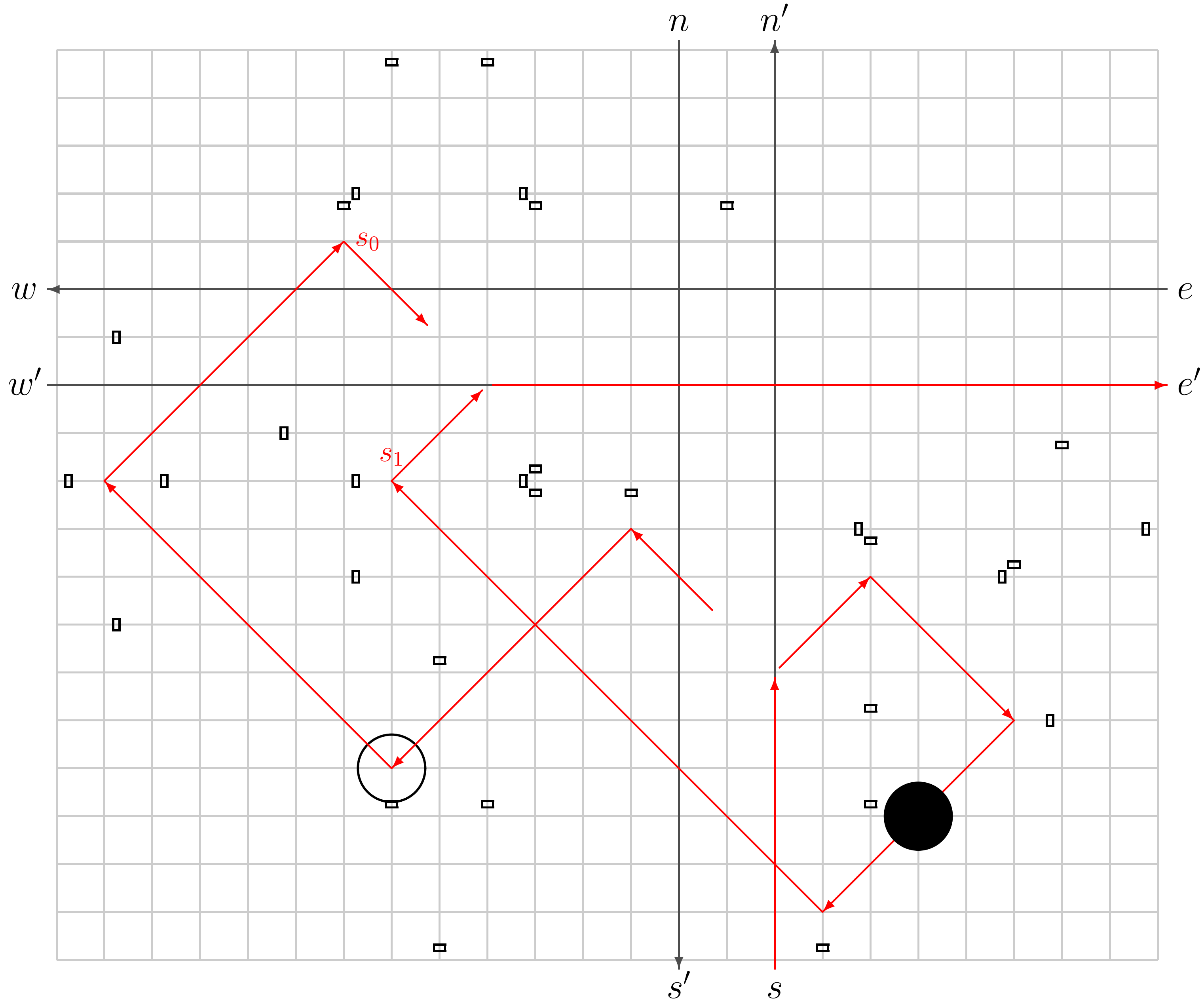
Movements of Balls (State: H, Input: s)



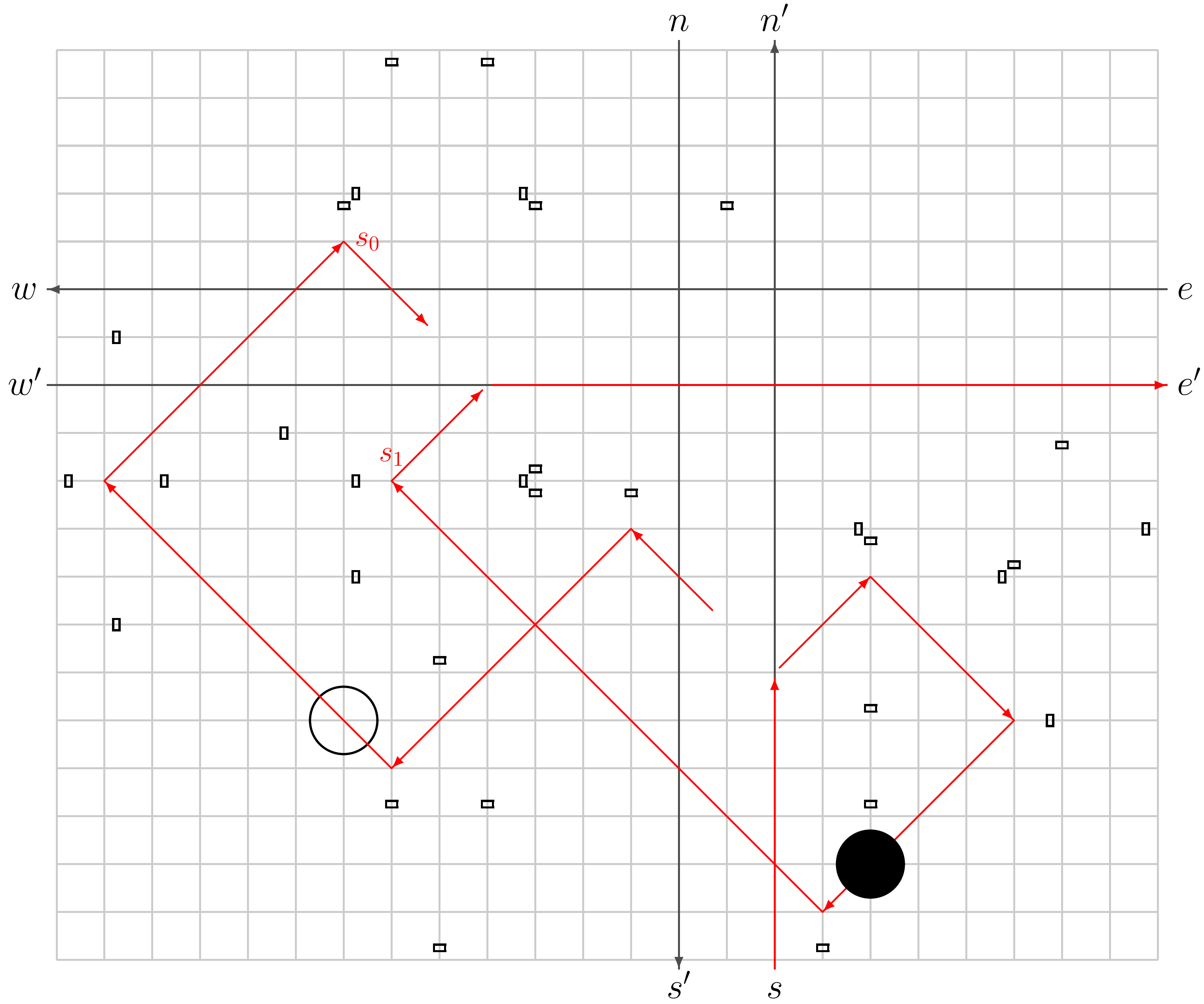
Movements of Balls (State: H, Input: s)



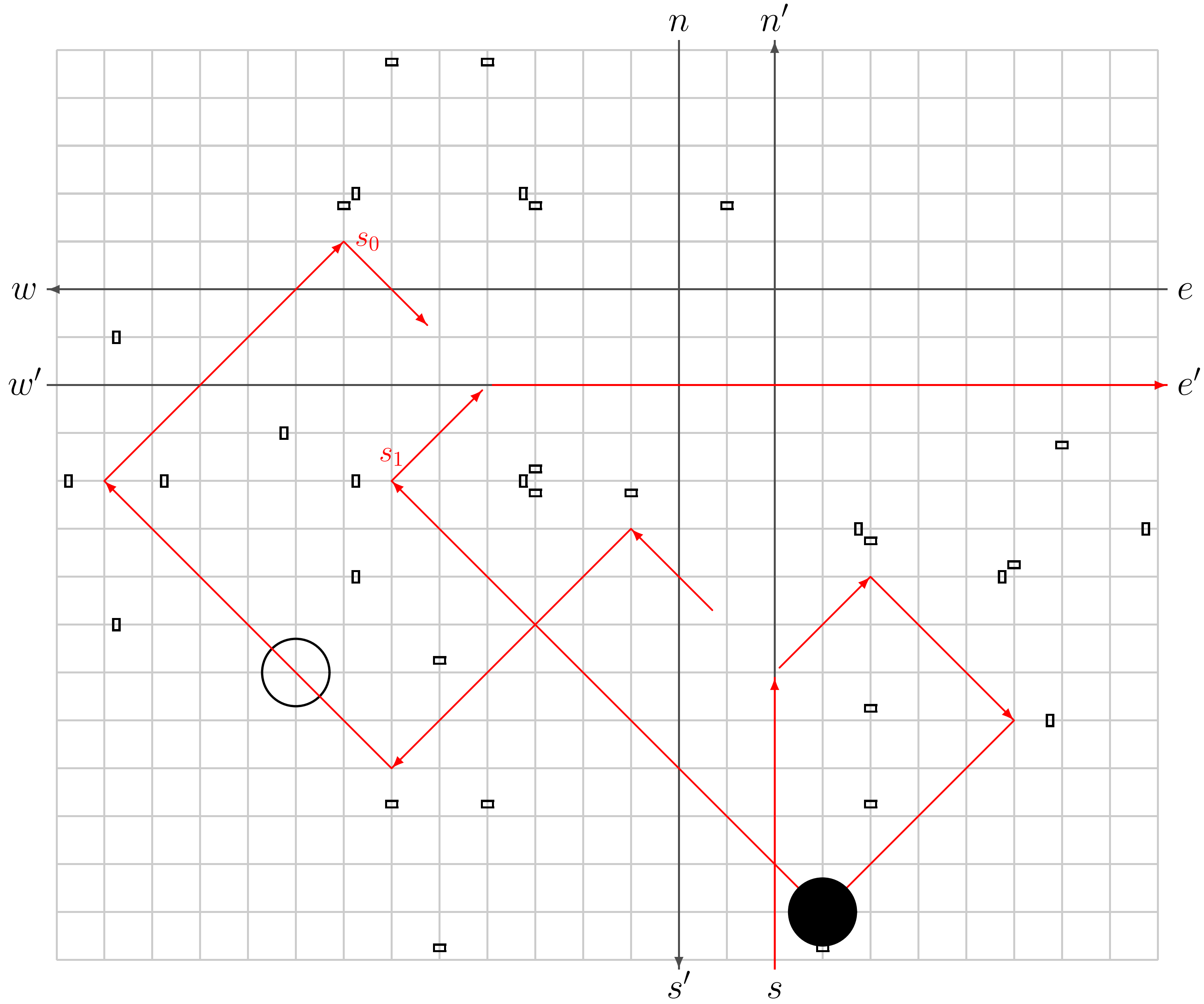
Movements of Balls (State: H, Input: s)



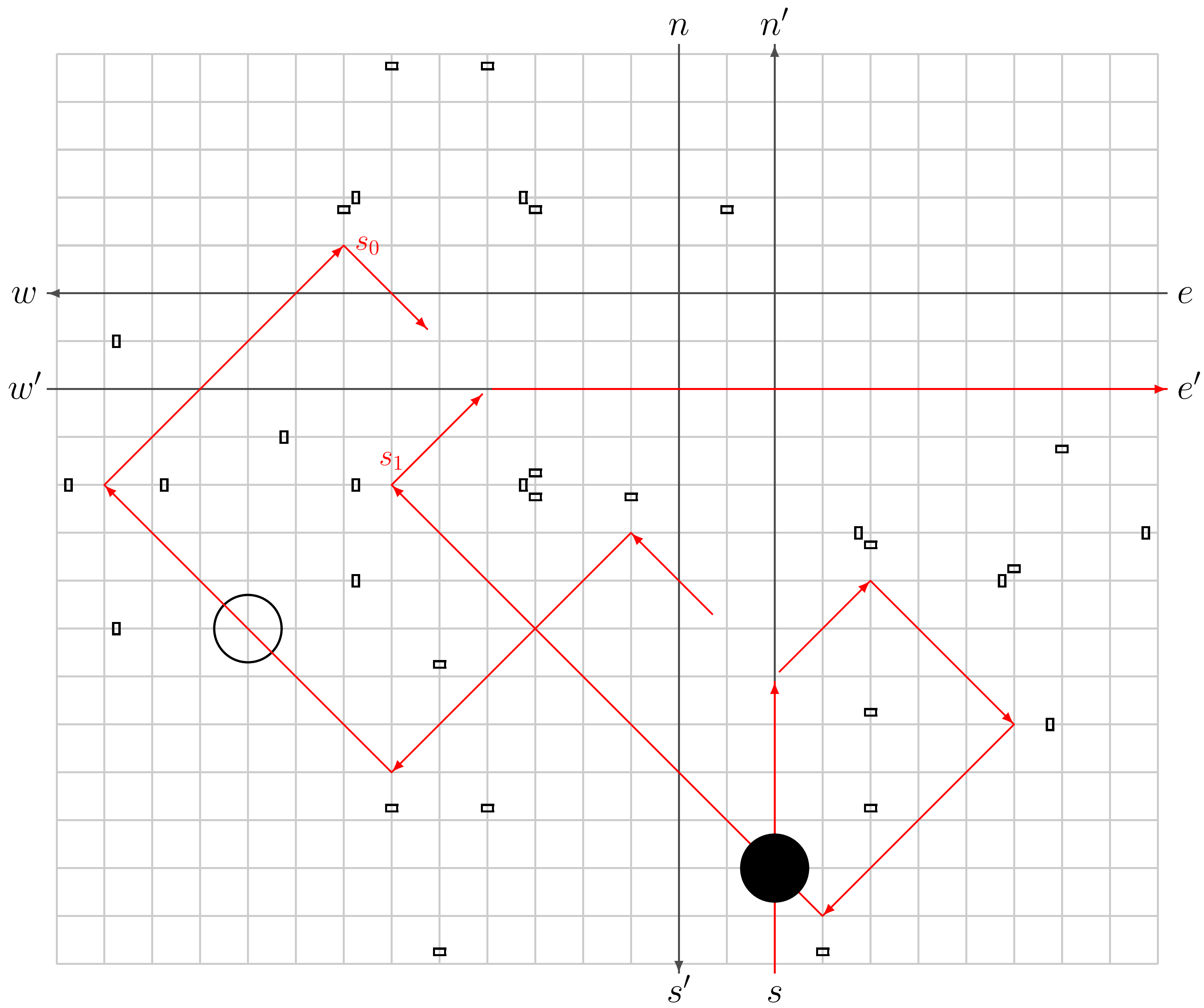
Movements of Balls (State: H, Input: s)



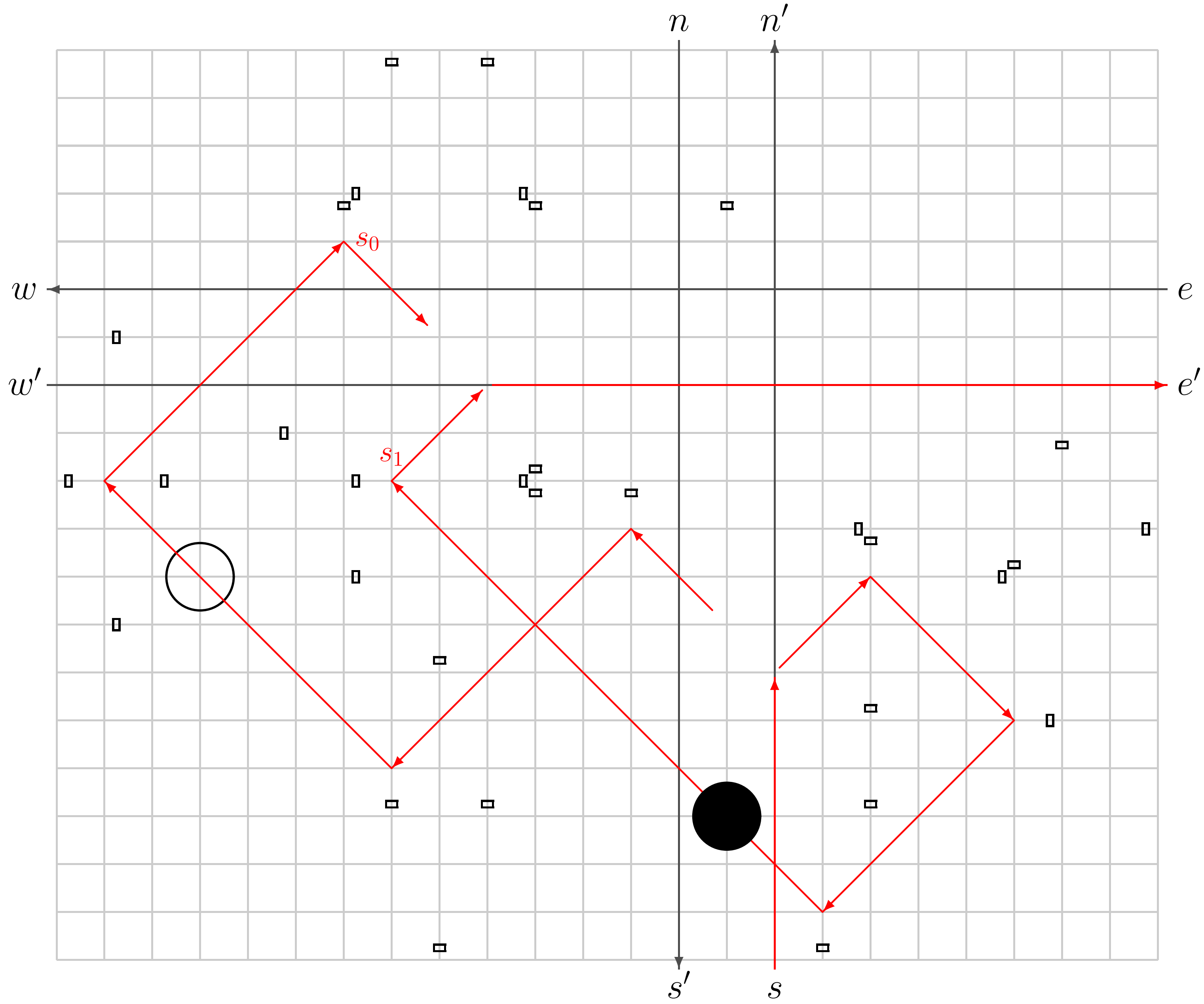
Movements of Balls (State: H, Input: s)



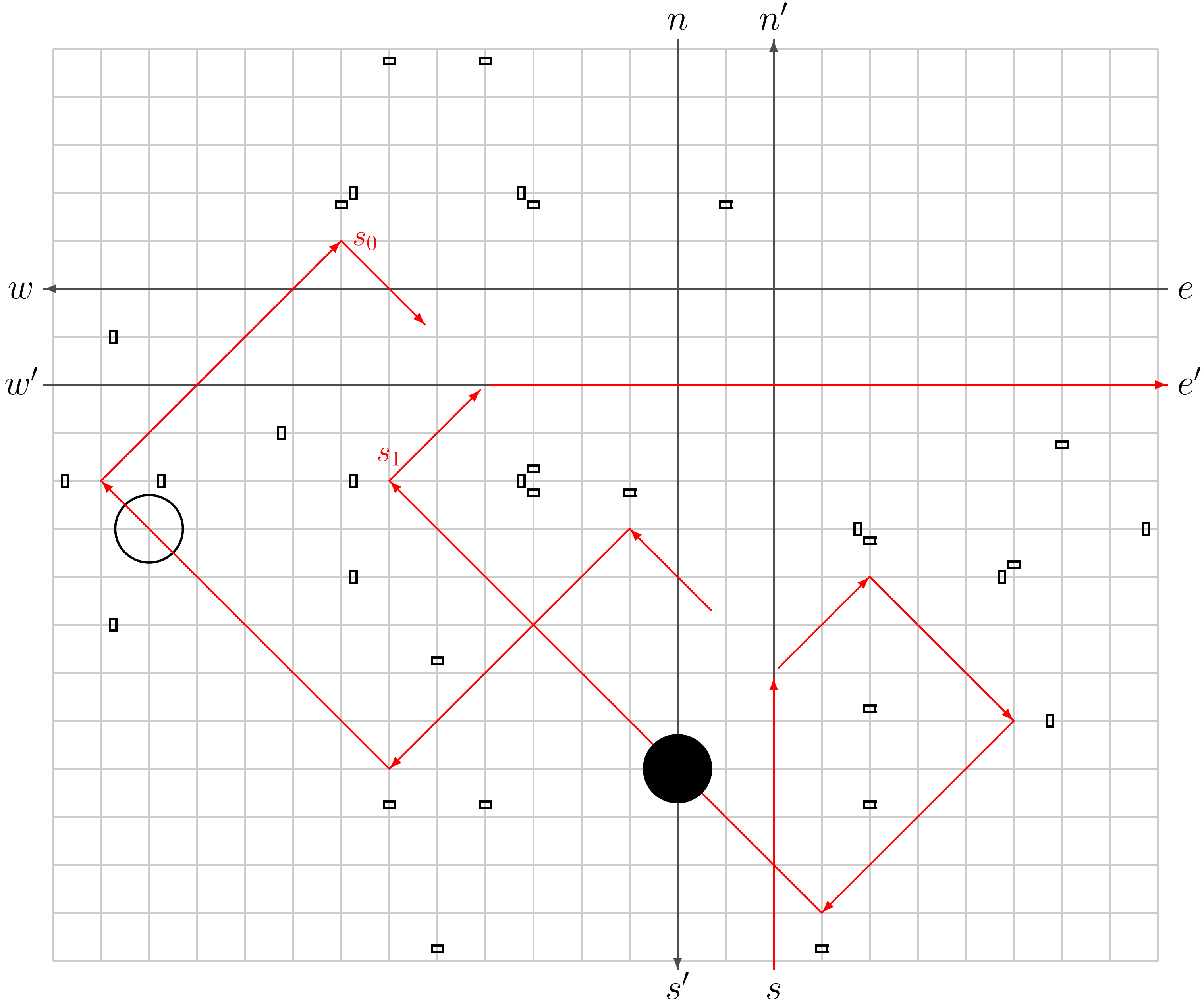
Movements of Balls (State: H, Input: s)



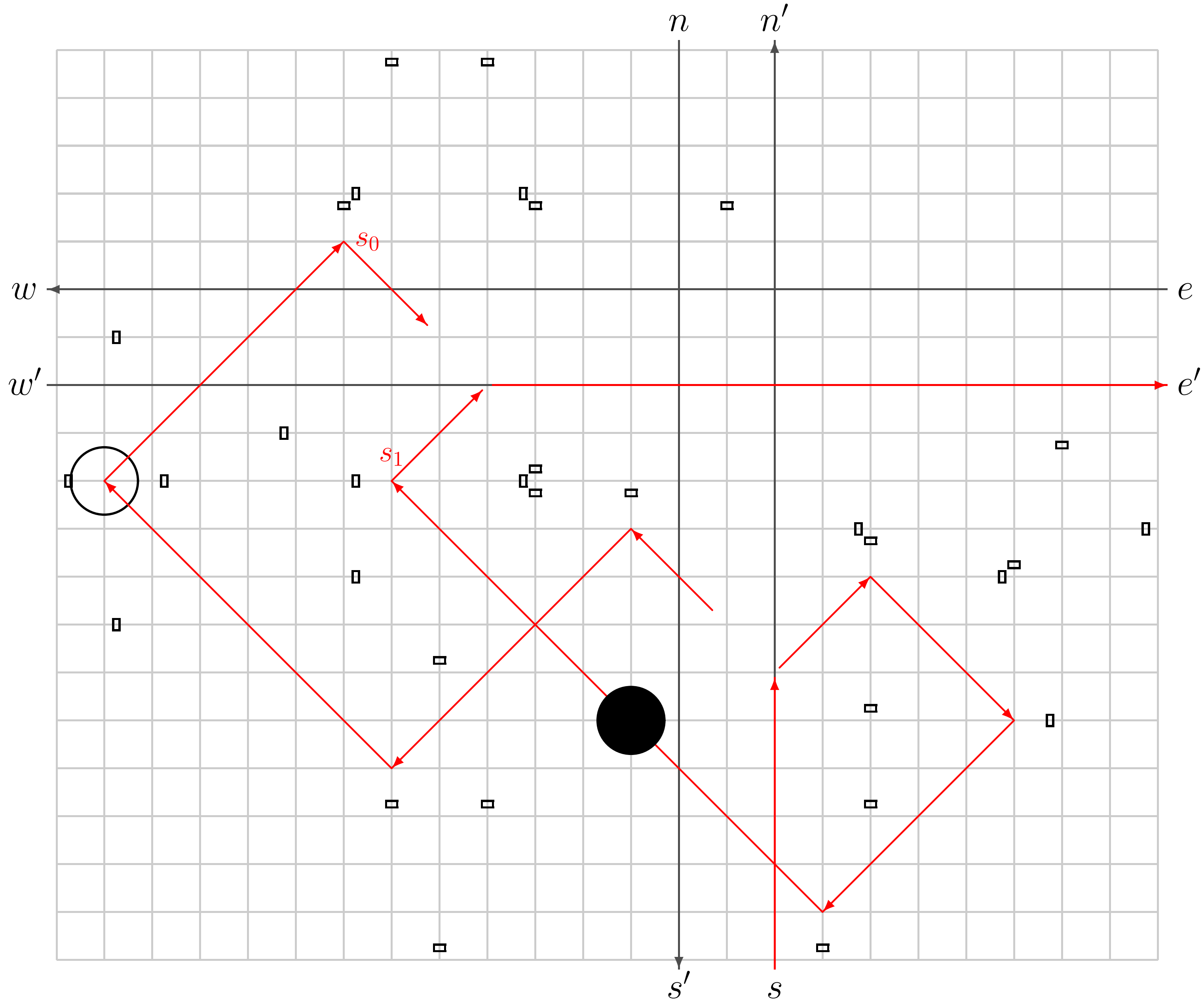
Movements of Balls (State: H, Input: s)



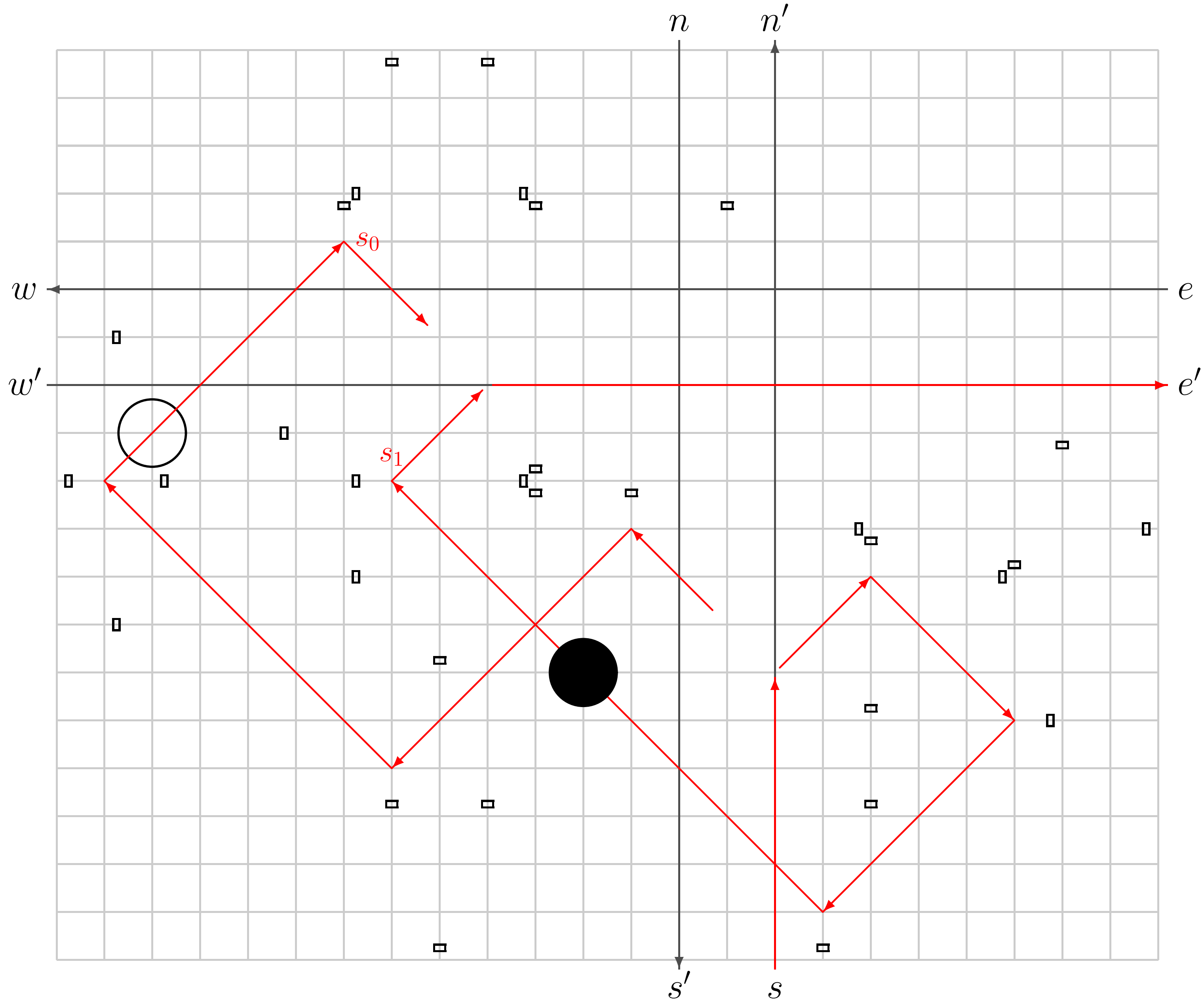
Movements of Balls (State: H, Input: s)



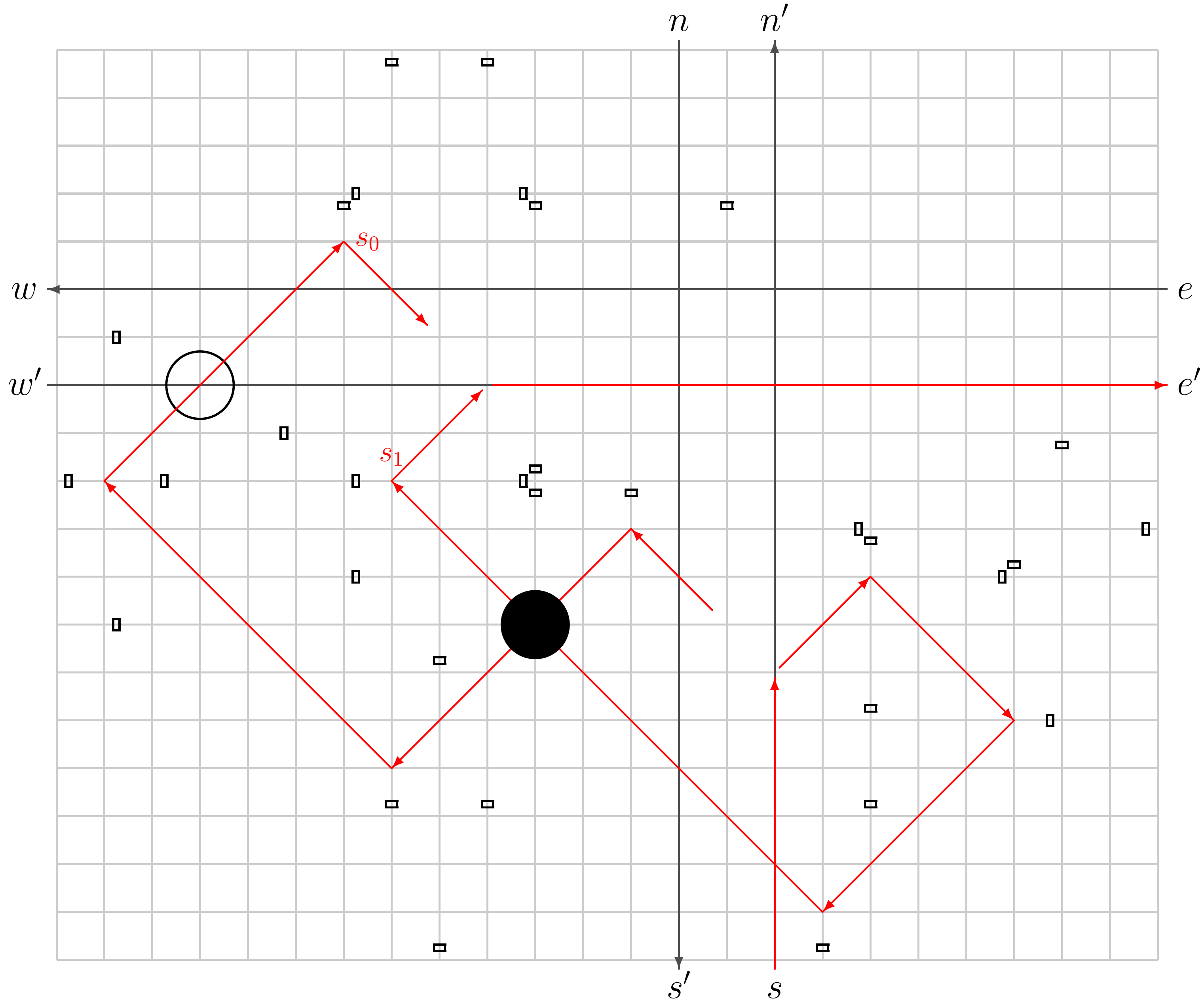
Movements of Balls (State: H, Input: s)



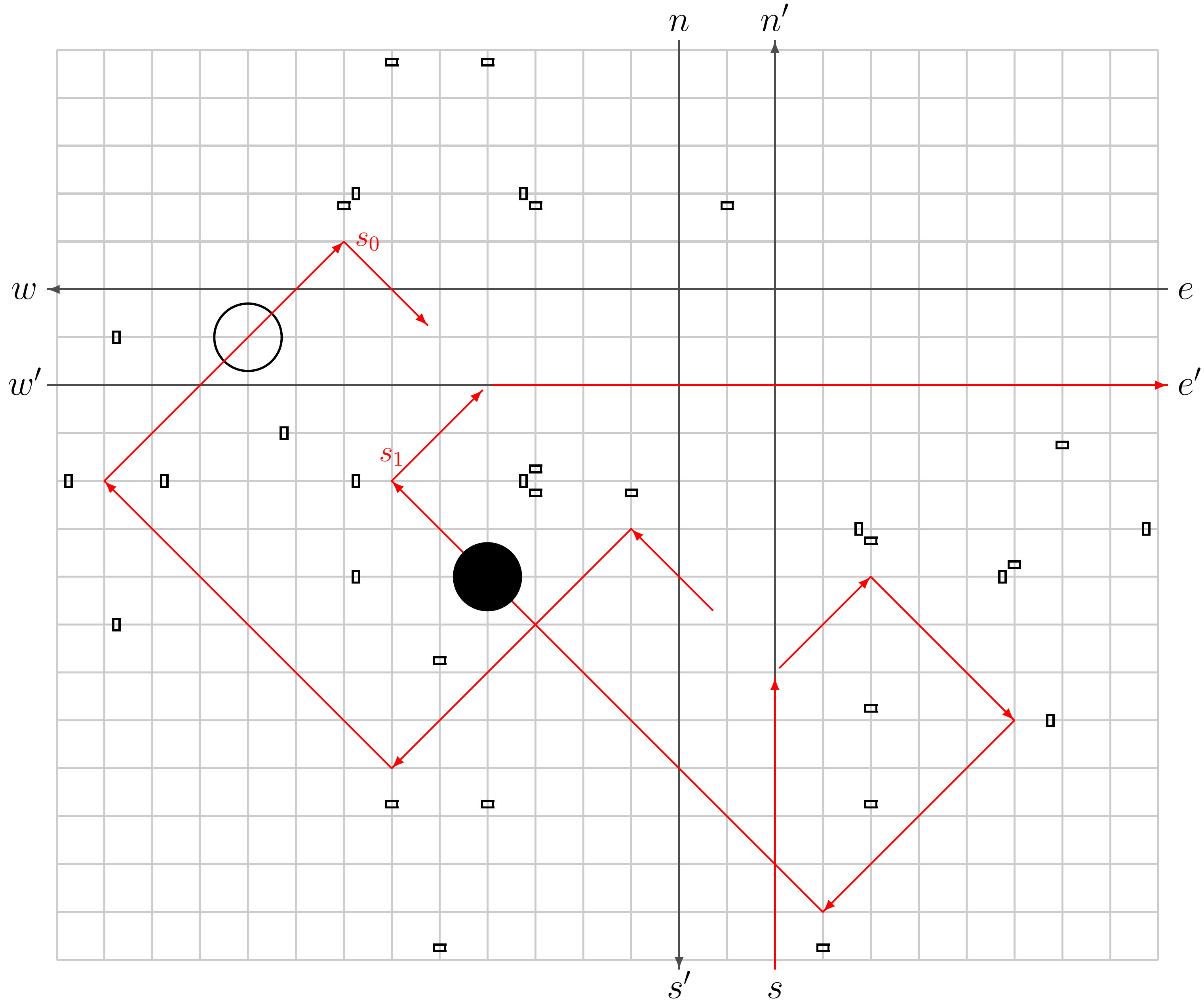
Movements of Balls (State: H, Input: s)



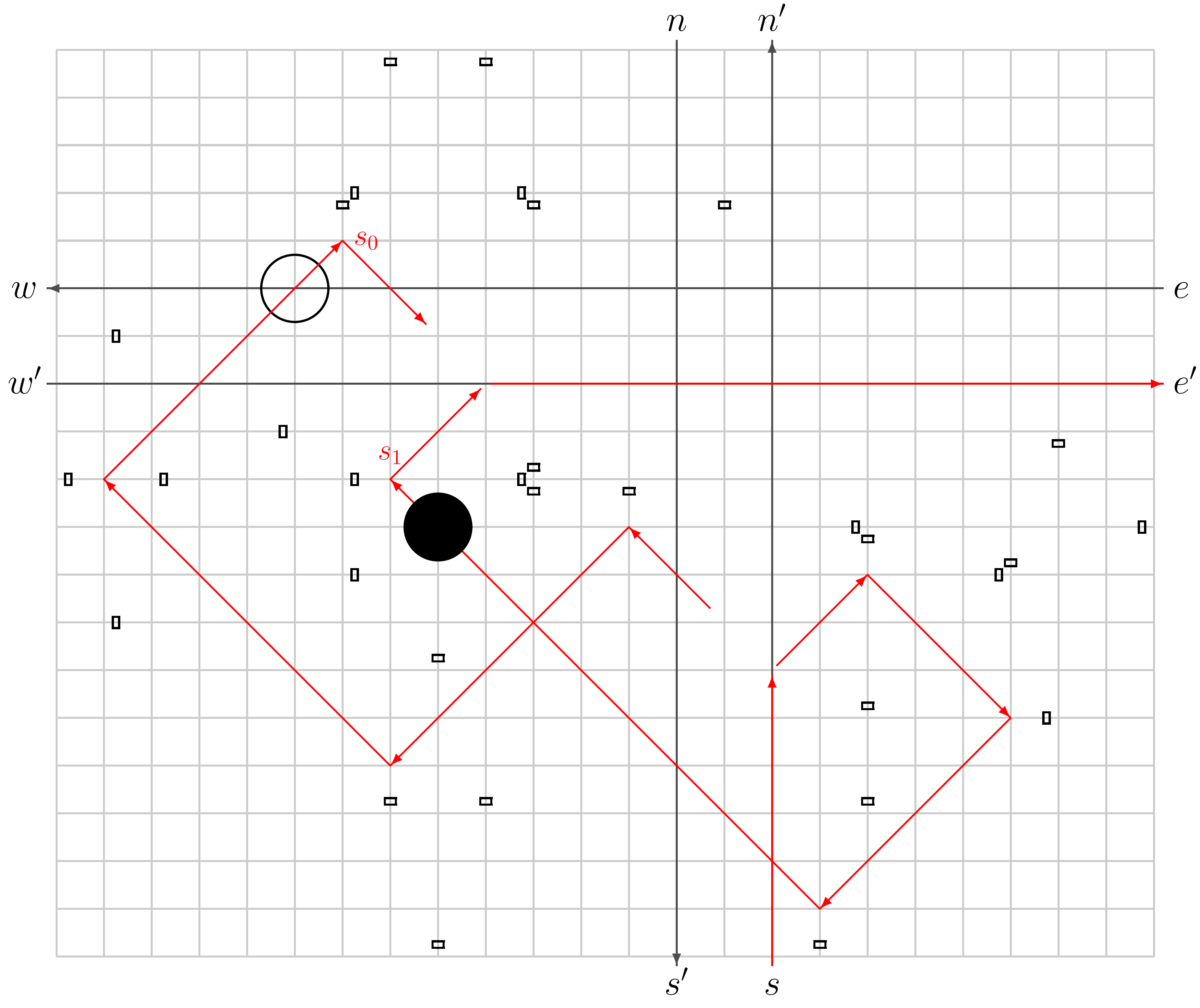
Movements of Balls (State: H, Input: s)



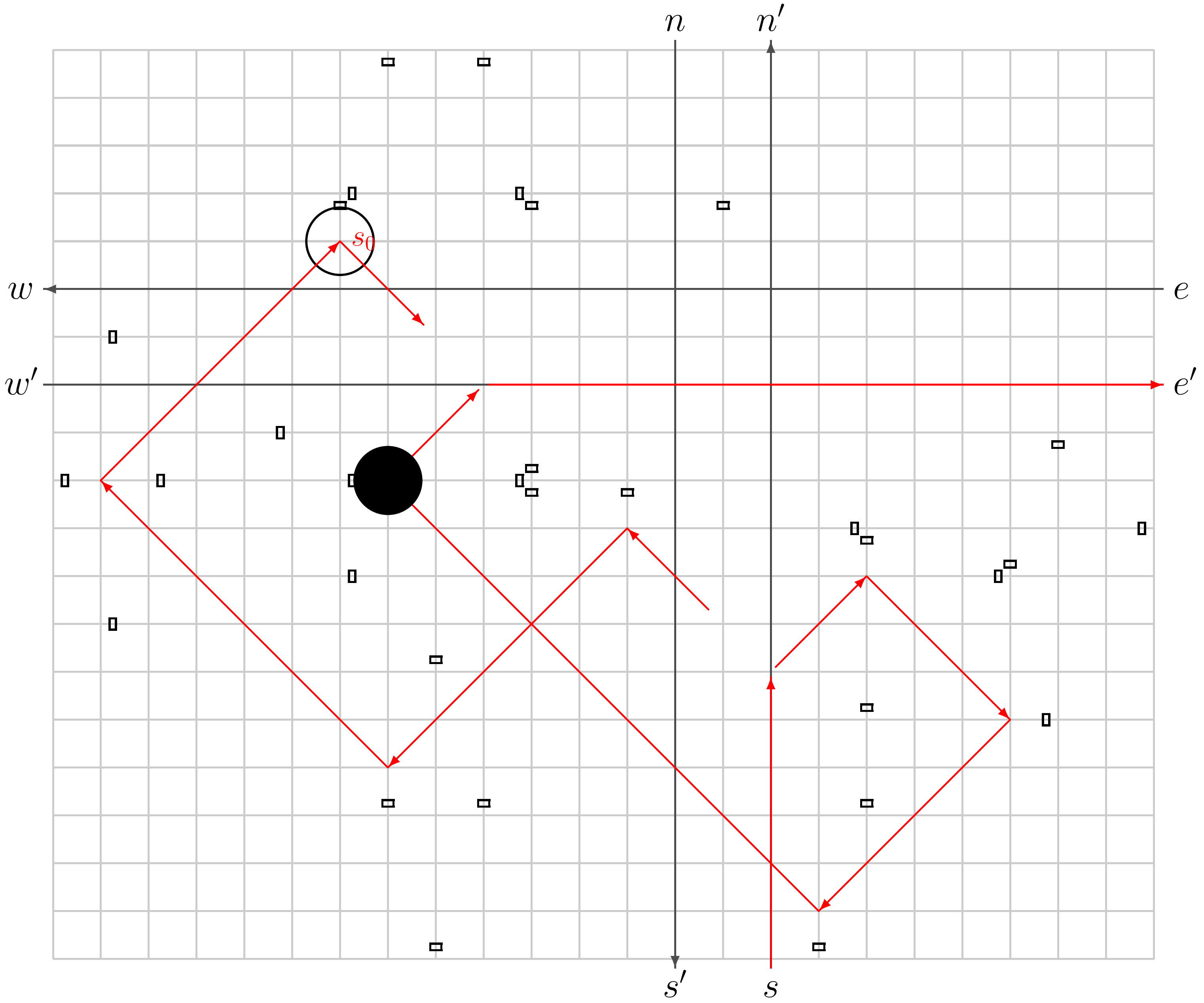
Movements of Balls (State: H, Input: s)



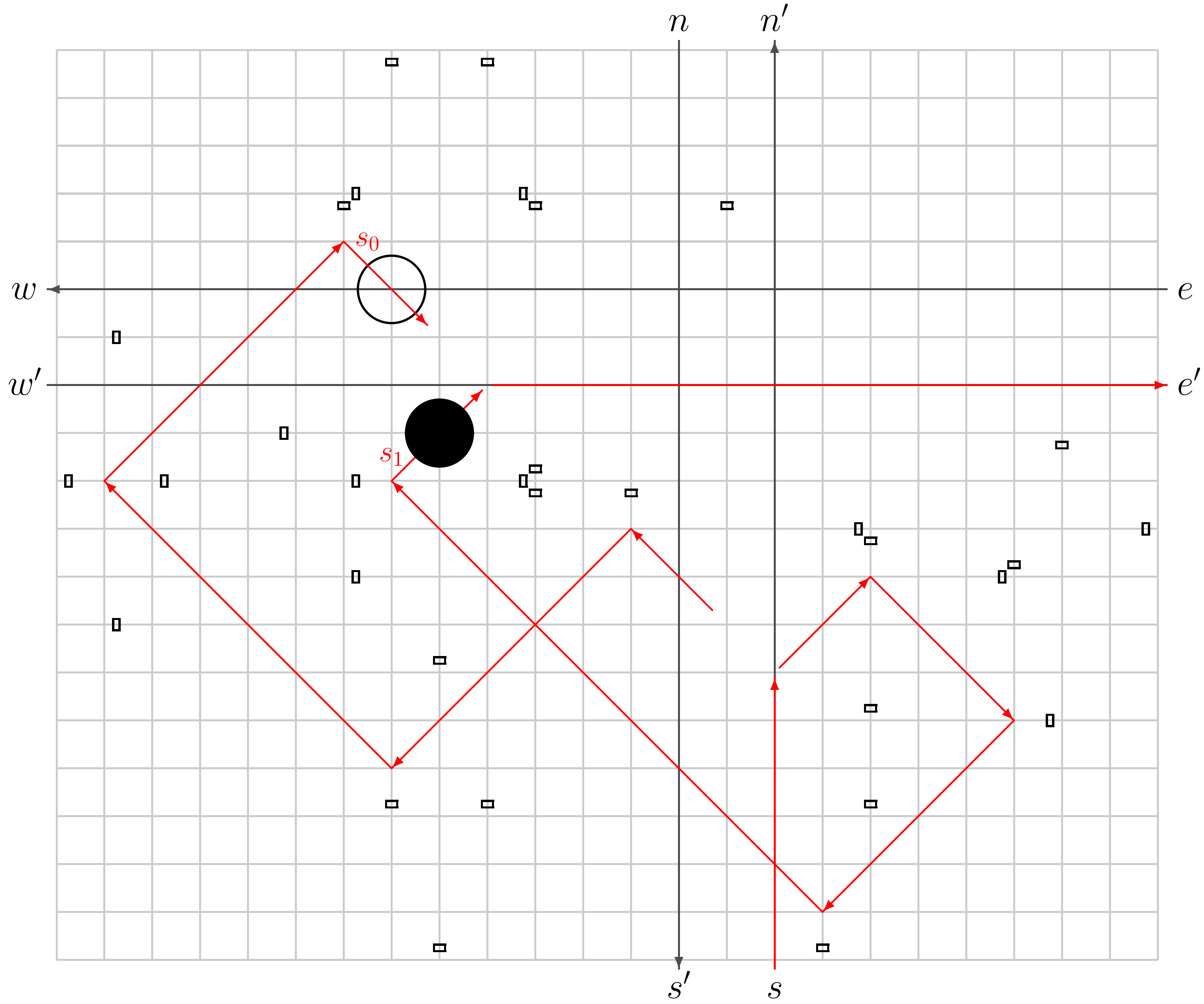
Movements of Balls (State: H, Input: s)



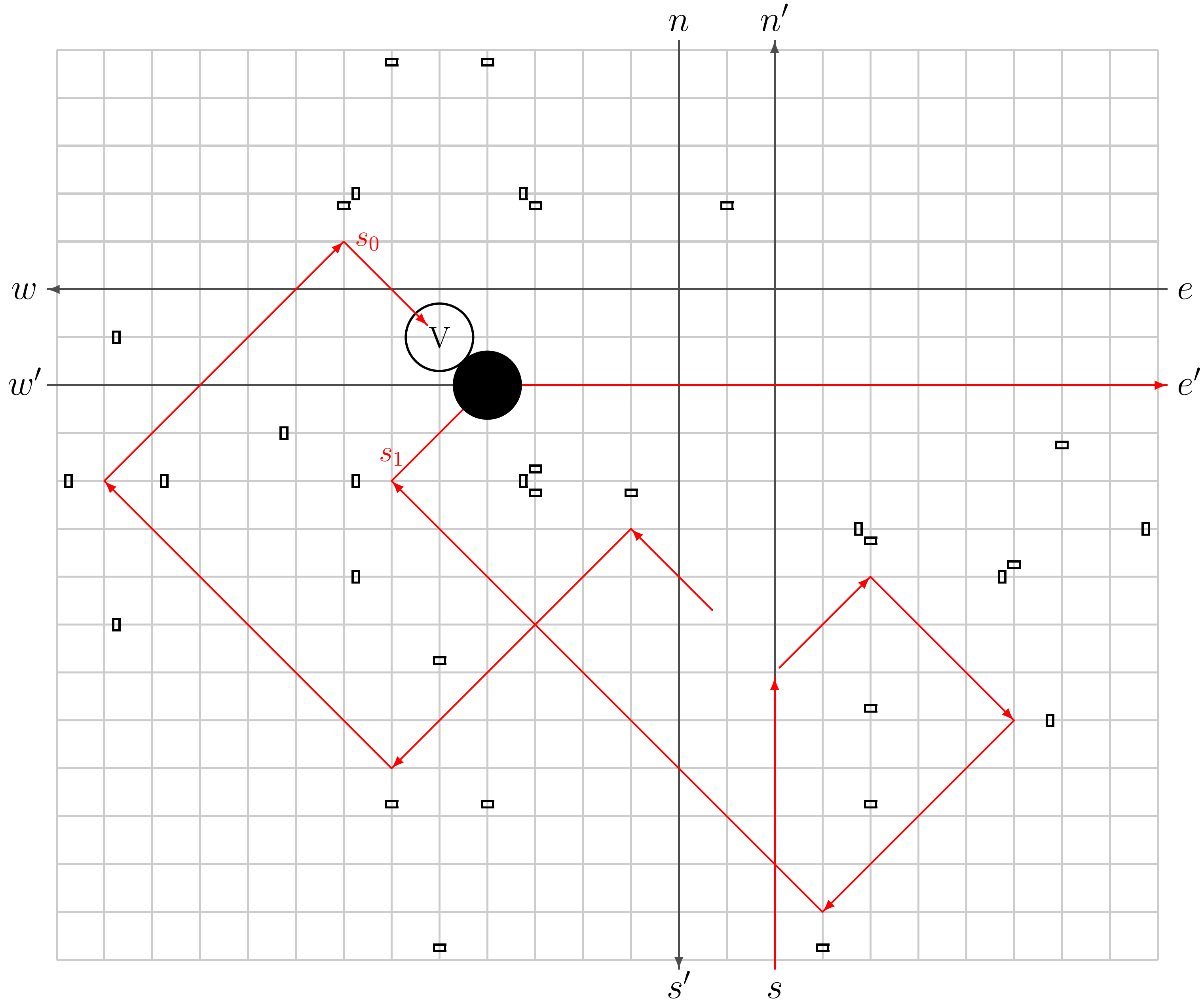
Movements of Balls (State: H, Input: s)



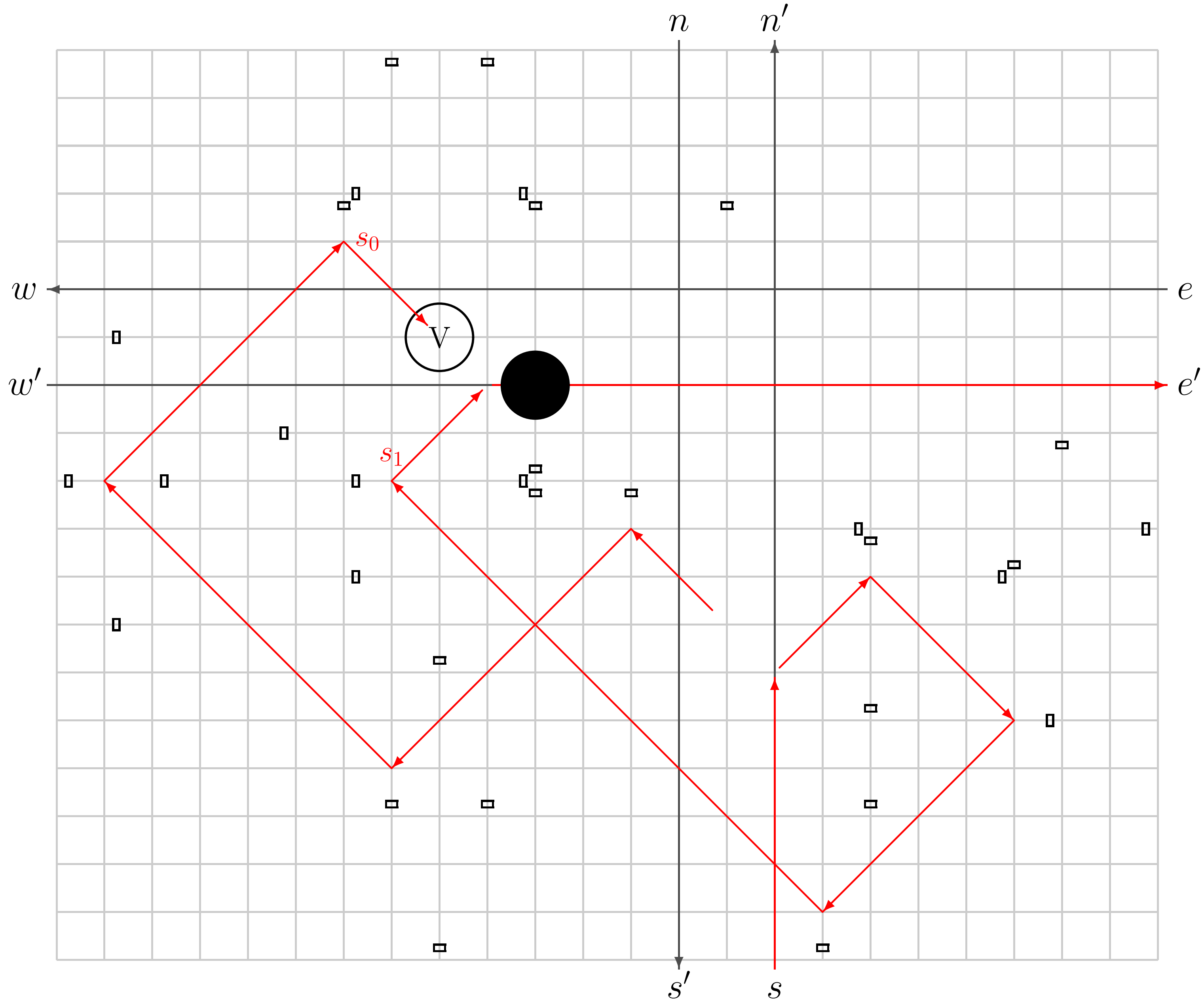
Movements of Balls (State: H, Input: s)



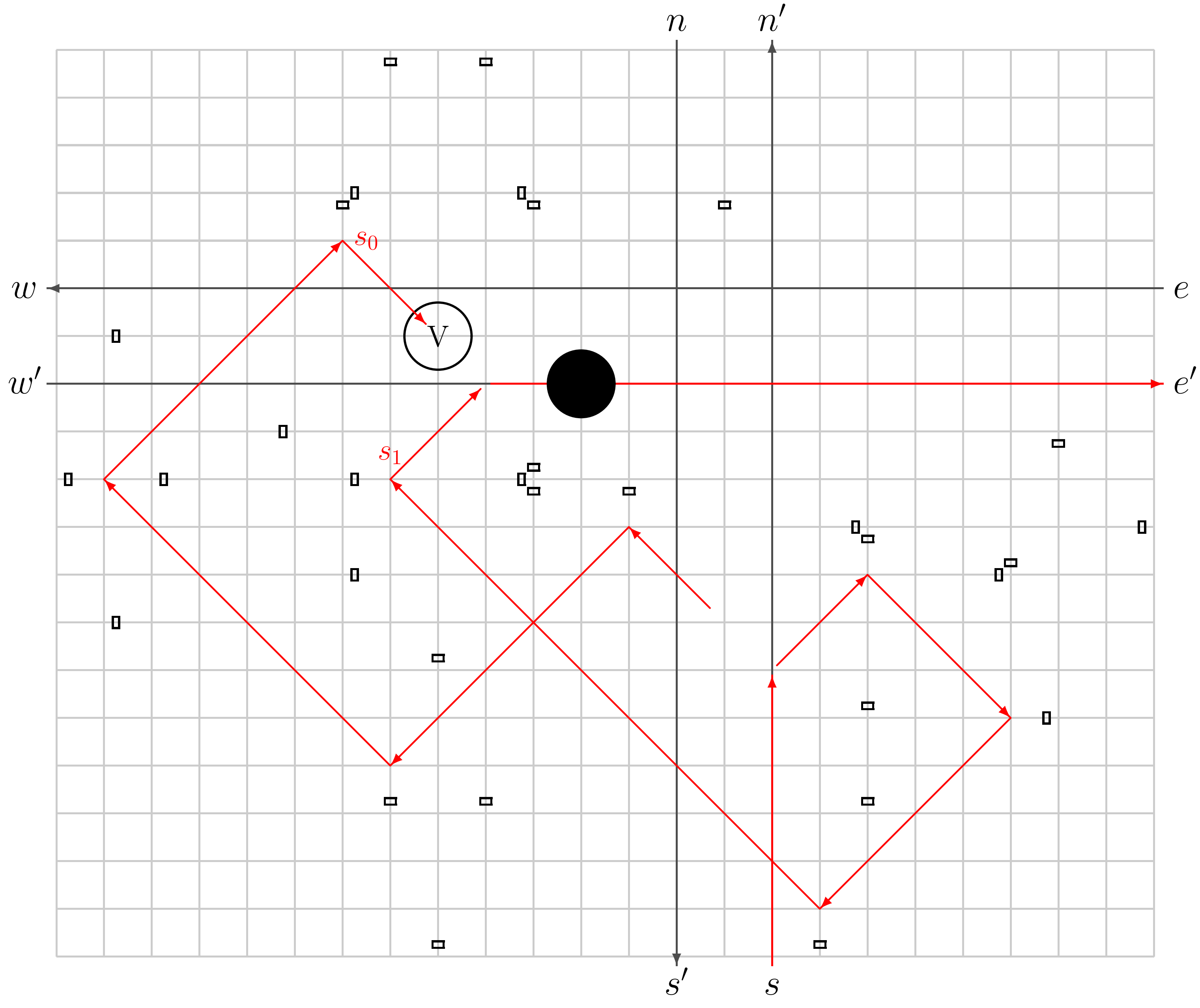
Movements of Balls (State: H, Input: s)



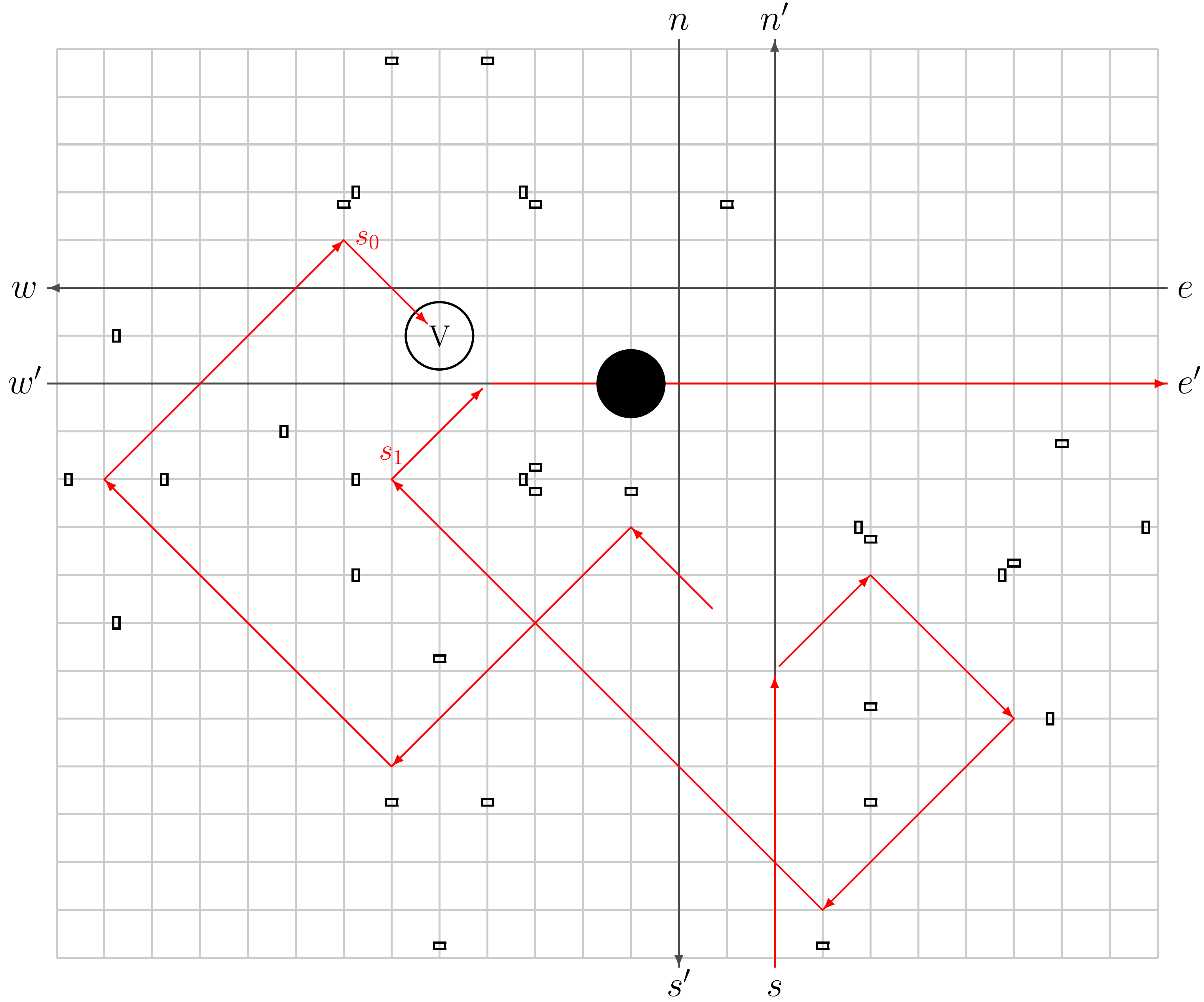
Movements of Balls (State: H, Input: s)



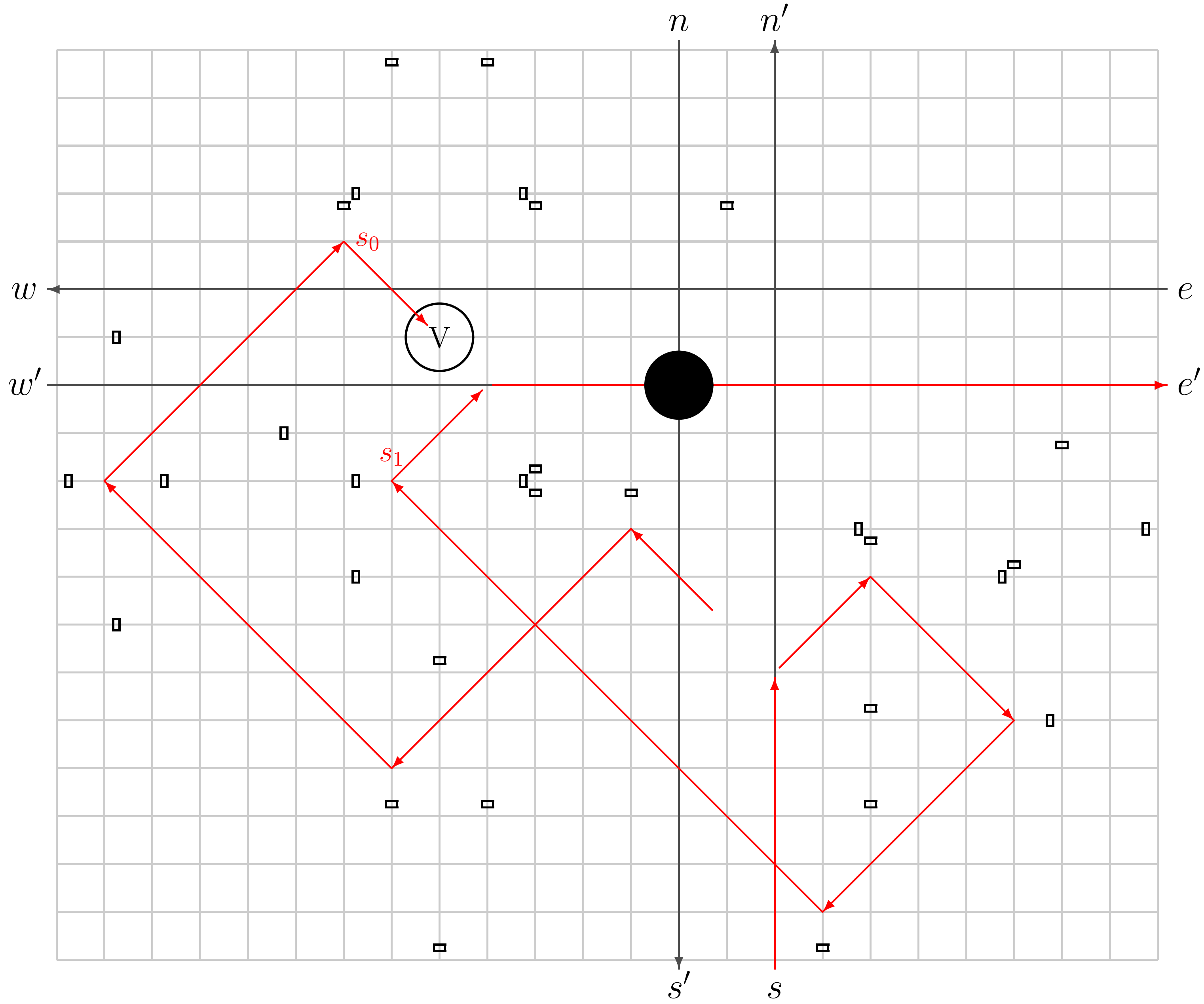
Movements of Balls (State: H, Input: s)



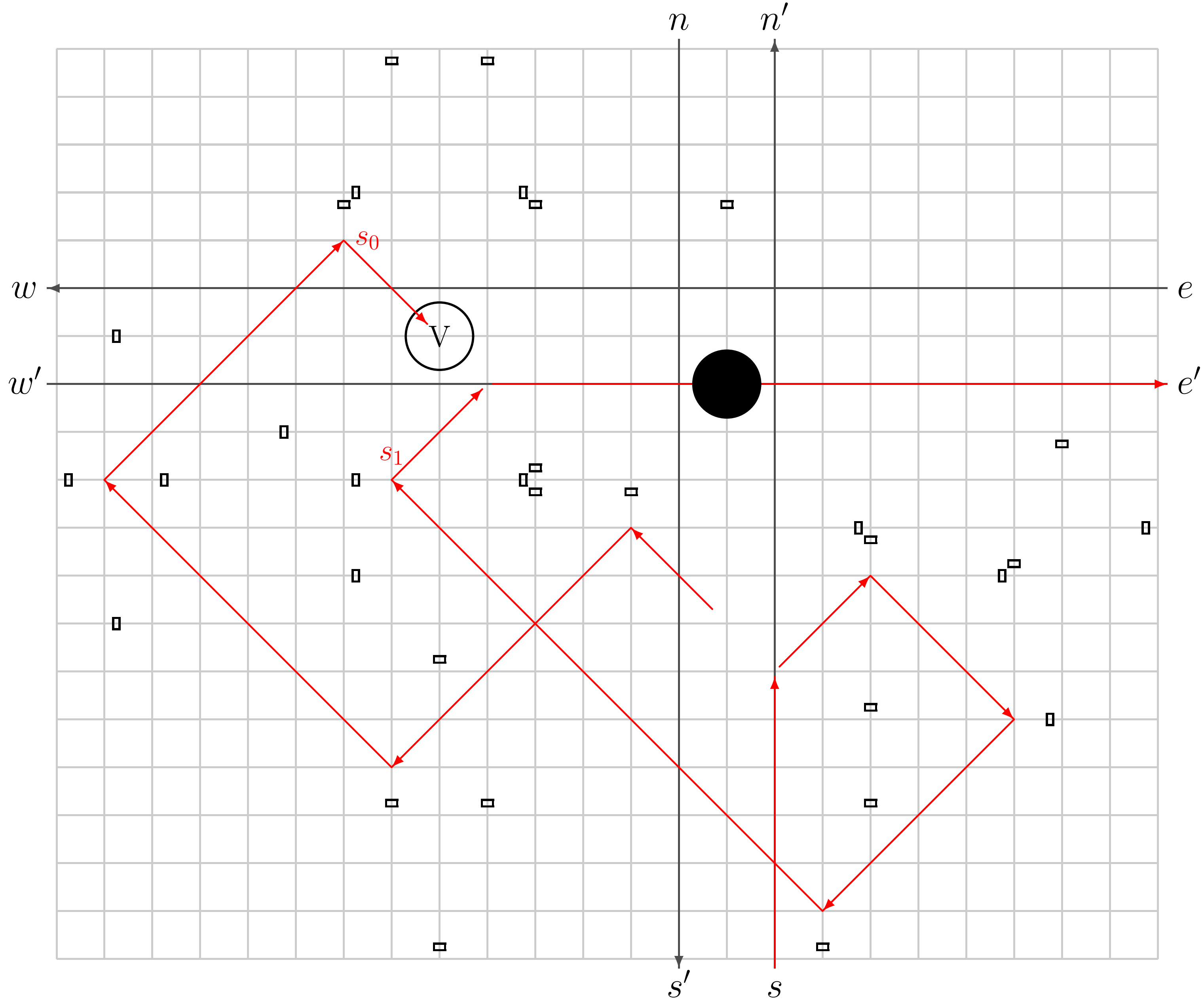
Movements of Balls (State: H, Input: s)



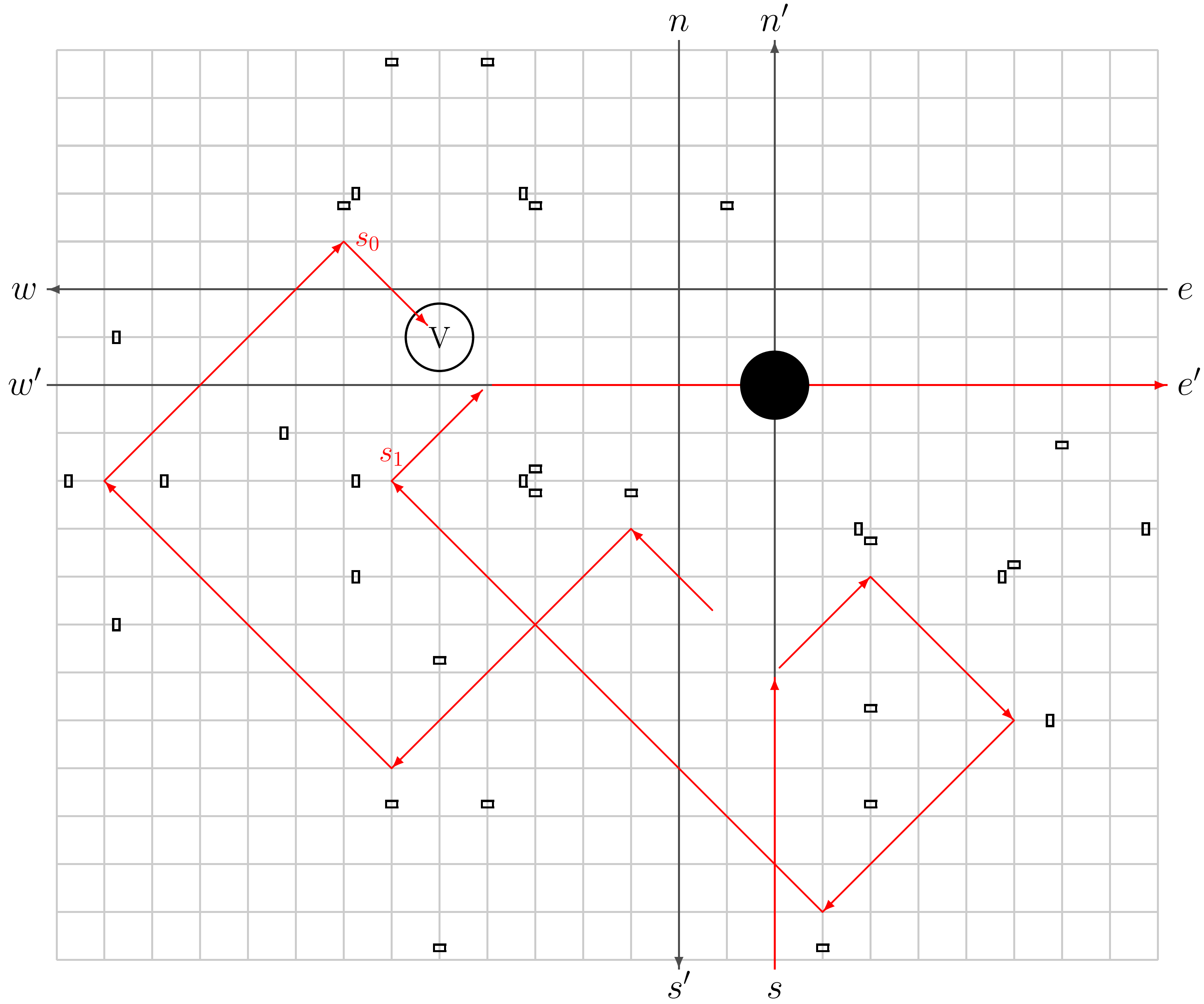
Movements of Balls (State: H, Input: s)



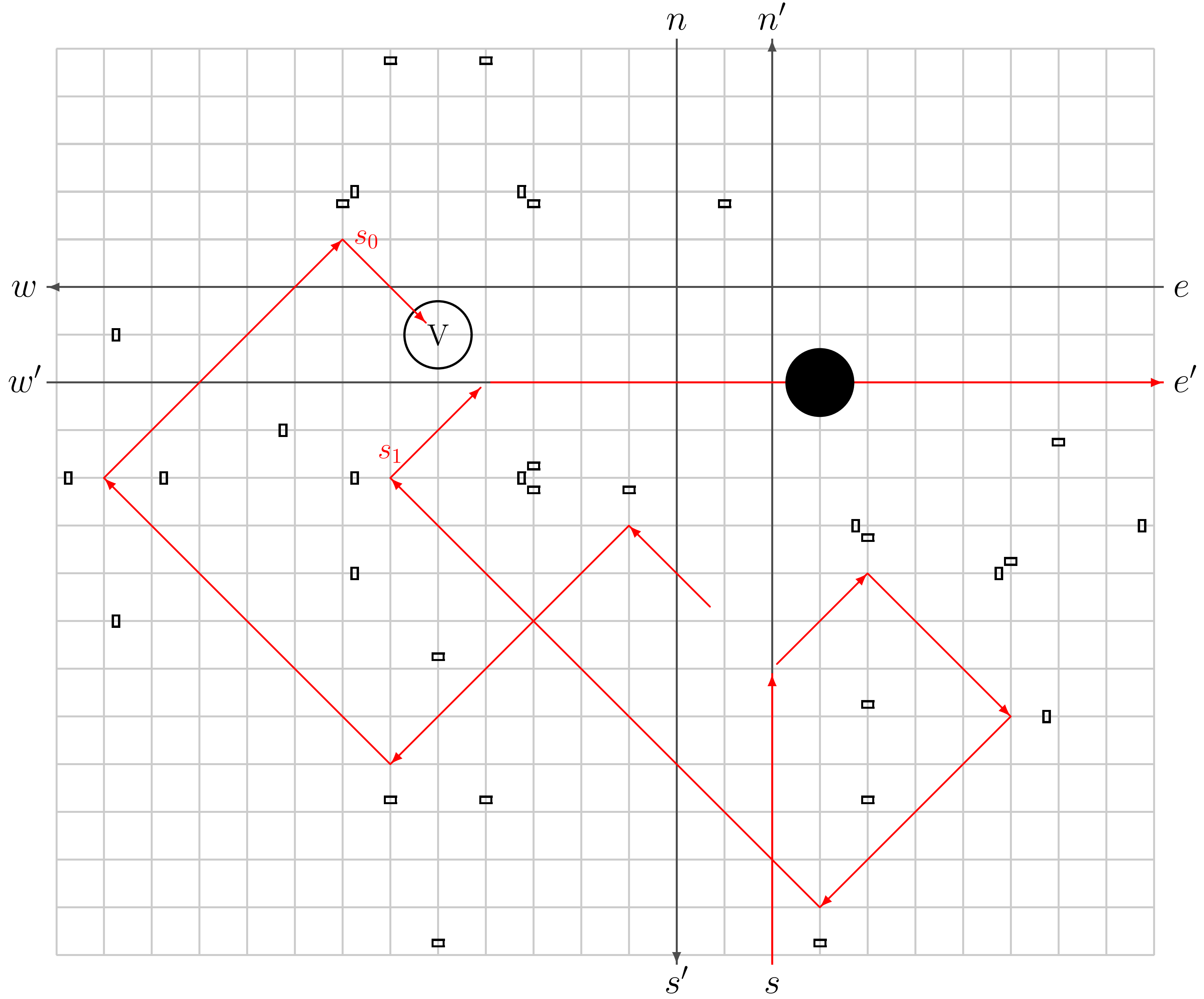
Movements of Balls (State: H, Input: s)



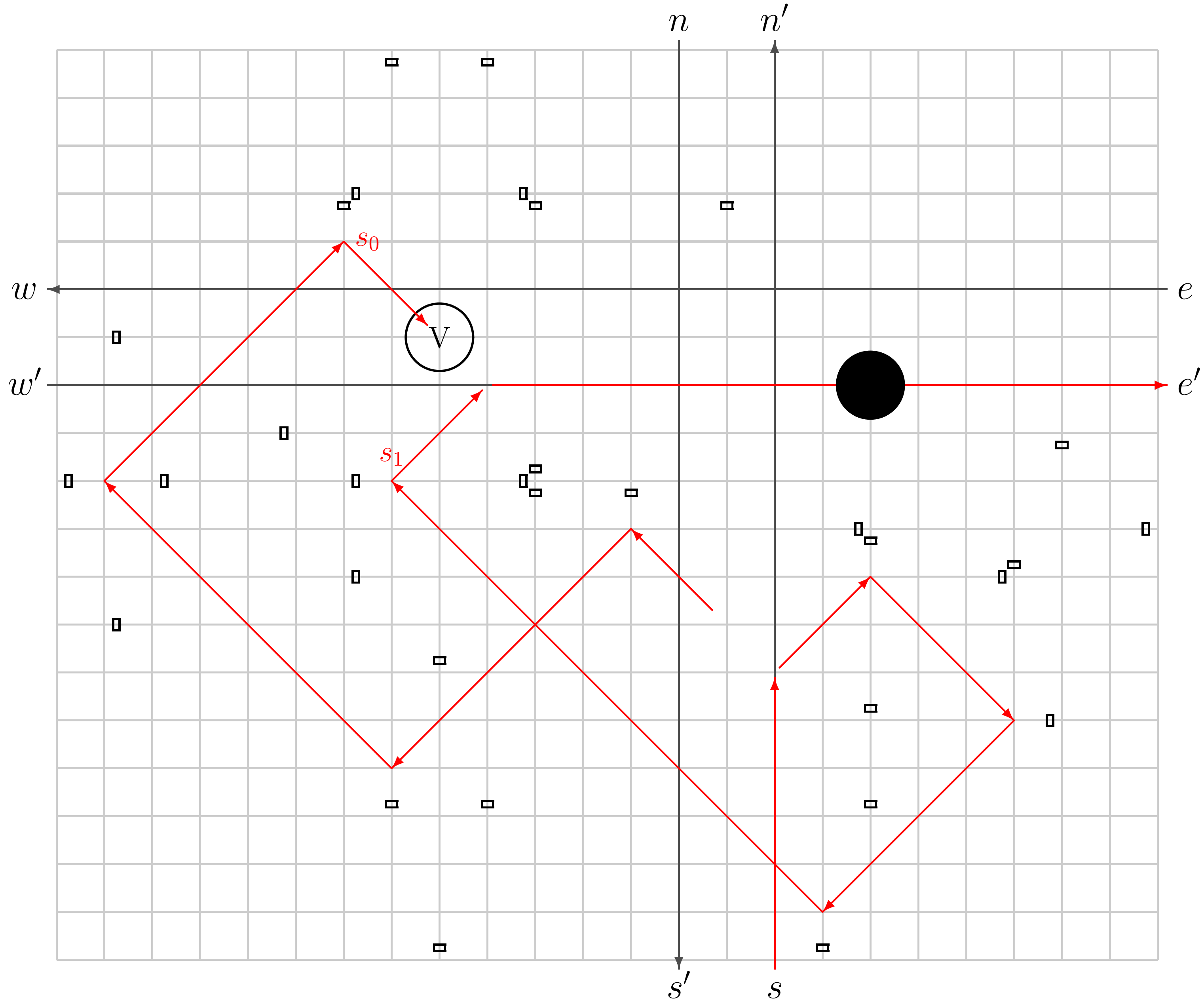
Movements of Balls (State: H, Input: s)



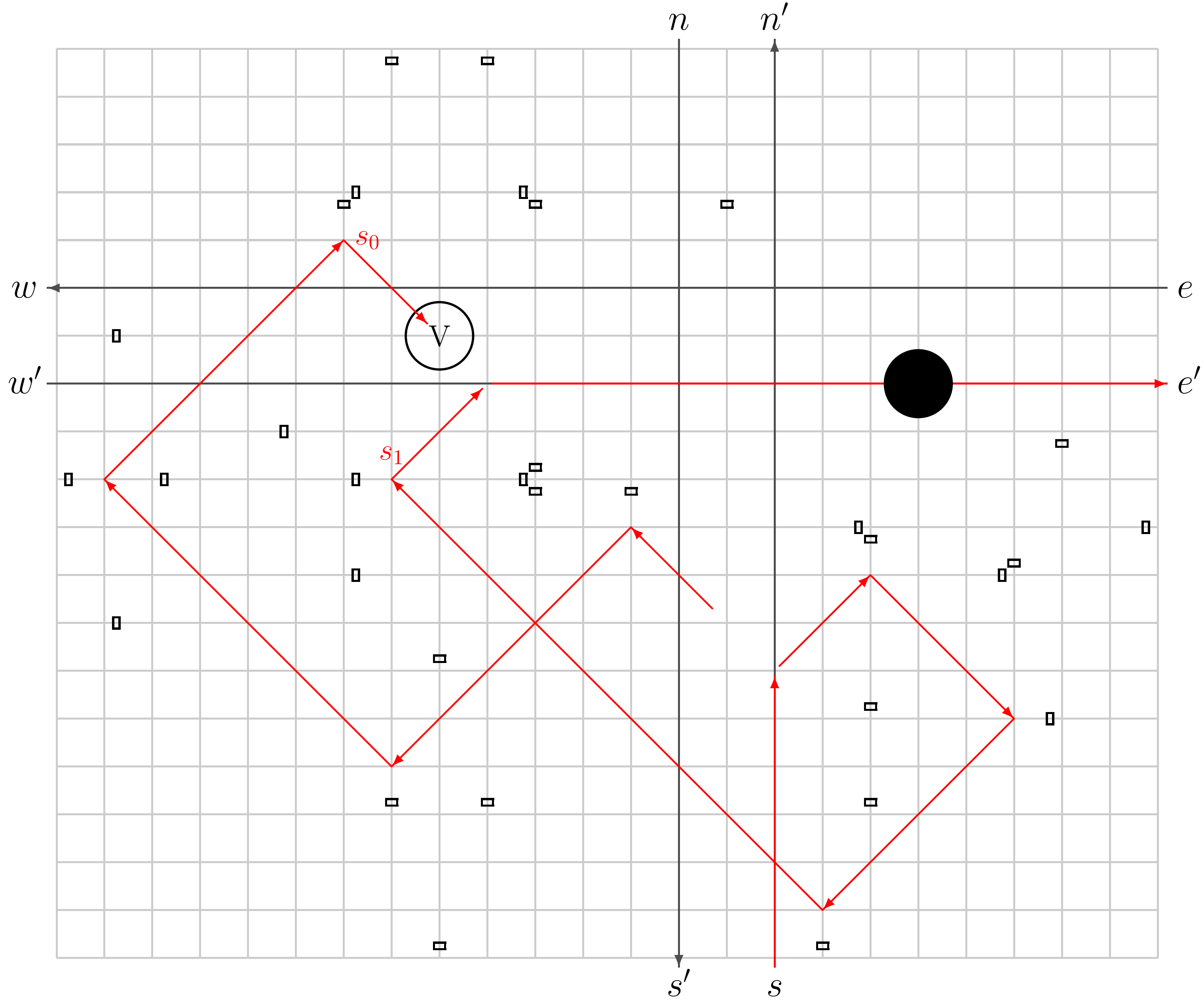
Movements of Balls (State: H, Input: s)



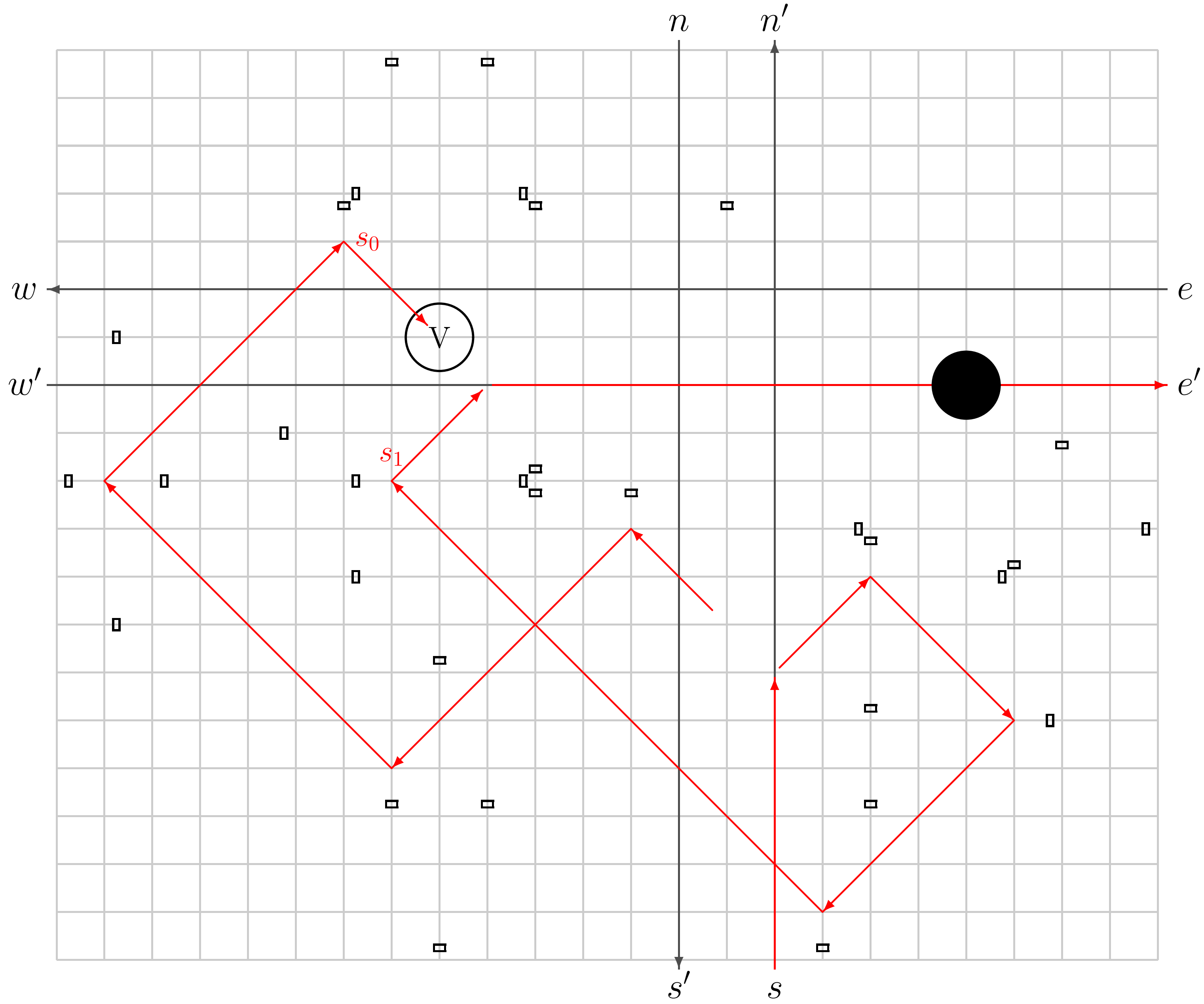
Movements of Balls (State: H, Input: s)



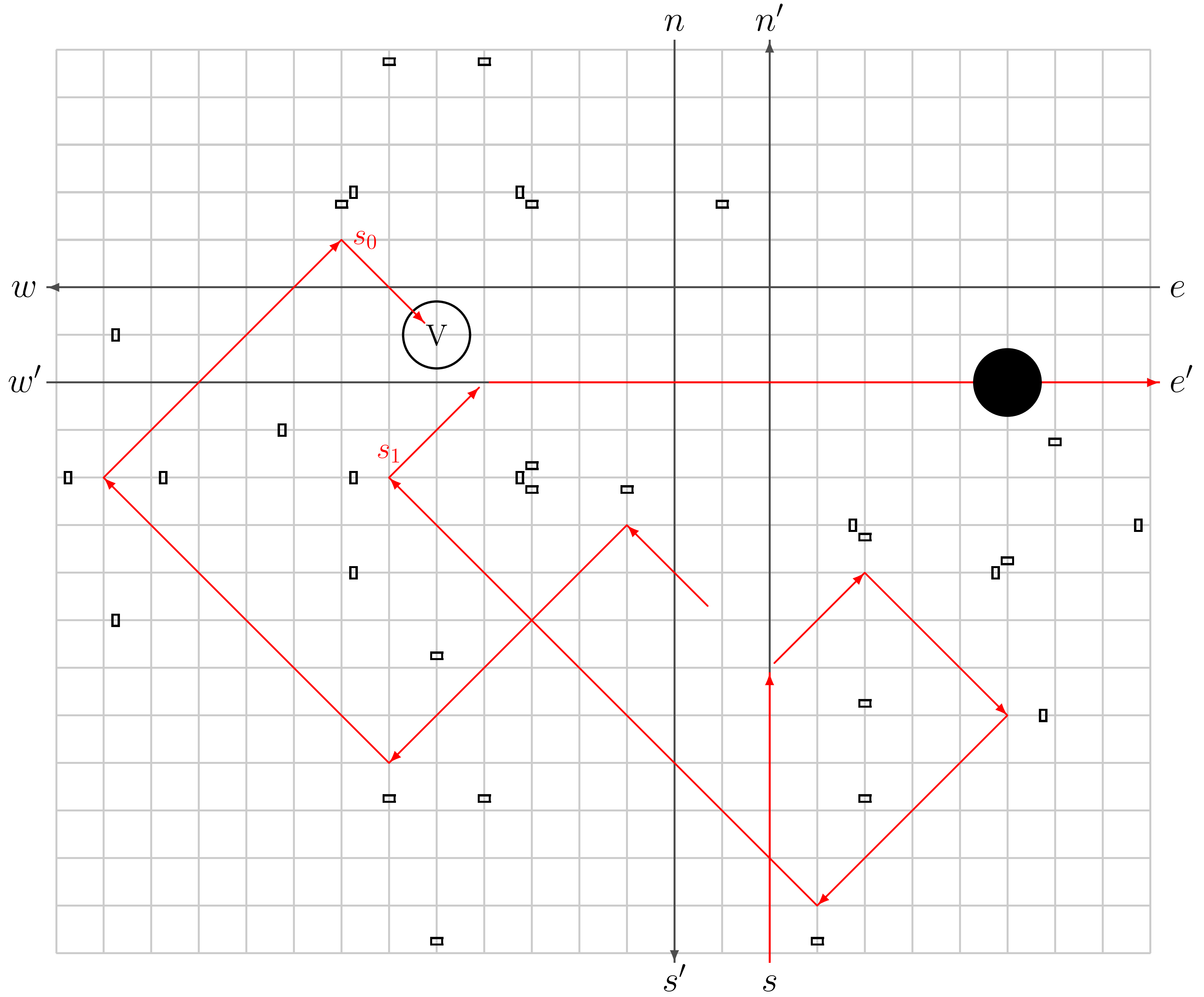
Movements of Balls (State: H, Input: s)



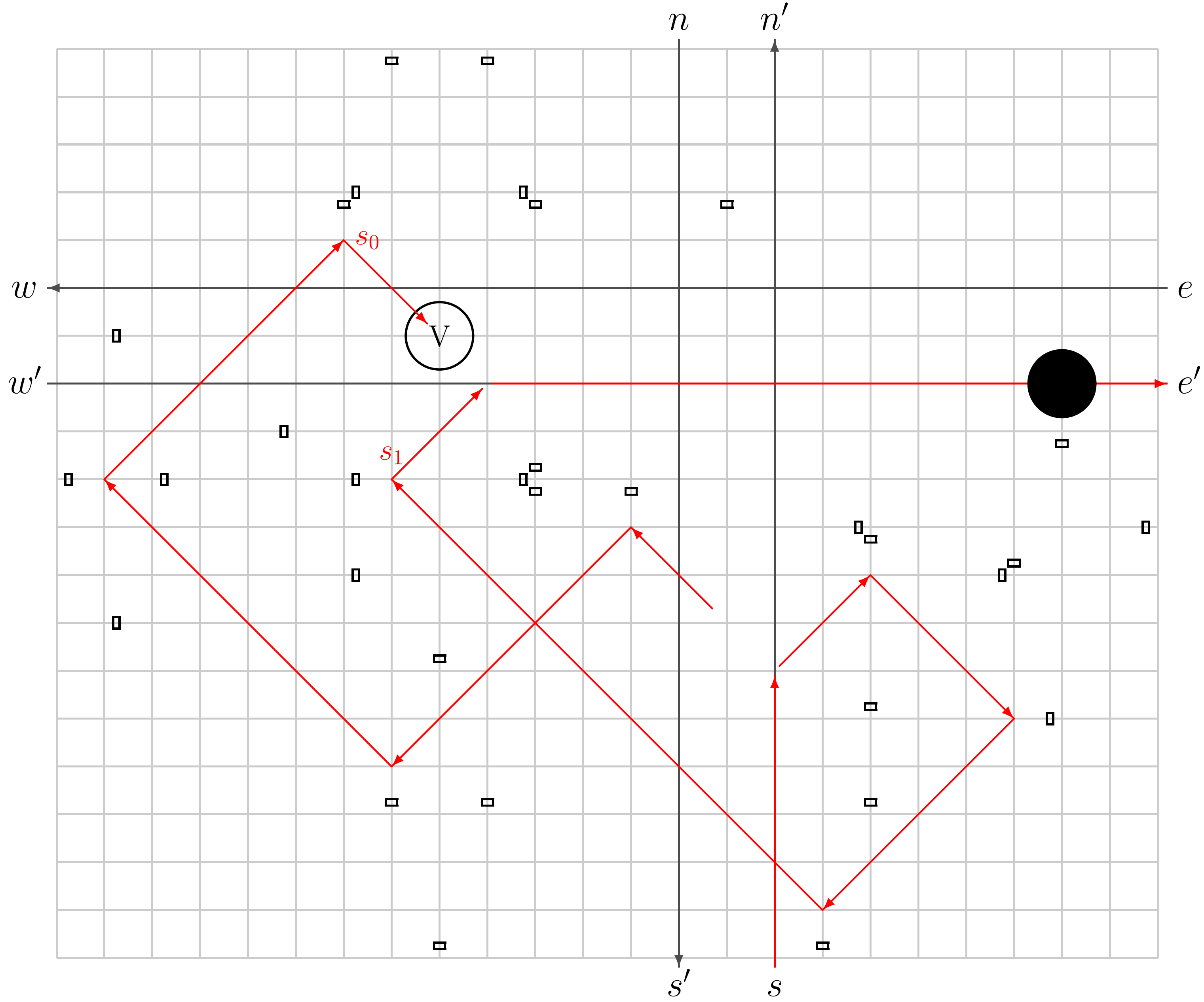
Movements of Balls (State: H, Input: s)



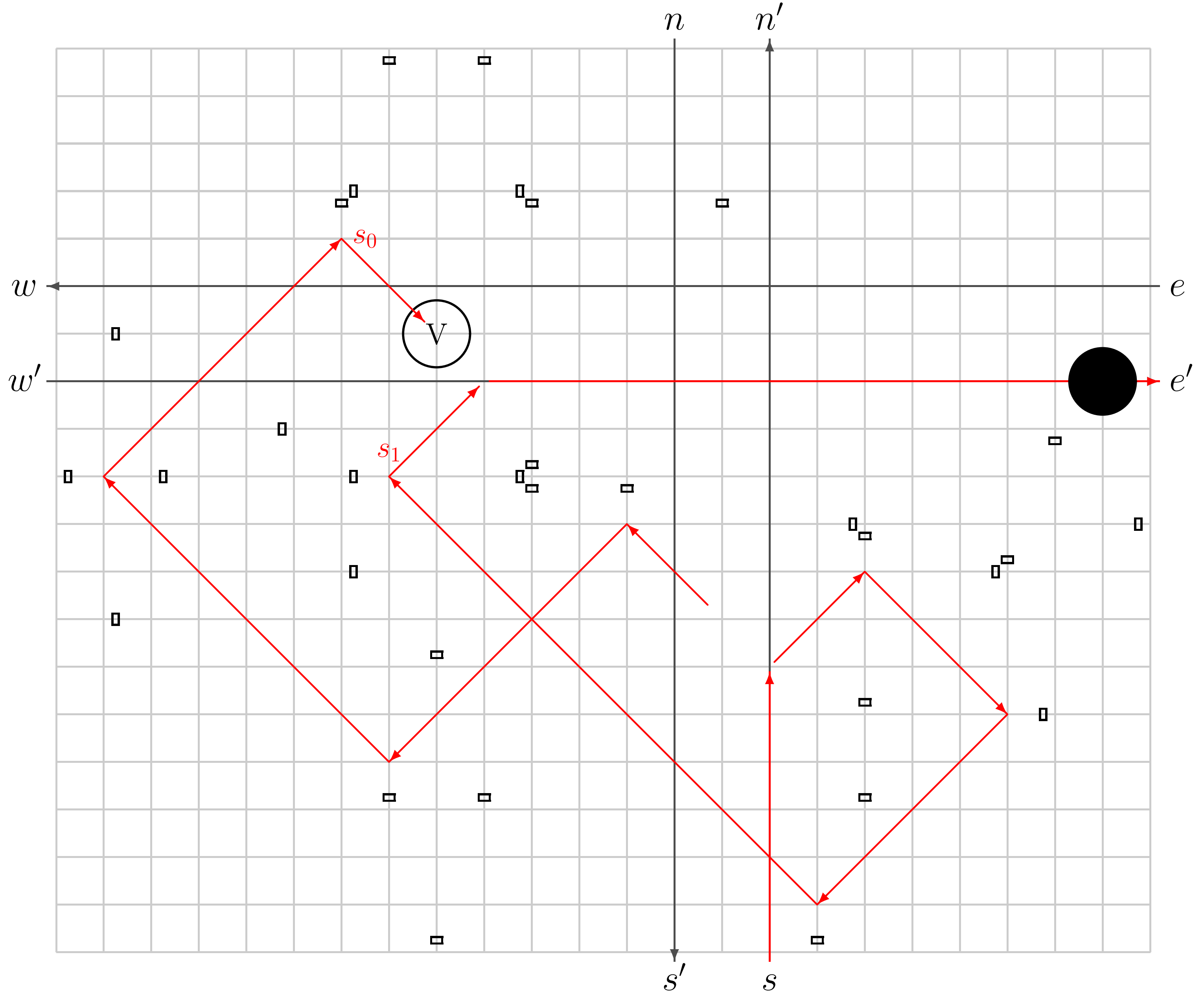
Movements of Balls (State: H, Input: s)



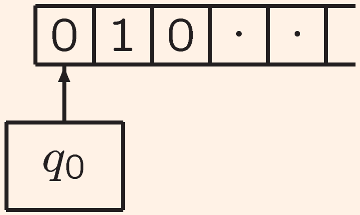
Movements of Balls (State: H, Input: s)



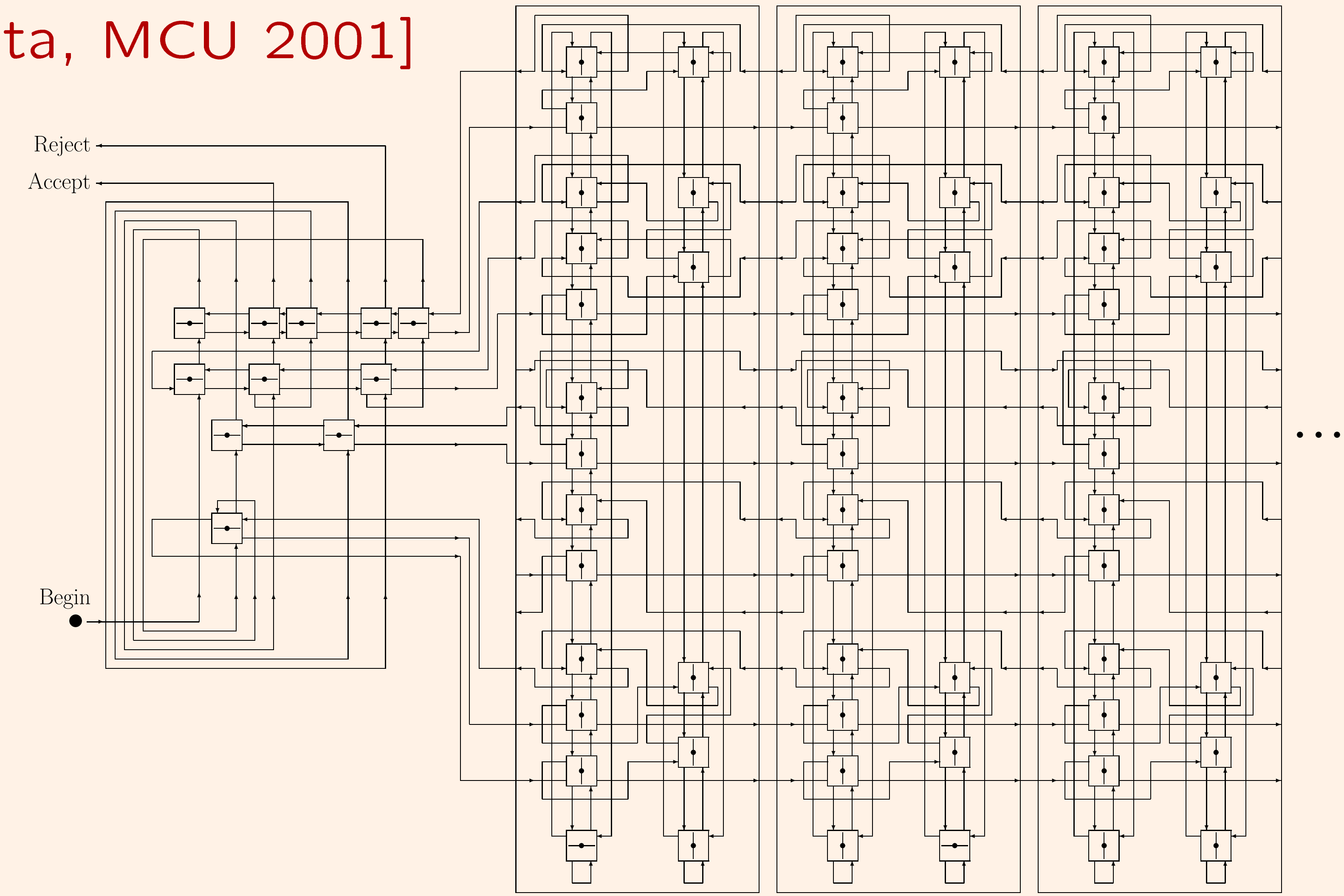
Movements of Balls (State: H, Input: s)



Any Reversible Turing Machine Can Be Composed Only of REs



[Morita, MCU 2001]

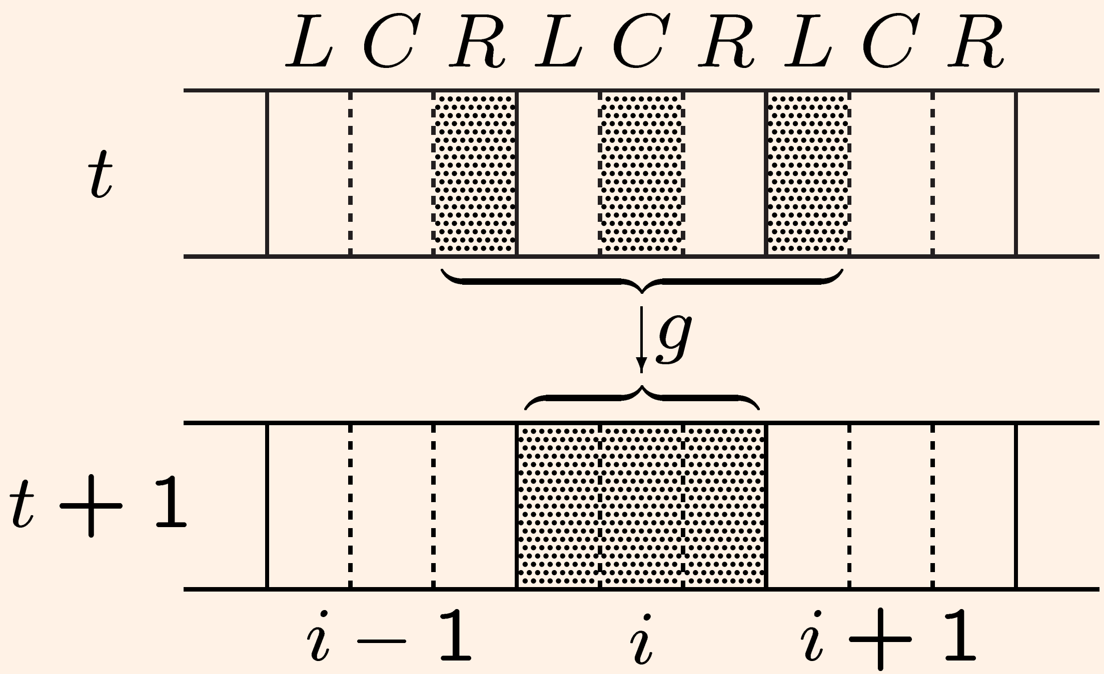


2.4 Reversible Cellular Automaton (RCA)

- It is a CA whose global function is one-to-one.
- A kind of spatio-temporal model of a physically reversible space.

Partitioned Cellular Automata

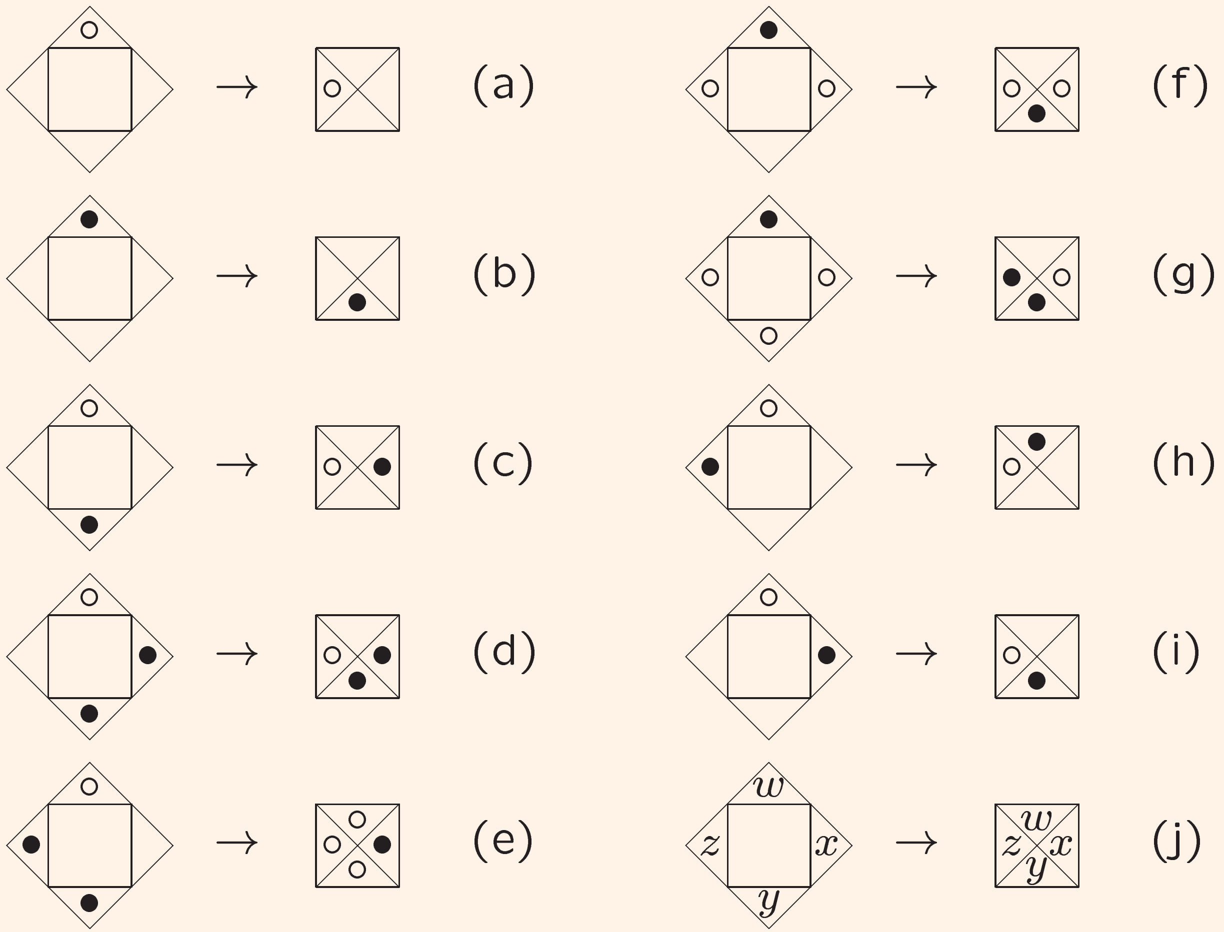
- 1D Partitioned CA (PCA)



The local function g of a 1D PCA.

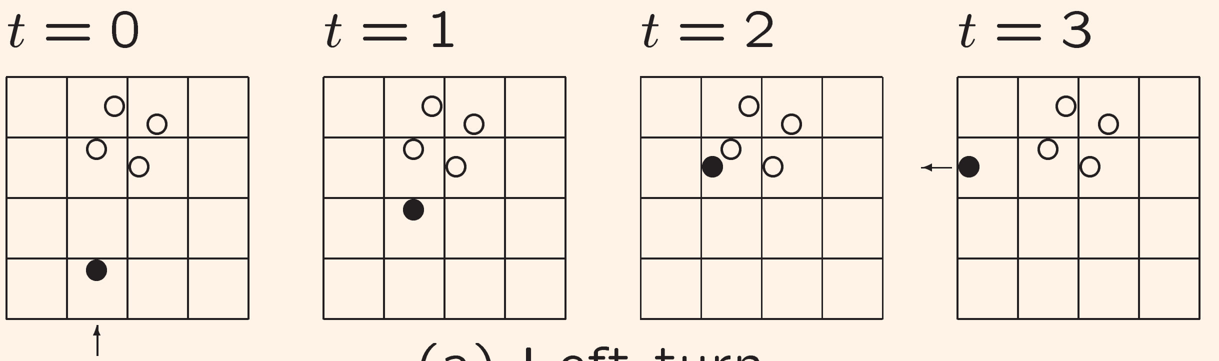
For any PCA, the global function is one-to-one iff the local function is one-to-one.

A 3^4-State 2D Reversible PCA [Morita, Ogiro, 2004]

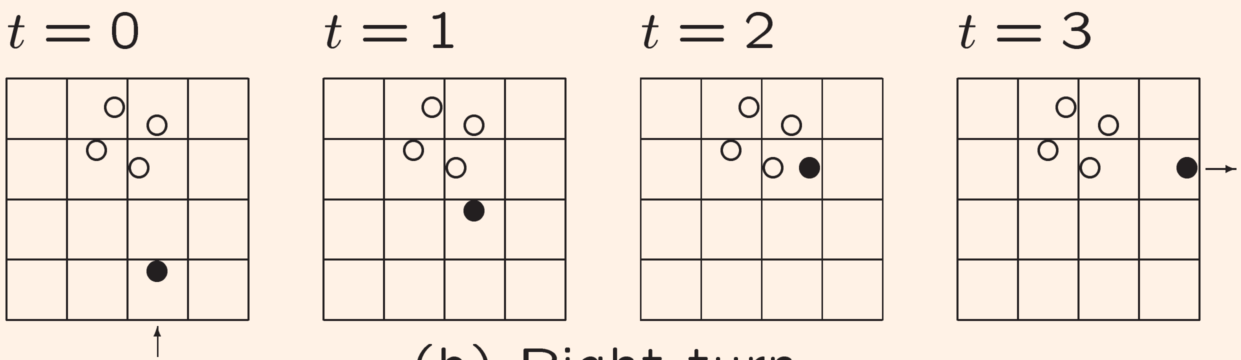


Rotation-symmetric rule schemes. The scheme (j) represents 45 rules not specified by the schemes (a)–(i). ($w, x, y, z \in$ { blank, ○, ● }).

Realizing Left/Right-Turn of a Signal

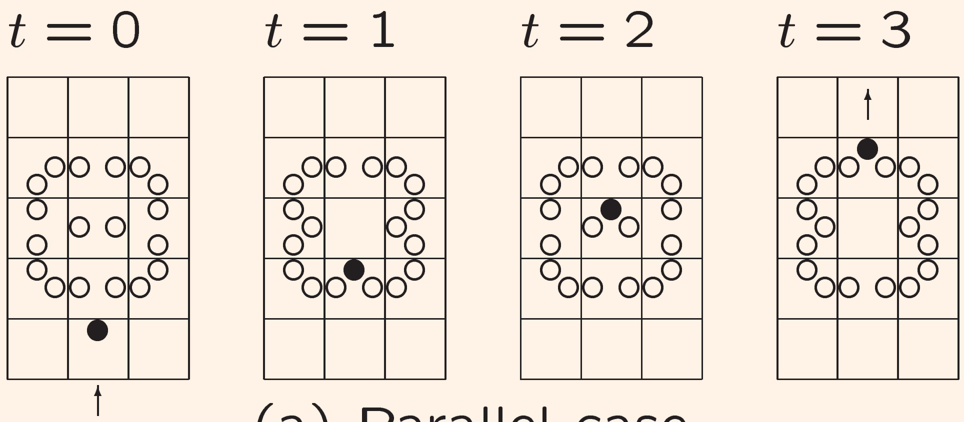


(a) Left-turn.

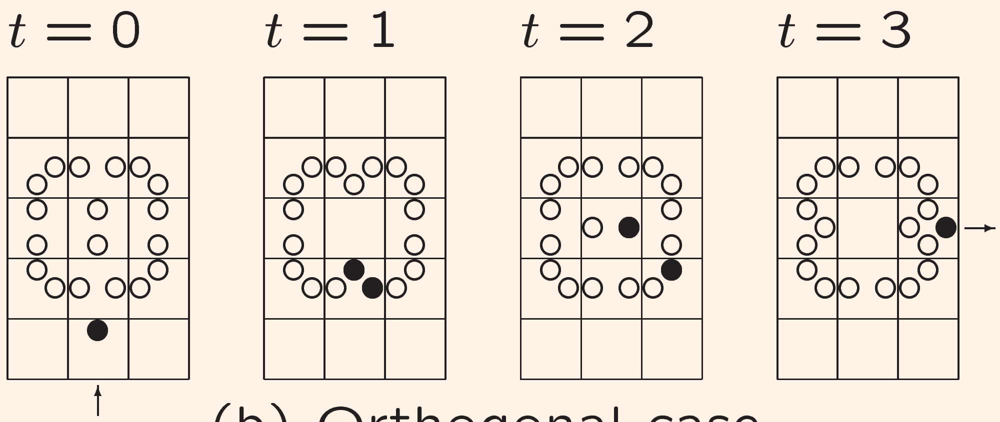


(b) Right-turn.

Realizing a Rotary Element

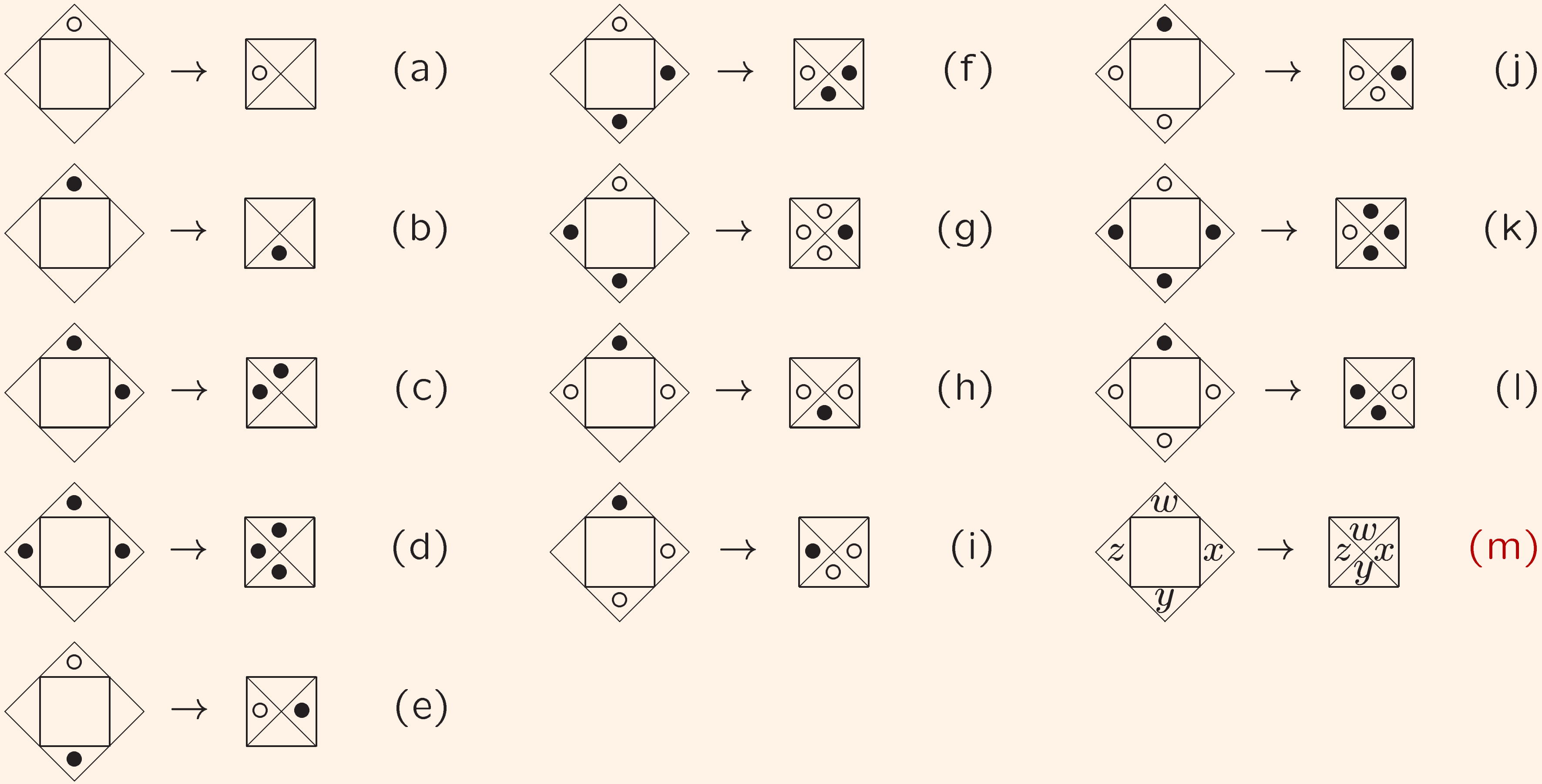


(a) Parallel case.



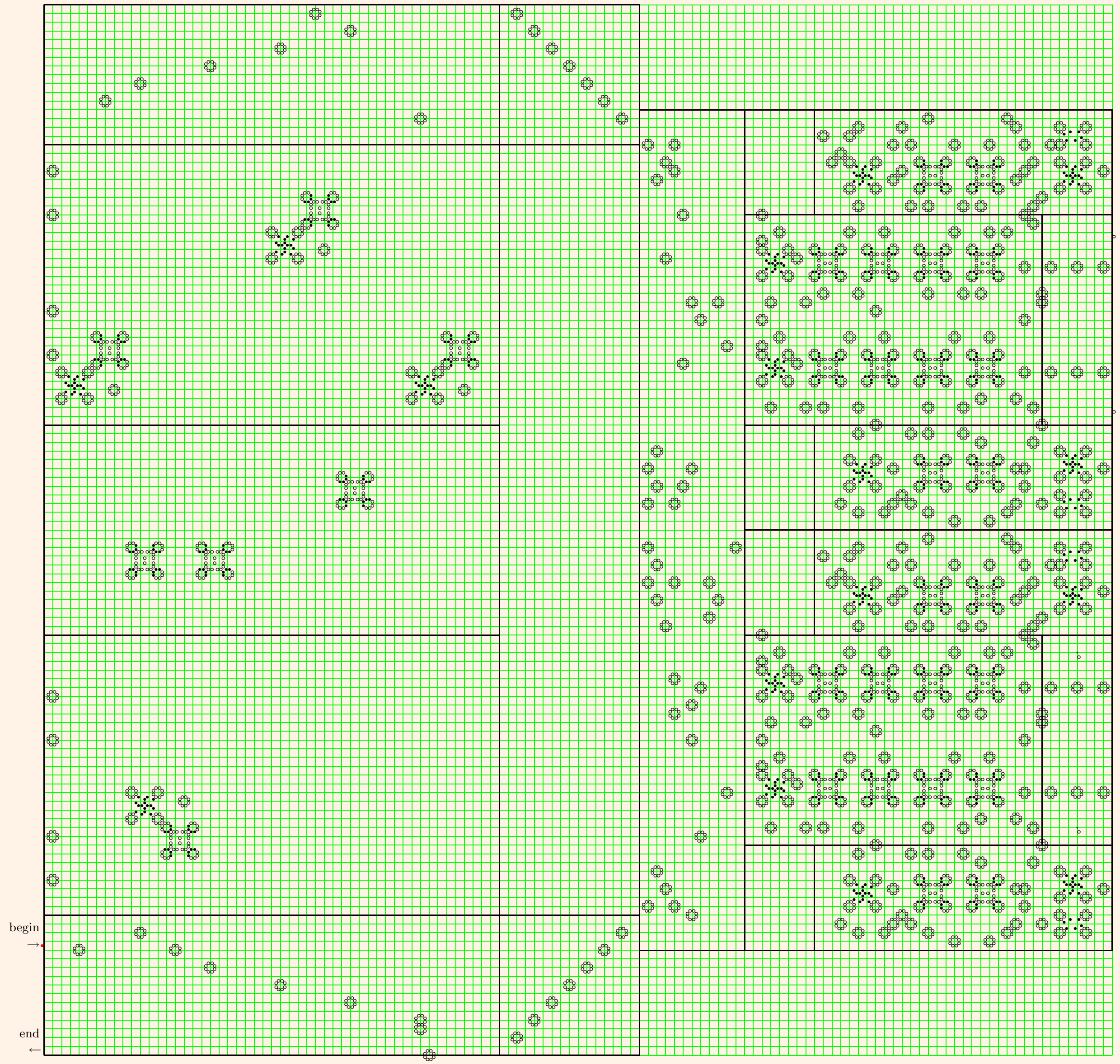
(b) Orthogonal case.

Another 3^4-State 2D RPCA P_3 [Morita, et al., 2002]



The rule scheme (m) represents 33 rules not specified by (a)–(l)
($w, x, y, z \in$ { blank, ○, ● } $= \{0, 1, 2\}$).

Reversible Counter Machine in P_3 Space



Movie of an RCM(2) in P_3

3. Universal Reversible Turing Machine

Definition of a TM

$$T = (Q, S, q_0, q_f, s_0, \delta)$$

Q: a finite set of states.

S: a finite set of tape symbols.

q_0: an initial state $q_0 \in Q$.

q_f: a final state $q_f \in Q$.

s_0: a blank symbol $s_0 \in S$.

δ: a move relation given by a set of *quadruples*, or *quintuples*.

Quadruple Form Vs. Quintuple Form

(1) Quadruple form:

Read/write: $[p, s, s', q] \in Q \times S \times S \times Q$, or

Head-shift: $[p, /, d, q] \in Q \times \{/\} \times \{-, 0, +\} \times Q$

The "inverse" TM can be easily defined for a given reversible TM [Bennett, 1973].

(2) Quintuple form:

Read/write/head-shift:

$$[p, s, s', d, q] \in Q \times S \times S \times \{-, 0, +\} \times Q$$

The number of states can be fewer than (1). (Most classical universal TMs are in this form.)

Definition of a Reversible TM

A TM $T = (Q, S, q_0, q_f, s_0, \delta)$ is called *reversible* iff the following condition holds.

- Quadruple Case:
  For any pair of distinct quadruples
  $[p_1, b_1, c_1, q_1]$ and $[p_2, b_2, c_2, q_2]$,
  if $q_1 = q_2$, then $b_1 \neq / \ \wedge \ b_2 \neq / \ \wedge \ c_1 \neq c_2$.
- Quintuple Case:
  For any pair of distinct quintuples
  $[p_1, s_1, s'_1, d_1, q_1]$ and $[p_2, s_2, s'_2, d_2, q_2]$,
  if $q_1 = q_2$, then $s'_1 \neq s'_2 \ \wedge \ d_1 = d_2$.

Conversion between Quadruple and Quintuple Forms

Proposition 1 For any RTM in the quintuple form, there is an RTM in the quadruple form that simulates each step of the former in two steps.

(Note: The converse of the Proposition 1 can be shown also easily.)

A Small Universal RTM (URTM)

A *URTM* is an RTM that can compute *any* recursive function.

Main Result: There is a 17-state 5-symbol RTM that can simulate any cyclic tag system.

Since cyclic tag systems are known to be universal [Cook, 2004], the above RTM is universal.

Cyclic Tag System (CTAG) [Cook, 2004]

$$C = (k, \{Y, N\}, (\mathsf{halt}, p_1, \cdots, p_{k-1}))$$

- k: the length of a cycle (positive integer).
- $\{Y, N\}$: the alphabet used in a CTAG.
- $(p_1, \cdots, p_{k-1}) \in (\{Y, N\}^*)^{k-1}$: production rules.

An *instantaneous description* (ID) is a pair (v, i), where $v \in \{Y, N\}^*$ and $i \in \{0, \cdots, k-1\}$.

For any $(v, i), (w, j) \in \{Y, N\}^* \times \{0, \cdots, k-1\}$,

$$\begin{aligned}(Yv, i) \Rightarrow (w, j) \text{ iff } & [m \neq 0] \wedge [j = i + 1 \bmod k] \\ & \wedge [w = vp_i], \\ (Nv, i) \Rightarrow (w, j) \text{ iff } & [j = i + 1 \bmod k] \wedge [w = v].\end{aligned}$$

A Simple Example of a CTAG

$$C_0 = (3, \{Y, N\}, (Y \to \text{halt},\ Y \to NN,\ Y \to YY))$$

If an initial word NYY is given, the computing on C_0 proceeds as follows:

$$
\begin{array}{rl}
 & (NYY, 0) \\
\Rightarrow & (YY, 1) \\
\Rightarrow & (YNN, 2) \\
\Rightarrow & (NNYY, 0) \\
\Rightarrow & (NYY, 1) \\
\Rightarrow & (YY, 2) \\
\Rightarrow & (YYY, 0)
\end{array}
$$

A Simple Example of a CTAG

$$C_0 = (3, \{Y, N\}, (\text{halt}, \ NN, \ YY))$$

If an initial word NYY is given, the computing on C_0 proceeds as follows:

$$
\begin{aligned}
& (\ N\ Y\ Y\ , \quad 0\ ) \\
\Rightarrow\ & (\ Y\ Y\ , \quad 1\ ) \\
\Rightarrow\ & (\ Y\ N\ N\ , \quad 2\ ) \\
\Rightarrow\ & (\ N\ N\ Y\ Y\ , \quad 0\ ) \\
\Rightarrow\ & (\ N\ Y\ Y\ , \quad 1\ ) \\
\Rightarrow\ & (\ Y\ Y\ , \quad 2\ ) \\
\Rightarrow\ & (\ Y\ Y\ Y\ , 0\ )
\end{aligned}
$$

Simulating a 2-Tag System by a CTAG
[Cook, 2004]

An Example

- A given 2-tag system:

$$T_1 = (2, \{a_0, a_1, a_2\}, \{a_0 : \text{halt},\ a_1 \rightarrow a_2,\ a_2 \rightarrow a_0 a_1\})$$

- A cyclic tag system that simulates T_1:

$$C_1 = (6, \{Y, N\}, (\text{halt},\ NNY,\ YNNNYN,\ \varepsilon, \varepsilon, \varepsilon)),$$

where a_0, a_1, and a_2 in T_1 are coded by YNN, NYN, and NNY in C_1.

Simulating T_1 by C_1

$$T_1 = (2, \{a_0, a_1, a_2\}, \{a_0 : \text{halt},\ a_1 \to a_2,\ a_2 \to a_0 a_1\})$$
$$C_1 = (6, \{Y, N\}, (\text{halt},\ NNY,\ YNN\,NYN,\ \varepsilon, \varepsilon, \varepsilon))$$

Example:

$$T_1 : \quad a_2 a_1 \ \Rightarrow\ a_0 a_1$$

$$\begin{array}{lll}
C_1 : & (\ N\ N\ Y\ \ N\ Y\ N\ , & 0\ ) \\
\Rightarrow & (\ N\ Y\ \ N\ Y\ N\ , & 1\ ) \\
\Rightarrow & (\ Y\ \ N\ Y\ N\ , & 2\ ) \\
\Rightarrow & (\ N\ Y\ N\ \ Y\ N\ N\ \ N\ Y\ N\ , & 3\ ) \\
\Rightarrow & (\ Y\ N\ \ Y\ N\ N\ \ N\ Y\ N\ , & 4\ ) \\
\Rightarrow & (\ N\ \ Y\ N\ N\ \ N\ Y\ N\ , & 5\ ) \\
\Rightarrow & (\ Y\ N\ N\ \ N\ Y\ N\ , & 0\ )
\end{array}$$

A 17-State 5-Symbol URTM (URTM(17,5)) That Simulates Any CTAG

$$T_{17,5} = (\{q_0, \cdots, q_{16}\}, \{b, Y, N, *, \$\}, q_0, b, \delta),$$

(δ is shown in the next page.)

- Since $T_{17,5}$ simulates a CTAG without erasing the given information, it is in principle reversible.
- But, technically, it is not easy to design a TM to be reversible while keeping its size small.

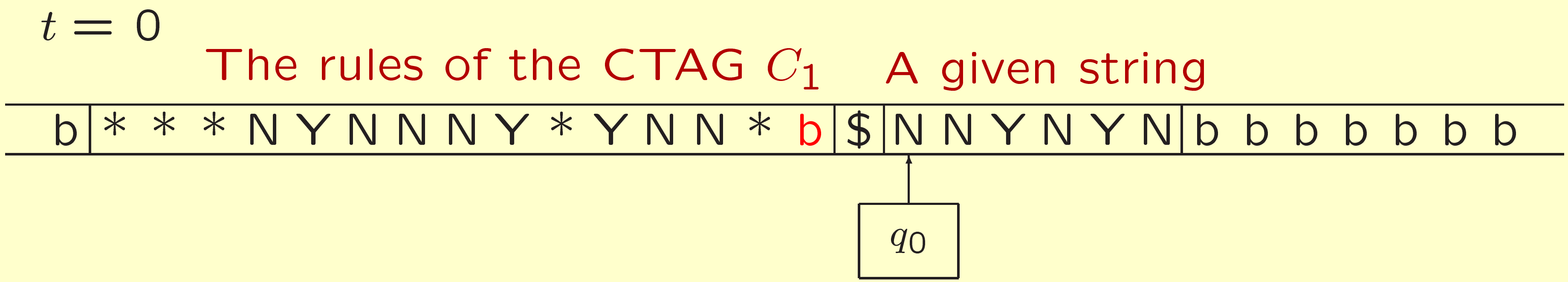
The quintuple set of the URTM(17,5)

	b	Y	N	$*$	$\$$
q_0	$\$ - q_2$	$\$ - q_1$	$b - q_{13}$		
q_1	halt	$Y - q_1$	$N - q_1$	$* + q_0$	$b - q_1$
q_2	$* - q_3$	$Y - q_2$	$N - q_2$	$* - q_2$	null
q_3	$b + q_{12}$	$b + q_4$	$b + q_7$	$b + q_{10}$	
q_4	$Y + q_5$	$Y + q_4$	$N + q_4$	$* + q_4$	$\$ + q_4$
q_5	$b - q_6$				
q_6	$Y - q_3$	$Y - q_6$	$N - q_6$	$* - q_6$	$\$ - q_6$
q_7	$N + q_8$	$Y + q_7$	$N + q_7$	$* + q_7$	$\$ + q_7$
q_8	$b - q_9$				
q_9	$N - q_3$	$Y - q_9$	$N - q_9$	$* - q_9$	$\$ - q_9$
q_{10}		$Y + q_{10}$	$N + q_{10}$	$* + q_{10}$	$\$ + q_{11}$
q_{11}		$Y + q_{11}$	$N + q_{11}$	$* + q_{11}$	$Y + q_0$
q_{12}		$Y + q_{12}$	$N + q_{12}$	$* + q_{12}$	$\$ - q_3$
q_{13}	$* - q_{14}$	$Y - q_{13}$	$N - q_{13}$	$* - q_{13}$	$\$ - q_{13}$
q_{14}	$b + q_{16}$	$Y - q_{14}$	$N - q_{14}$	$b + q_{15}$	
q_{15}	$N + q_0$	$Y + q_{15}$	$N + q_{15}$	$* + q_{15}$	$\$ + q_{15}$
q_{16}		$Y + q_{16}$	$N + q_{16}$	$* + q_{16}$	$\$ - q_{14}$

Simulating the CTAG C_1 by the URTM(17,5)

$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\,NYN, \varepsilon, \varepsilon, \varepsilon)),$$

Initial ID: $(NNY\ \ NYN,\ 0)$

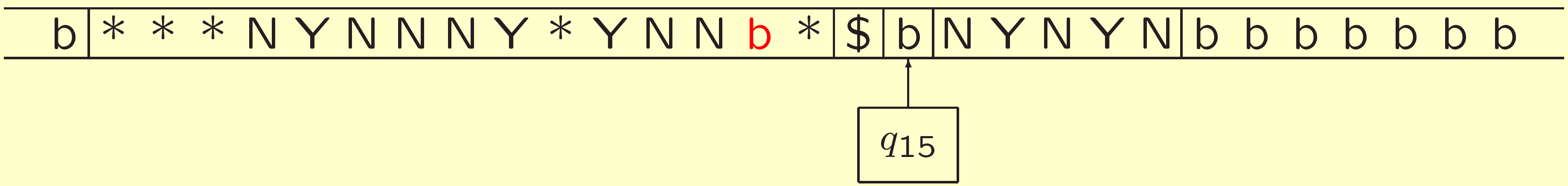


Simulating the CTAG C_1 by the URTM(17,5)

$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\,NYN, \varepsilon, \varepsilon, \varepsilon)),$$

$$(NNY\ \ NYN,\ 0)\ \Rightarrow\ (NY\ \ NYN,\ 1)$$

$t = 6$

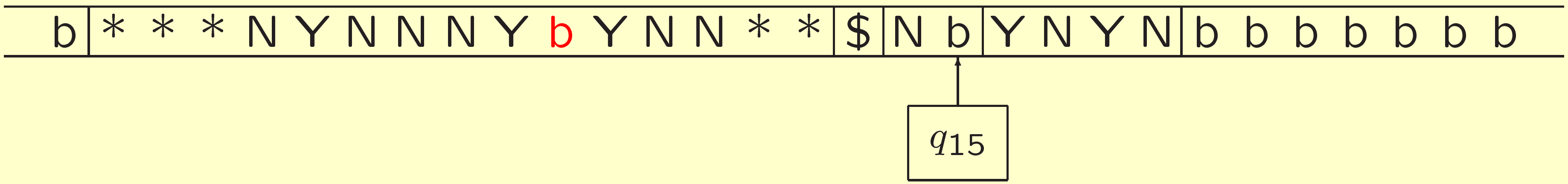


Simulating the CTAG C_1 by the URTM(17,5)

$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\,NYN, \varepsilon, \varepsilon, \varepsilon)),$$

$$(NY\ \ NYN,\ 1) \Rightarrow\ (Y\ \ NYN,\ 2)$$

$t = 23$



Simulating the CTAG C_1 by the URTM(17,5)

$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\,NYN, \varepsilon, \varepsilon, \varepsilon)),$$

$$(Y \;\; NYN,\;\; 2) \;\Rightarrow\; (NYN \;\; YNN \;\; NYN,\;\; {\color{red}3})$$

$t = 298$

b	* * b N Y N N N Y * Y N N * *	$	N N $	N Y N Y N N N Y N	b

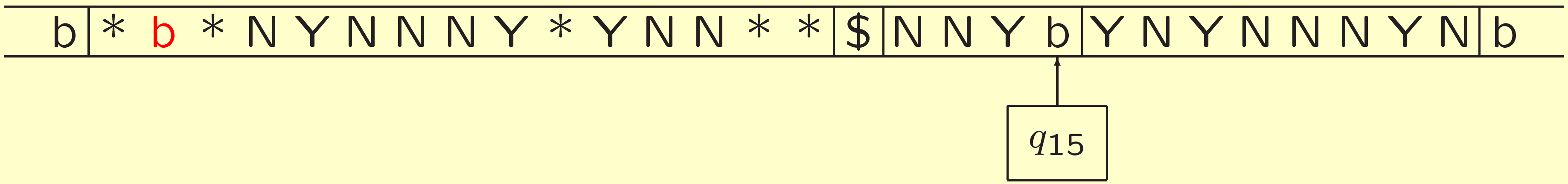
q_{11}

Simulating the CTAG C_1 by the URTM(17,5)

$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\,NYN, \varepsilon, \varepsilon, \varepsilon)),$$

$$(NYN\ YNN\ NYN,\ 3) \Rightarrow (YN\ YNN\ NYN,\ 4)$$

$t = 335$

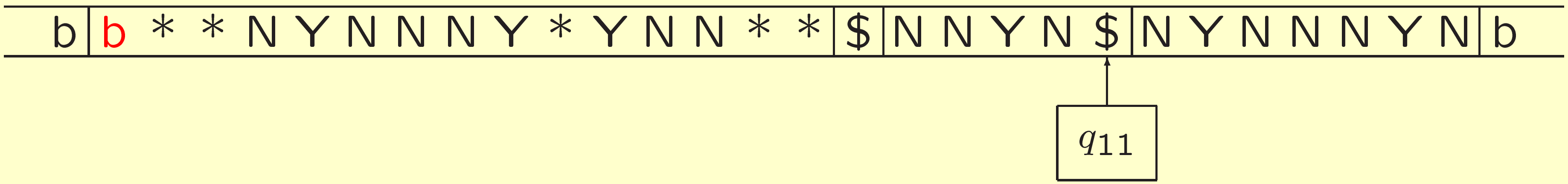


Simulating the CTAG C_1 by the URTM(17,5)

$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\,NYN, \varepsilon, \varepsilon, \varepsilon)),$$

$$(YN\ YNN\ NYN,\ 4) \Rightarrow (N\ YNN\ NYN,\ 5)$$

$t = 378$

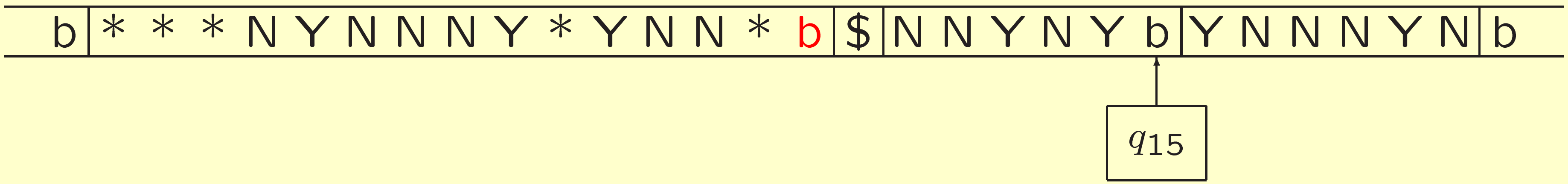


Simulating the CTAG C_1 by the URTM(17,5)

$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\ NYN, \varepsilon, \varepsilon, \varepsilon)),$$

$$(N\ \ YNN\ \ NYN,\ \ 5)\ \Rightarrow\ (YNN\ \ NYN,\ \ 0)$$

$t = 425$

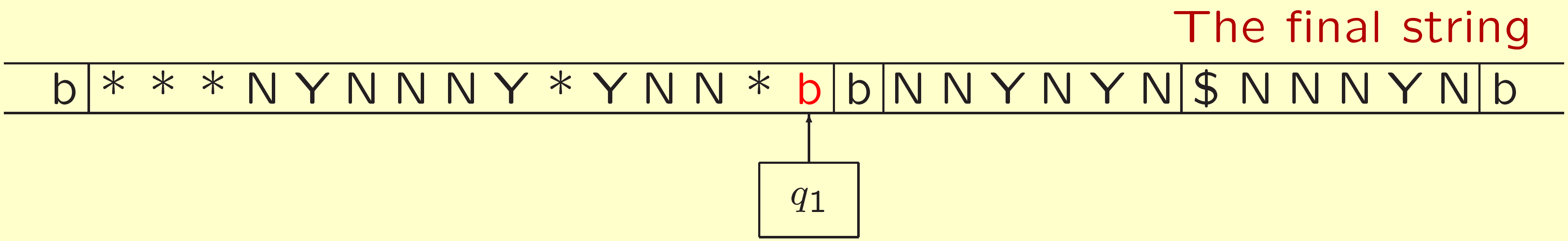


Simulating the CTAG C_1 by the URTM(17,5)

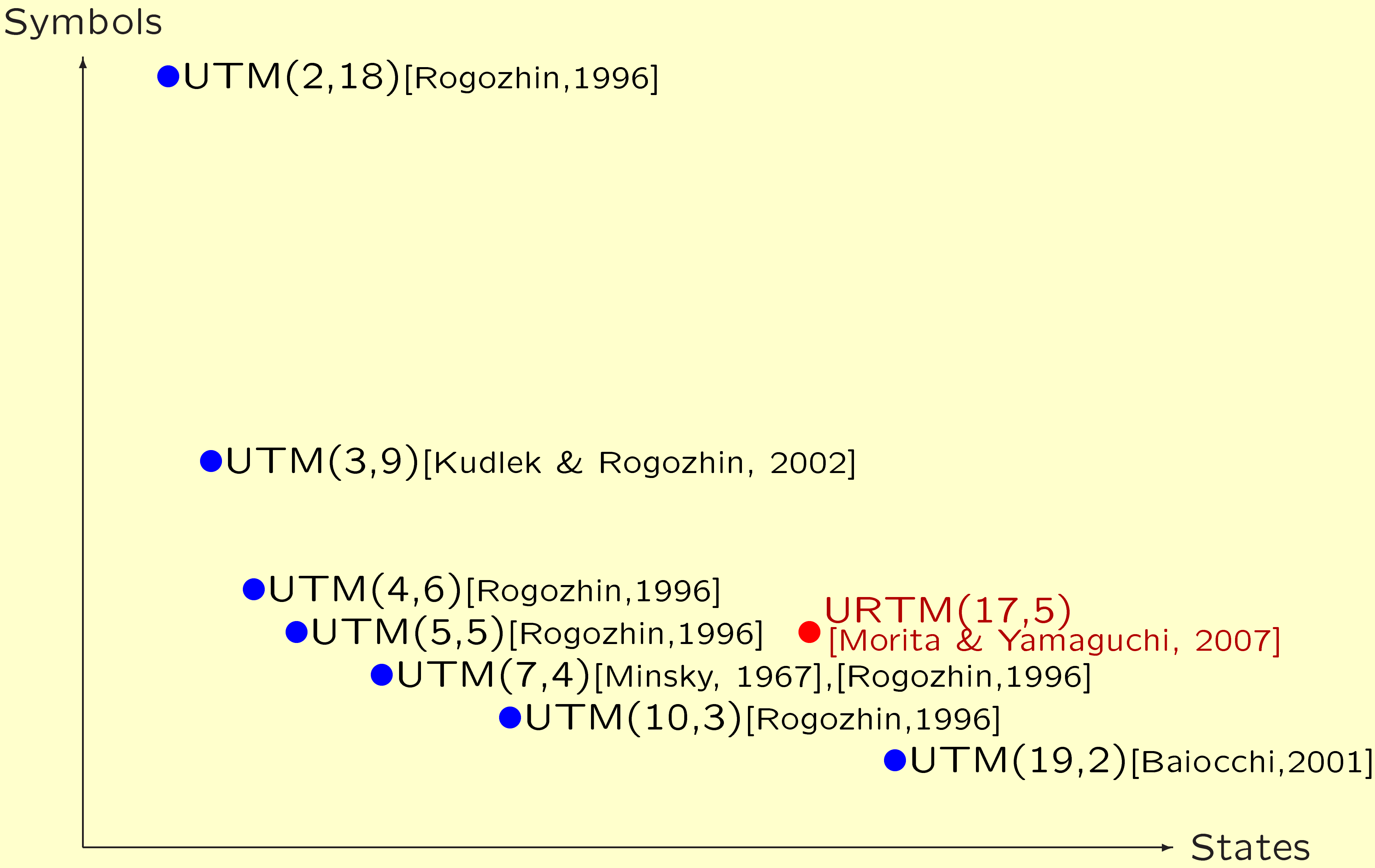
$$C_1 = (6, \{Y, N\}, (\text{halt}, NNY, YNN\,NYN, \varepsilon, \varepsilon, \varepsilon)),$$

$(YNN\ \ NYN,\ 0)$: Halting ID

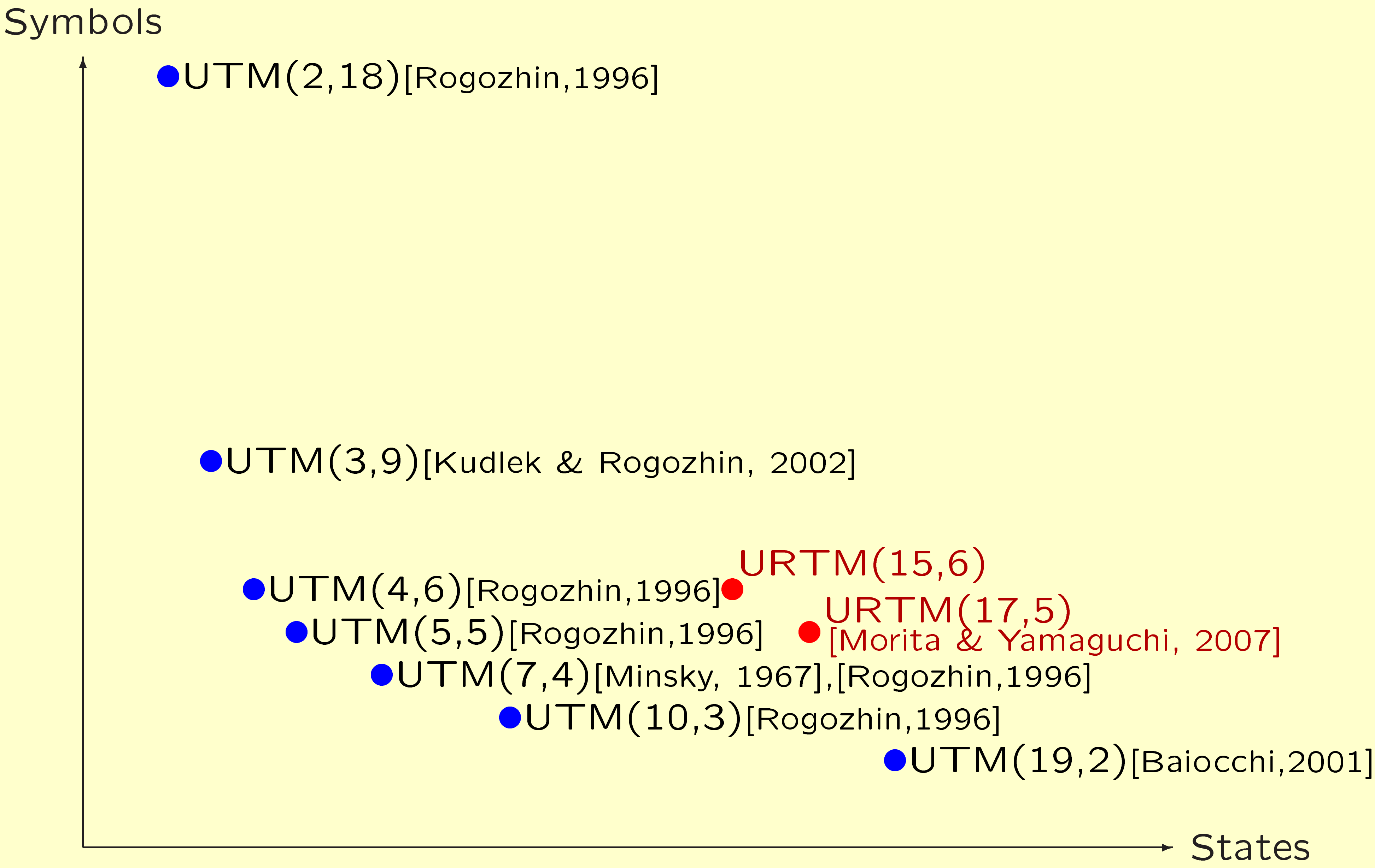
$t = 434$



Small UTMs and URTMs



Small UTMs and URTMs



Concluding Remarks

- We gave a 17-state 5-symbol URTM that simulates any cyclic tag system.
- We suppose the size of a URTM can be further reduced. Hence, this is a start point of the study of small URTMs.

Thank you!