Feder and Vardi's Non-Dichotomy Theorem Revisited

Alexey Barsukov

Toulouse, 31.03.2023

Table of Contents

Background

Monotone Monadic SNP without Inequality

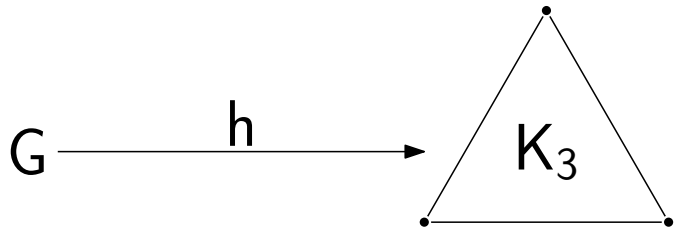
Definition

The MMSNP logic consists of ESO sentences of the form

$$\exists X_1, \ldots, X_s \; \forall x_1, \ldots, x_n \bigwedge_{i=1}^{m} \neg\big(\alpha_i \wedge \beta_i \wedge \varepsilon_i\big), \text{ where}$$

- every α_i is a conjunction of input atomic formulas,
- every β_i is a conjunction of existential atomic formulas,
- every ε_i is a conjunction of inequalities ($x_i \neq x_j$),
- all atomic formulas of α_i must be non negated (monotone),
- all existential relations $X_1, \ldots, X_s$ have arity 1 (monadic),
- every ε_i is empty (without inequality).

CSP is a Subclass of MMSNP

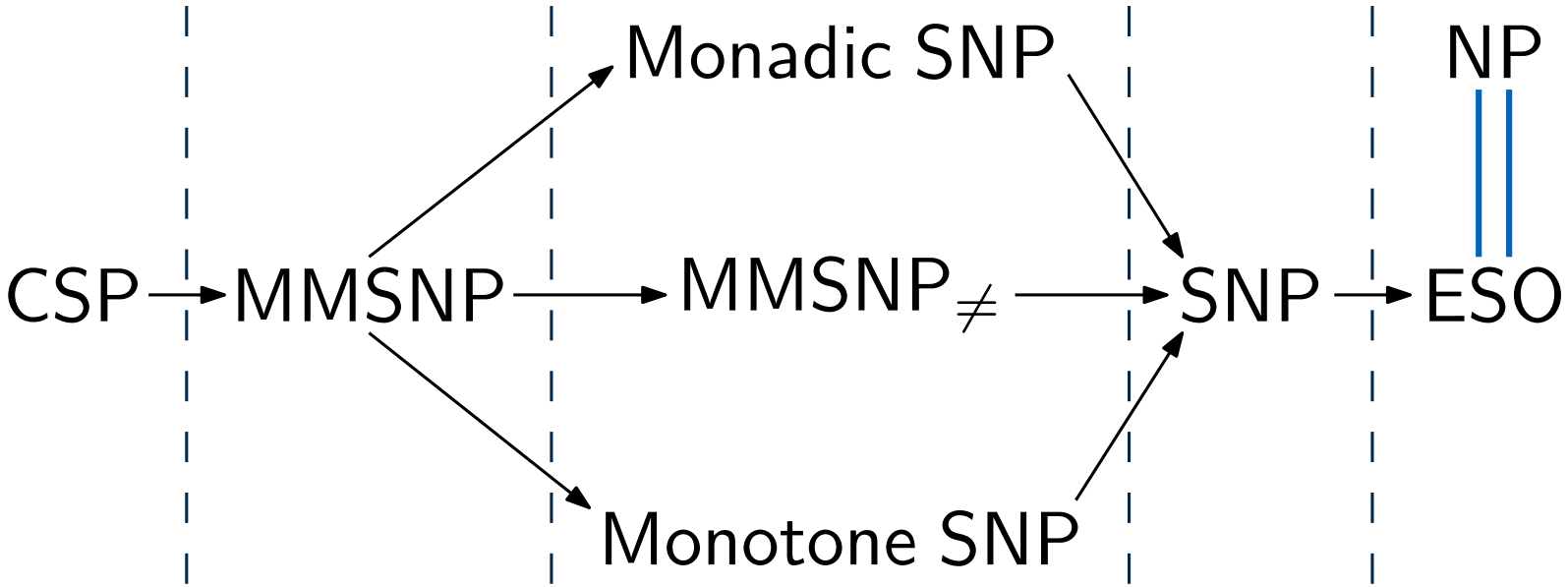


Example

The decision problem of mapping a digraph to K_3 is described by the following MMSNP sentence:

$$\exists R, G, B\ \forall x, y$$
$$\neg\big(\neg R(x) \wedge \neg G(x) \wedge \neg B(x)\big) \wedge$$
$$\neg\big(R(x) \wedge G(x)\big) \wedge \neg\big(G(x) \wedge B(x)\big) \wedge \neg\big(B(x) \wedge R(x)\big) \wedge$$
$$\neg\big(E(x,y) \wedge R(x) \wedge R(y)\big) \wedge$$
$$\neg\big(E(x,y) \wedge G(x) \wedge G(y)\big) \wedge \neg\big(E(x,y) \wedge B(x) \wedge B(y)\big)$$

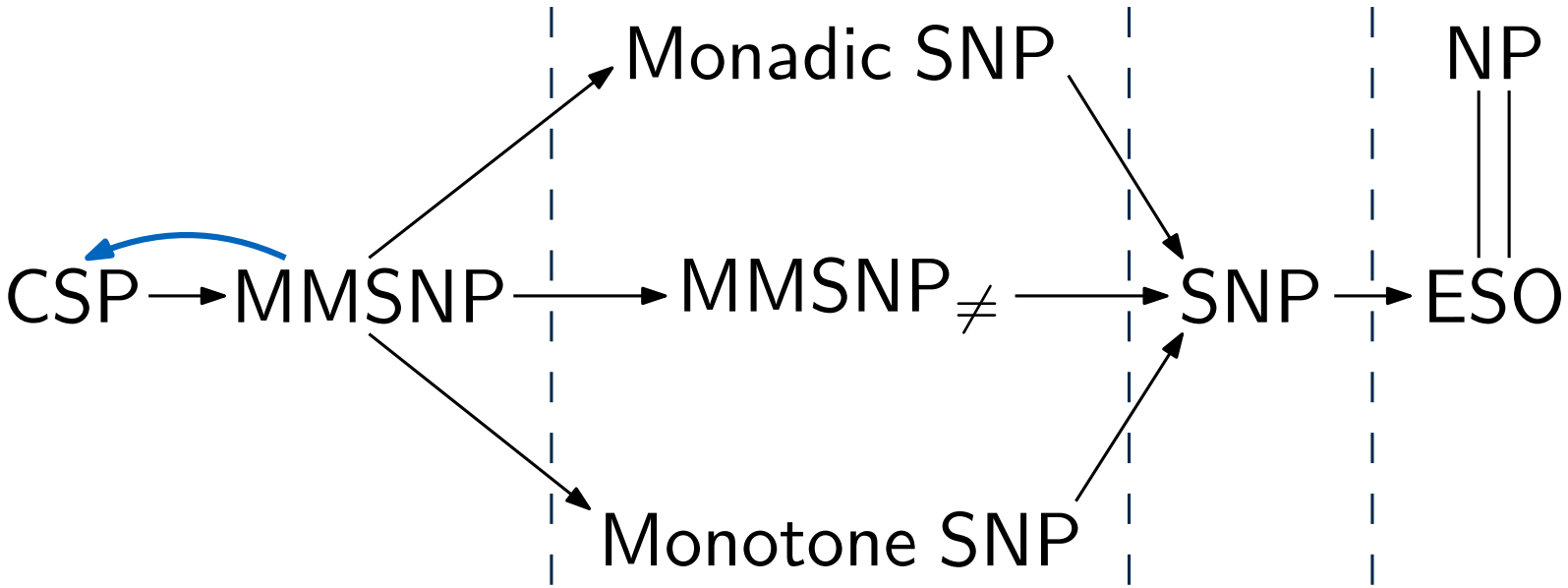
Fagin's Theorem



Theorem ([Fagin, 1974])

Any ESO sentence describes a problem in NP, and any NP problem can be described by an ESO sentence.

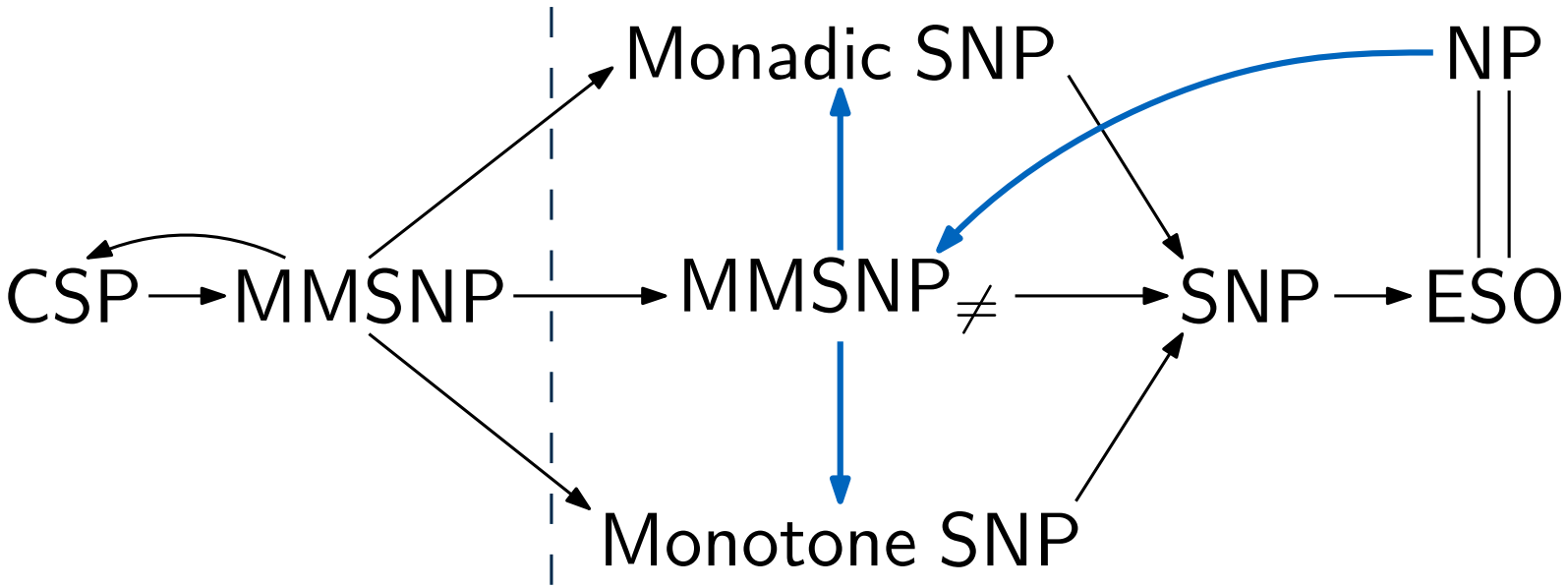
MMSNP and CSP



Theorem ([Feder, Vardi, 1998])

For any MMSNP problem there is an equivalent finite CSP problem up to polynomial time equivalences.

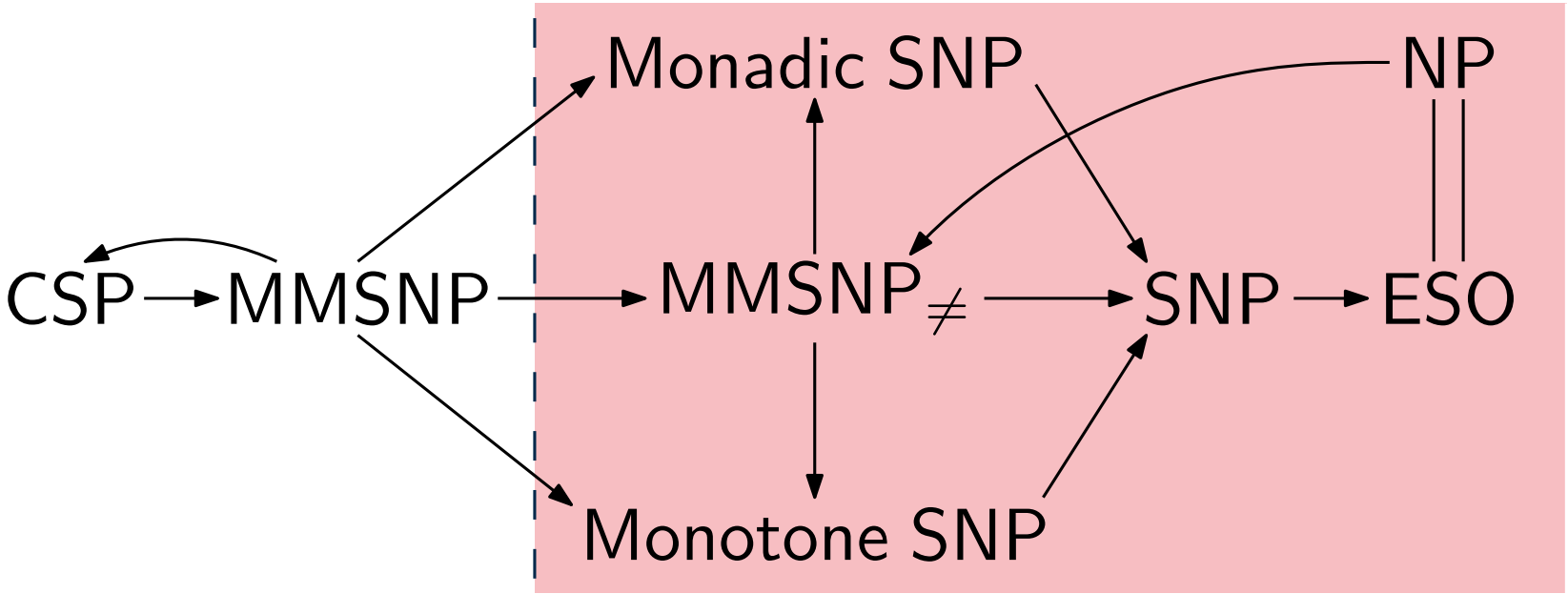
Superclasses of MMSNP



Theorem ([Feder, Vardi, 1998])

For any NP problem there are sentences in each of the three superclasses of MMSNP that describe Ptime equivalent problems.

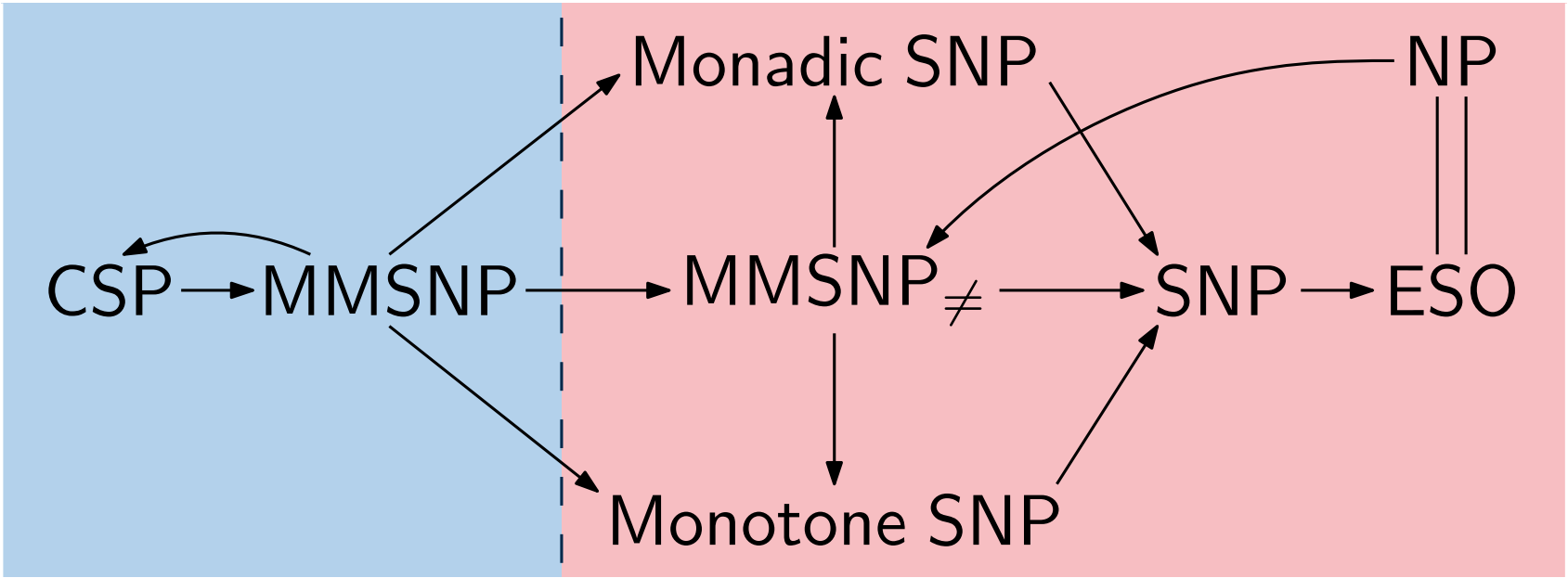
NP has no (P vs NP-complete) Dichotomy



Theorem ([Ladner, 1975])

If P≠NP, then NP contains a problem which is neither solvable in polynomial time nor NP-complete.

CSP has a (P vs NP-complete) Dichotomy

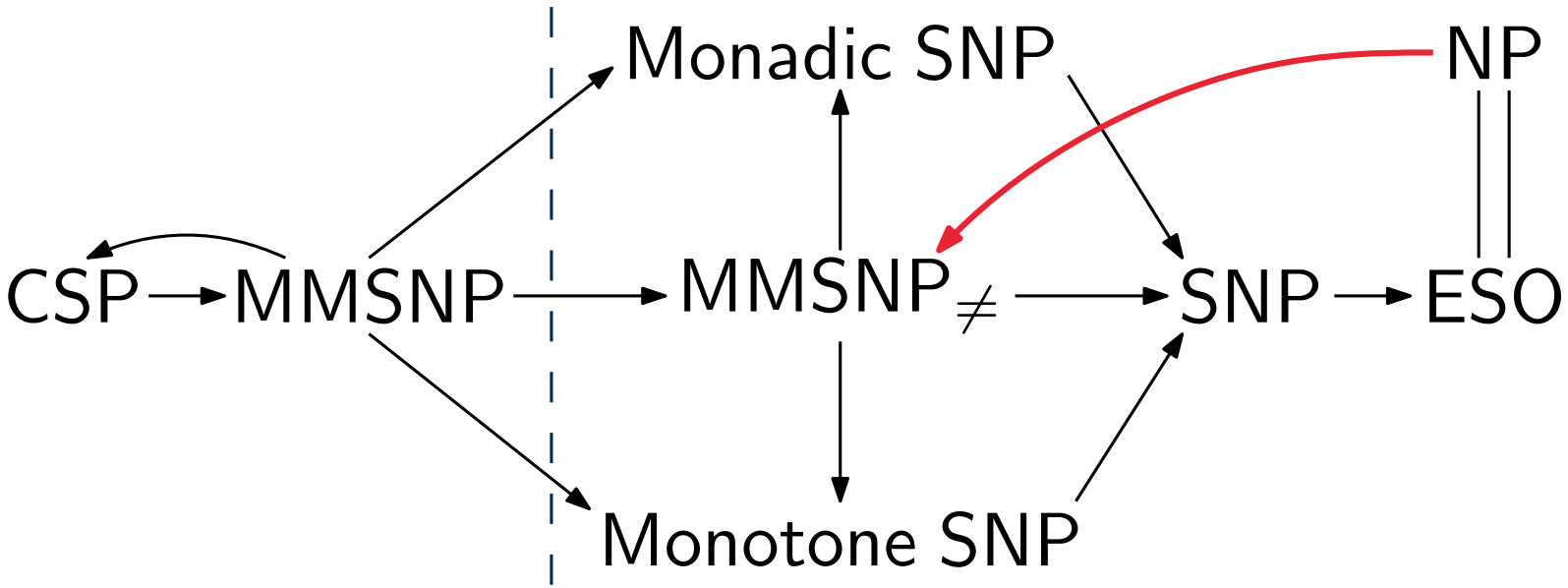


Theorem ([Zhuk, 2020])

Any finite CSP problem is either solvable in polynomial time or NP-complete.

Feder and Vardi's Theorem

MMSNP with Inequality



Theorem ([Feder, Vardi, 1998])

For any NP problem there is a sentence in MMSNP with inequality that describes a polynomial time equivalent problem.

Oblivious Turing Machines

Definition

A (non deterministic) Turing machine M is oblivious if the head movement depends only on the input size, *i.e.*, if for any two input strings s_1, s_2 of the same length,

- the execution times for s_1, s_2 are the same and
- at any moment t of the execution for s_1, the head of M is at the same position as at the moment t of the execution for s_2.

Theorem ([Pippenger, Fischer, 1979])

Any non deterministic Turing machine can be simulated by an oblivious Turing machine.

Proof Sketch

Proof.

Any oblivious Turing machine is simulated by an $\text{MMSNP}_{\neq}$ sentence such that

- the movement of the head is given as part of the input,
- only the states of the machine and cell values used during the computation are quantified existentially,
- it rejects an instance if it does not describe an input followed by the correct movement of the head,
- it accepts the instance if the number of cells allowed for the computation is smaller than the execution time,
- otherwise it accepts precisely when the machine accepts.

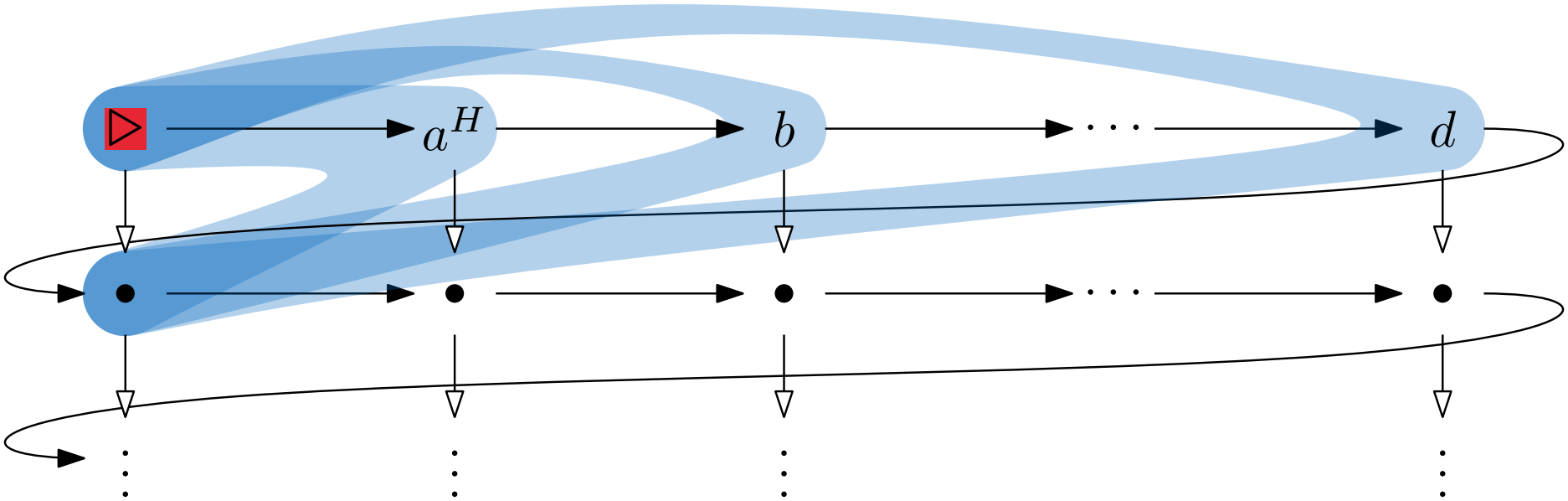
□

Holes in the Proof

- The precise composition (relations, construction) of the input is not provided.
- It is unclear how do they manage to reject an instance when the head movement is not correct.
- To sum up, Feder and Vardi showed that any NP problem can be reduced in polynomial time to a problem in $\text{MMSNP}_{\neq}$ but the backwards direction is unclear.

Construction

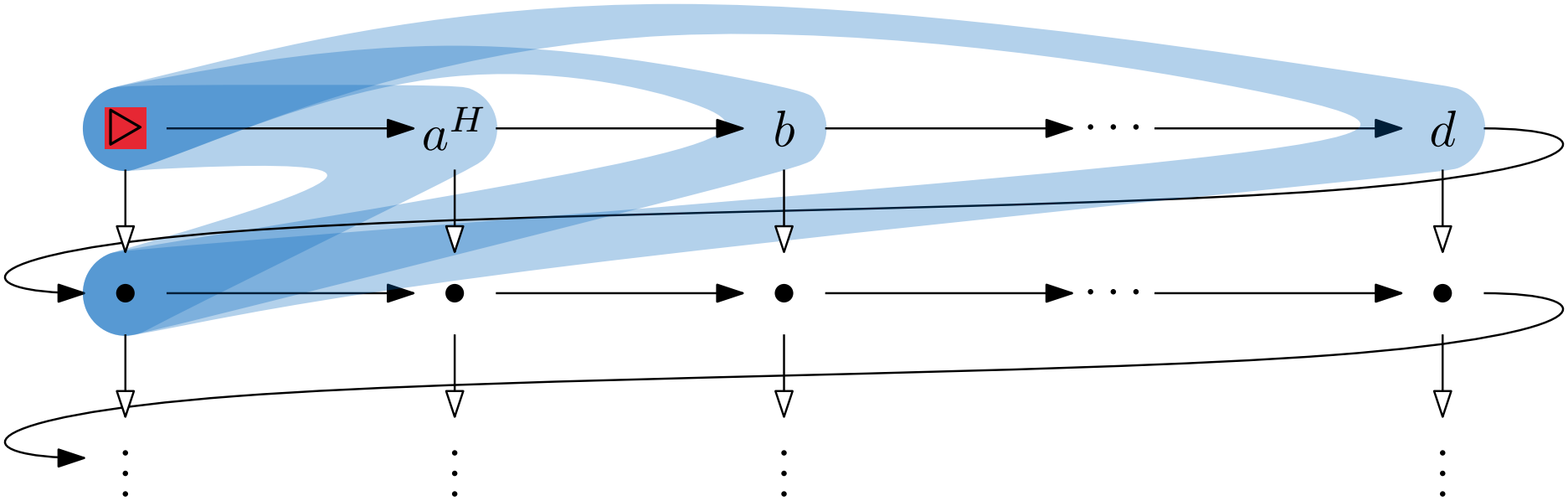
Input Relations



An input string is reduced to a two dimensional grid, where the ith line represents the string at the time i. The input relations are

- ■ – the constant highlighting the first element of the first line,
- for any $s \in \Sigma$ there are two input relations $s(\cdot), s^H(\cdot)$, the H-superscript keeps the head position of the original machine,

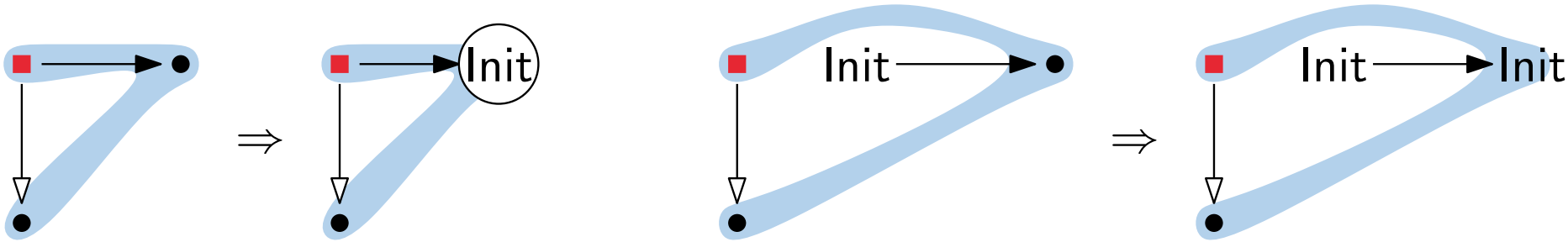
Input Relations



An input string is reduced to a two dimensional grid, where the ith line represents the string at the time i. The input relations are

- $E_{\rightarrow}, E_{\downarrow}$ – the grid arcs representing space and time,
- there is a ternary relation $row(\mathsf{x}, \mathsf{y}, \mathsf{z})$, it means that y is somewhere in the row that starts at x and ends at z,

Existential Relations



The Init elements represent the first row of the grid. The Head is positioned right next to ■.

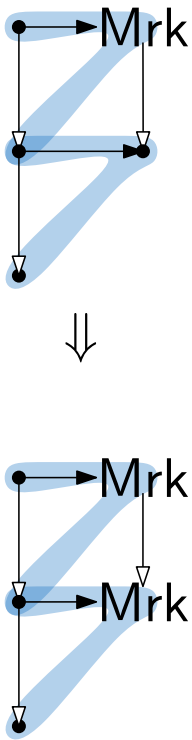
$$\neg\big(\blacksquare(\mathsf{x}) \wedge E_{\rightarrow}(\mathsf{x},\mathsf{y}) \wedge E_{\downarrow}(\mathsf{x},\mathsf{z}) \wedge row(\mathsf{x},\mathsf{y},\mathsf{z}) \wedge \neg\mathsf{Init}(\mathsf{y})\big)$$

$$\big(\blacksquare(\mathsf{x}) \wedge E_{\rightarrow}(\mathsf{x},\mathsf{y}) \wedge E_{\downarrow}(\mathsf{x},\mathsf{z}) \wedge row(\mathsf{x},\mathsf{y},\mathsf{z})\big) \Rightarrow \mathsf{Head}(\mathsf{y})$$

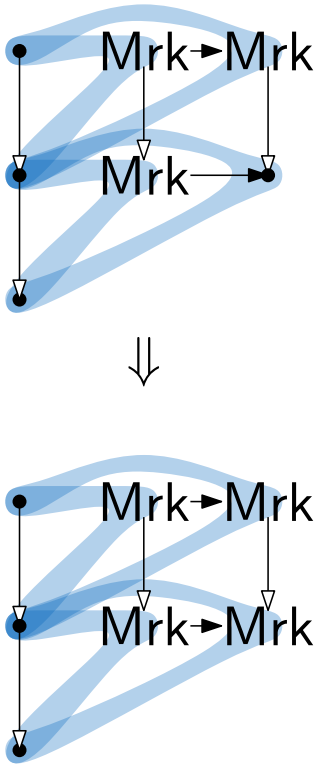
Existential Relations

Mrk is set by Init: $\big(\mathsf{Init}(x) \wedge s(x)\big) \Rightarrow \mathsf{Mrk}(x)$, and spreads like that:

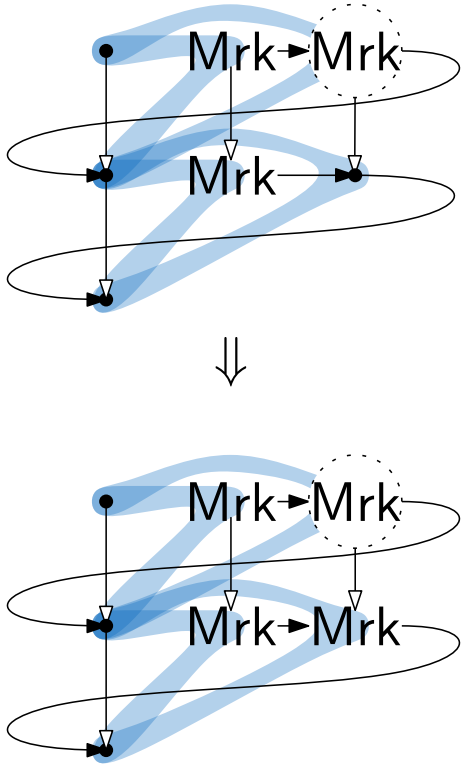
Left End



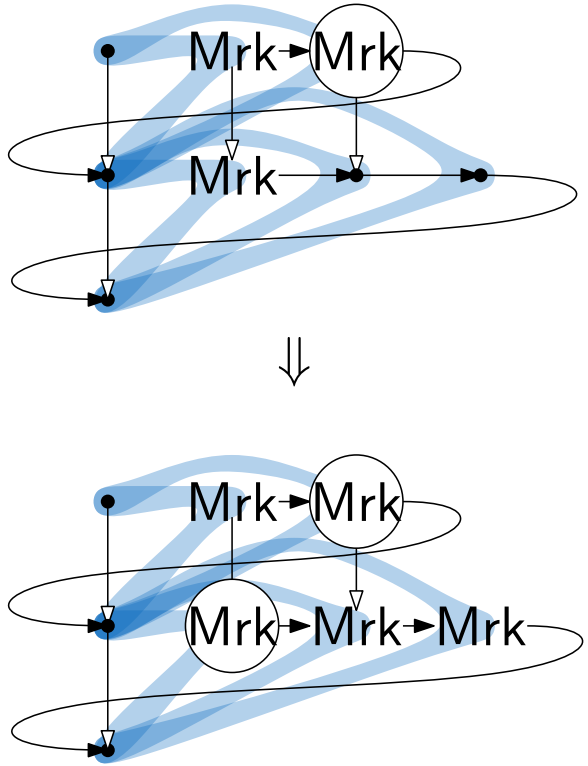
Middle



Right, No Head



Right, Head



Existential Relations

Head

The Head is obliged to move further only if it is at a Mrk element. It starts at the top-left corner and moves bottom-right until no longer possible, then starts moving bottom-left, etc.

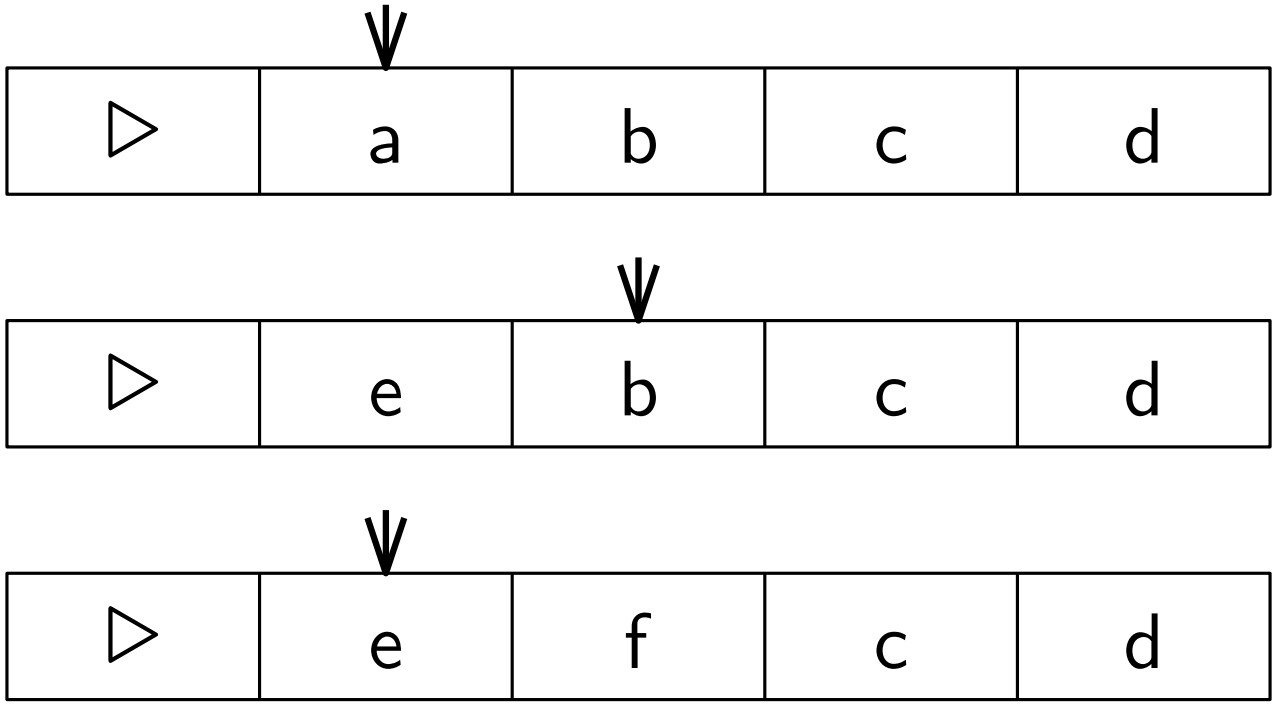
Symbols of the Alphabet

For every input $s(\cdot)$ there is $\mathsf{s}(\cdot)$, it is launched by Init: $\big(s(\mathsf{x}) \wedge \mathsf{Init}(\mathsf{x})\big) \Rightarrow \mathsf{s}(\mathsf{x})$, and spreads down unless Head rewrites it.

Machine States

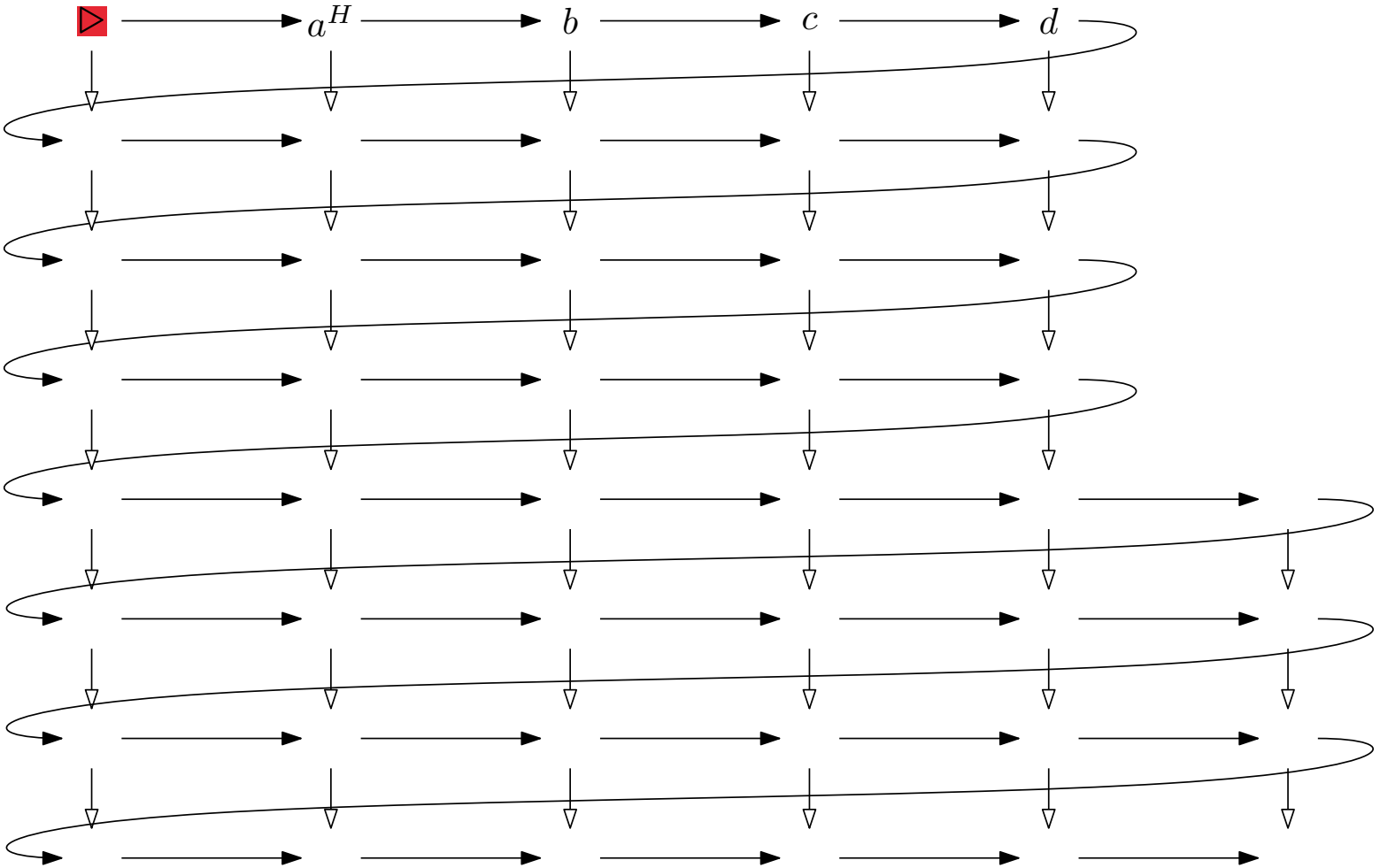
For every state q there is $\mathsf{q}(\cdot)$. All points of the same row must have the same state: $\big(row(x, y, z) \wedge row(x, y', z) \wedge \mathsf{q}(y)\big) \Rightarrow \mathsf{q}(y')$. The Head cannot be in a rejecting state: $\neg\big(\mathsf{Head}(\mathsf{x}) \wedge \mathsf{q}_{\mathsf{reject}}(\mathsf{x})\big)$.

Example of a Construction

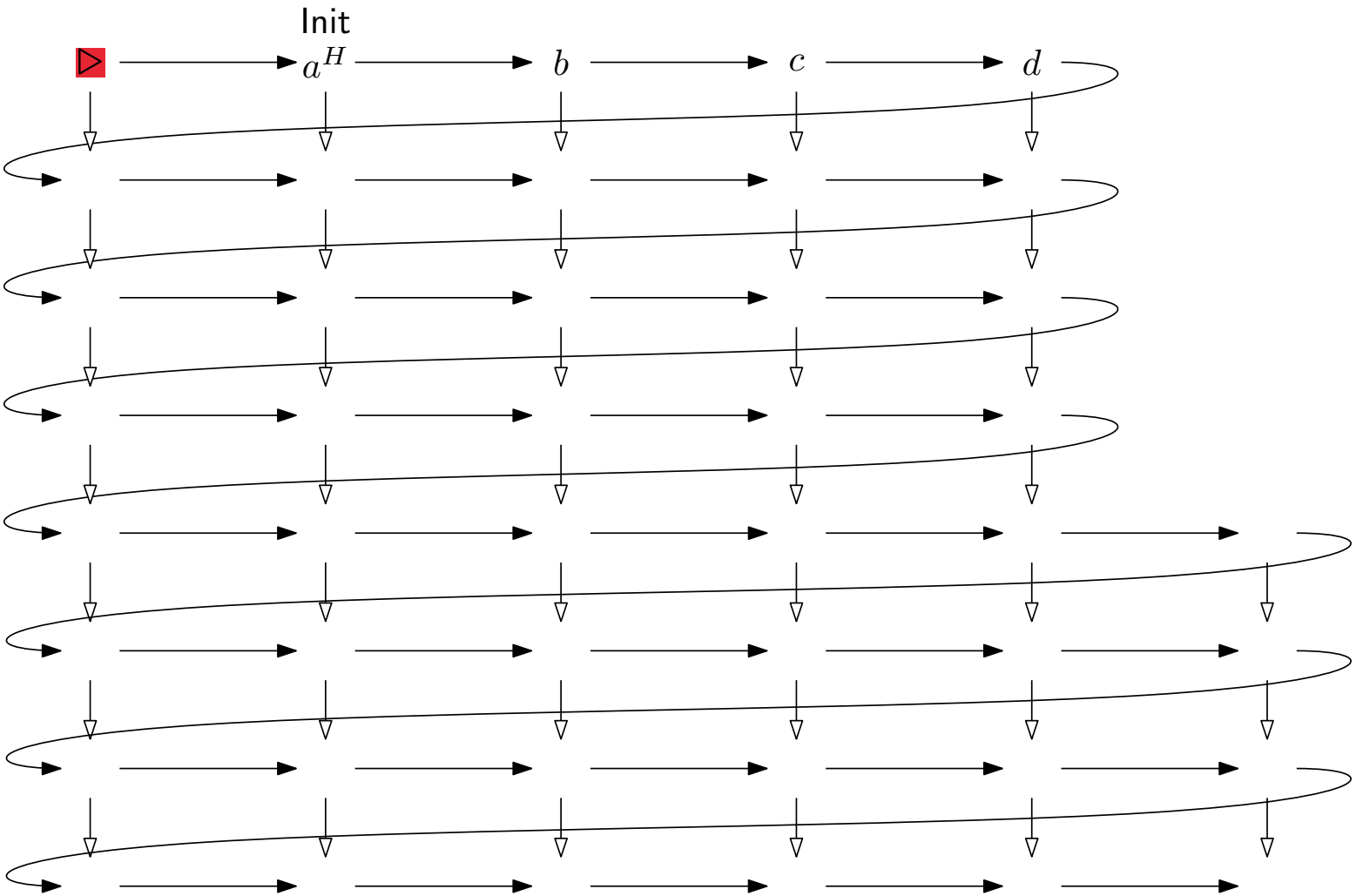


Let M be a Turing machine that does these two transitions.

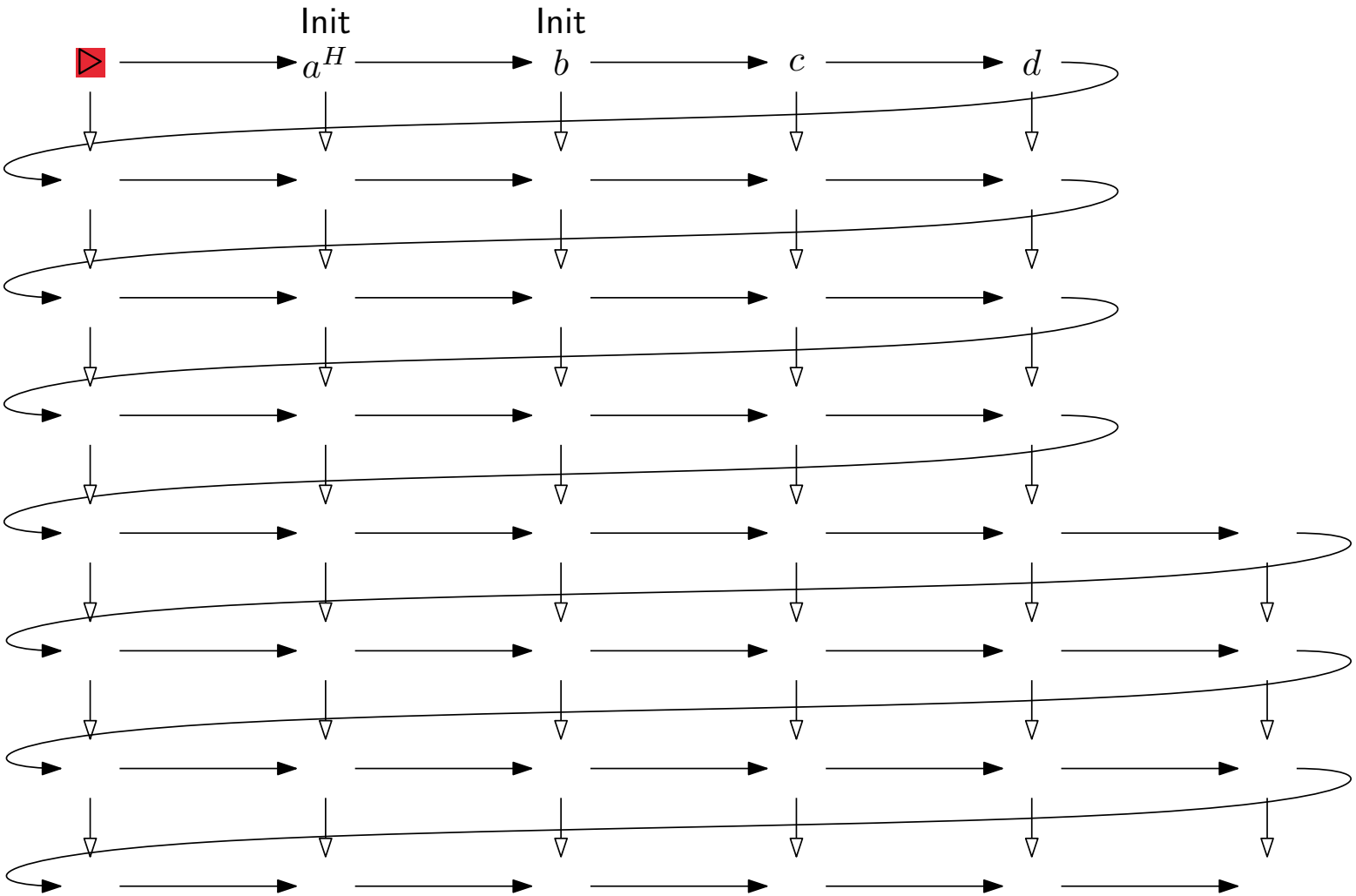
Example of a Construction



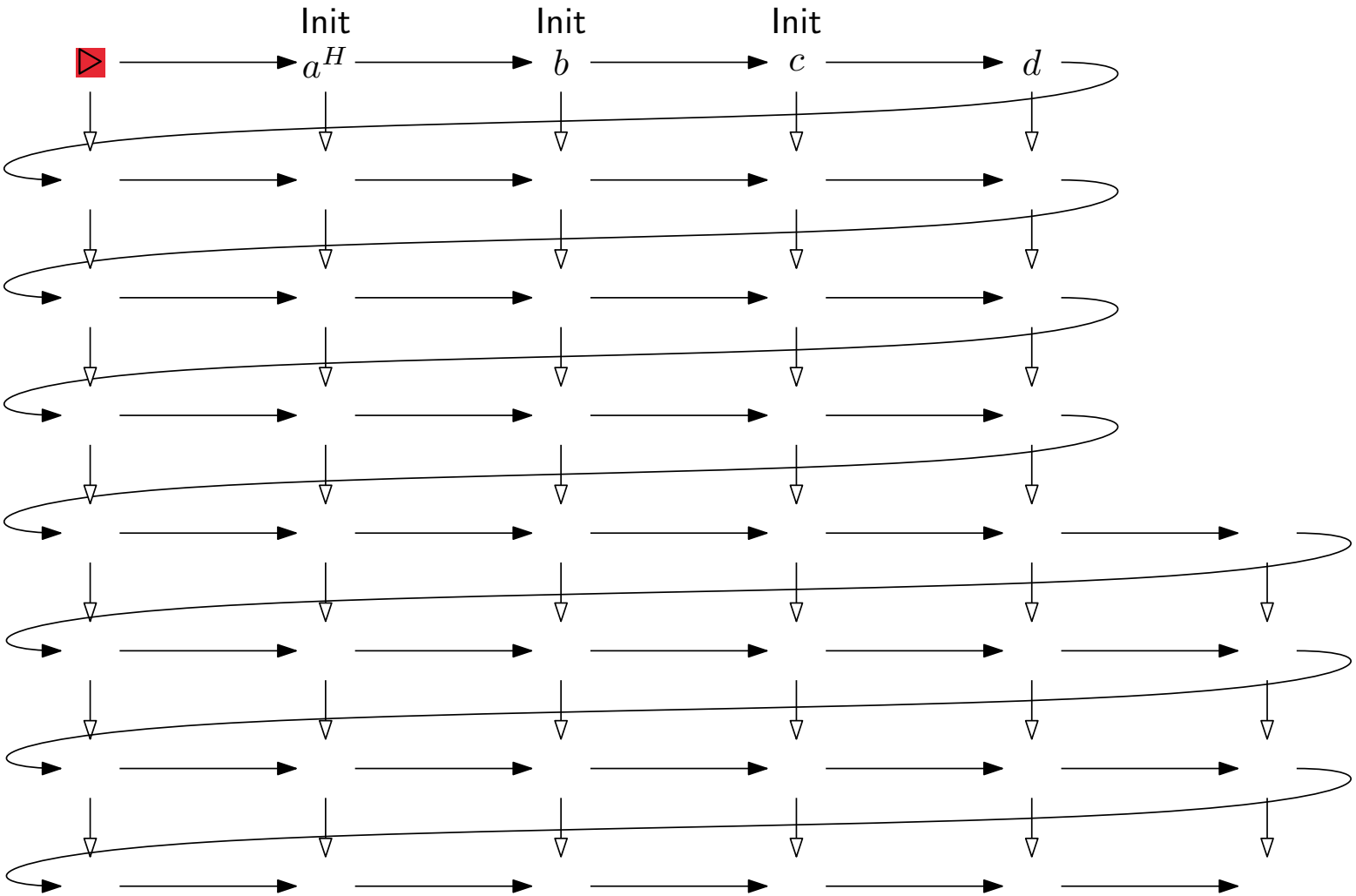
Example of a Construction



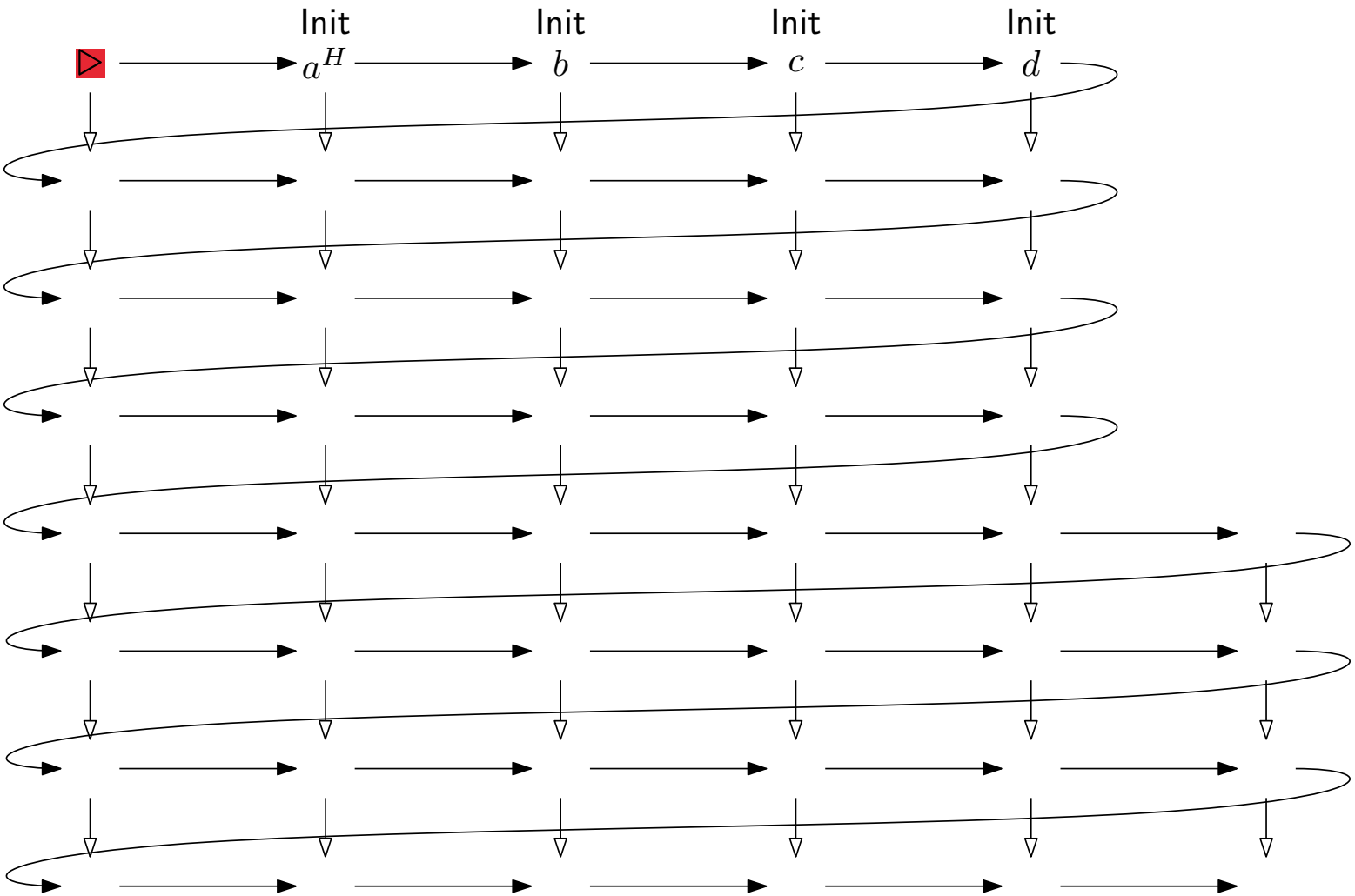
Example of a Construction



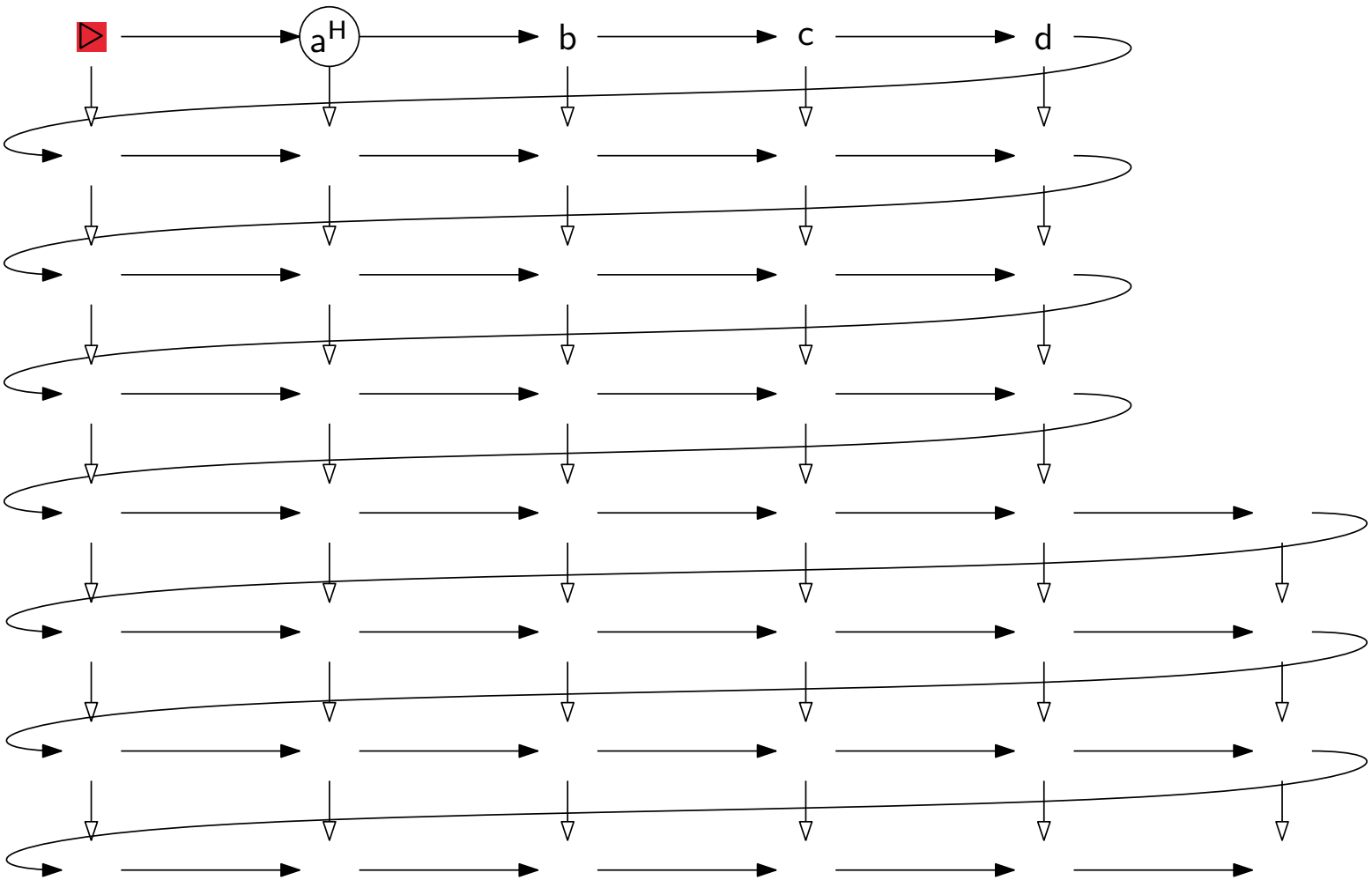
Example of a Construction



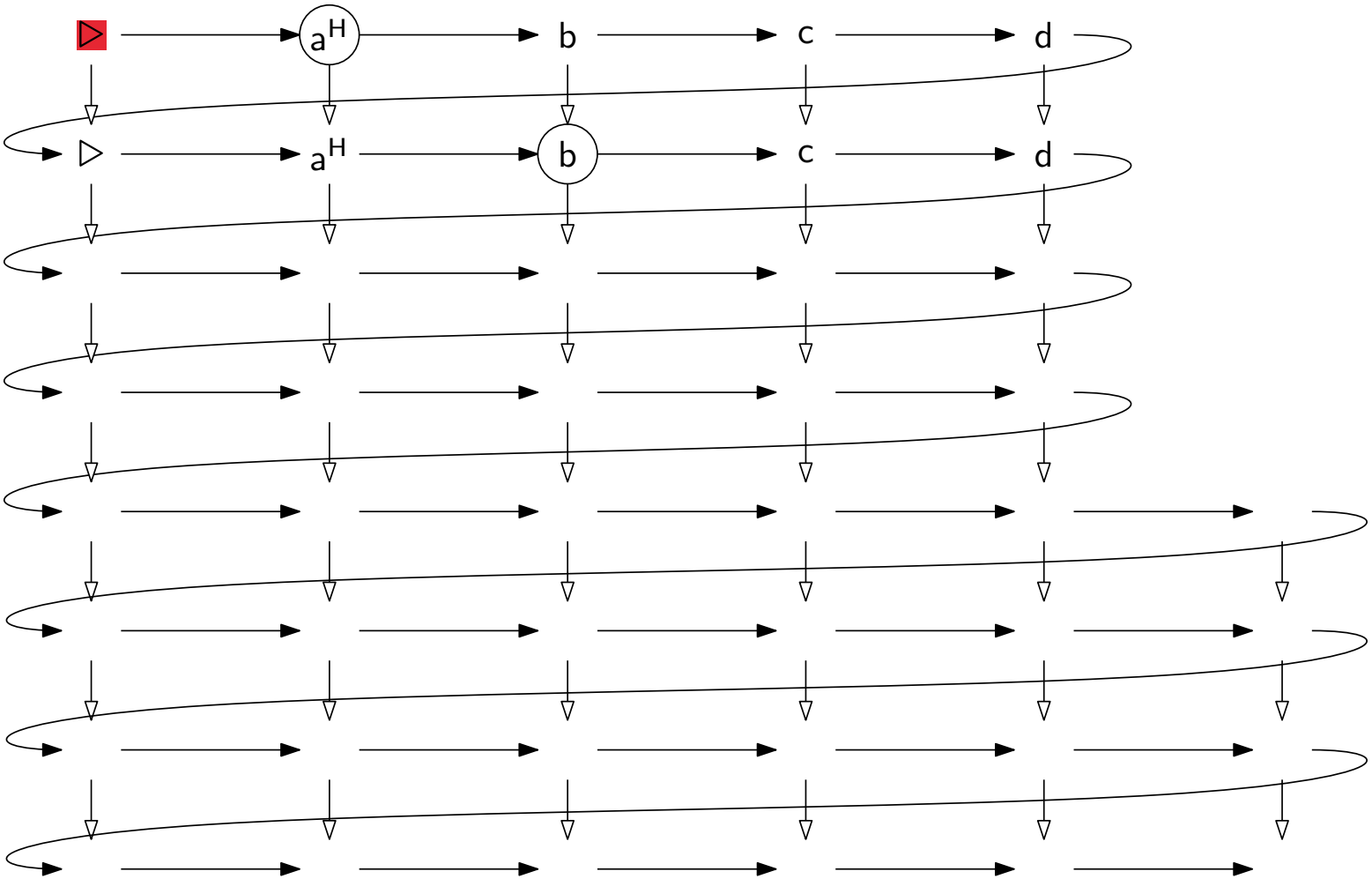
Example of a Construction



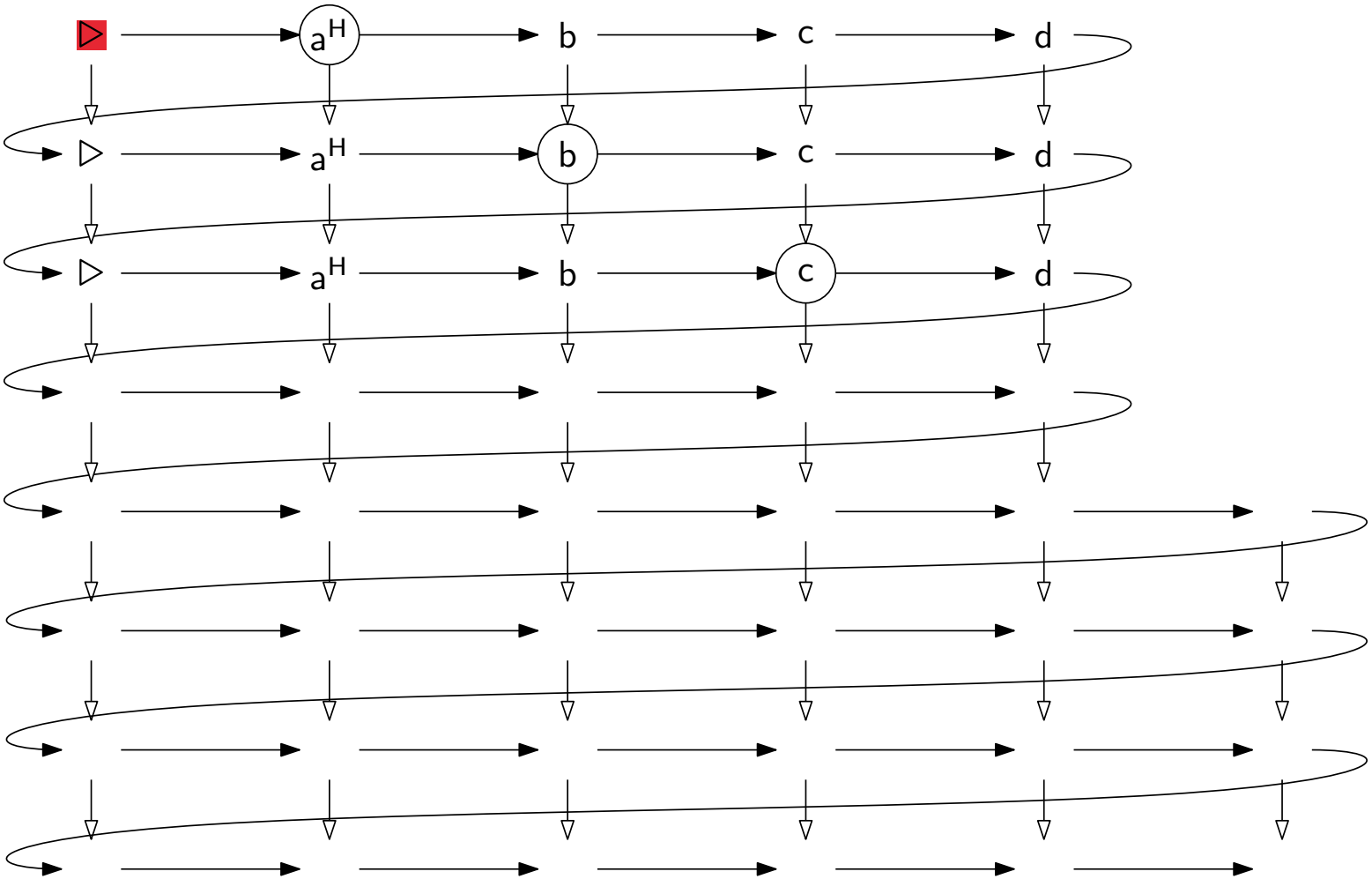
Example of a Construction



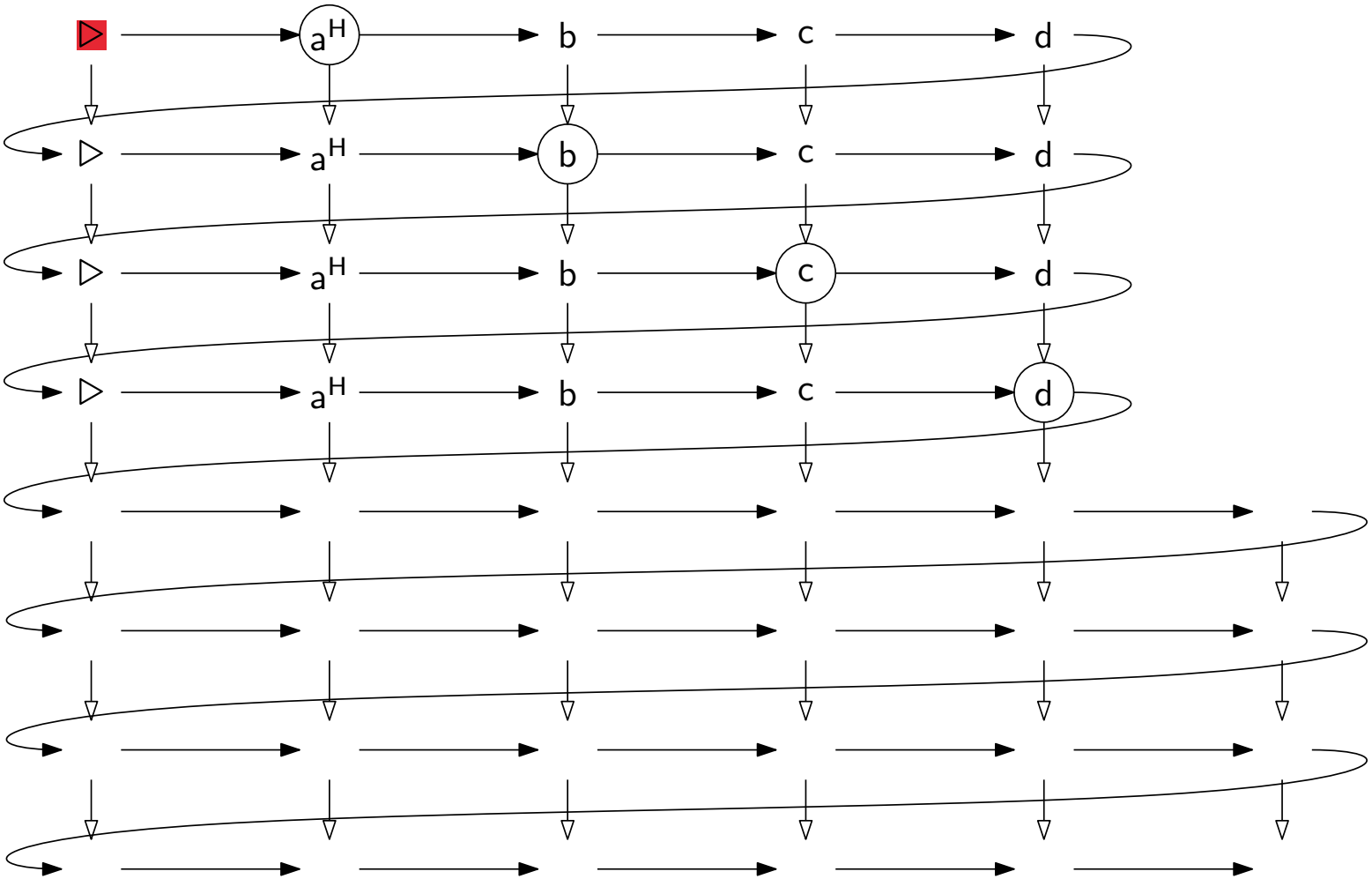
Example of a Construction



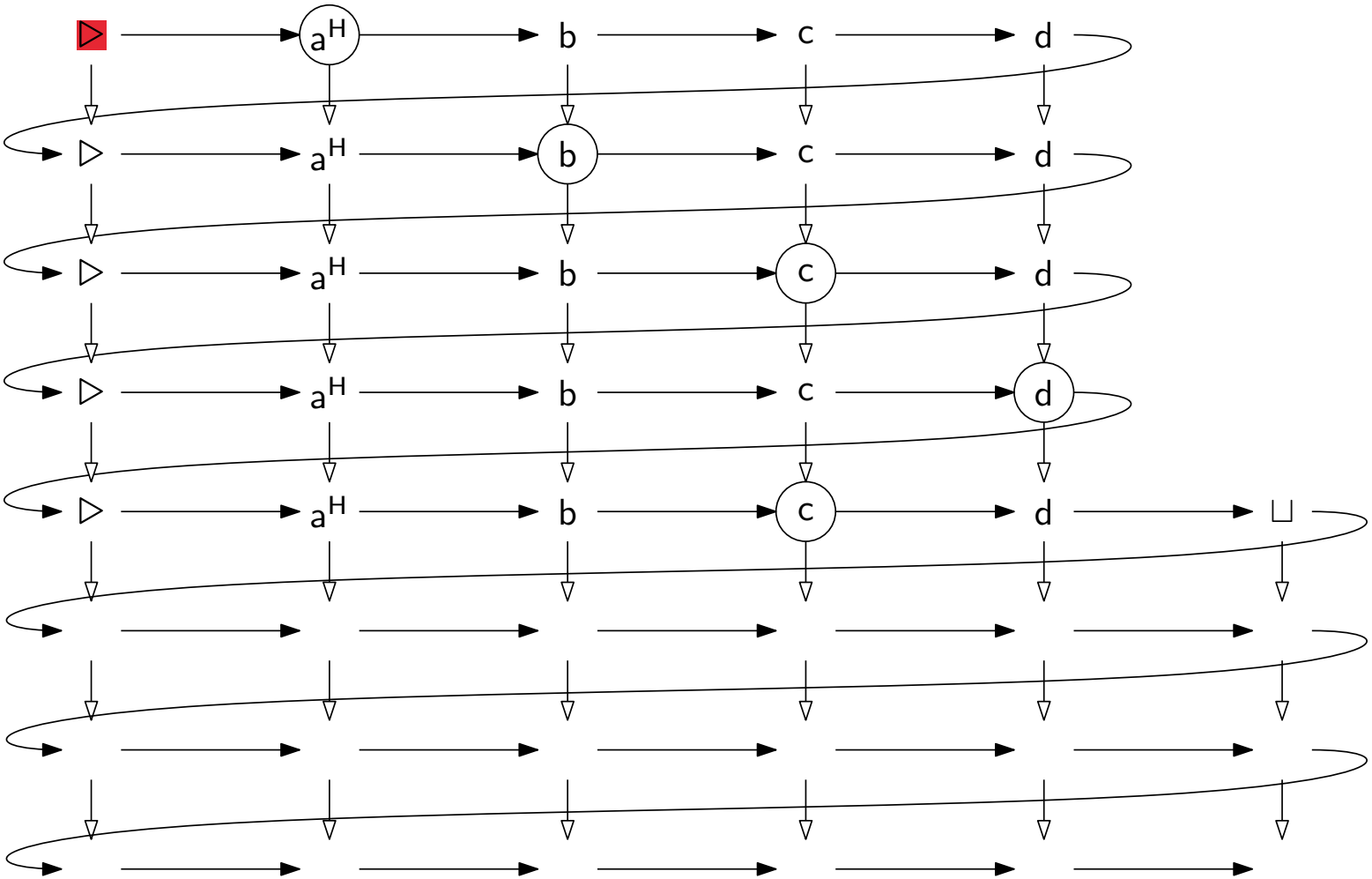
Example of a Construction



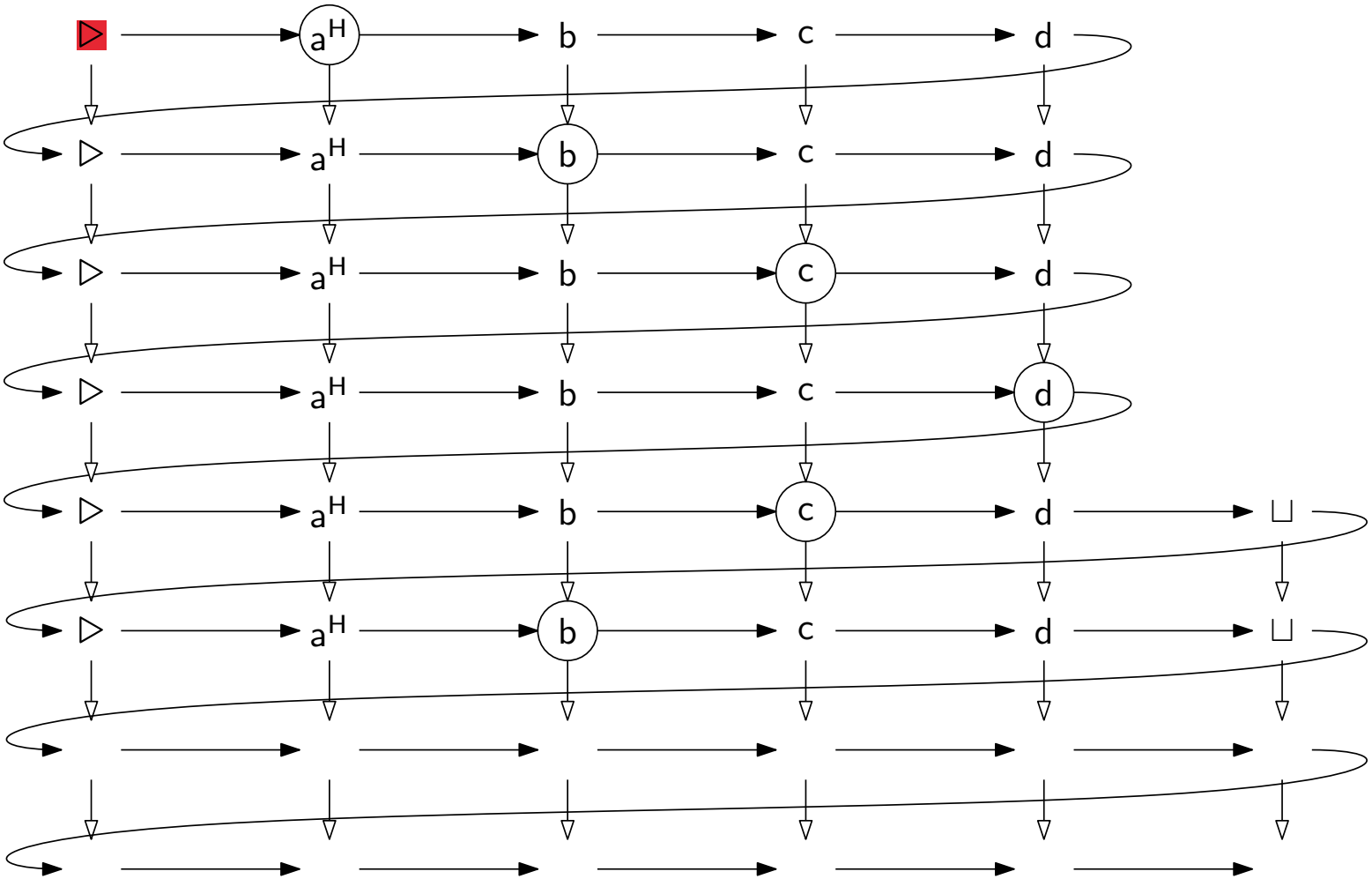
Example of a Construction



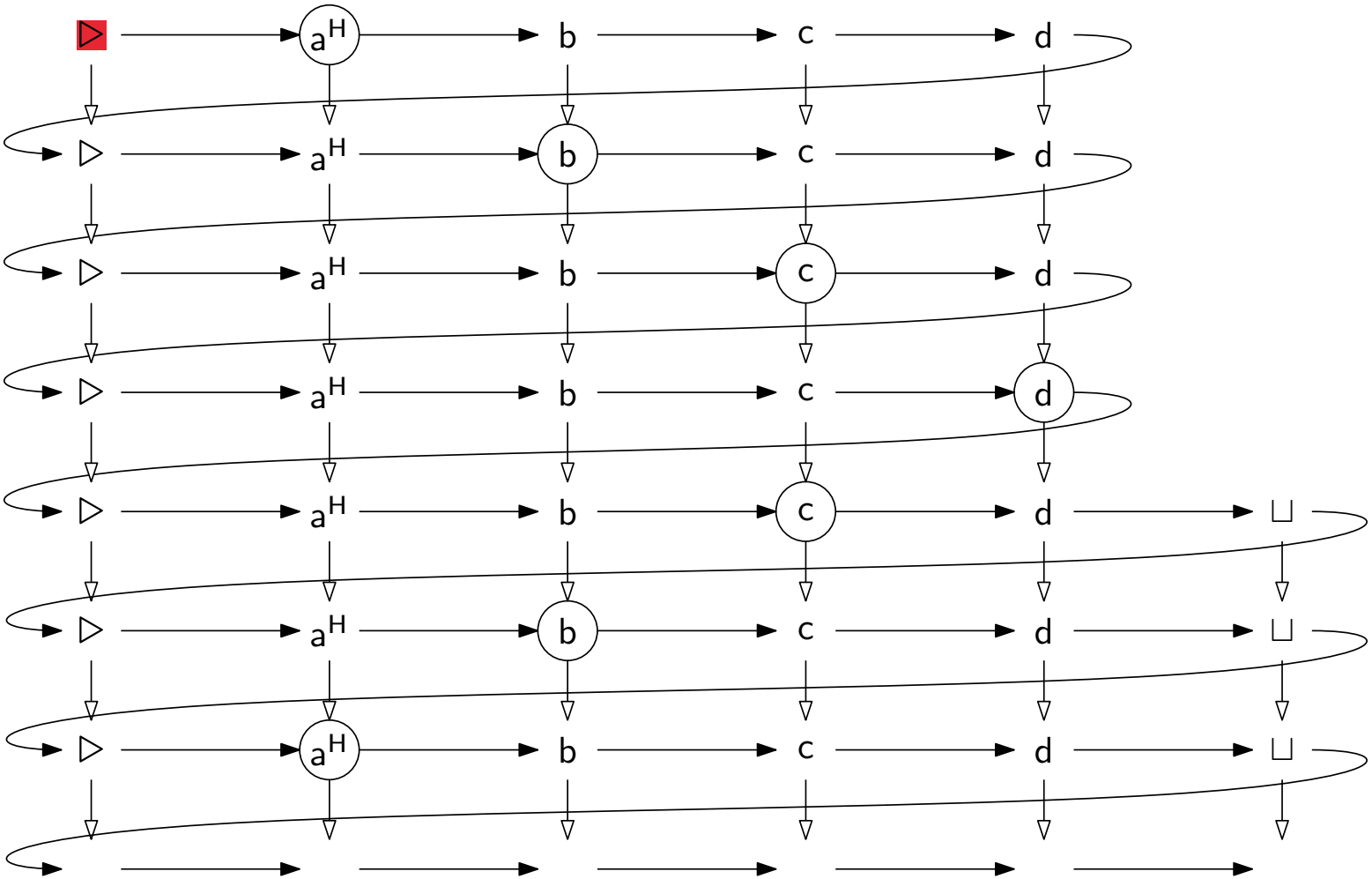
Example of a Construction



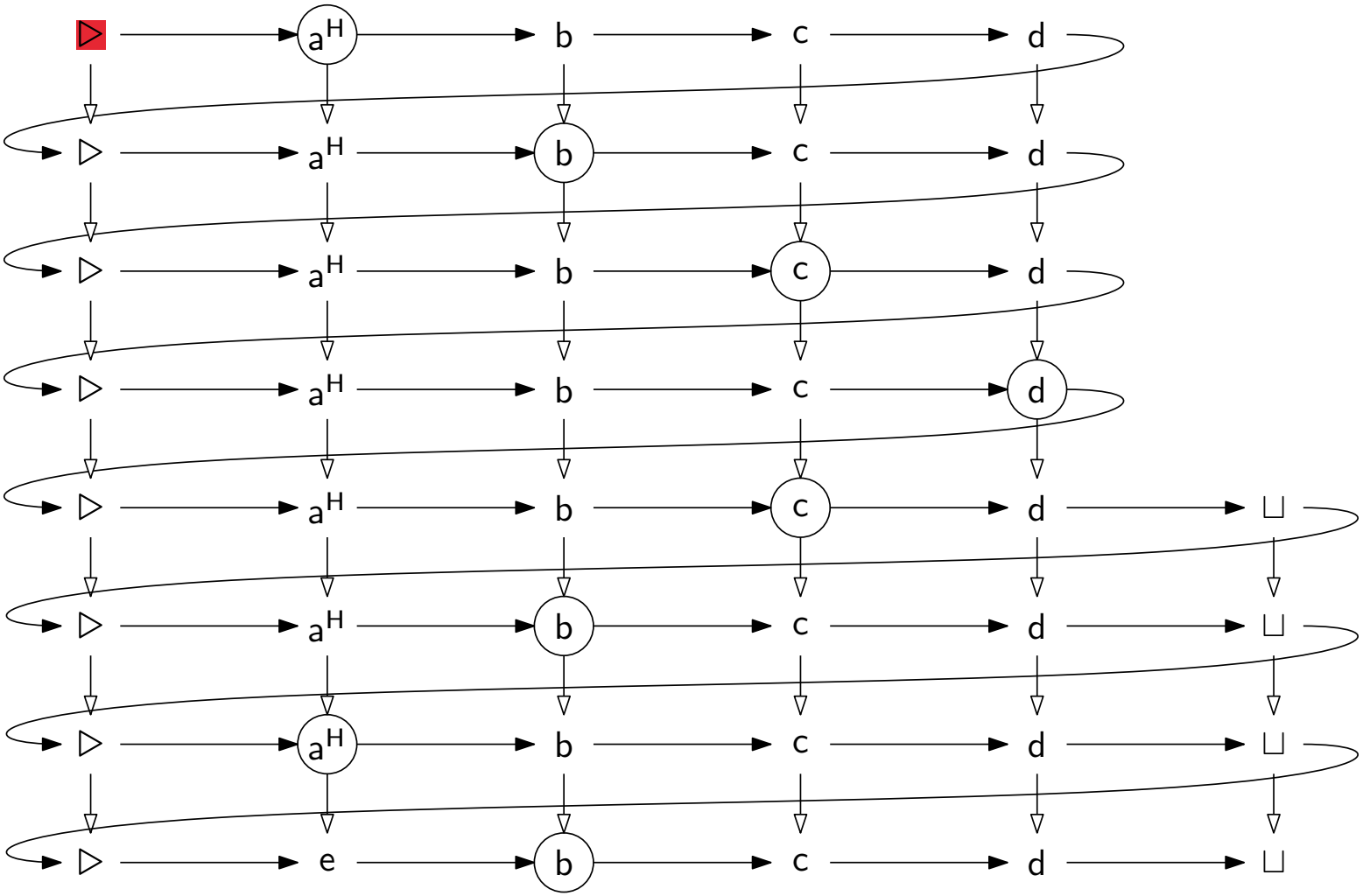
Example of a Construction



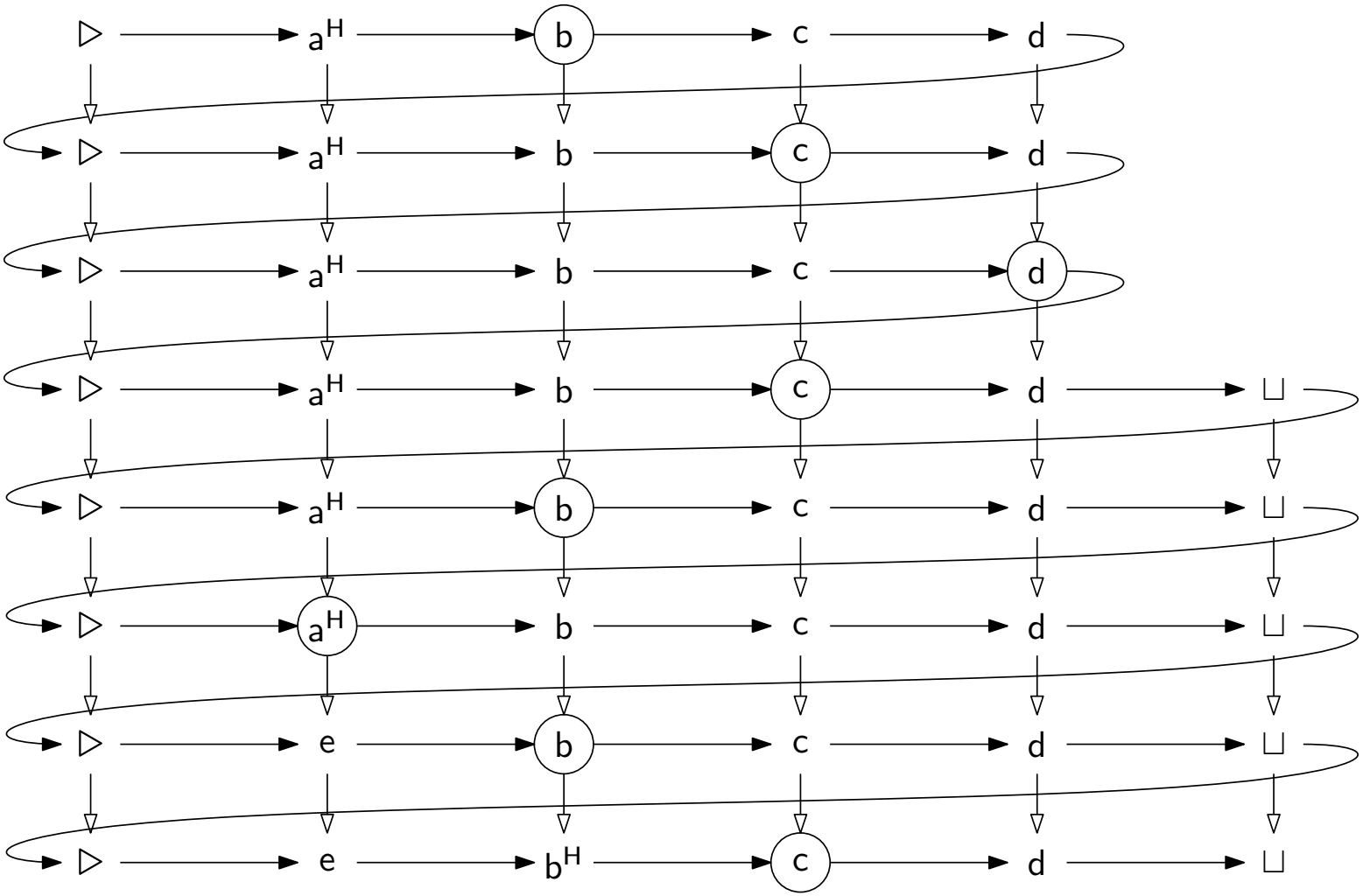
Example of a Construction



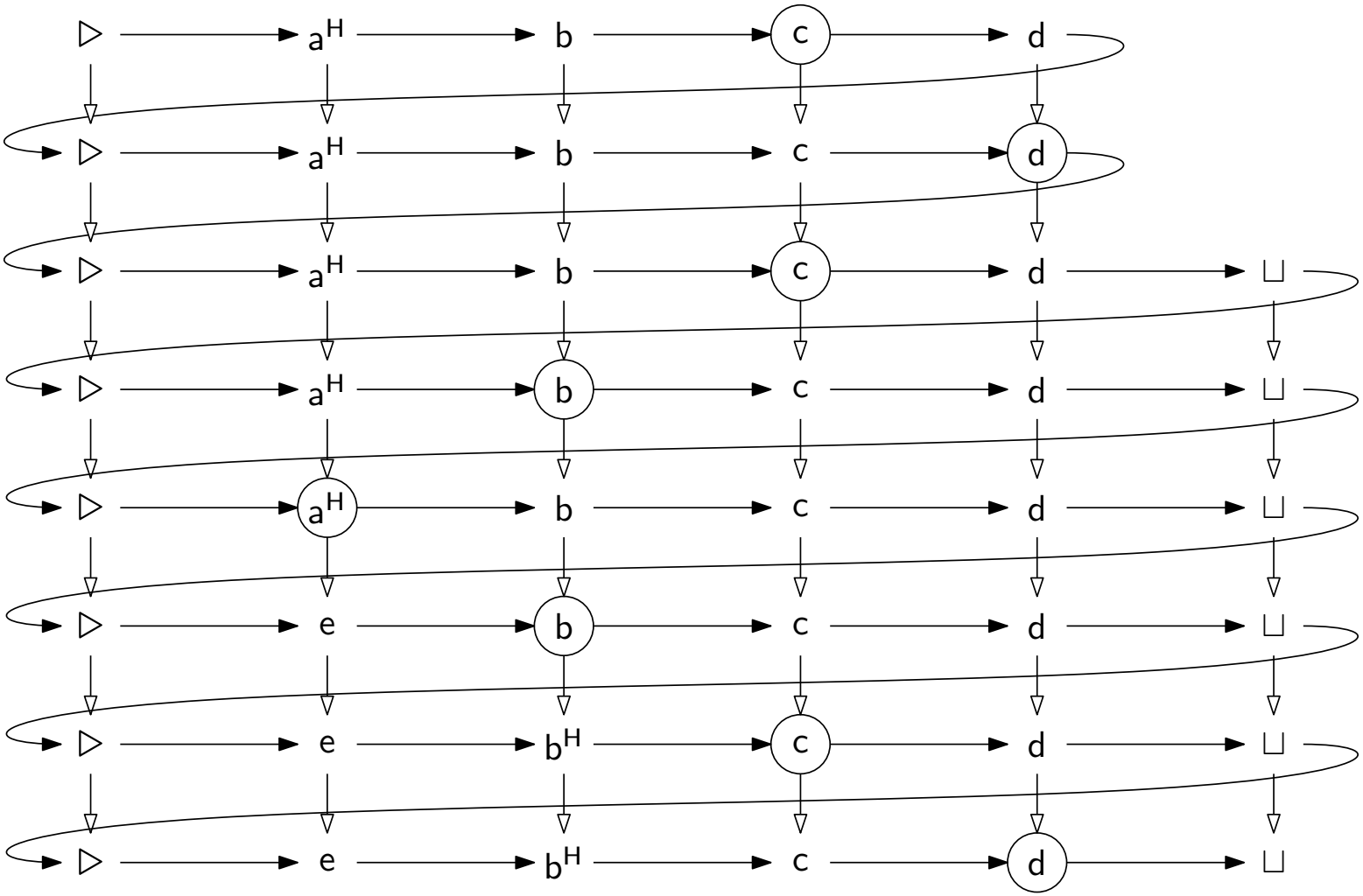
Example of a Construction



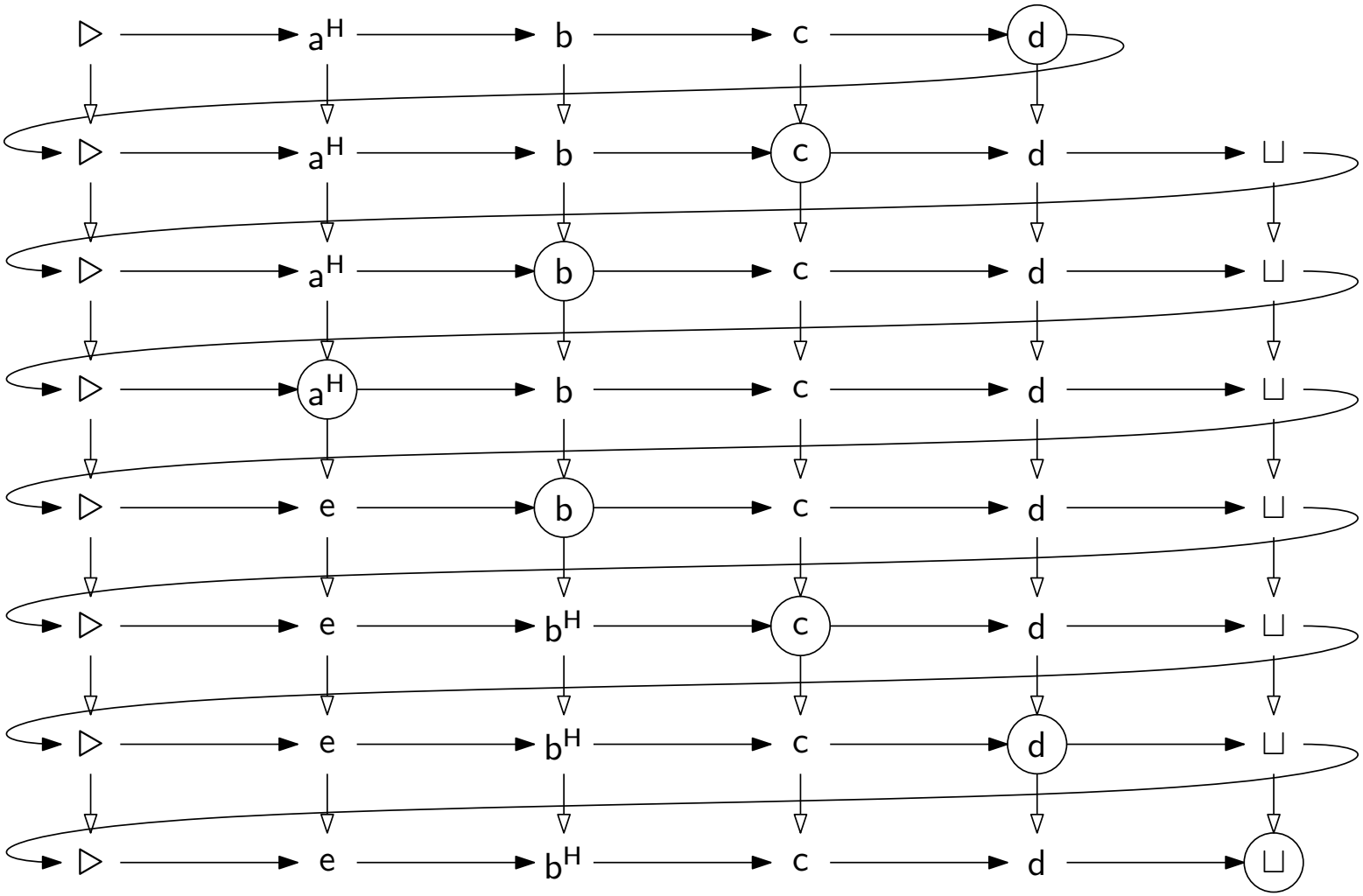
Example of a Construction



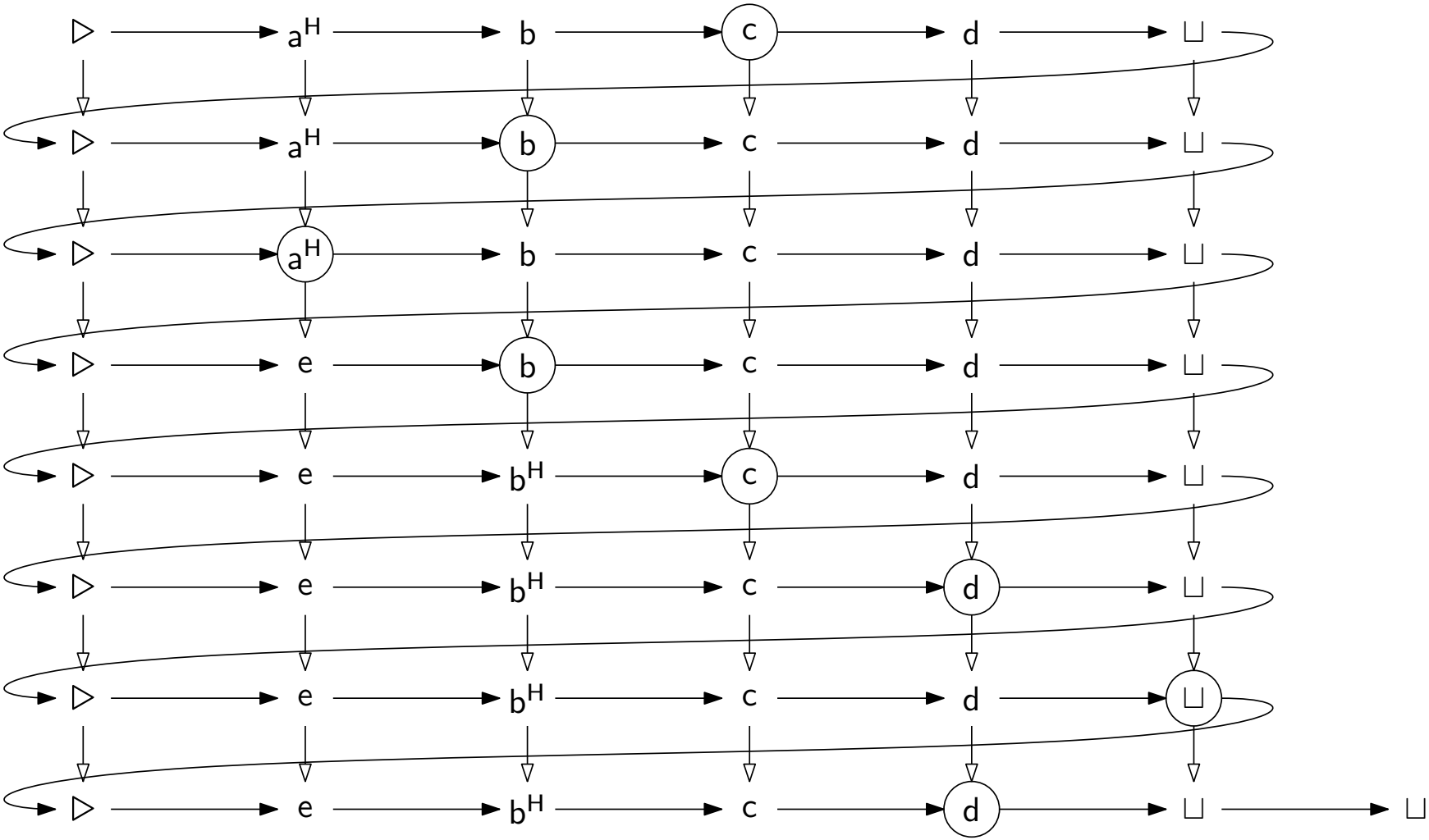
Example of a Construction



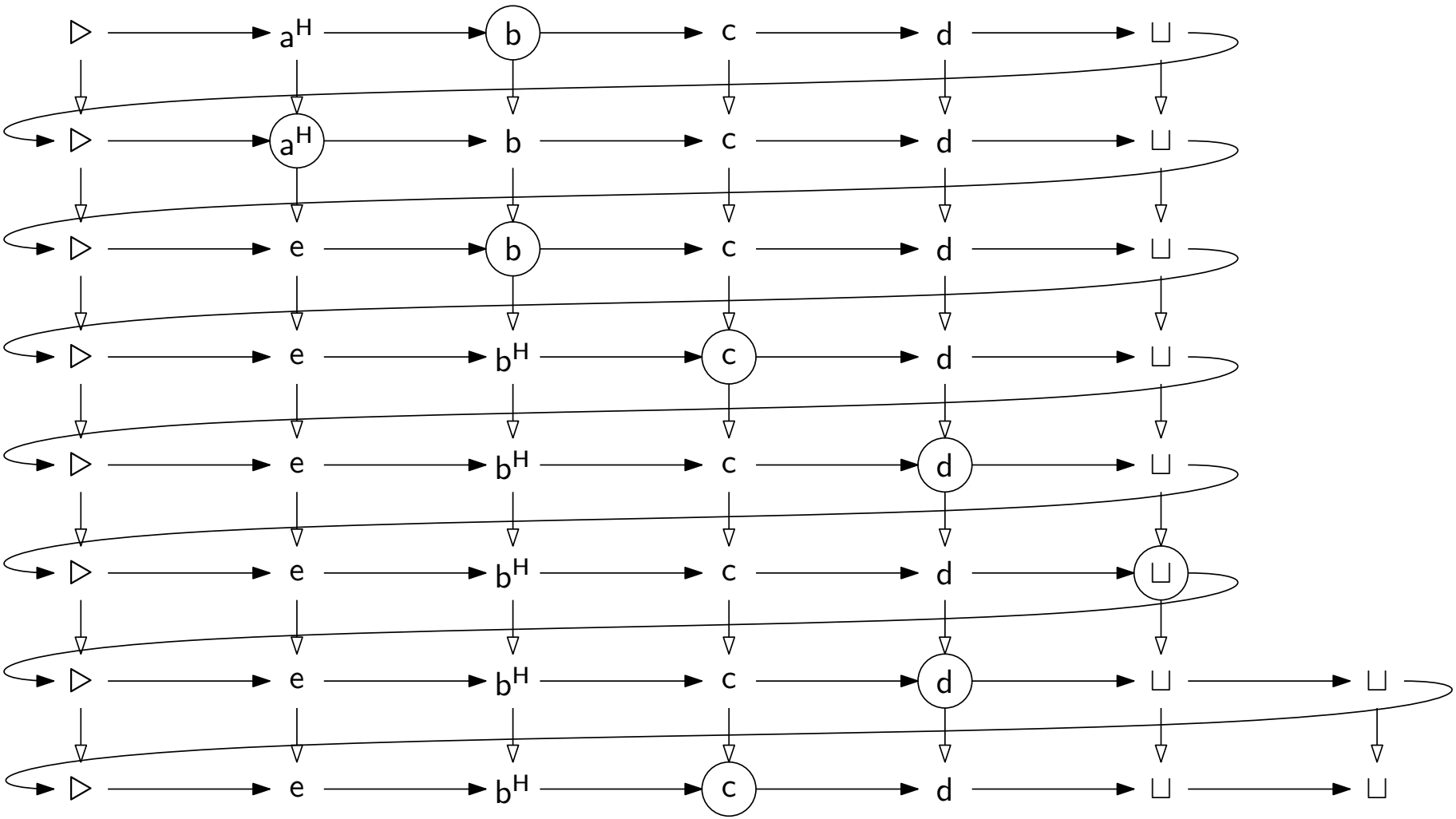
Example of a Construction



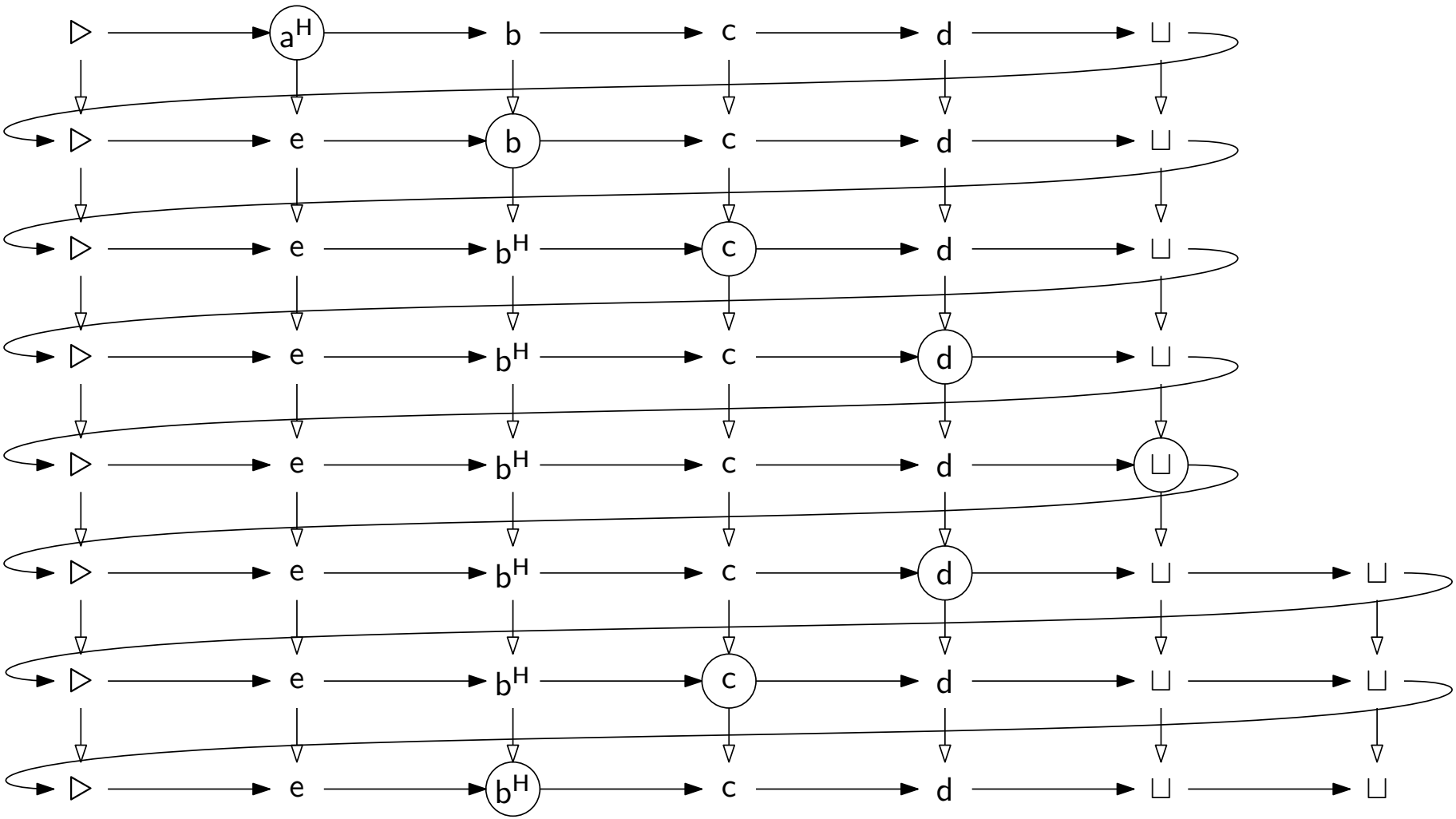
Example of a Construction



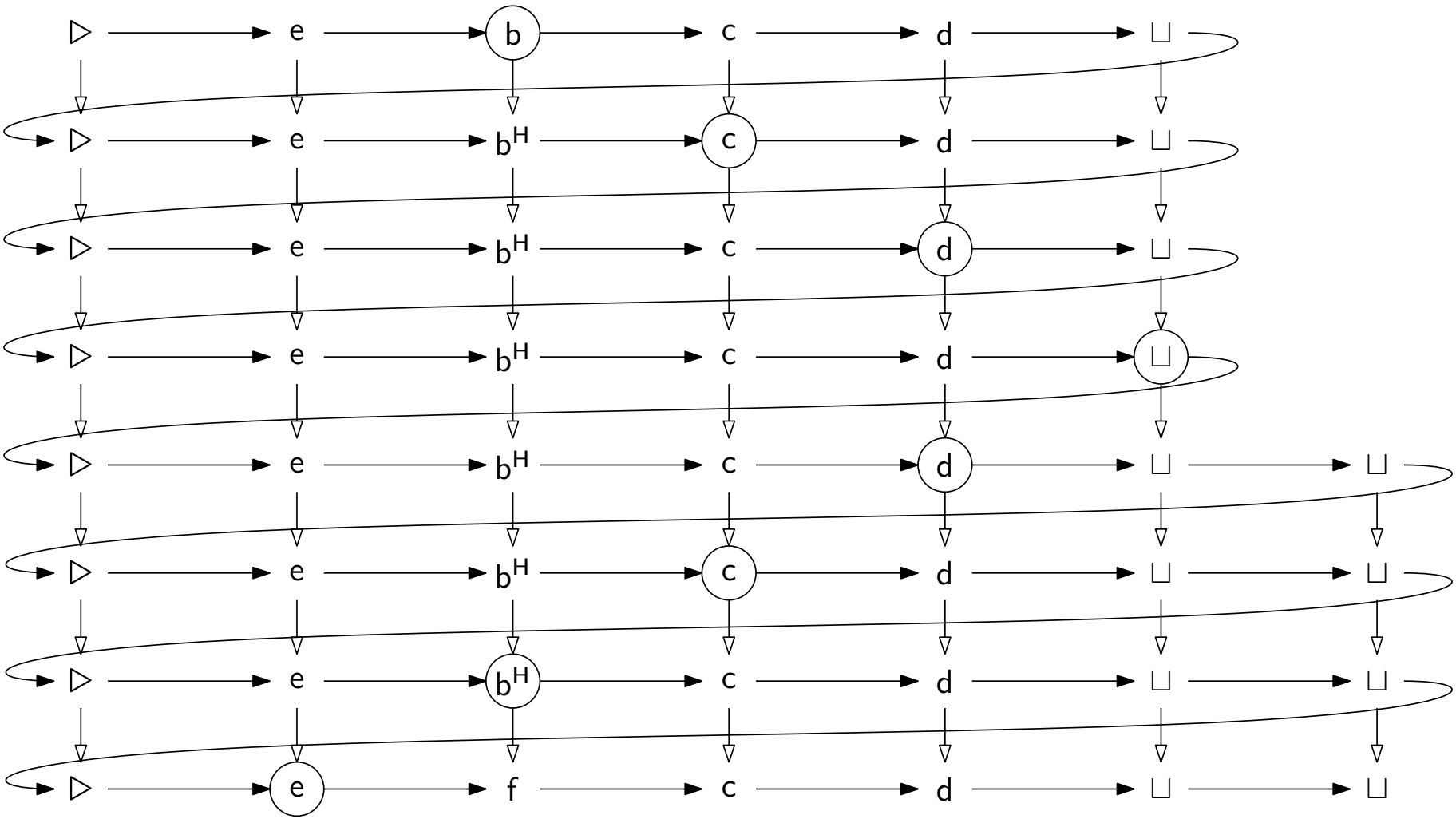
Example of a Construction



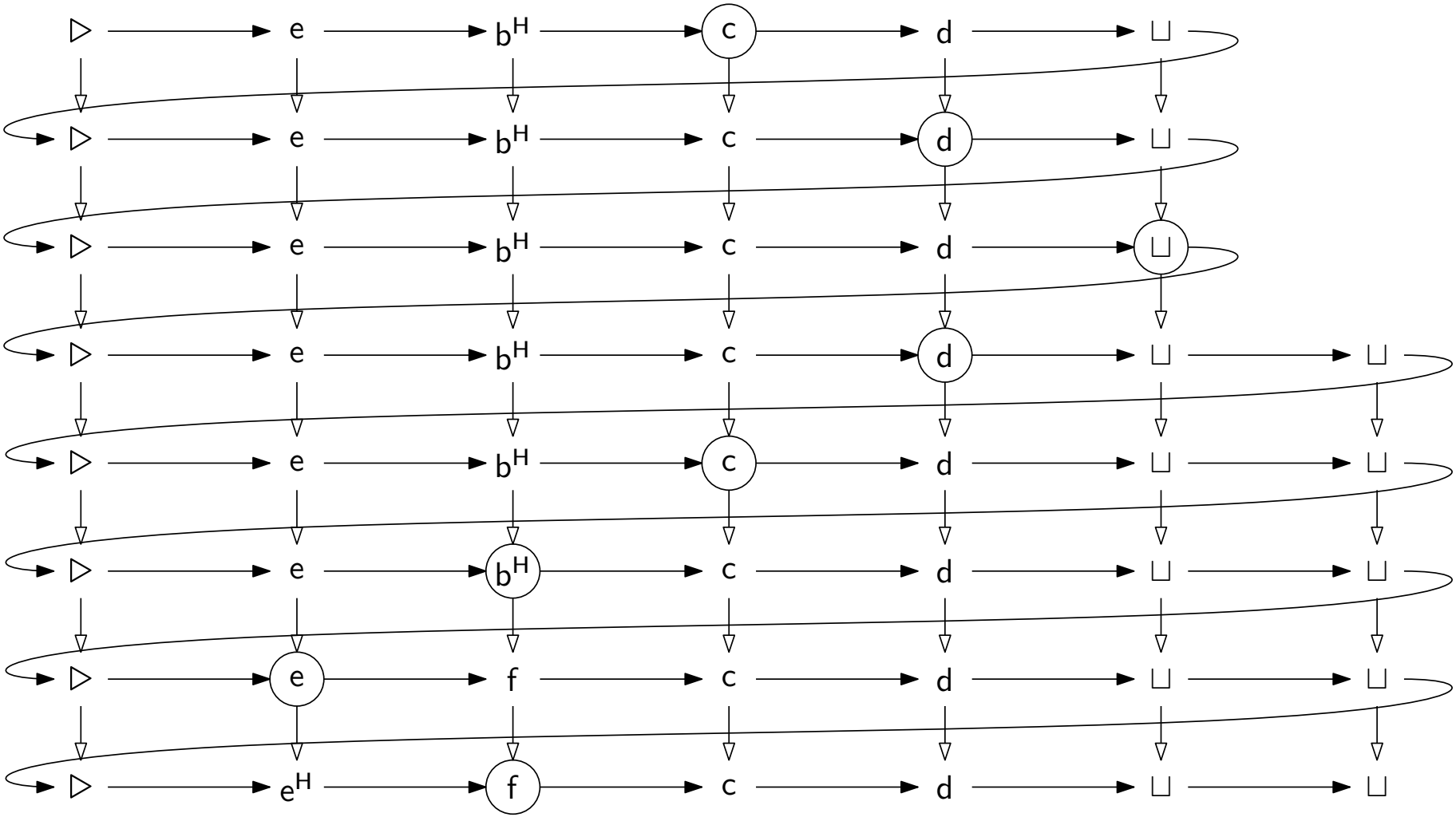
Example of a Construction



Example of a Construction



Example of a Construction



Forward Direction Proof

- If an oblivious Turing machine runs in time $f(n)$ for any input string of length n, then we reduce it to the grid with $f(n)$ rows, where the length of each row corresponds to the number of round trips of the head: first n rows have length $n+1$, next $2n-1$ rows have length $n+2$, etc.
- The input symbol relations correspond to the symbols written on the string. Every element of the first row is Init, and every element of the structure is Mrk.
- If, at the end, Head is in a rejecting state, then the sentence is not satisfied. Otherwise, the sentence is satisfied.

Backwards Direction

Ugly Input Types

Over-Complete Input

The input contains loops or more relations than needed: the $E_{\rightarrow}$- or $E_{\downarrow}$-degrees are greater than 1, or many different ■ elements, or a cell with 2 different symbols $s(\mathrm{x}), s'(\mathrm{x})$.

Incomplete Input

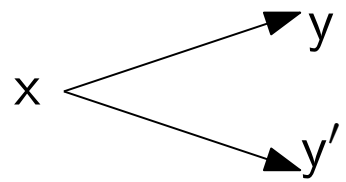
The input size does not permit to simulate the machine run: the rows are too short or there are too few of them.

Complete Input

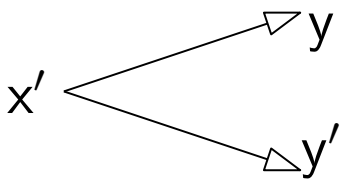
The input contains sufficiently many rows of the right size so that the machine run can be simulated.

Over-Complete Input is Rejected

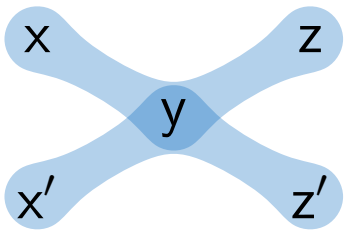
$$\neg(\blacksquare(\mathsf{x}) \wedge \blacksquare(\mathsf{x}') \wedge \mathsf{x} \neq \mathsf{x}')$$



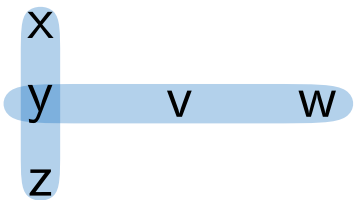
$$\neg(E_{\rightarrow}(\mathsf{x},\mathsf{y}) \wedge E_{\rightarrow}(\mathsf{x},\mathsf{y}') \wedge \mathsf{y} \neq \mathsf{y}')$$



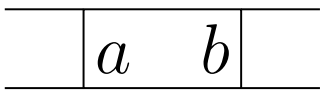
$$\neg(E_{\downarrow}(\mathsf{x},\mathsf{y}) \wedge E_{\downarrow}(\mathsf{x},\mathsf{y}') \wedge \mathsf{y} \neq \mathsf{y}')$$



$$\neg(row(\mathsf{x},\mathsf{y},\mathsf{z}) \wedge row(\mathsf{x}',\mathsf{y},\mathsf{z}') \wedge \mathsf{x} \neq \mathsf{x}')$$



$$\neg(row(\mathsf{x},\mathsf{y},\mathsf{z}) \wedge row(\mathsf{y},\mathsf{v},\mathsf{w}))$$



$$\neg(s_a(\mathsf{x}) \wedge s_b(\mathsf{x}))$$

Incomplete Input

Lemma

One can find in Ptime choices for the relations Init, Mrk, and Head such that if the input instance is accepted, then it is accepted with this choice of relations.

Claim

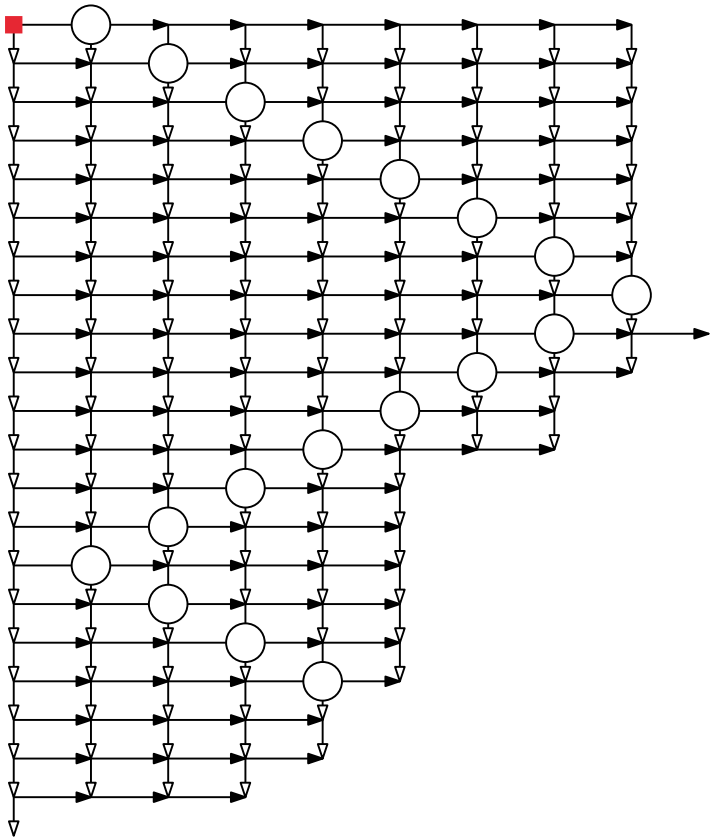
Each of the sets of possible choices of Init, Mrk, Head contains a unique choice minimal by inclusion. These minimal choices satisfy the lemma above.

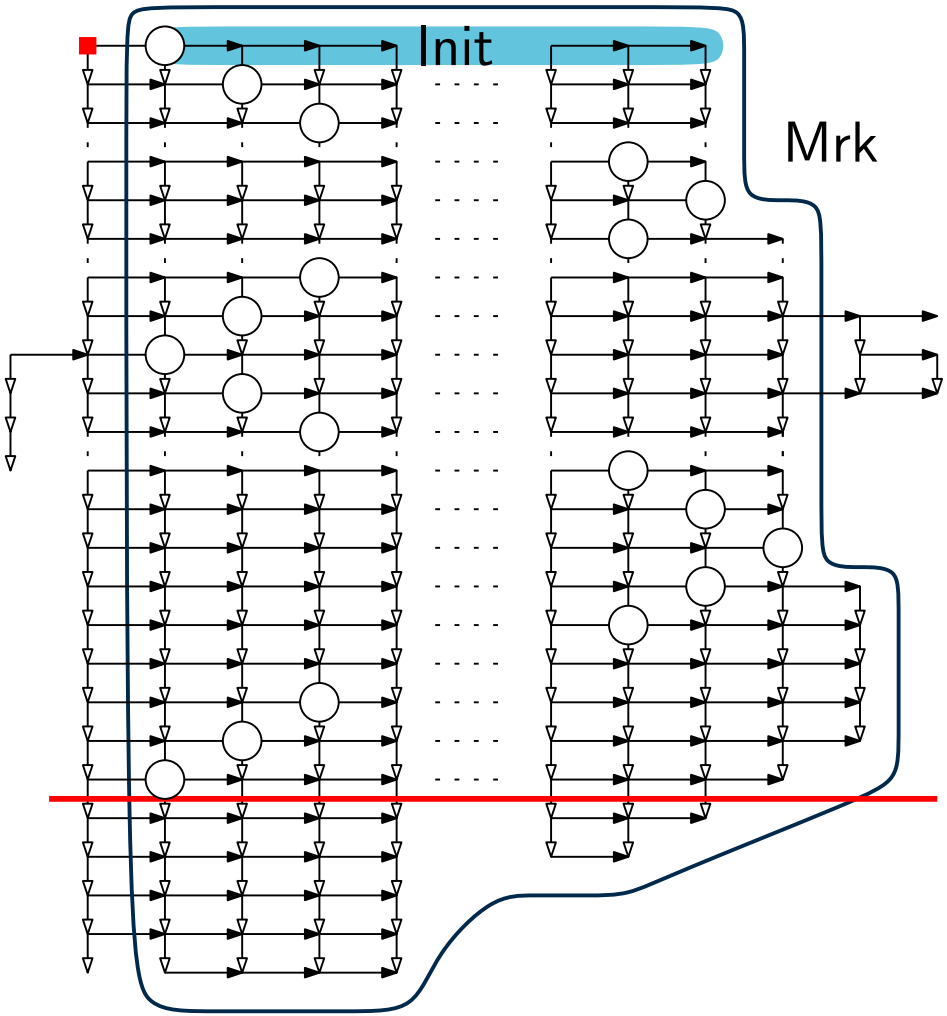
Dealing with Incomplete Input

Knowing which elements are Init and Mrk, and knowing the positions of the Head, we understand whether the execution can be finished. If it cannot be finished, then the Head never falls into a rejecting state, so every incomplete input is accepted.

Complete Input

If the input can simulate the execution, then we reduce it to the string of symbols of Init elements (the first row of the grid).





References

- Ronald Fagin
  Generalized first-order spectra, and polynomial-time recognizable sets
  SIAM-AMS Proc., 1974
- Tomás Feder and Moshe Y. Vardi
  The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory
  SIAM J. Comput., 1998, 10.1137/S0097539794266766
- Richard E. Ladner
  On the Structure of Polynomial Time Reducibility
  J. ACM, 1975, 10.1145/321864.321877
- Dmitriy Zhuk
  A Proof of the CSP Dichotomy Conjecture
  J. ACM, 2020, 10.1145/3402029
- Nicholas Pippenger and Michael J. Fischer
  Relations Among Complexity Measures
  J. ACM, 1979, 10.1145/322123.322138