Consensus Acceptor Games

Marie van den Bogaard

joint work with Dietmar Berwanger

LSV - ENS Cachan

GT-ALGA March 2015

Games with imperfect information on finite graphs:

One player (against Nature): regular strategies are sufficient

Two or more players: what kind of strategies can they require?

Consensus acceptor games

Simplest game with imperfect information:

- two players against Nature on a finite graph
- every play is finite
- players have only one choice: yes/no when the play ends
- winning condition: consensus $+$ in v_+ play $+$, in v_- play $-$!

A concurrent safety game.

Acceptor Games:

Game over observation alphabet Γ

$$G = (V, E, (\beta^1, \beta^2), v_0, F)$$

- (V, E) finite graph (states, transitions)
- observation functions $\beta^1, \beta^2 : V \to \Gamma$
- initial state v_0, final states $F \subseteq \{v_+, v_-, v_=\}$

Play:

- Nature chooses a path π from v_0 to a final state
  $\rightsquigarrow$ observations $\beta^1(\pi)$, $\beta^2(\pi)$
- Players choose decision $d^1, d^2 \in \{+, -\}$ simultaneously and independently

Consensus strategies

Winning condition:

- $d^1 = d^2$
- if final state is v_+, then $d^1 = d^2 = +$; same for v_-

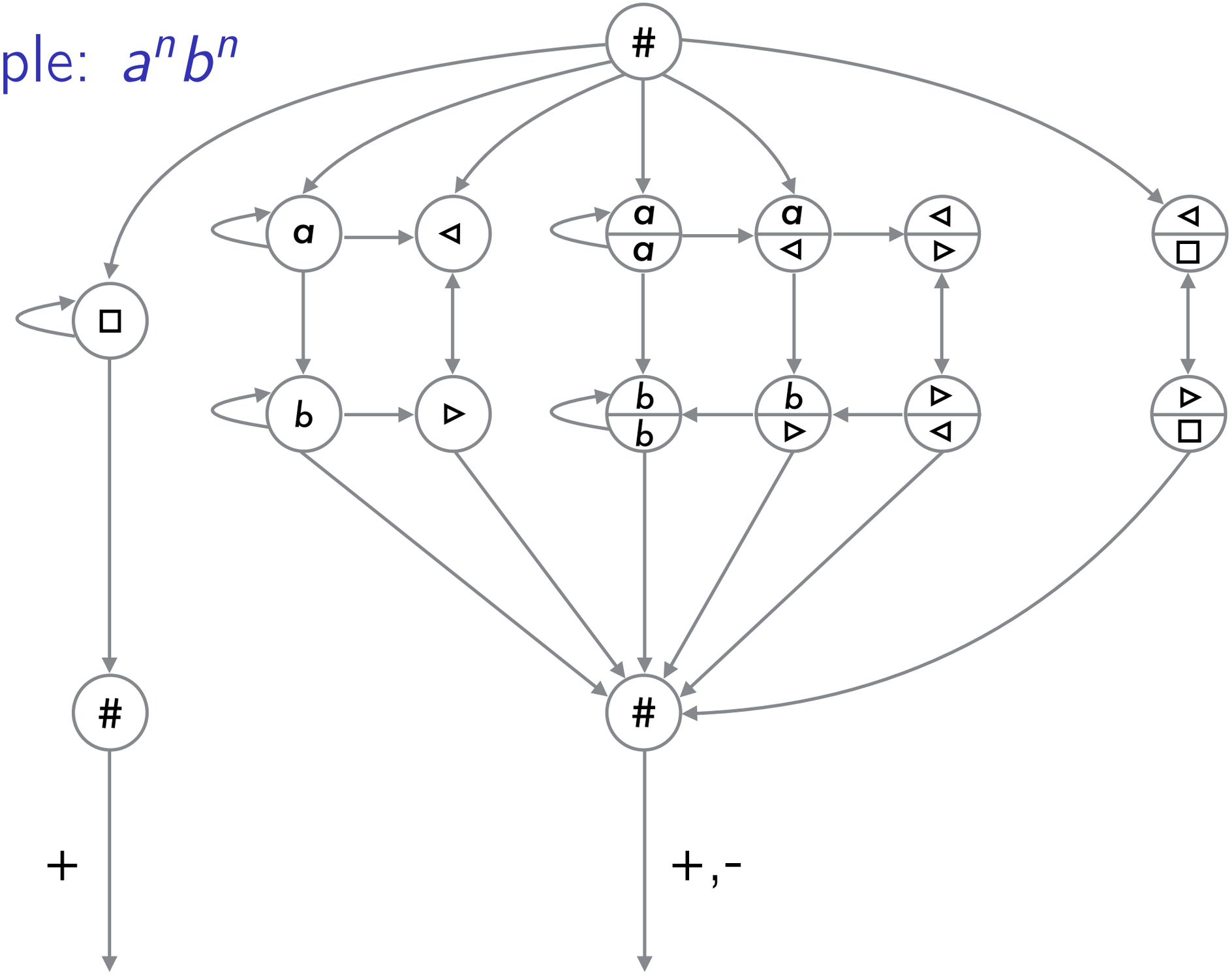
Strategy s^i maps plays to $\{+, -\}$, s.t.
respects observations: if $\beta^i(\pi) = \beta^i(\pi')$ then $s^i(\pi) = s^i(\pi')$

write $\pi \sim^i \pi'$

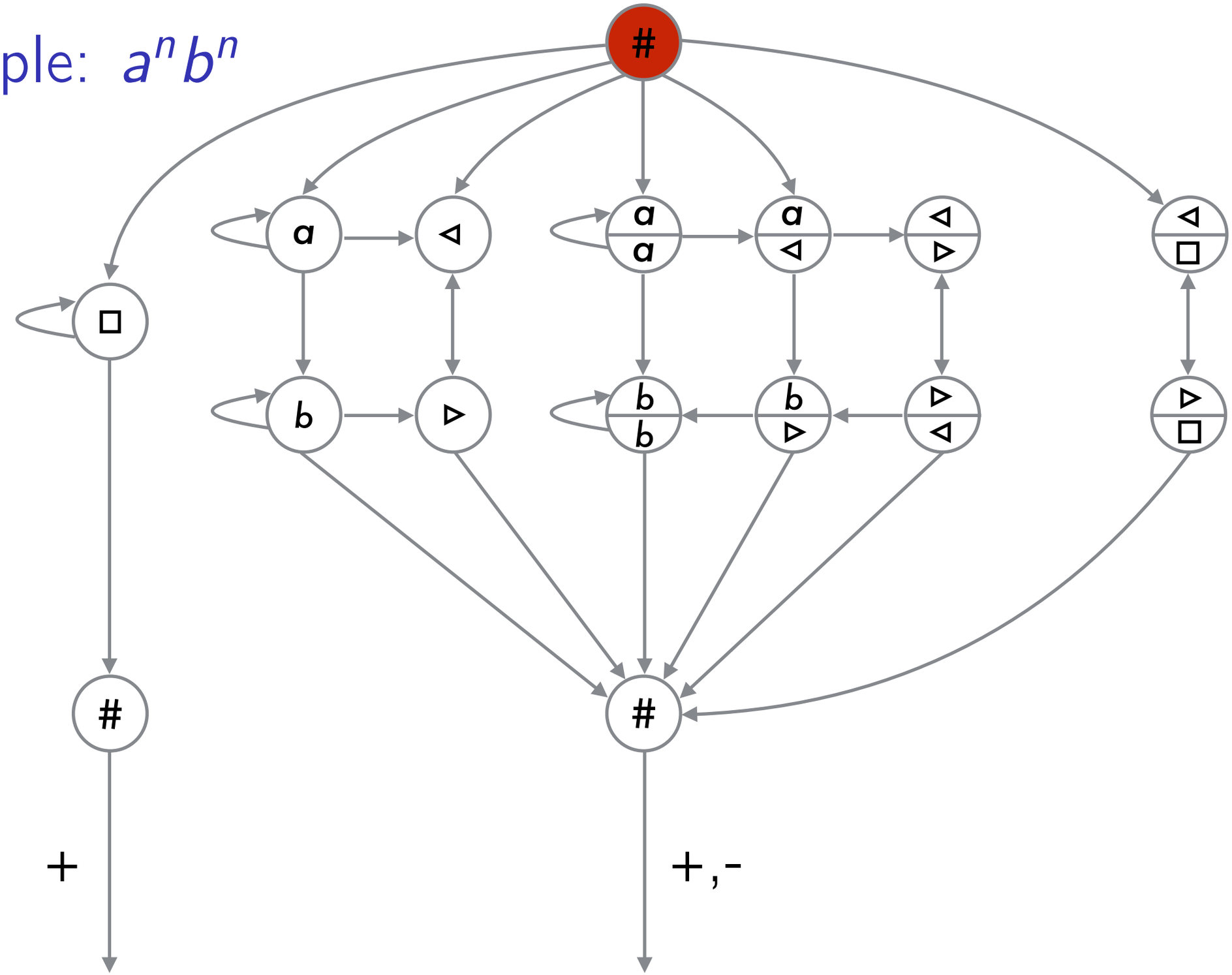
Set $\sim^* := (\sim^1 \cup \sim^2)^*$
necessary for consensus: if $\pi \sim^* \pi'$, then $s(\pi) = s(\pi')$

"play +" must be common knowledge

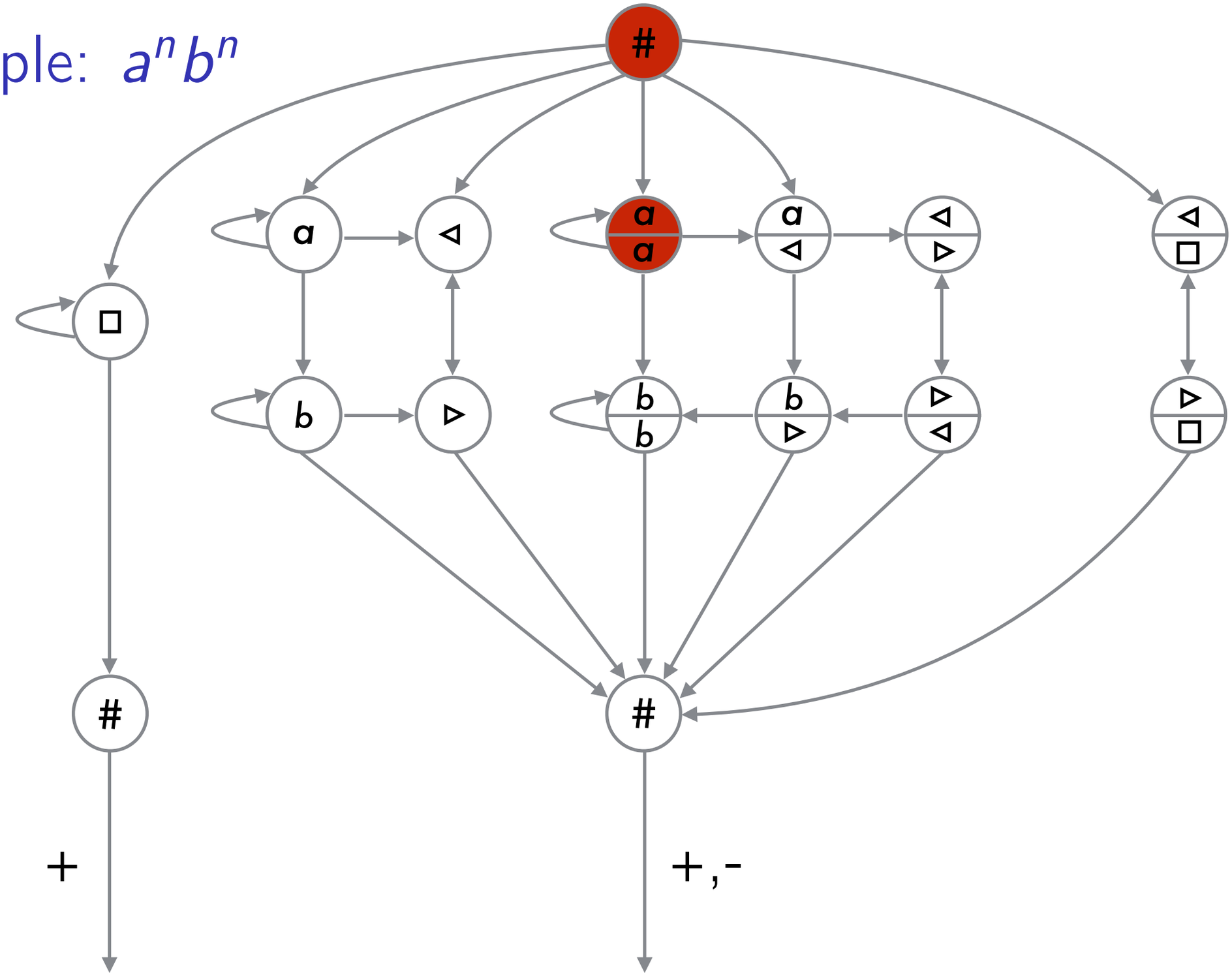
Example: $a^n b^n$



Example: $a^n b^n$

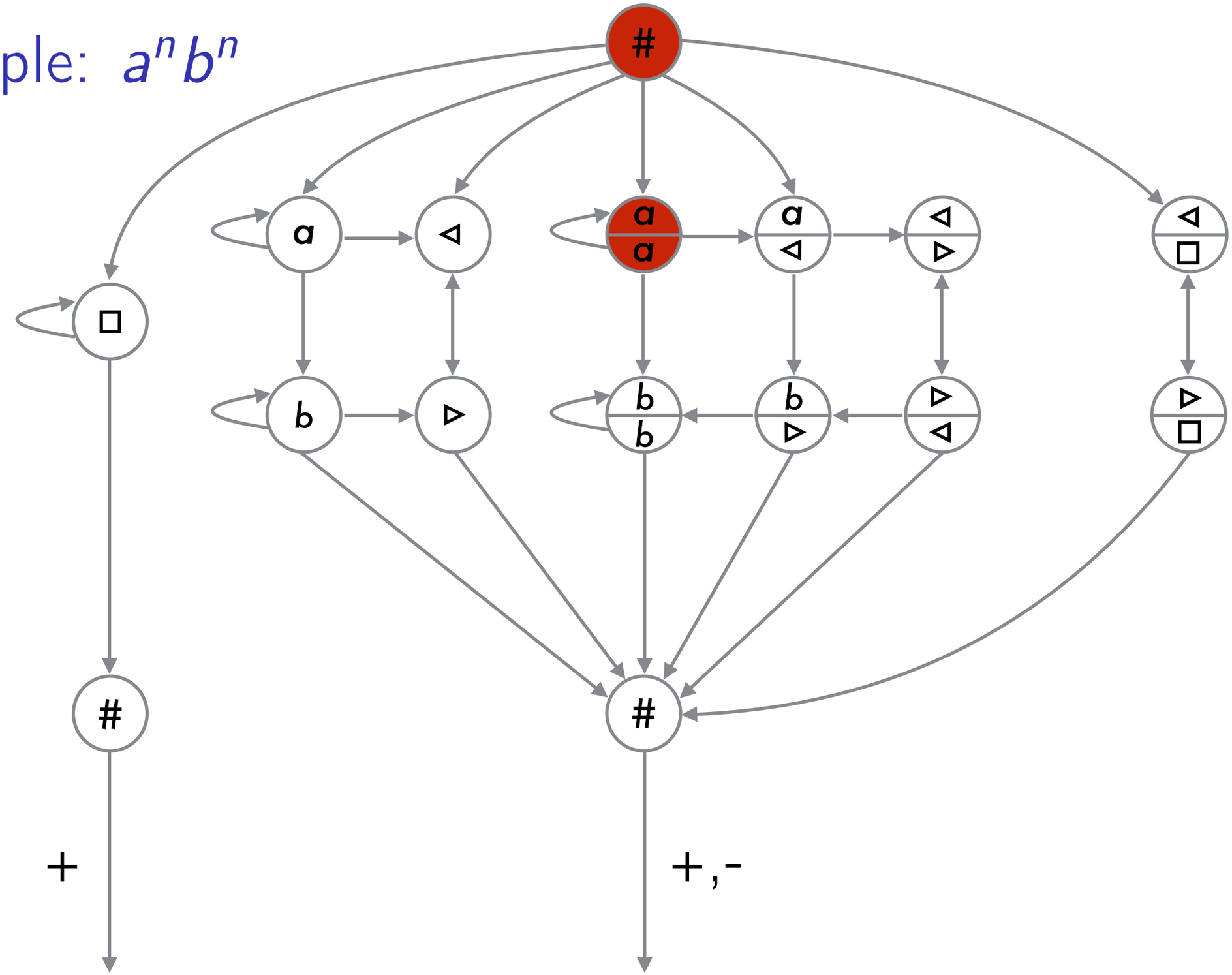


Example: $a^n b^n$



a a

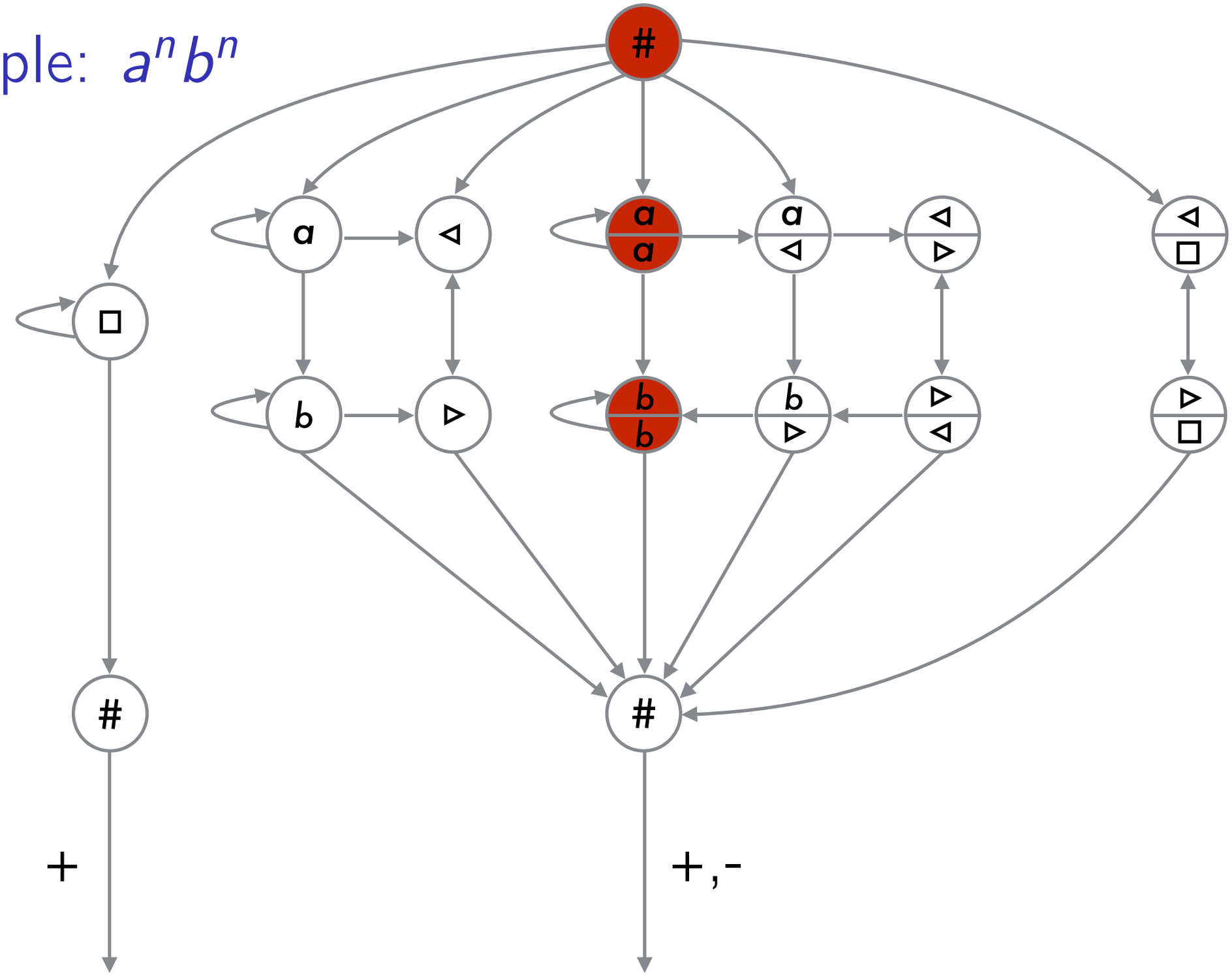
Example: $a^n b^n$



a a

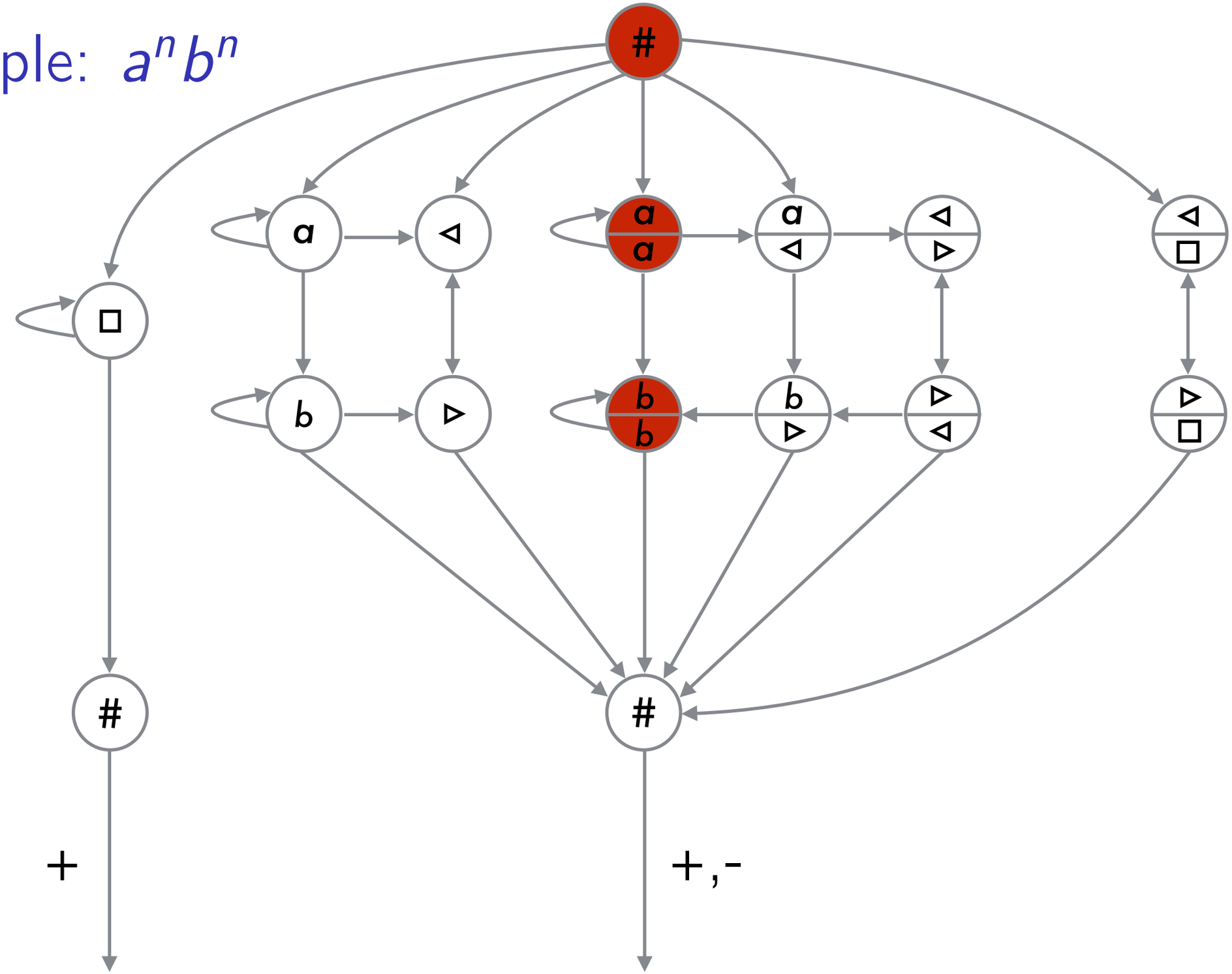
a a

Example: $a^n b^n$



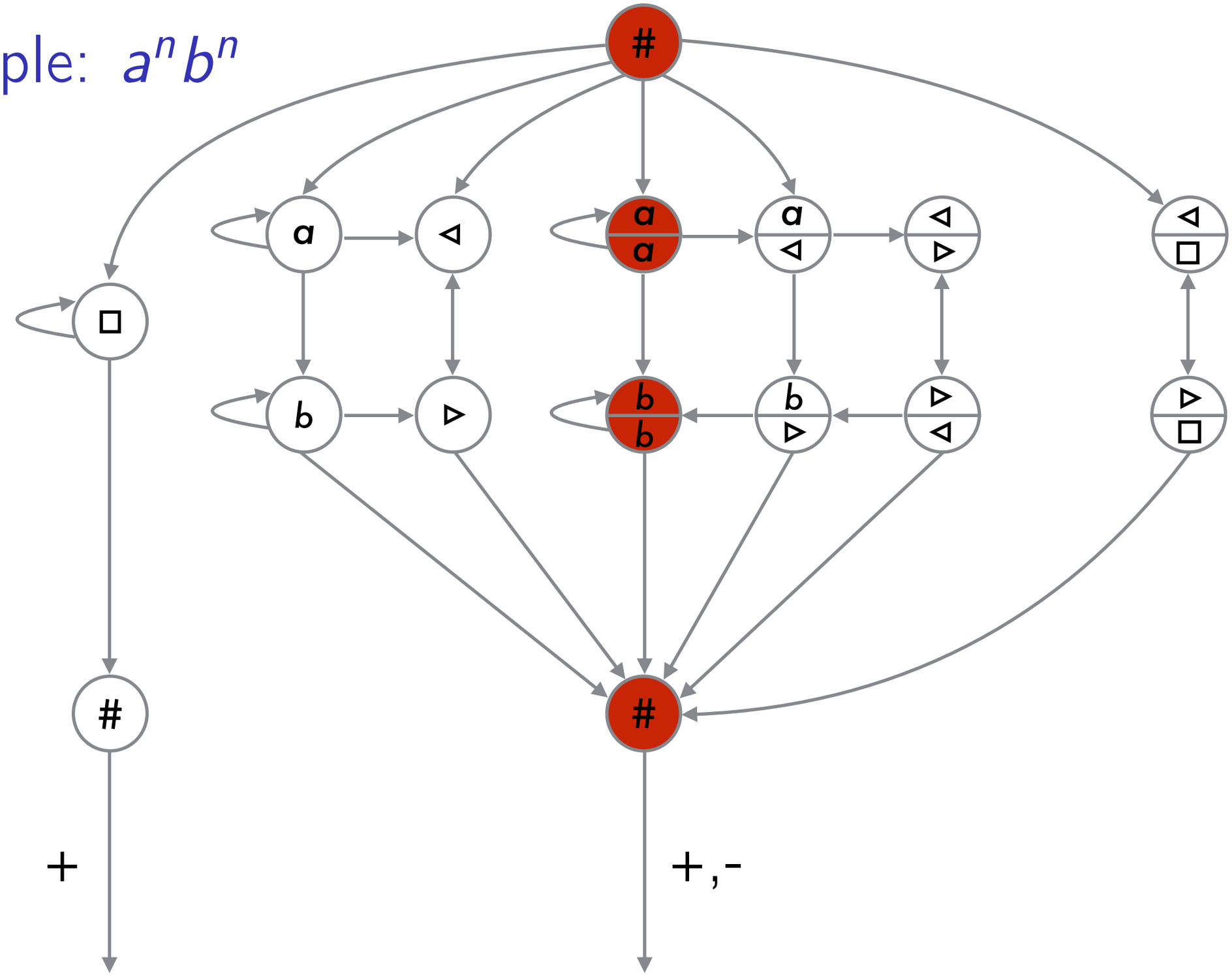
a a
a a
b b

Example: $a^n b^n$



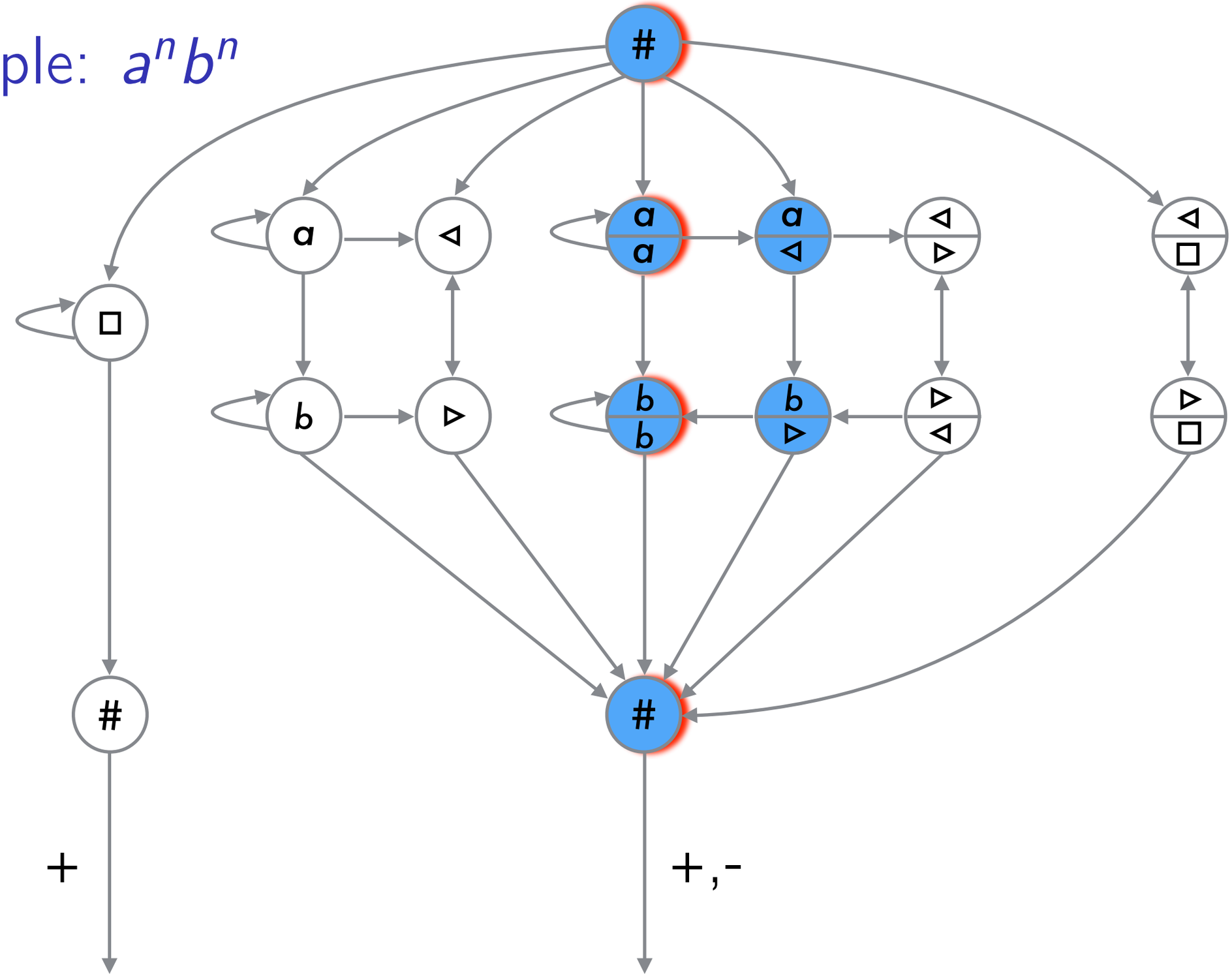
a a
a a
b b
b b

Example: $a^n b^n$



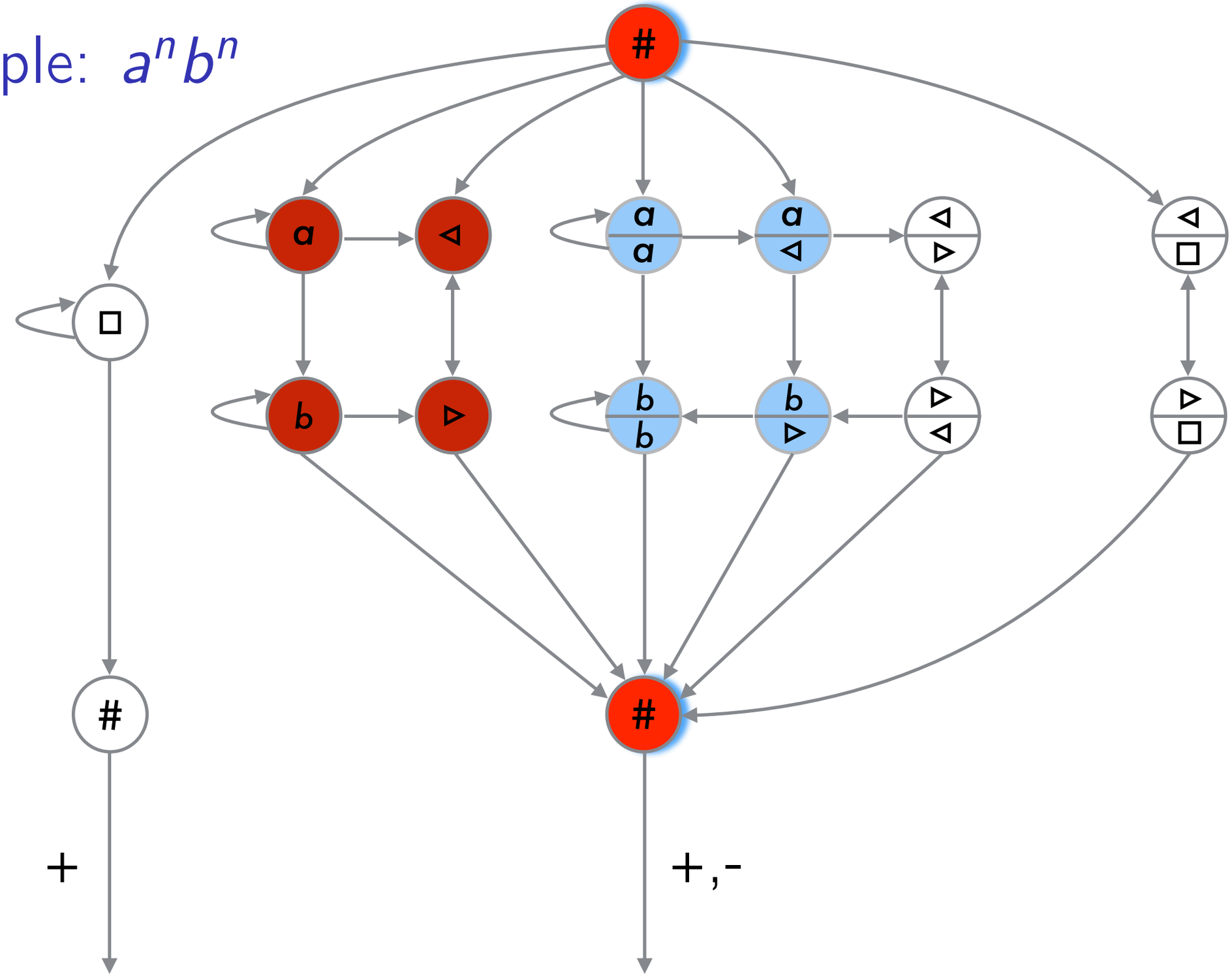
a a
a a
b b
b b

Example: $a^n b^n$



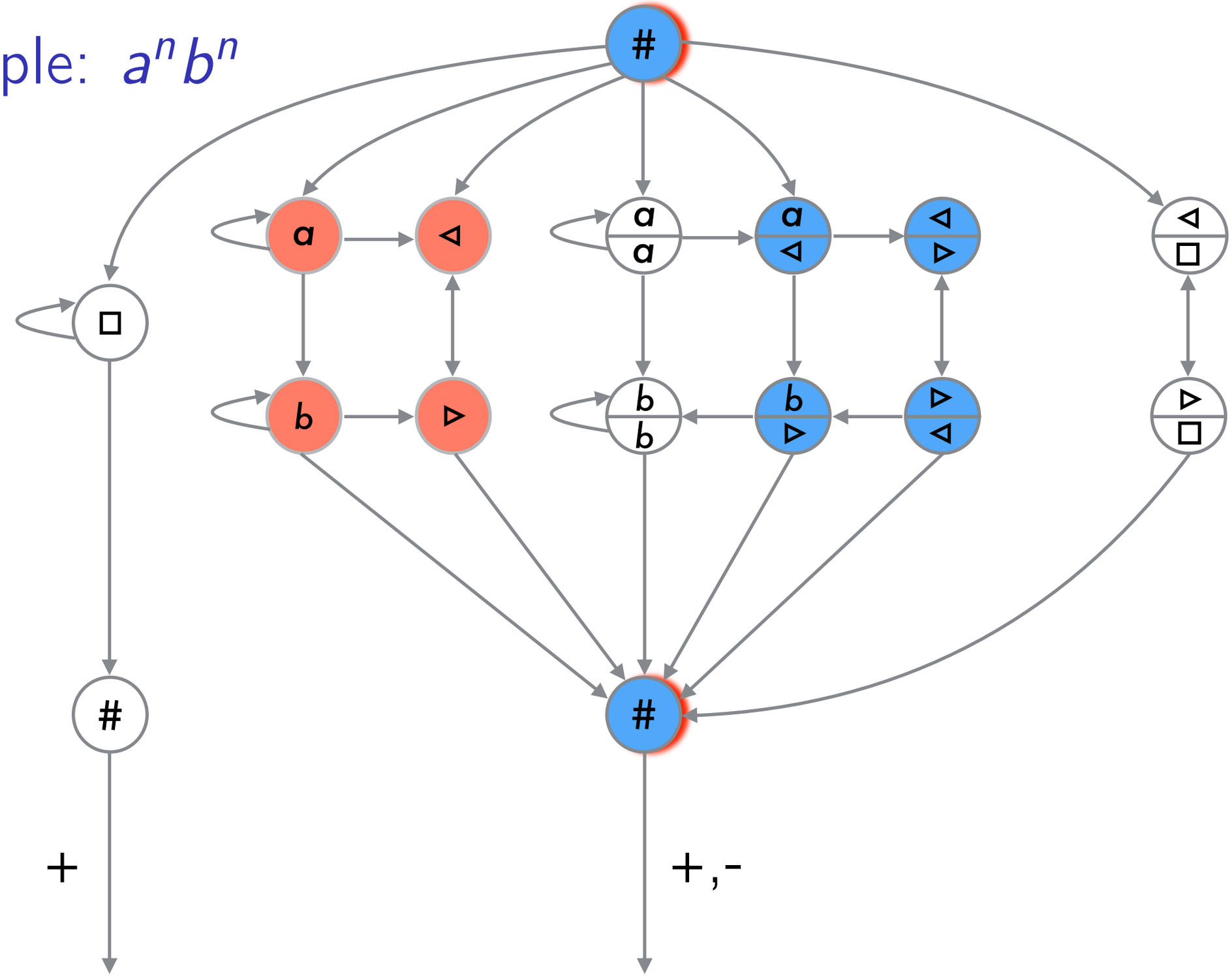
$$\begin{matrix} a & a \\ a & a \\ b & b \\ b & b \end{matrix} \sim_1 \begin{matrix} a & a \\ a & \triangleleft \\ b & \triangleright \\ b & b \end{matrix}$$

Example: $a^n b^n$



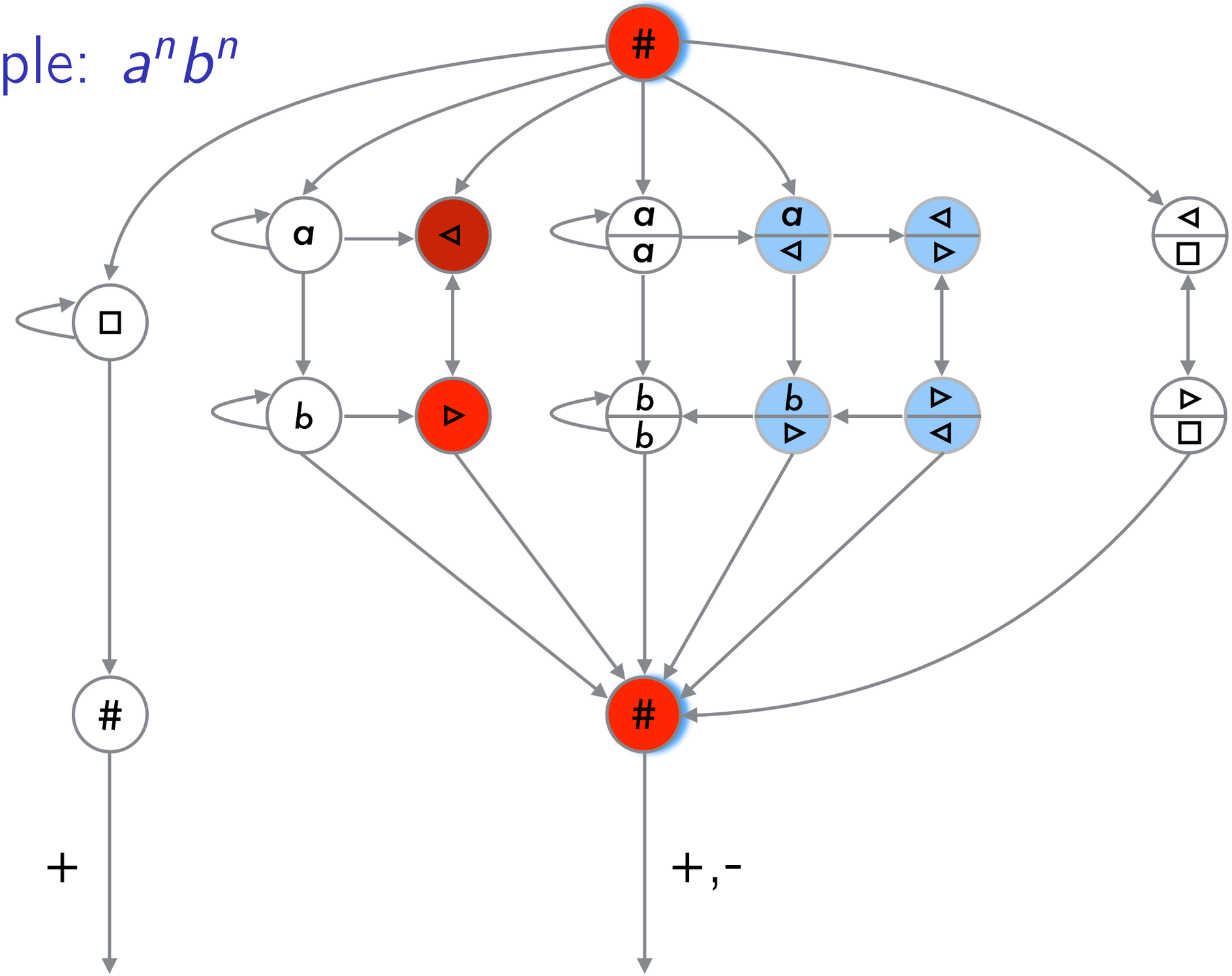
$$\begin{matrix} a & a \\ a & a \\ b & b \\ b & b \end{matrix} \sim_1 \begin{matrix} a & a \\ a & \triangleleft \\ b & \triangleright \\ b & b \end{matrix} \sim_2 \begin{matrix} a & a \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ b & b \end{matrix}$$

Example: $a^n b^n$



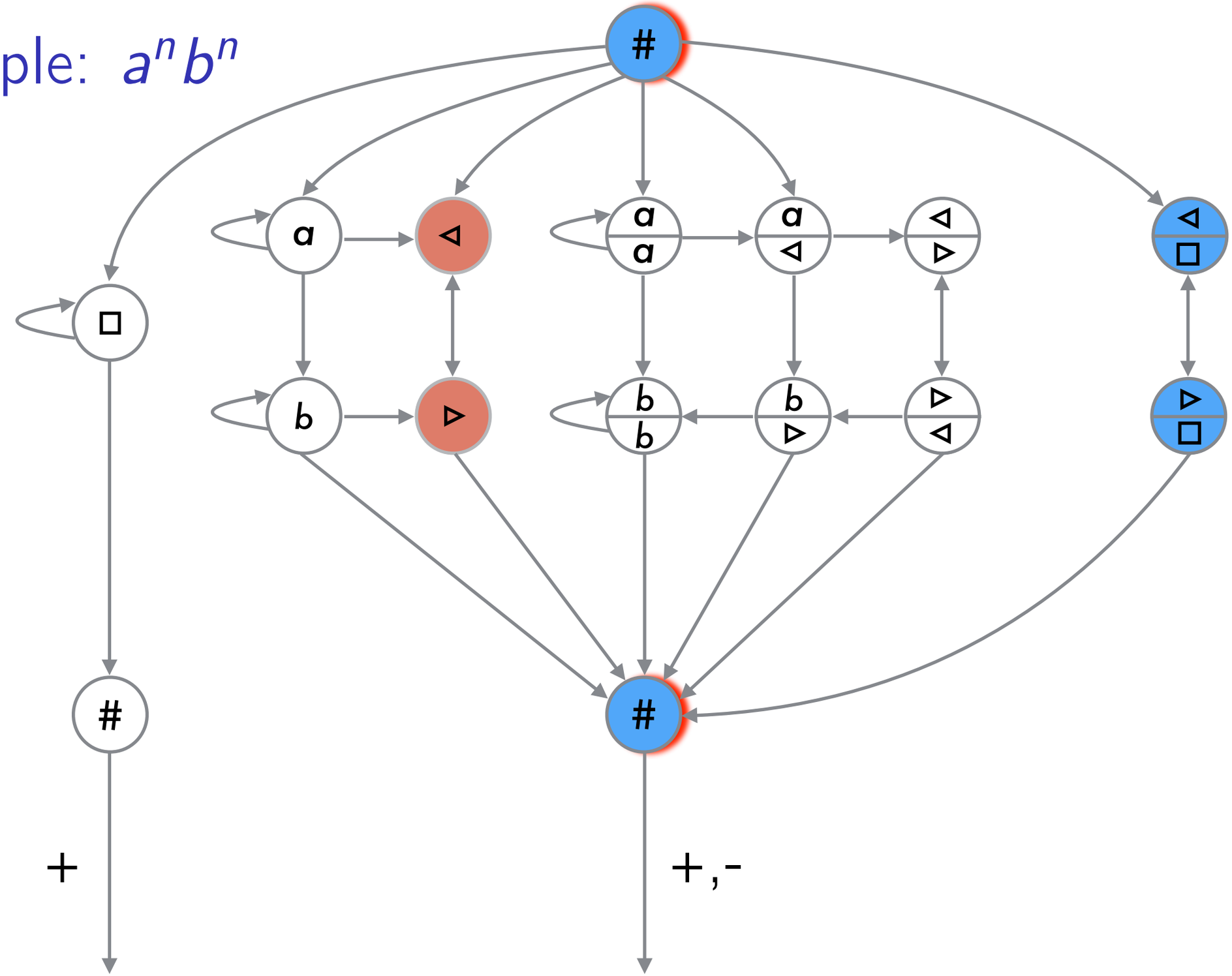
$$\begin{matrix} a & a \\ a & a \\ b & b \\ b & b \end{matrix} \sim_1 \begin{matrix} a & a \\ a & \triangleleft \\ b & \triangleright \\ b & b \end{matrix} \sim_2 \begin{matrix} a & a \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ b & b \end{matrix} \sim_1 \begin{matrix} a & \triangleleft \\ \triangleleft & \triangleright \\ \triangleright & \triangleleft \\ b & \triangleright \end{matrix}$$

Example: $a^n b^n$



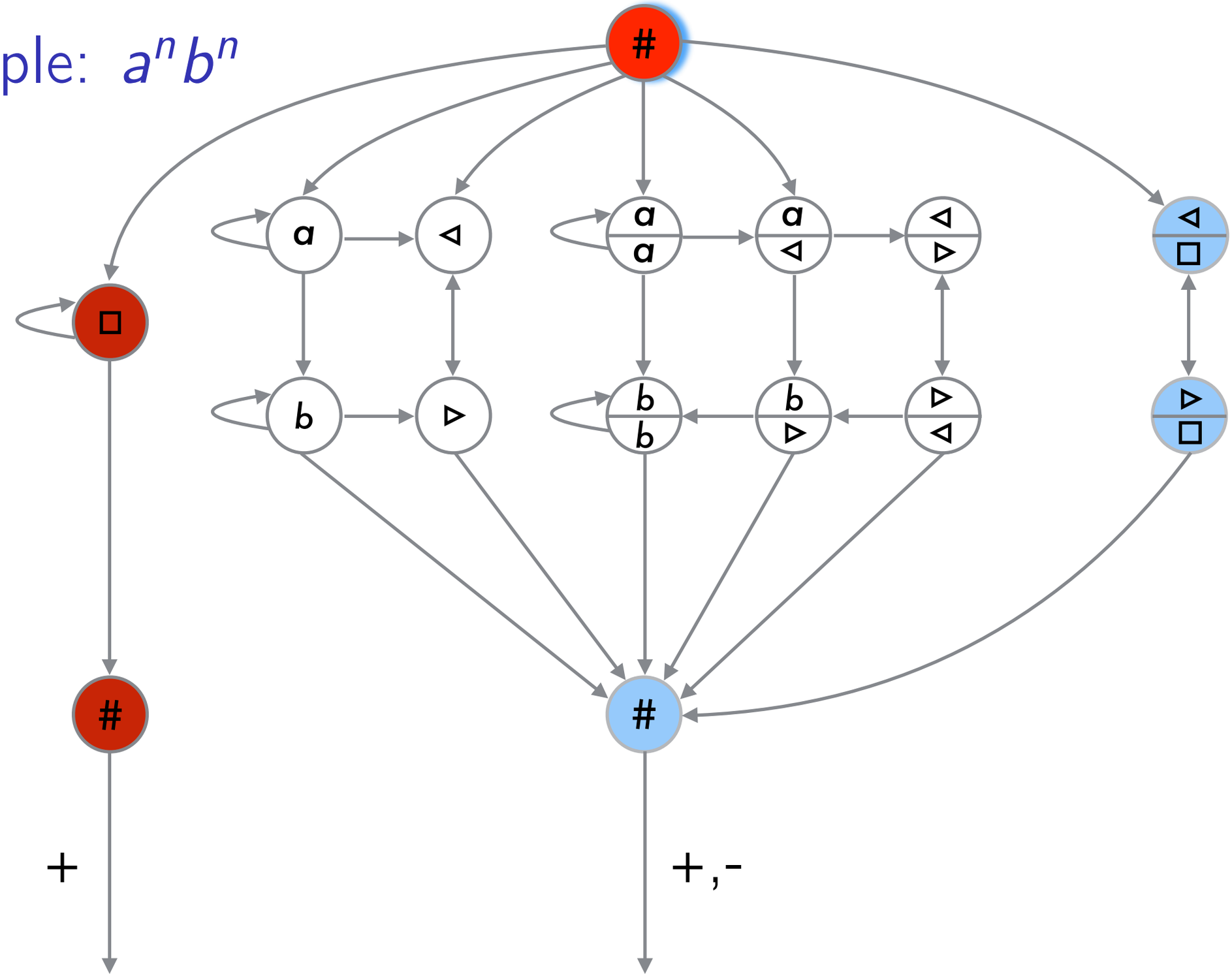
$$\begin{matrix} a & a \\ a & a \\ b & b \\ b & b \end{matrix} \sim_1 \begin{matrix} a & a \\ a & \triangleleft \\ b & \triangleright \\ b & b \end{matrix} \sim_2 \begin{matrix} a & a \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ b & b \end{matrix} \sim_1 \begin{matrix} a & \triangleleft \\ \triangleleft & \triangleright \\ \triangleright & \triangleleft \\ b & \triangleright \end{matrix} \sim_2 \begin{matrix} \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \end{matrix}$$

Example: $a^n b^n$



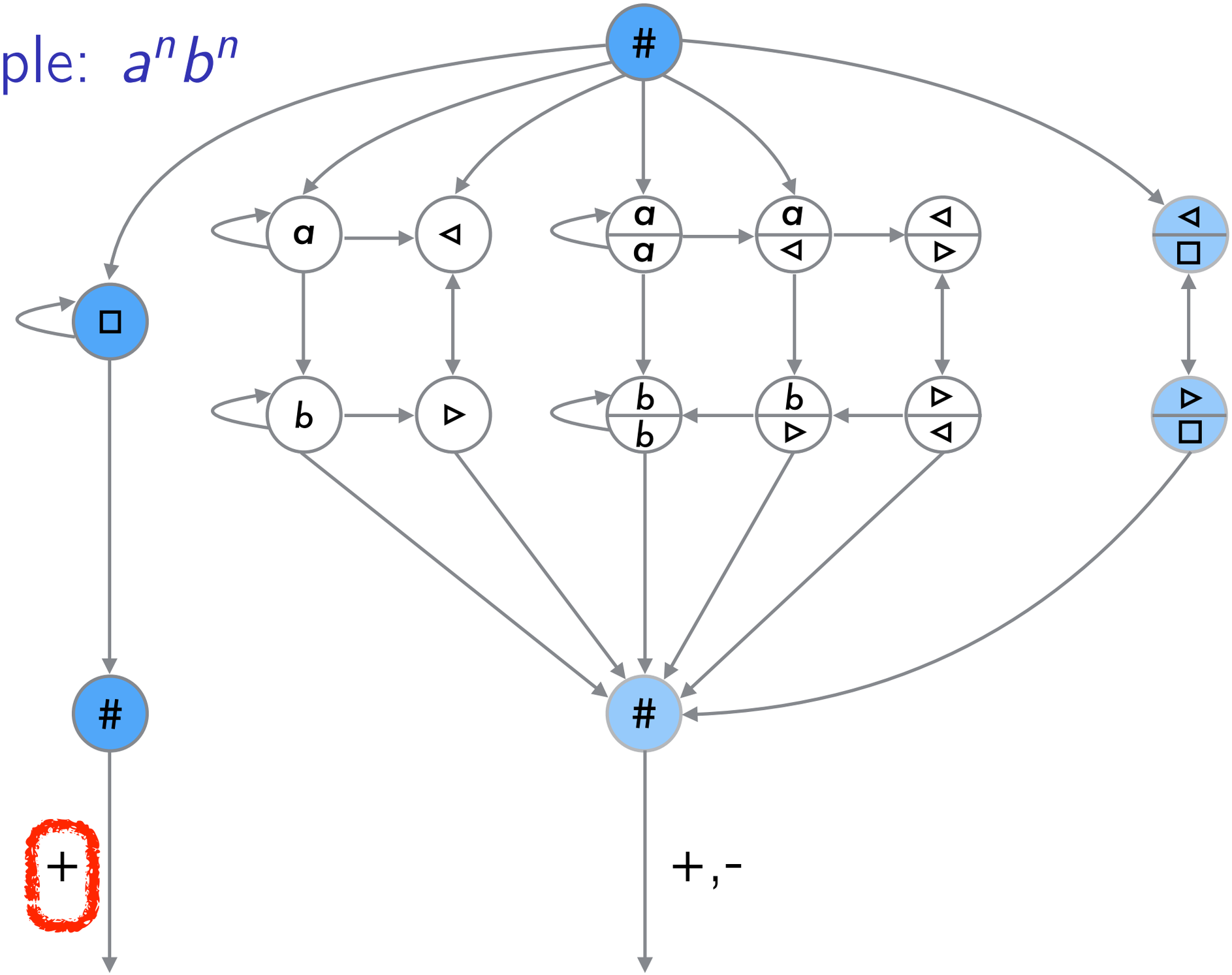
$$\begin{matrix} a & a \\ a & a \\ b & b \\ b & b \end{matrix} \sim_1 \begin{matrix} a & a \\ a & \triangleleft \\ b & \triangleright \\ b & b \end{matrix} \sim_2 \begin{matrix} a & a \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ b & b \end{matrix} \sim_1 \begin{matrix} a & \triangleleft \\ \triangleleft & \triangleright \\ \triangleright & \triangleleft \\ b & \triangleright \end{matrix} \sim_2 \begin{matrix} \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \end{matrix} \sim_1 \begin{matrix} \triangleleft & \square \\ \triangleright & \square \\ \triangleleft & \square \\ \triangleright & \square \end{matrix}$$

Example: $a^n b^n$



$$\begin{matrix} a & a \\ a & a \\ b & b \\ b & b \end{matrix} \sim_1 \begin{matrix} a & a \\ a & \triangleleft \\ b & \triangleright \\ b & b \end{matrix} \sim_2 \begin{matrix} a & a \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ b & b \end{matrix} \sim_1 \begin{matrix} a & \triangleleft \\ \triangleleft & \triangleright \\ \triangleright & \triangleleft \\ b & \triangleright \end{matrix} \sim_2 \begin{matrix} \triangleleft & \triangleleft \\ \triangleright & \triangleright \\ \triangleleft & \triangleleft \\ \triangleright & \triangleright \end{matrix} \sim_1 \begin{matrix} \triangleleft & \square \\ \triangleright & \square \\ \triangleleft & \square \\ \triangleright & \square \end{matrix} \sim_2 \begin{matrix} \square & \square \\ \square & \square \\ \square & \square \\ \square & \square \end{matrix}$$

Example: $a^n b^n$



$$\begin{array}{cccccccccccccc}
a & a & & a & a & & a & a & & a & \triangleleft & & \triangleleft & \triangleleft & & \triangleleft & \square & & \square & \square \\
a & a & \sim_1 & a & \triangleleft & \sim_2 & \triangleleft & \triangleleft & \sim_1 & \triangleleft & \triangleright & \sim_2 & \triangleright & \triangleright & \sim_1 & \triangleright & \square & \sim_2 & \square & \square \\
b & b & & b & \triangleright & & \triangleright & \triangleright & & \triangleright & \triangleleft & & \triangleleft & \triangleleft & & \triangleleft & \square & & \square & \square \\
b & b & & b & b & & b & b & & b & \triangleright & & \triangleright & \triangleright & & \triangleright & \square & & \square & \square
\end{array}$$

Games and Languages

Fix $\Sigma \subset \Gamma$.

A strategy s^1 induces $L(s^1) := \{w \in \Sigma^* \mid s(\#w\#) = 1\}$

A game G **covers** a language $L \subseteq \Sigma^*$ if :

- there exists a winning strategy s such that $L(s) = L$
- for every winning strategy s, we have $L \subseteq L(s)$

A game G **characterises** L if for every winning strategy s, we have $L = L(s)$.

Consensus acceptor games $\equiv$ context-sensitive languages

Theorem.

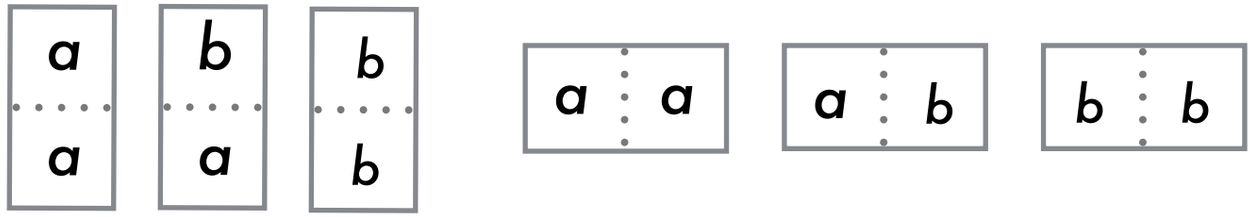
1. Every context-sensitive language is characterised by a game
2. Deciding which decision to take in a play is PSPACE-hard
3. It is undecidable whether a game admits a winning strategy
4. If yes, winning strategies can be implemented by linear bounded automata

Arguments

1. Characterisation: expressing corridor tilings with dominoes
2. Safe decision problem: membership in a context-sensitive language
3. Solvability problem: emptiness of a context-sensitive language (undecidable)
4. Implementation: Linear bounded automata can decide $\sim^*$

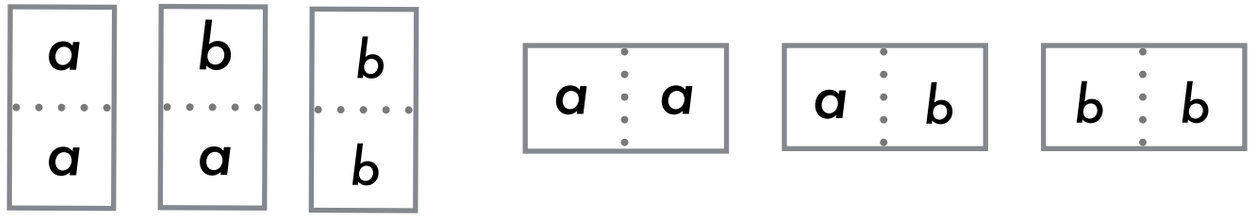
Domino systems and corridor tilings

Set of dominoes :

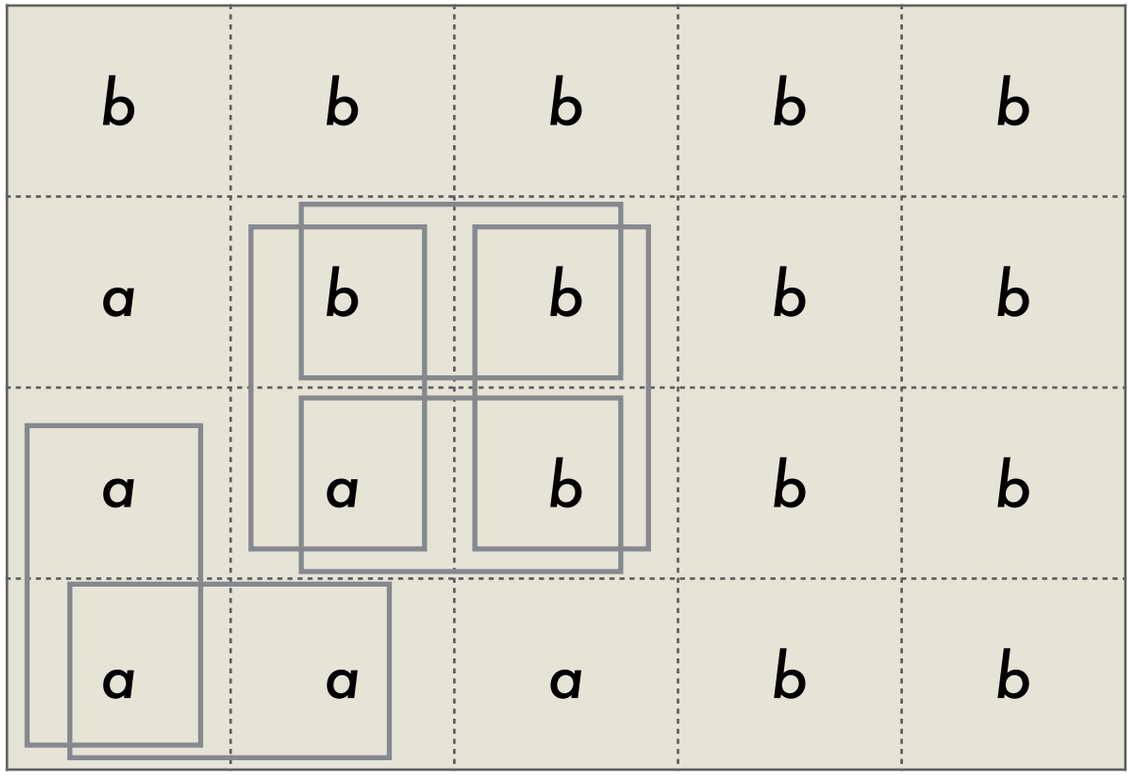


Domino systems and corridor tilings

Set of dominoes :

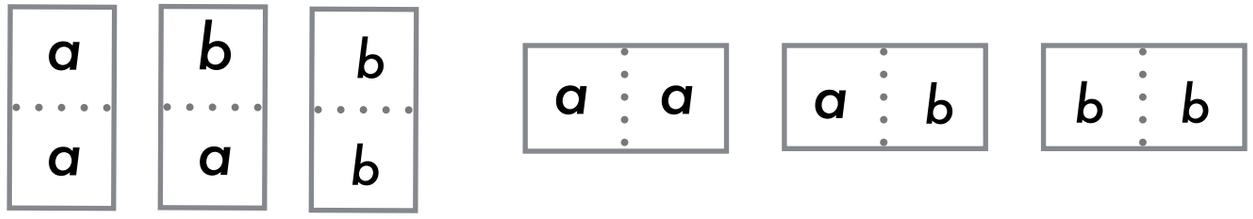


Tiling :

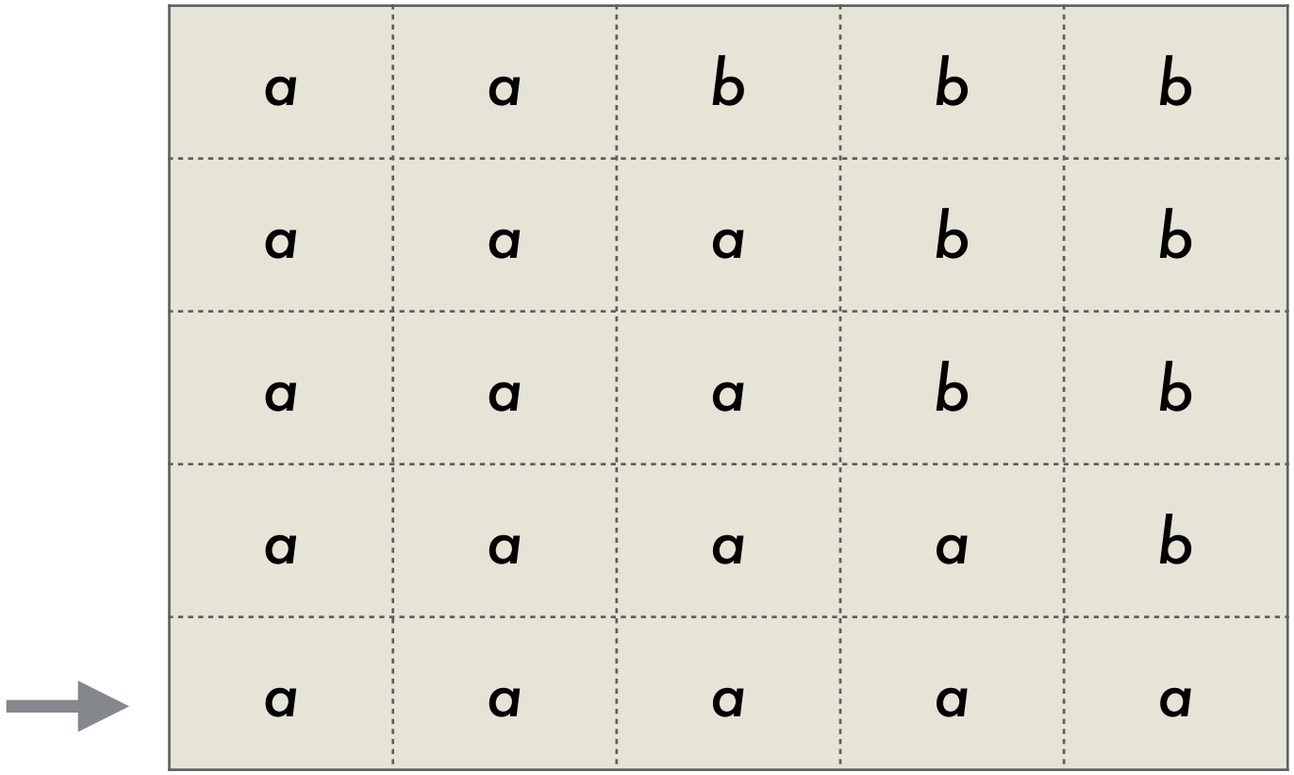


Domino systems and corridor tilings

Set of dominoes :

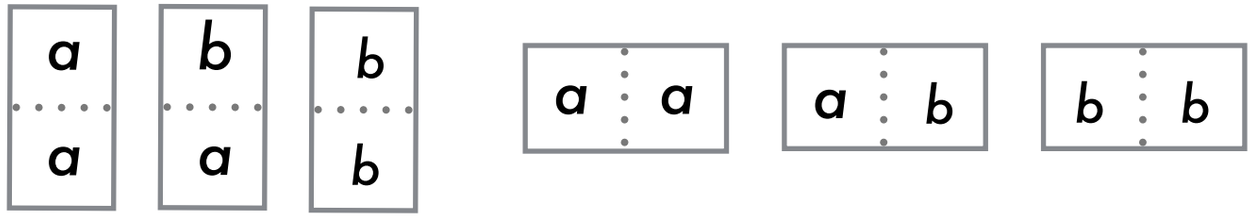


Corridor tiling :

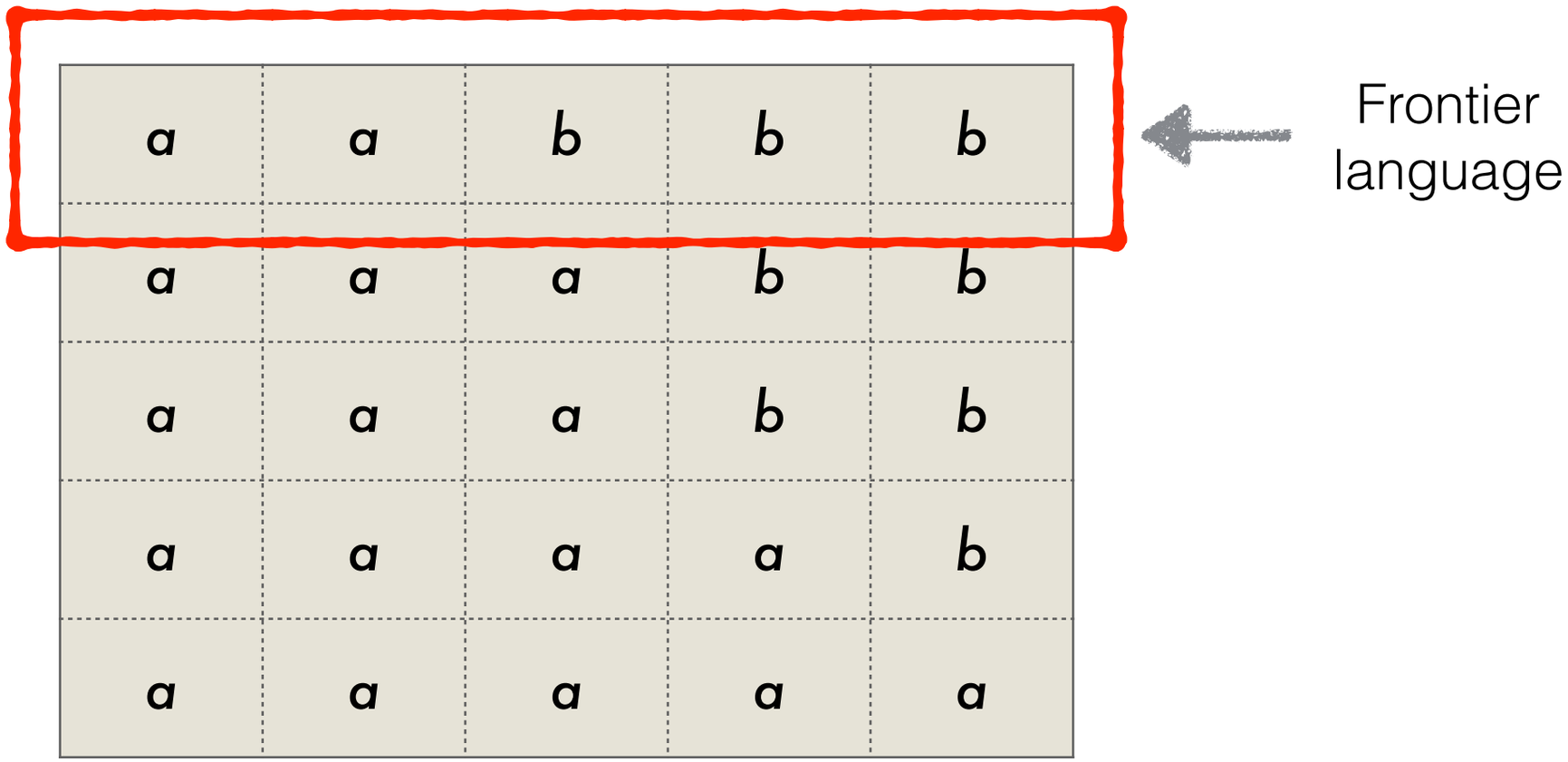


Domino systems and corridor tilings

Set of dominoes :

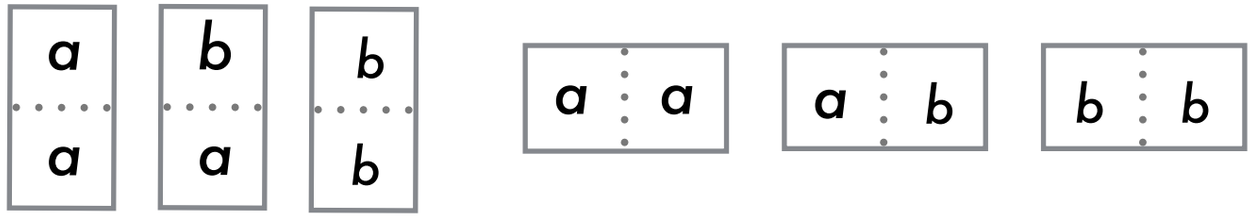


Corridor tiling :

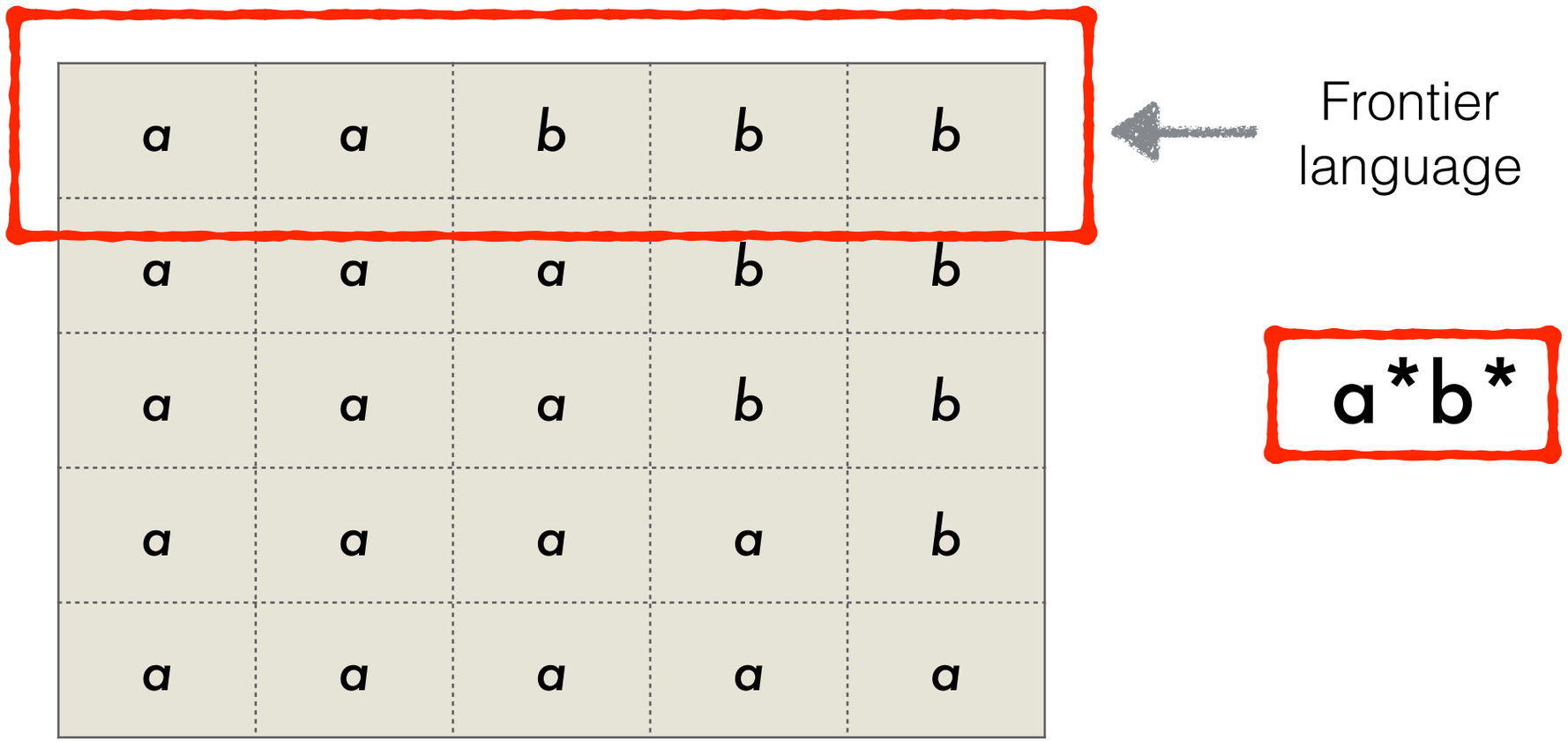


Domino systems and corridor tilings

Set of dominoes :



Corridor tiling :



Theorem.[Latteux, Simplot, 97]
Every context-sensitive language is the frontier language of a domino system.
and vice versa

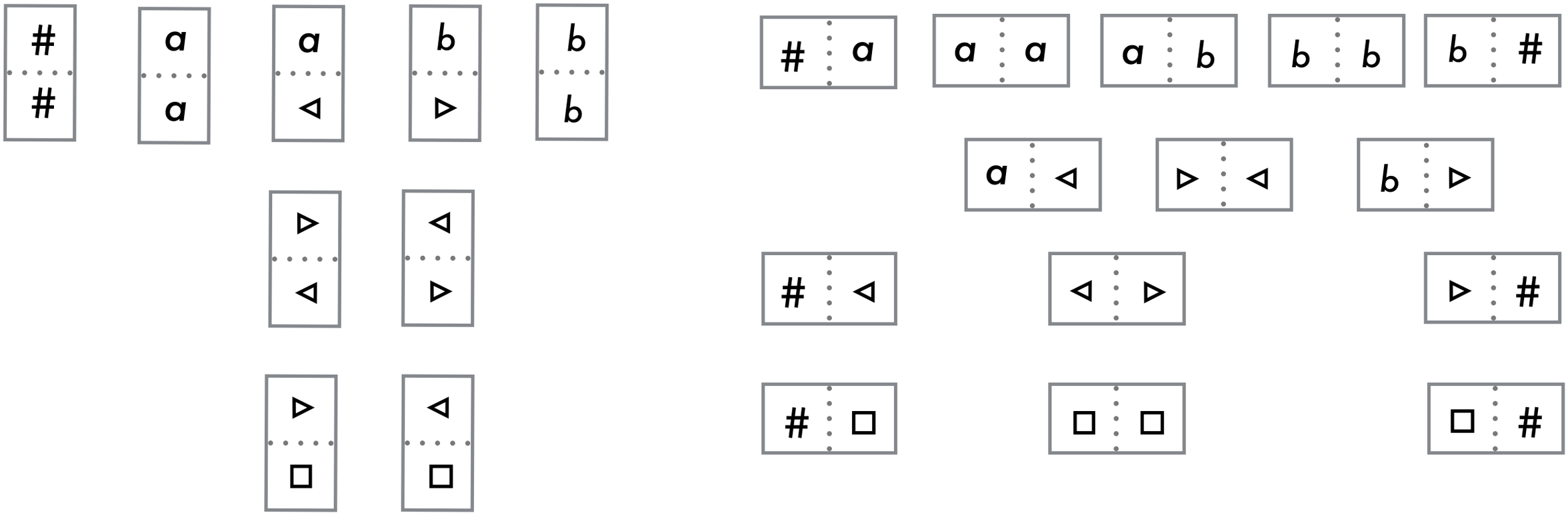
Given a domino system, construct a game G that covers the frontier language
For every observation history π, accept if there exists a corridor tiling from π

Example: $a^n b^n$

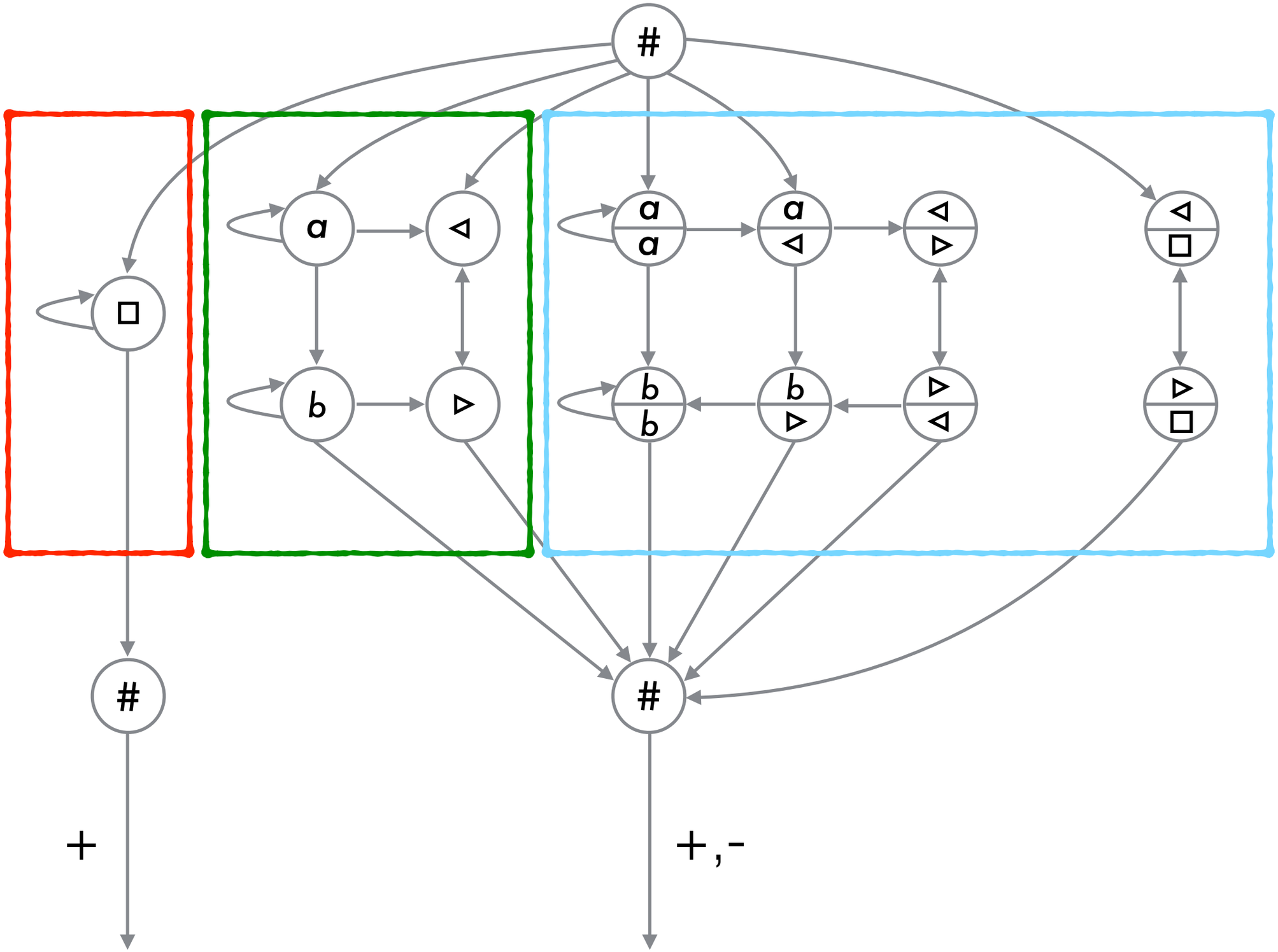
#	a	a	b	b	#
#	a	◂	▸	b	#
#	◂	▸	◂	▸	#
#	□	□	□	□	#

Example: $a^n b^n$

#	a	a	b	b	#
#	a	◁	▷	b	#
#	◁	▷	◁	▷	#
#	□	□	□	□	#

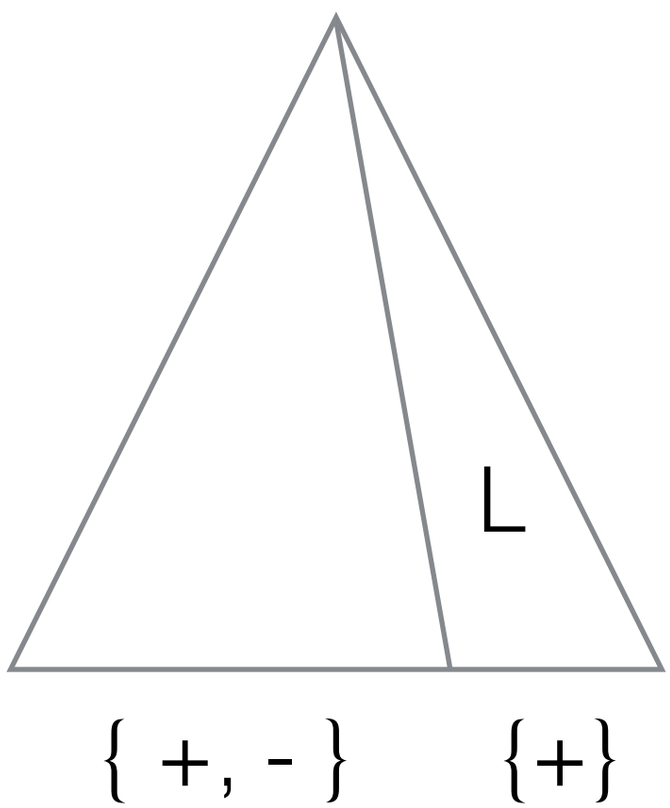


Example: $a^n b^n$



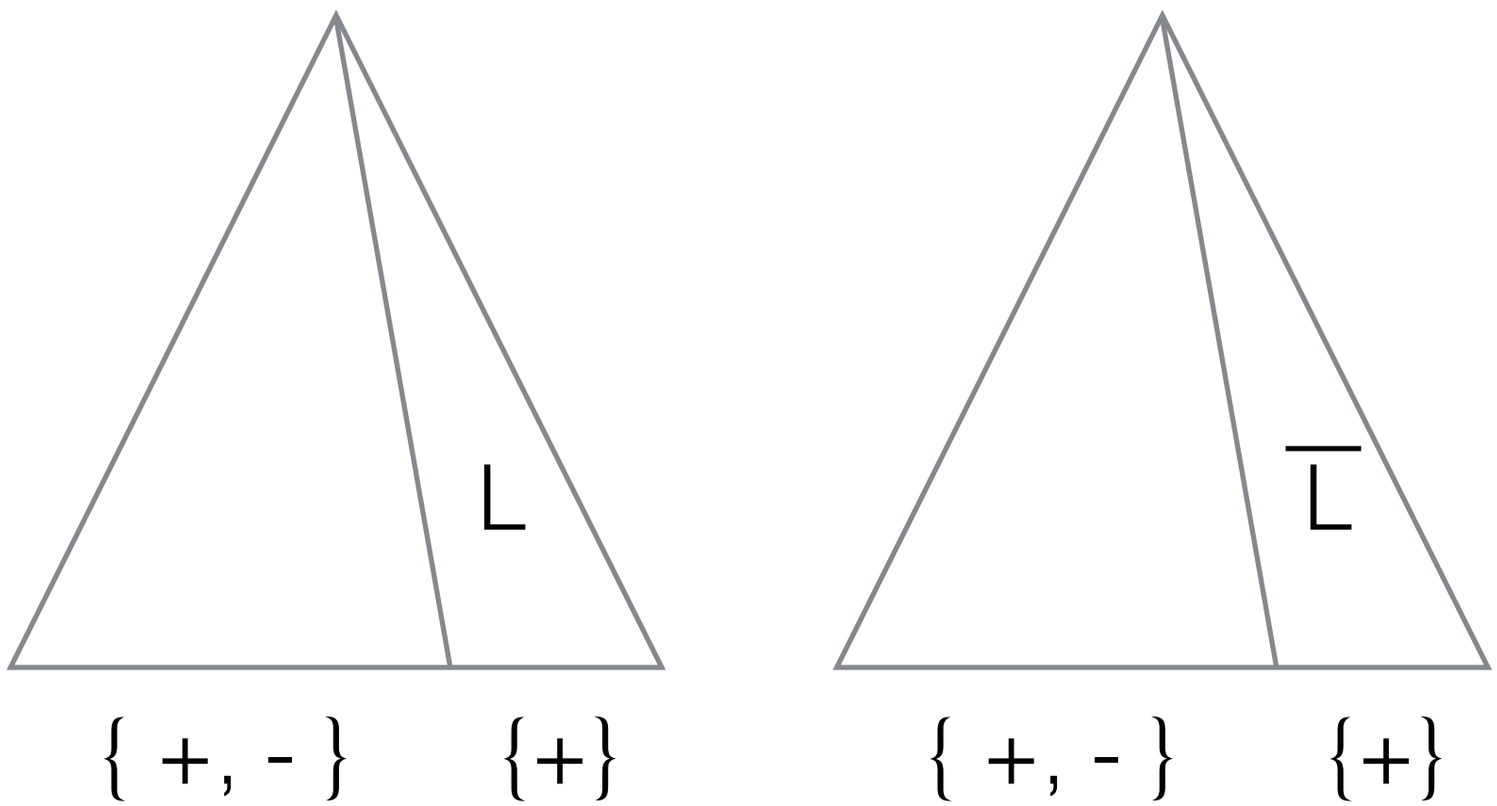
Characterising

Given L context-sensitive, construct G that covers it.
Construct G' to cover the complement.
Combine G and G' to characterise L.



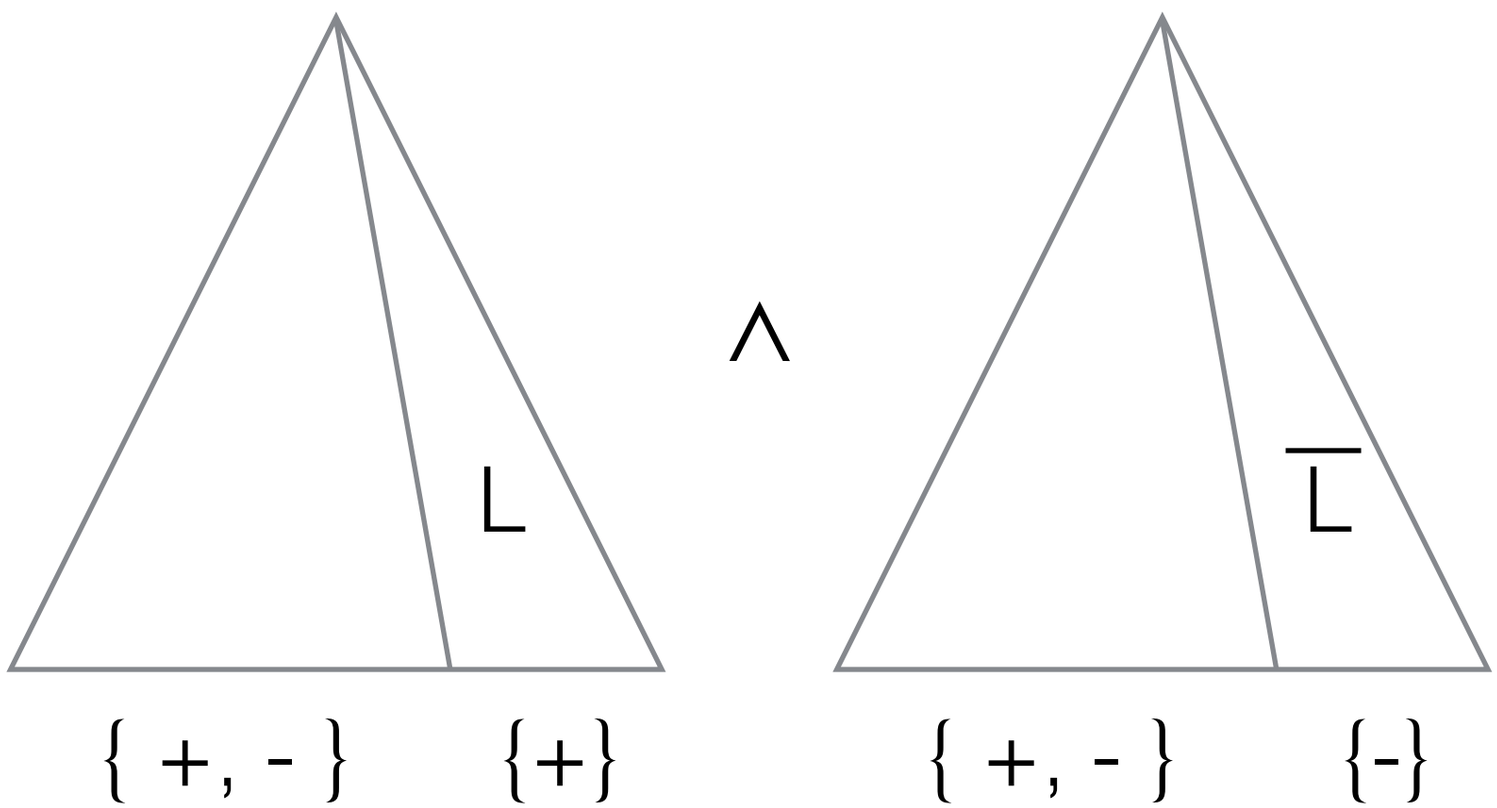
Characterising

Given L context-sensitive, construct G that covers it.
Construct G' to cover the complement.
Combine G and G' to characterise L.



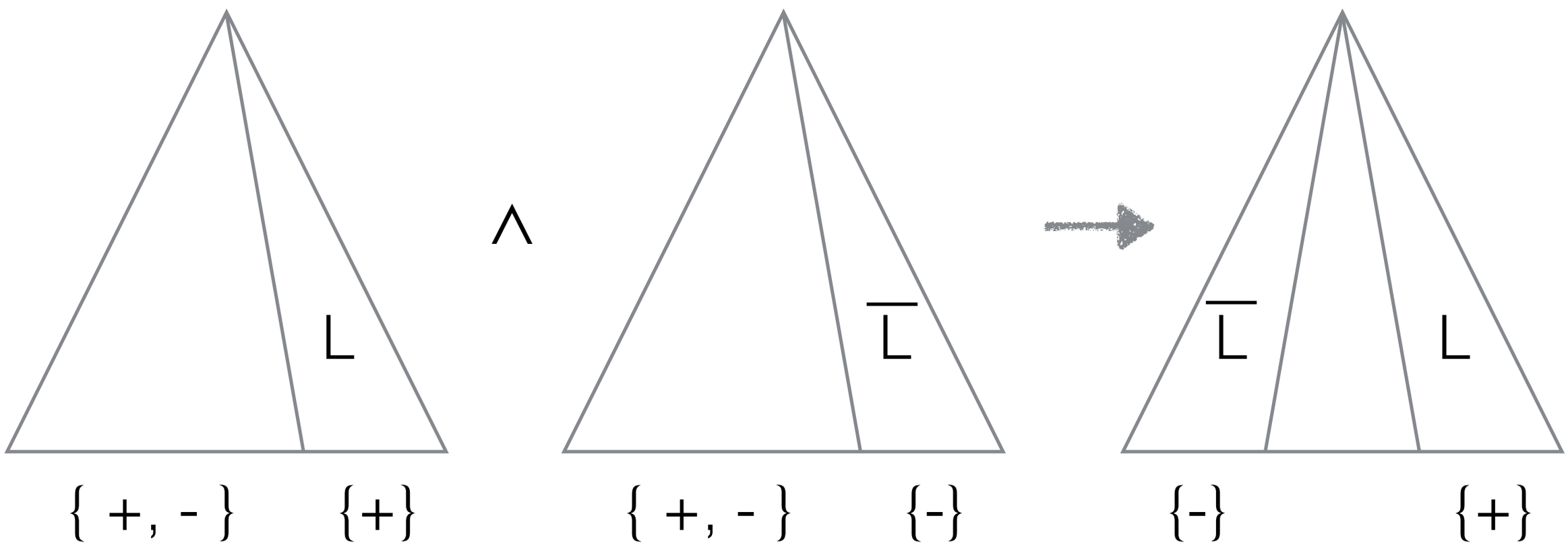
Characterising

Given L context-sensitive, construct G that covers it.
Construct G' to cover the complement.
Combine G and G' to characterise L.



Characterising

Given L context-sensitive, construct G that covers it.
Construct G' to cover the complement.
Combine G and G' to characterise L.



Consequences

(2) Take a game characterising a PSPACE-hard language L,

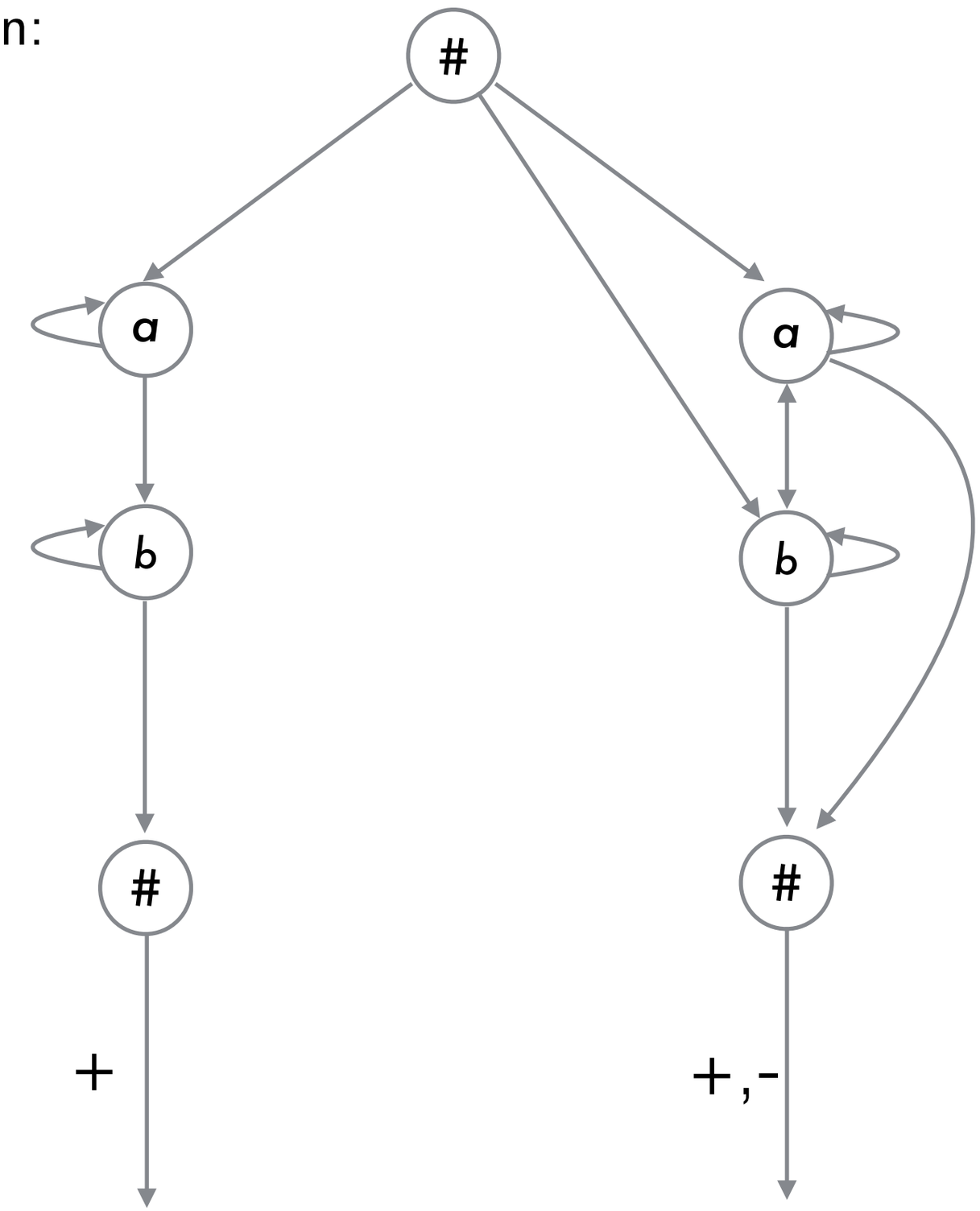
$$s(\pi) = 1 \iff \pi \in L$$

(3) Combine games characterising L and the empty language
$s(\pi) \equiv 0$ only candidate, winning if $L = \emptyset$

(4) At π, if exists $\pi' \sim^* \pi$ that ends at v_-, play $-$, else play $+$
winning strategy, (nondeterministic) linear space

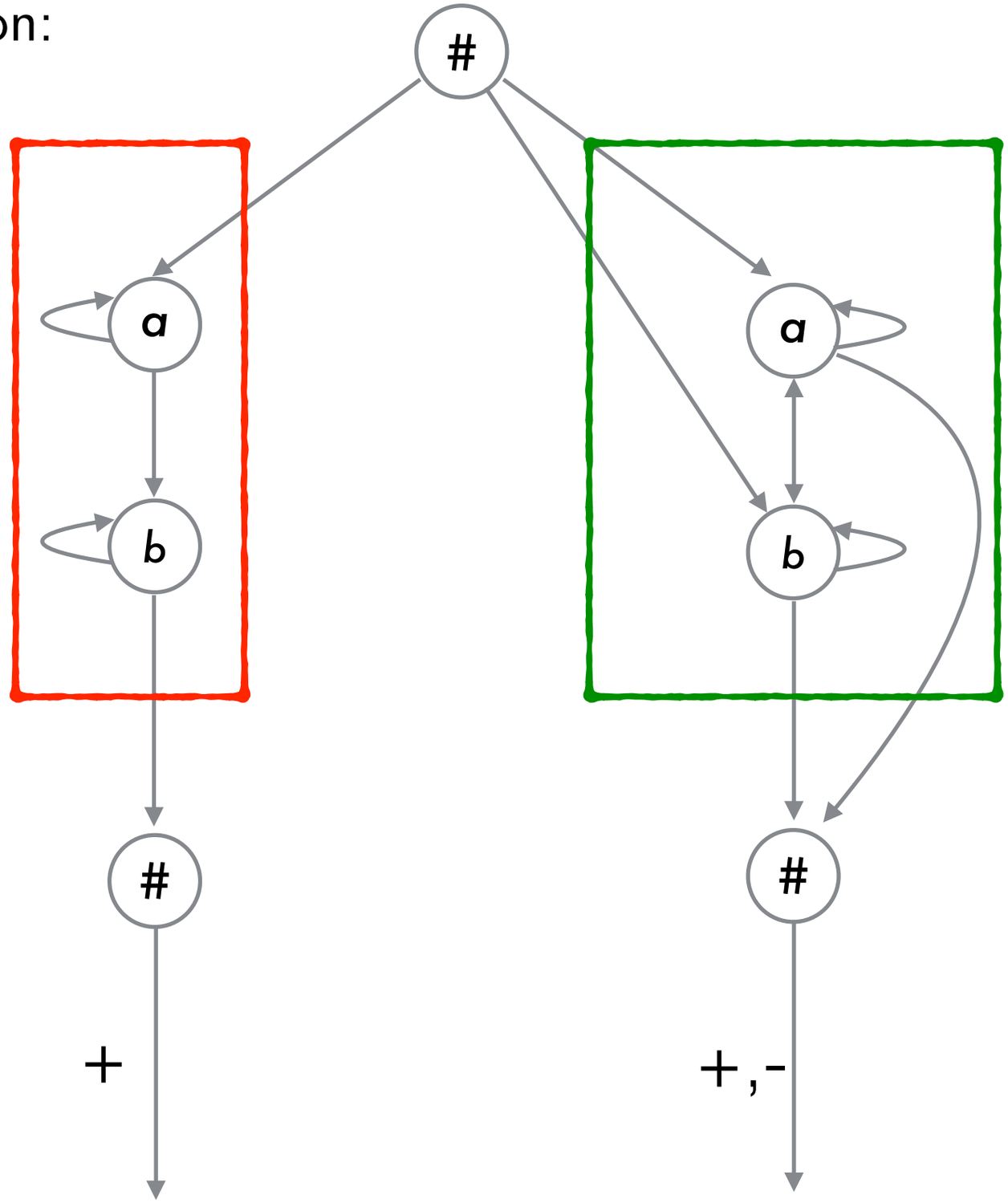
Cover regular languages

From a finite automaton:



Cover regular languages

From a finite automaton:



Cover Context-free languages

Chomsky-Schützenberger theorem.
Every context-free language L can be written as $h(R \cap D)$ with

1. h is a homomorphism
2. D is a Dyck language
3. R is a regular language

Theorem.[Okhotin 12] Also works with

1. h letter-to-letter homomorphism
2. D Dyck language with neutral symbols

Construction ingredients

Construction is possible with ordered observations.
Order on Γ: $\beta^1(v) \geq \beta^2(v)$ for all v

Construction:

1. Homomorphic image: add a copy of the clique thread with $\beta^1(v) = h(\beta^2(v)$
2. Dyck language: easy, threads to erase matching parentheses
3. Intersection with R: product with an NFA

Intersection closure

With ordered observations we can capture more than context-free languages:
Lemma. The class of games with ordered observations is closed under

intersection.

Nondeterministic linear time languages

Theorem.[Book, Greibach, 69] Every language in $NTIME(n)$ can be written as $h(K_1 \cap K_2 \cap K_3)$ with

1. h is a letter-to-letter homomorphism
2. K_1, K_2 and K_3 are context-free languages

Corollary. Every language in $NTIME(n)$ can be covered by a consensus acceptor game with ordered observations