Asynchronous approach in the plane: A deterministic polynomial algorithm[1]

Sébastien Bouchard[†] Marjorie Bournat[†] Yoann Dieudonné[‡]
Swan Dubois[†] Franck Petit[†]

[†]Sorbonne Universités, UPMC Univ Paris 06, CNRS, INRIA, LIP6 UMR 7606, Paris, France

[‡]MIS Lab., Université de Picardie Jules Verne, France

ESTATE Meeting August 31 2017

[1]This work was performed within Project ESTATE (Ref. ANR-16-CE25-0009-03), supported by French state funds managed by the ANR (Agence Nationale de la Recherche).

Deterministic asynchonous approach in the plane

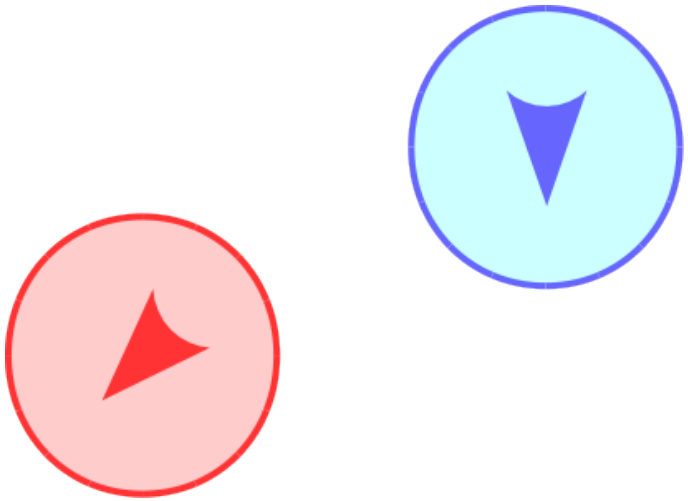
Deterministically bring two mobile asynchronous agents initially separated within each other's range of vision/communication

Deterministic asynchonous approach in the plane

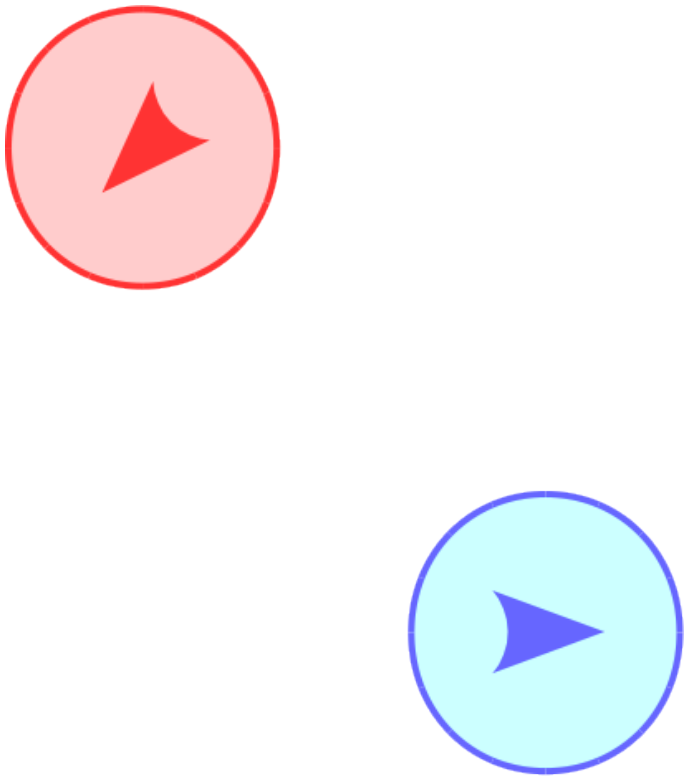
Deterministic asynchonous approach in the plane

Deterministic asynchonous approach in the plane

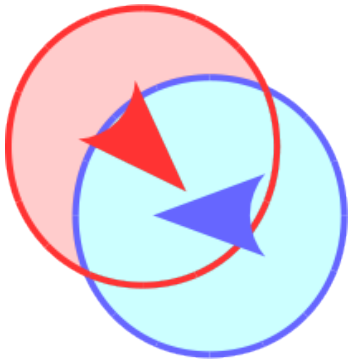
Deterministic asynchonous approach in the plane



Deterministic asynchonous approach in the plane



Deterministic asynchonous approach in the plane



A first exponential solution

Czyzowicz, Pelc, and Labourel

Deterministic approach algorithm whose cost (total distance traveled) is exponential in D, the initial distance between the agents.

A first exponential solution

Czyzowicz, Pelc, and Labourel

Deterministic approach algorithm whose cost (total distance traveled) is exponential in D, the initial distance between the agents.

Does a polynomial algorithm exist?

A first exponential solution

Czyzowicz, Pelc, and Labourel

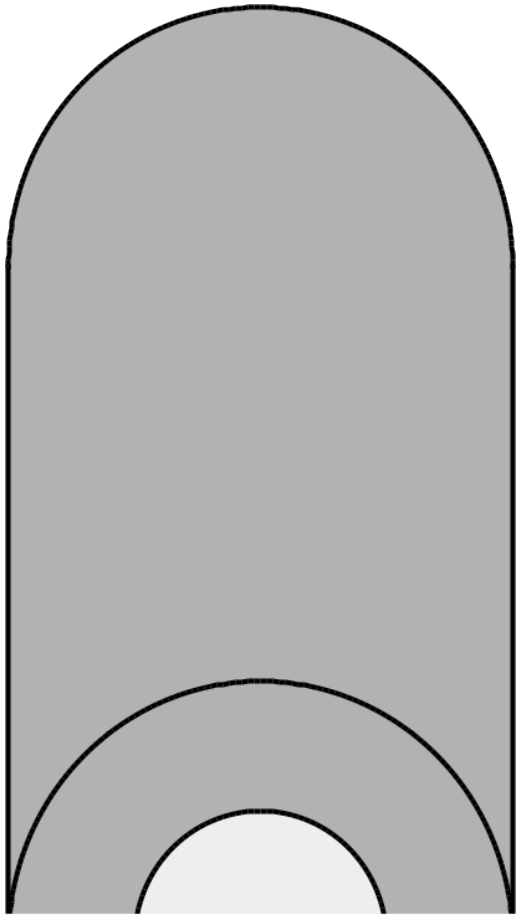
Deterministic approach algorithm whose cost (total distance traveled) is exponential in D, the initial distance between the agents.

Does a polynomial algorithm exist?

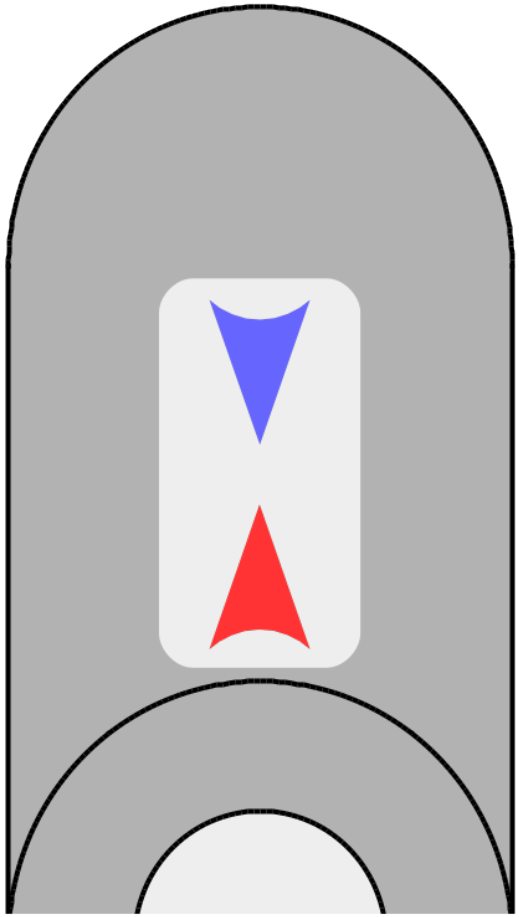
Tunnels

Two routes r_1 and r_2 respectively from nodes v_1 and v_2 form a *tunnel* iff r_1 has a prefix p which ends at v_2 and r_2 has a prefix which is $\overline{p}$ *i.e.*, the reverse of p.

A first exponential solution



A first exponential solution



Polynomial algorithms thanks to simplifying assumptions

Bampas, Czyzowicz, Gasieniec, Ilcinkas, and Labourel

The agents know the coordinates of their initial location in a common coordinate system (some kind of GPS).
Nearly optimal solution.

Polynomial algorithms thanks to simplifying assumptions

Bampas, Czyzowicz, Gasieniec, Ilcinkas, and Labourel

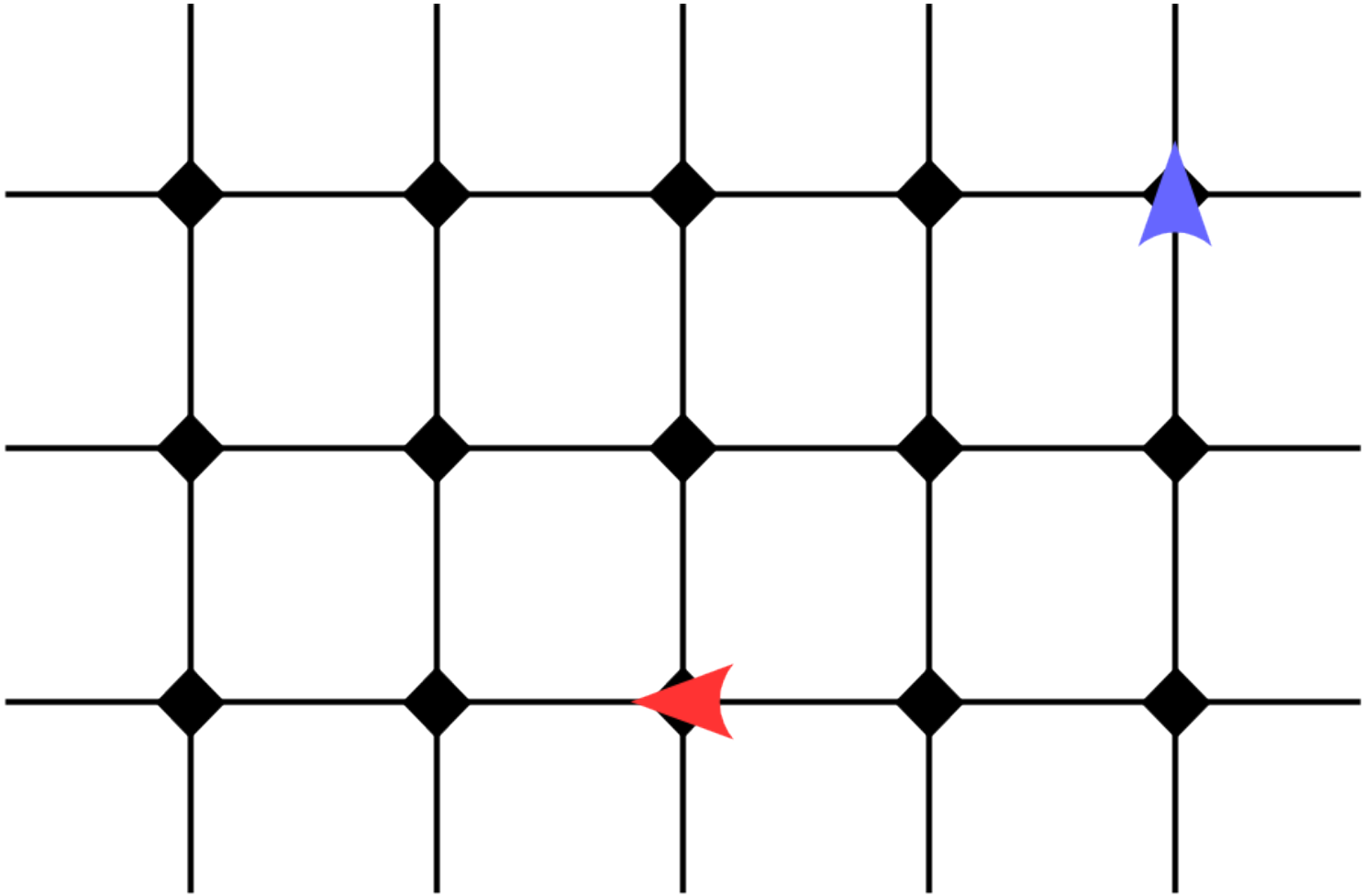
The agents know the coordinates of their initial location in a common coordinate system (some kind of GPS).
Nearly optimal solution.

Dieudonné and Pelc

Each agent is assigned a constant speed which cannot change afterwards.

Reduction to rendez-vous in the infinite grid

Reduction to rendez-vous in the infinite grid



Precisions about the model

- Have to meet at a same point (either node or inside edge)

Precisions about the model

- Have to meet at a same point (either node or inside edge)
- Fully asynchronous setting

Precisions about the model

- Have to meet at a same point (either node or inside edge)
- Fully asynchronous setting
- Execute locally our algorithm

Precisions about the model

- Have to meet at a same point (either node or inside edge)
- Fully asynchronous setting
- Execute locally our algorithm
- Vision limited to current node and no communication

Precisions about the model

- Have to meet at a same point (either node or inside edge)
- Fully asynchronous setting
- Execute locally our algorithm
- Vision limited to current node and no communication
- Unlimited memory

Precisions about the model

- Have to meet at a same point (either node or inside edge)
- Fully asynchronous setting
- Execute locally our algorithm
- Vision limited to current node and no communication
- Unlimited memory
- Any distinct inital location and integer label

Precisions about the model

- Have to meet at a same point (either node or inside edge)
- Fully asynchronous setting
- Execute locally our algorithm
- Vision limited to current node and no communication
- Unlimited memory
- Any distinct inital location and integer label
- Compass

Our results

Deterministic algorithm whose cost (total number of edge traversals) is polynomial in the initial distance D and the length of the shorter label.

Tackling the unknown distance

Simplifying assumption

Assume a synchronous setting

Tackling the unknown distance

Simplifying assumption

Assume a synchronous setting

First issue

Initial separating distance D unknown

Tackling the unknown distance

Simplifying assumption

Assume a synchronous setting

First issue

Initial separating distance D unknown

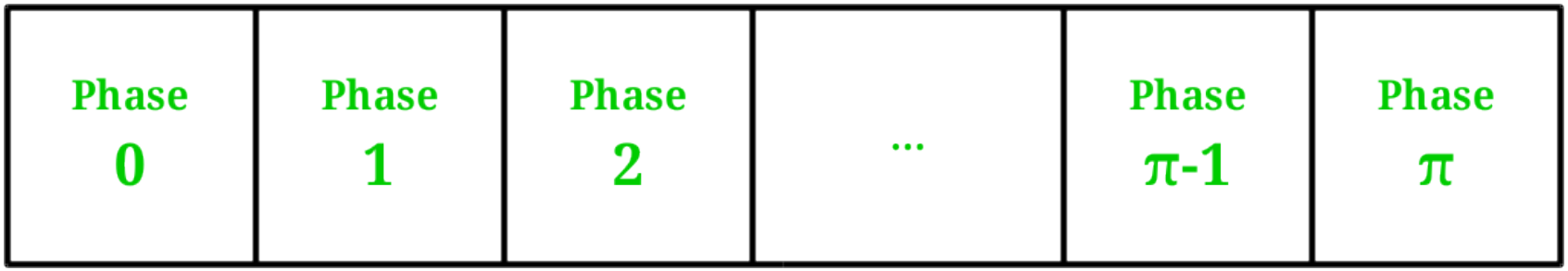
Solution: *phases*

Perform successive tests called phases:

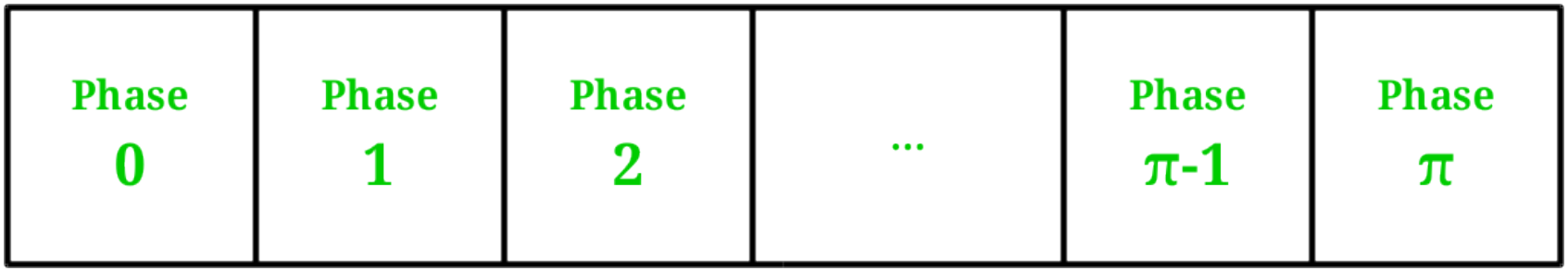
During each test, the agents act as if the tested value were the good one.

Phase $i \rightarrow$ Tested value 2^i

Tackling the unknown distance



Tackling the unknown distance



The good phase

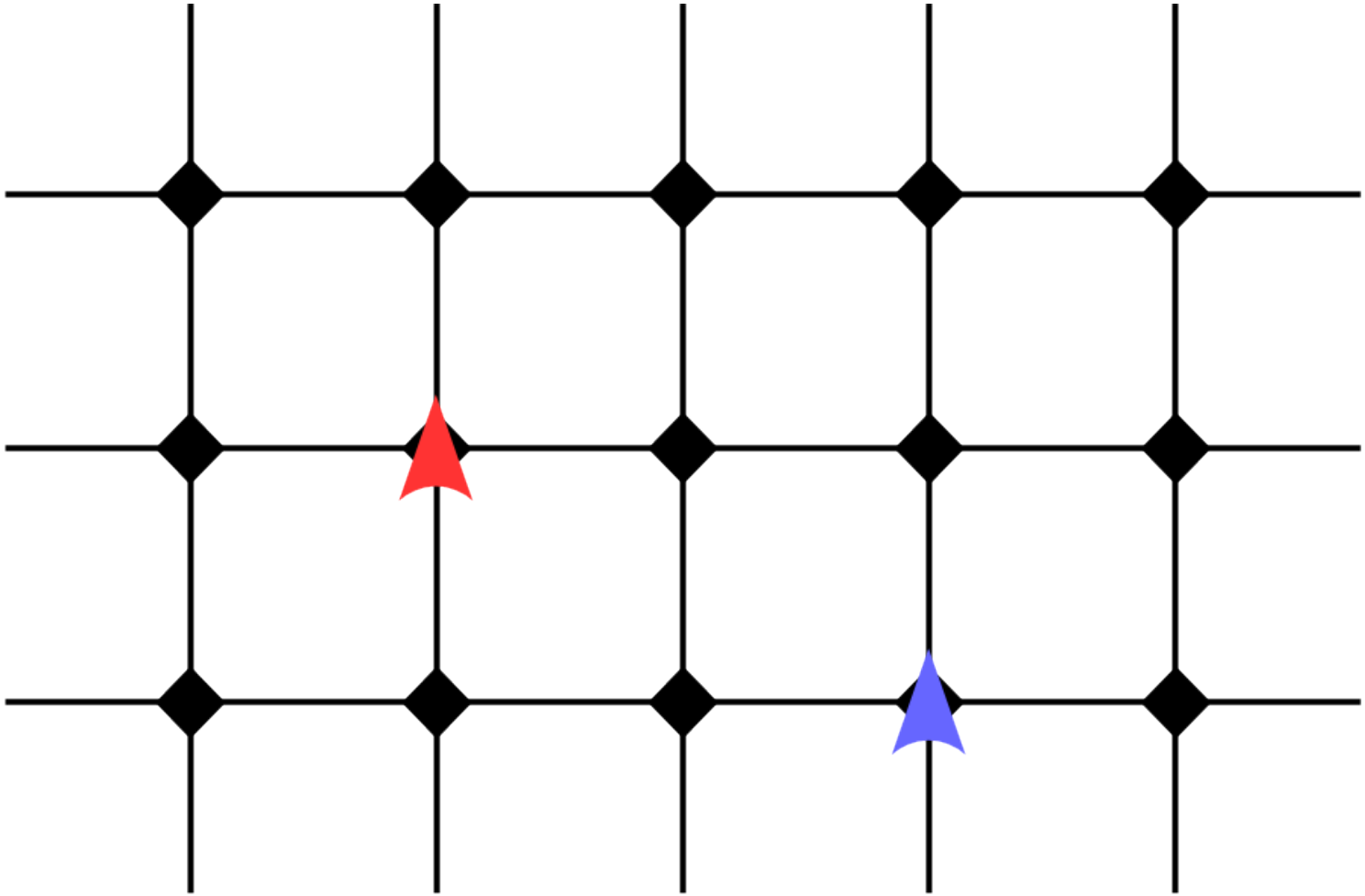
Phase π with $2^{\pi-1} < D \leq 2^{\pi}$

Tackling the symmetry

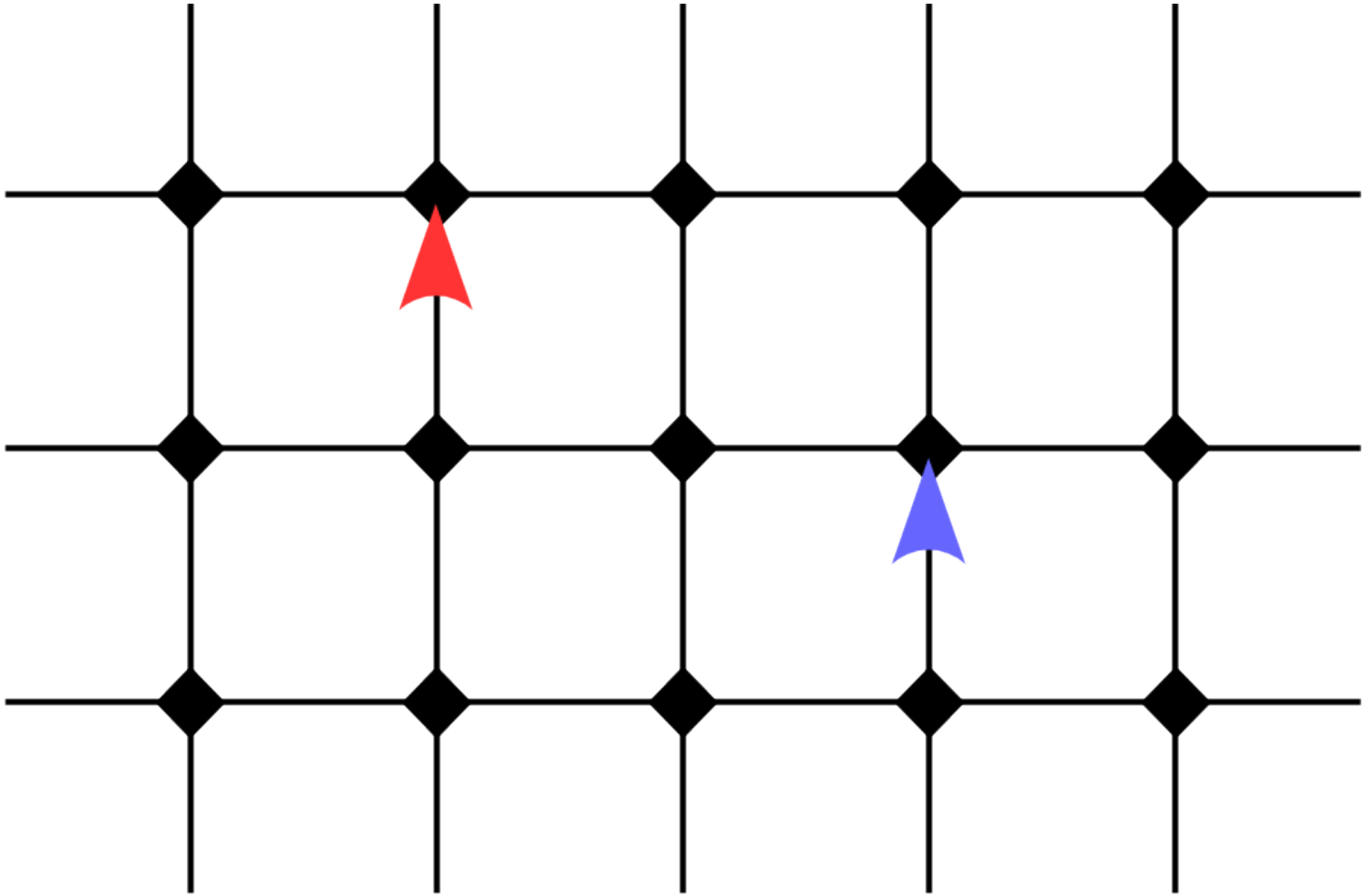
Another issue

The grid is symmetric: if the agents keep performing the same actions they never meet

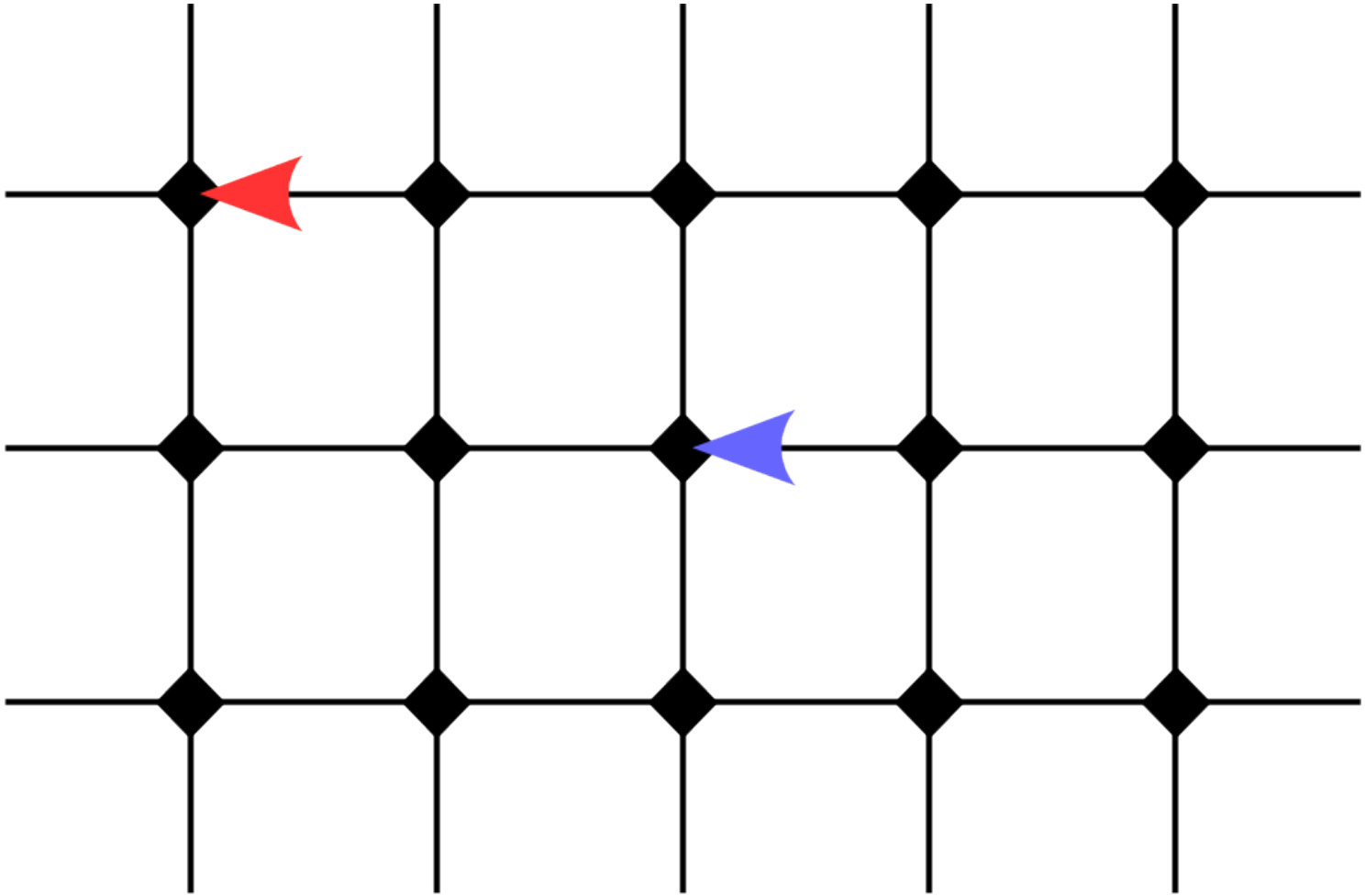
Tackling the symmetry



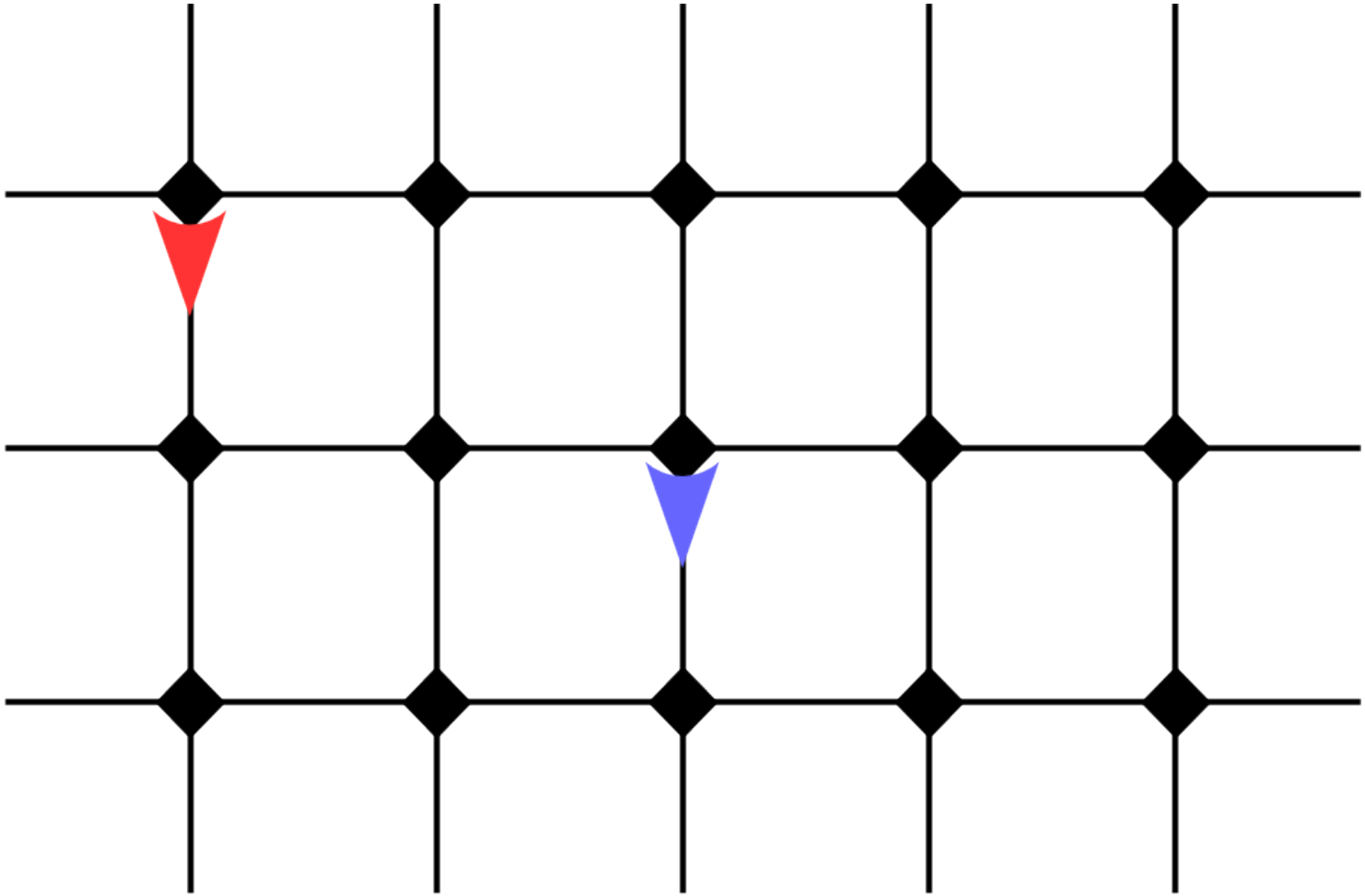
Tackling the symmetry



Tackling the symmetry



Tackling the symmetry



Tackling the symmetry

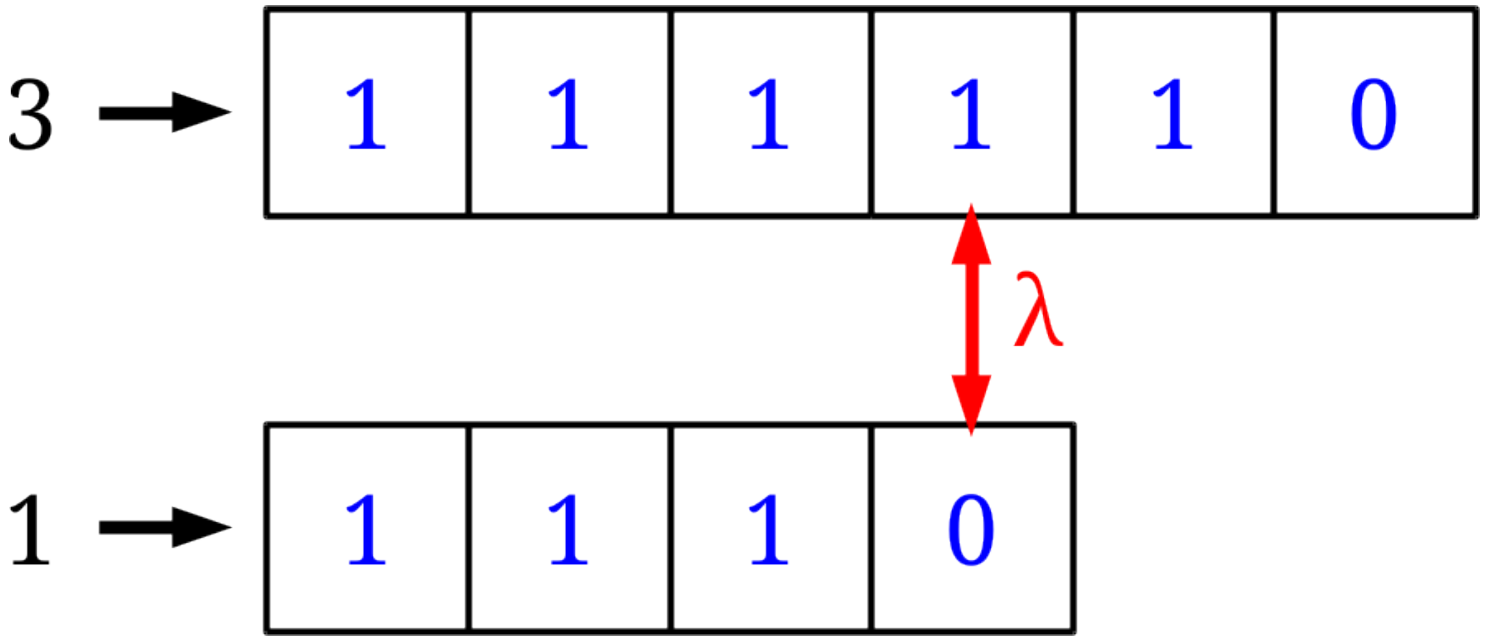
Another issue

The grid is symmetric: if the agents keep performing the same actions they never meet

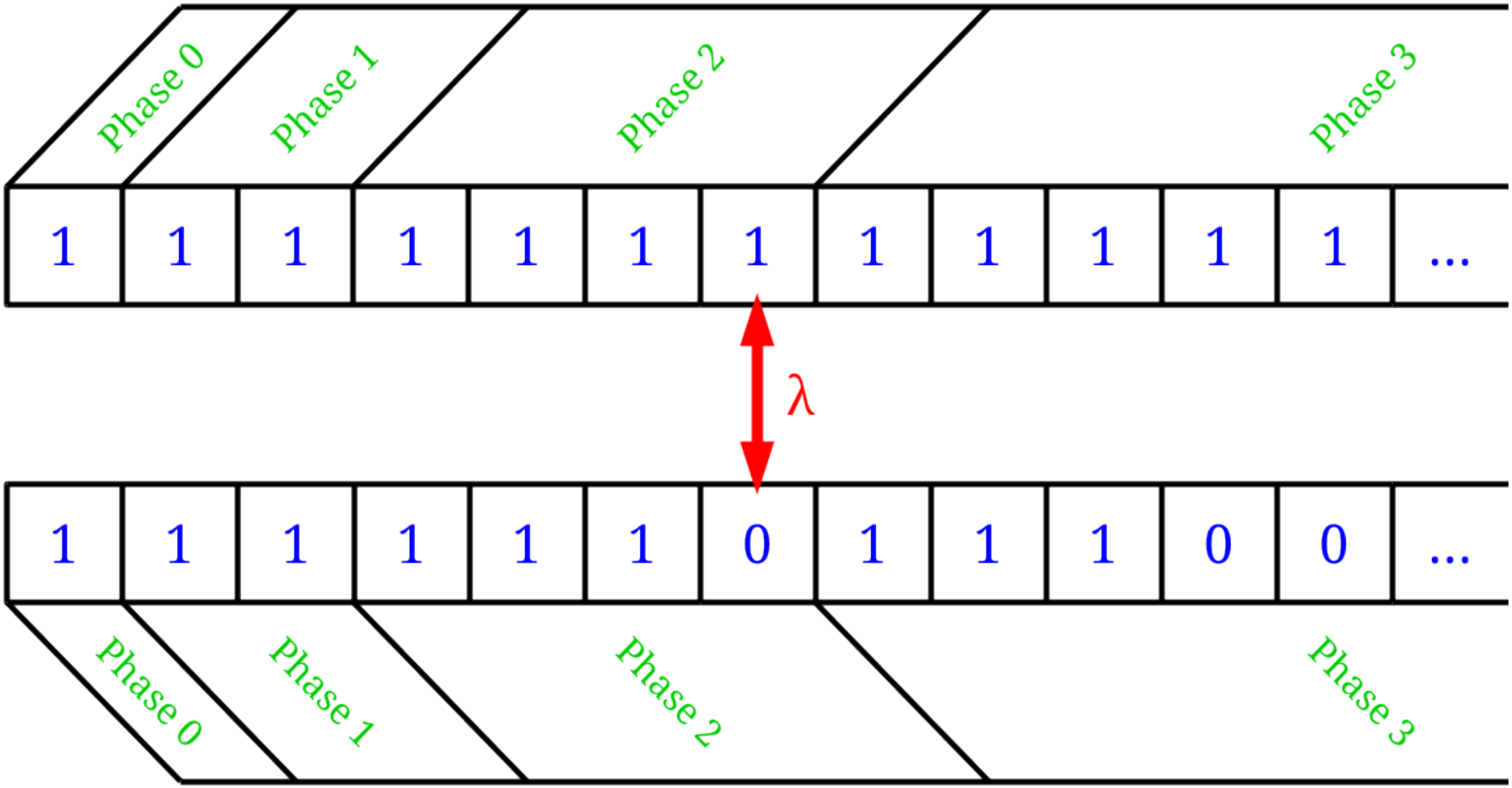
Dessmark, Fraigniaud, Kowalski, and Pelc

Roughly speaking: each agent reads a binary sequence obtained from its label until they read different bits.

Tackling the symmetry



Tackling the symmetry



Tackling the symmetry

Another issue

The grid is symmetric: if the agents keep performing the same actions they never meet

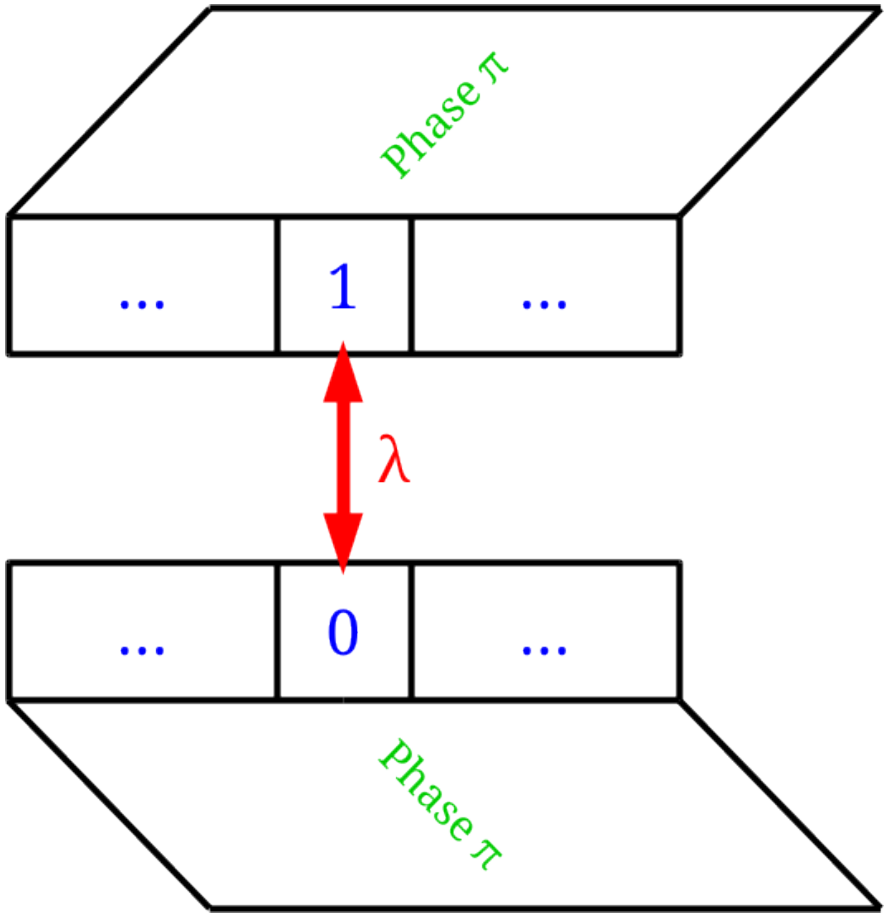
Dessmark, Fraigniaud, Kowalski, and Pelc

Roughly speaking: each agent reads a binary sequence obtained from its label until they read different bits.

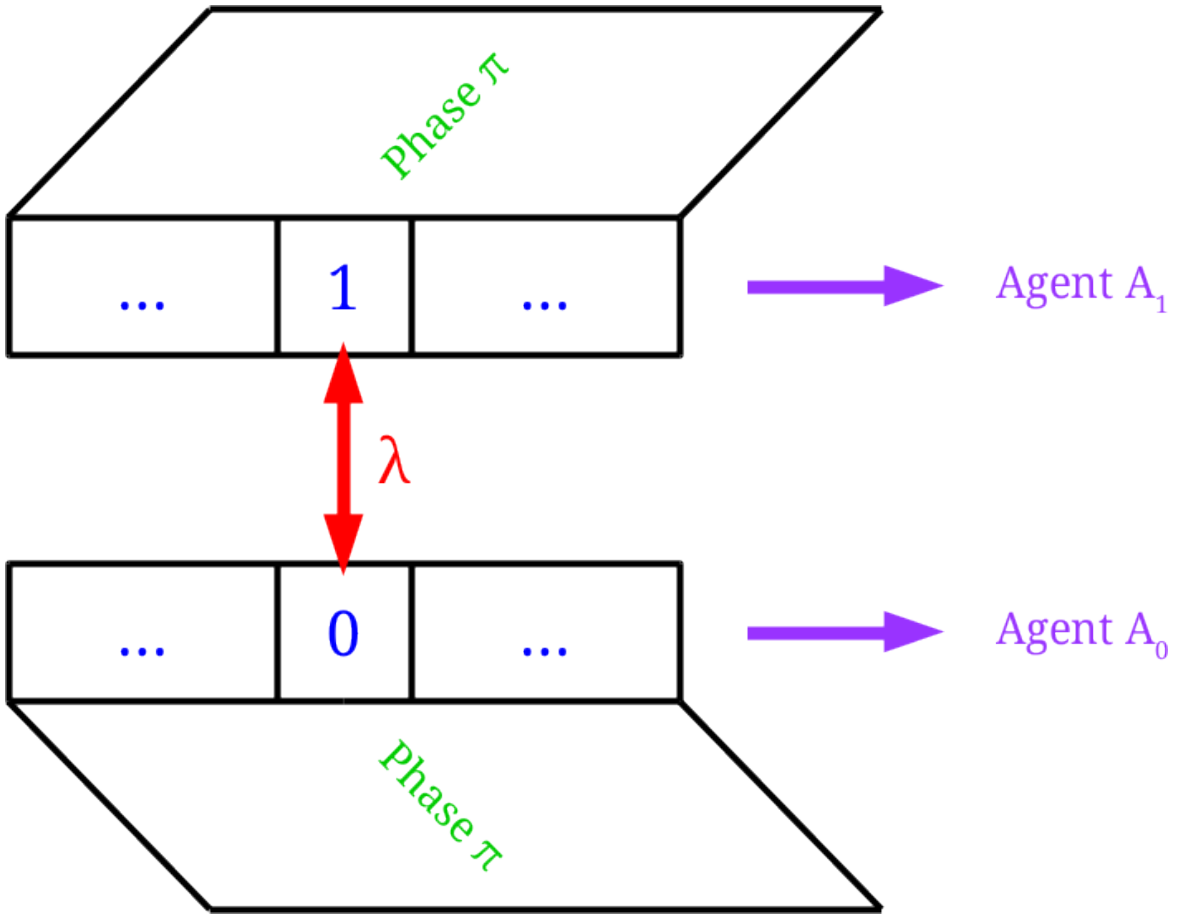
The good phase (update)

Phase π with $2^{\pi-1} < max\{D, \lambda\} \leq 2^{\pi}$

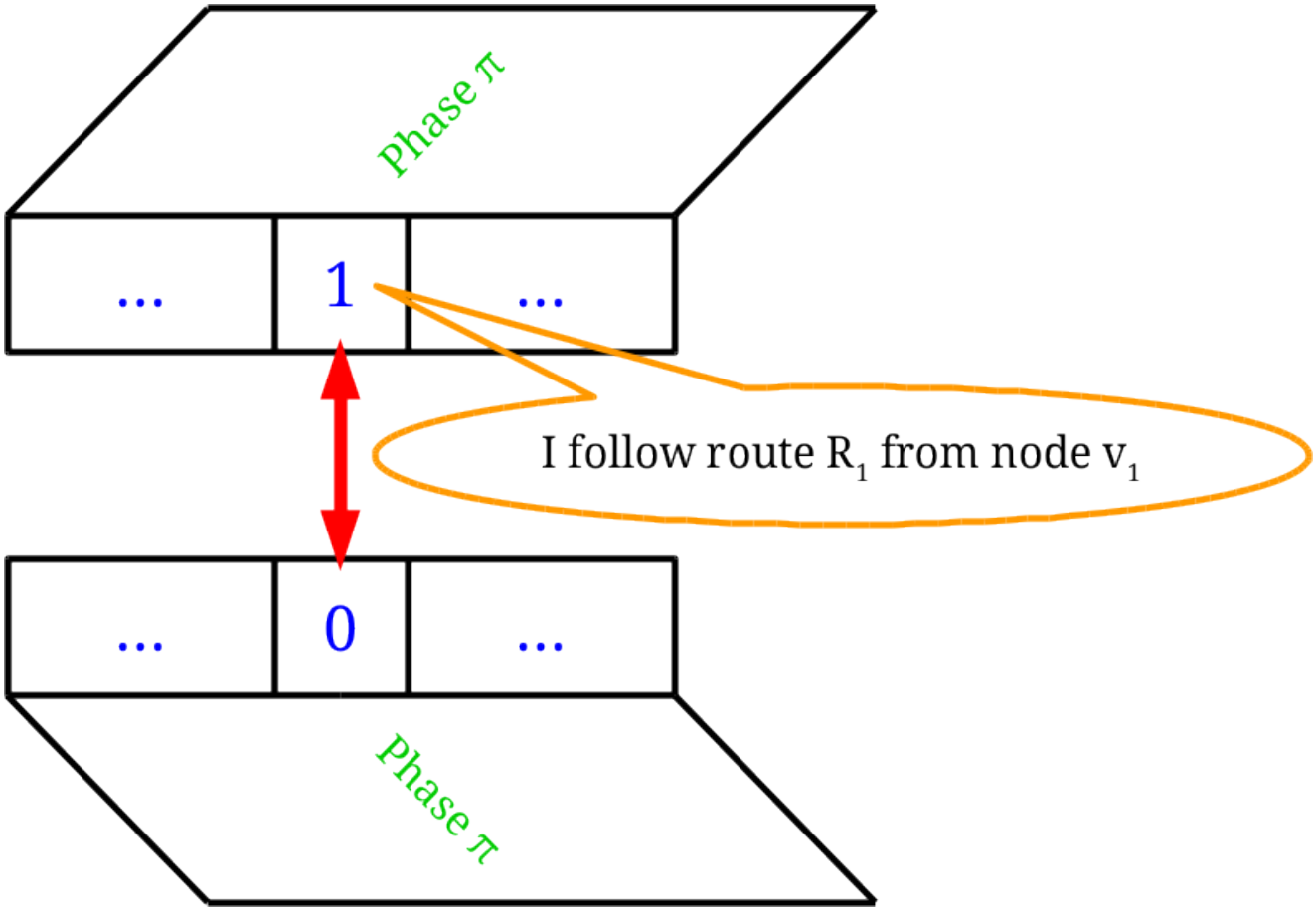
Focus and notations



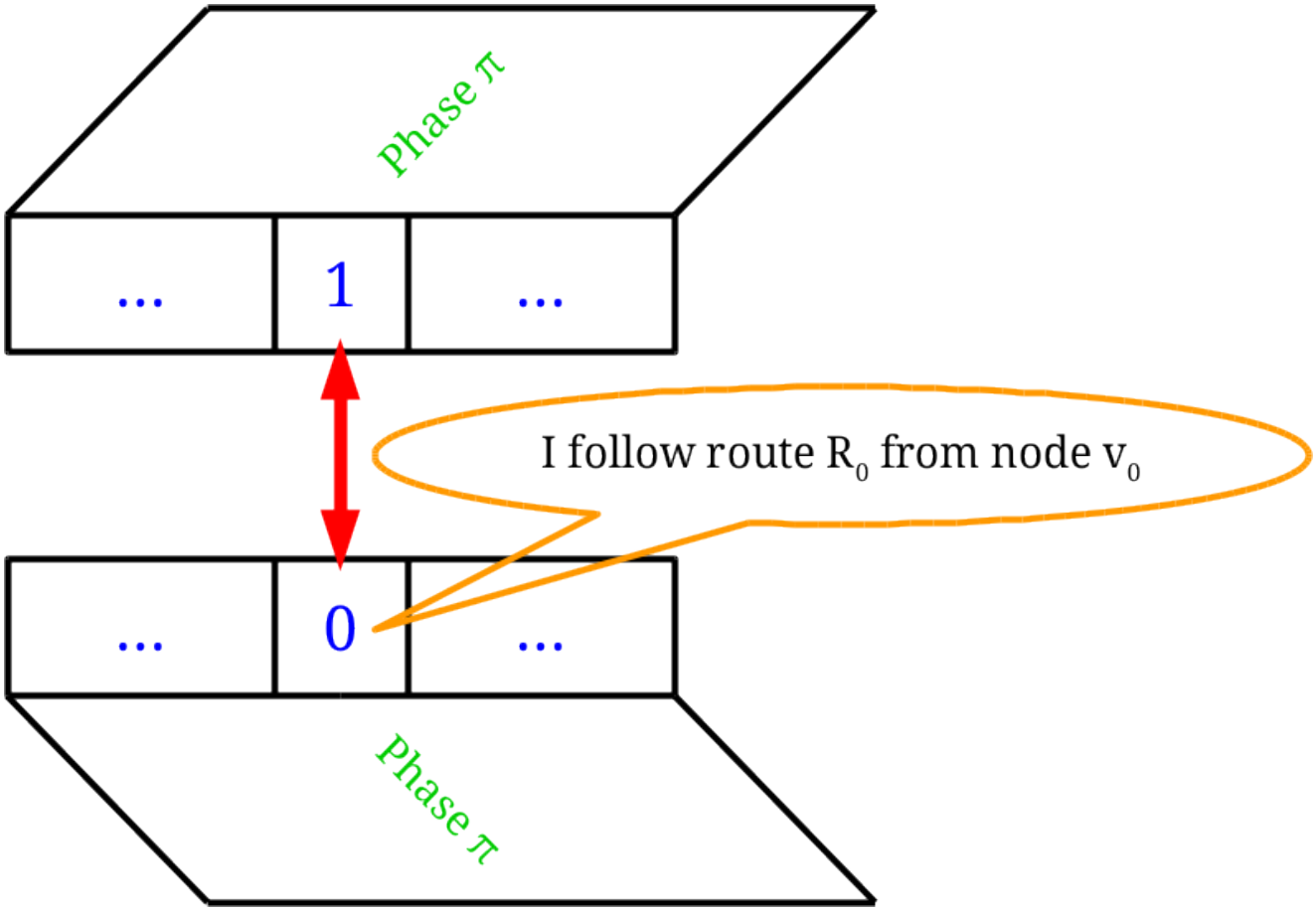
Focus and notations



Focus and notations



Focus and notations



Removing the synchronous setting assumption

Actual contribution

So far, well known techniques.

From now on, we get into our actual contribution.

Removing the synchronous setting assumption

Actual contribution

So far, well known techniques.

From now on, we get into our actual contribution.

Synchronisation mechanisms

The ahead agent *pushes* the other one:

Either they meet or the later agent makes some progress.

Removing the synchronous setting assumption

Actual contribution

So far, well known techniques.

From now on, we get into our actual contribution.

Synchronisation mechanisms

The ahead agent *pushes* the other one:

Either they meet or the later agent makes some progress.

Two kinds:

Removing the synchronous setting assumption

Actual contribution

So far, well known techniques.

From now on, we get into our actual contribution.

Synchronisation mechanisms

The ahead agent *pushes* the other one:

Either they meet or the later agent makes some progress.

Two kinds:

- First one: between bit readings, pushes the latest bit reading

Removing the synchronous setting assumption

Actual contribution

So far, well known techniques.

From now on, we get into our actual contribution.

Synchronisation mechanisms

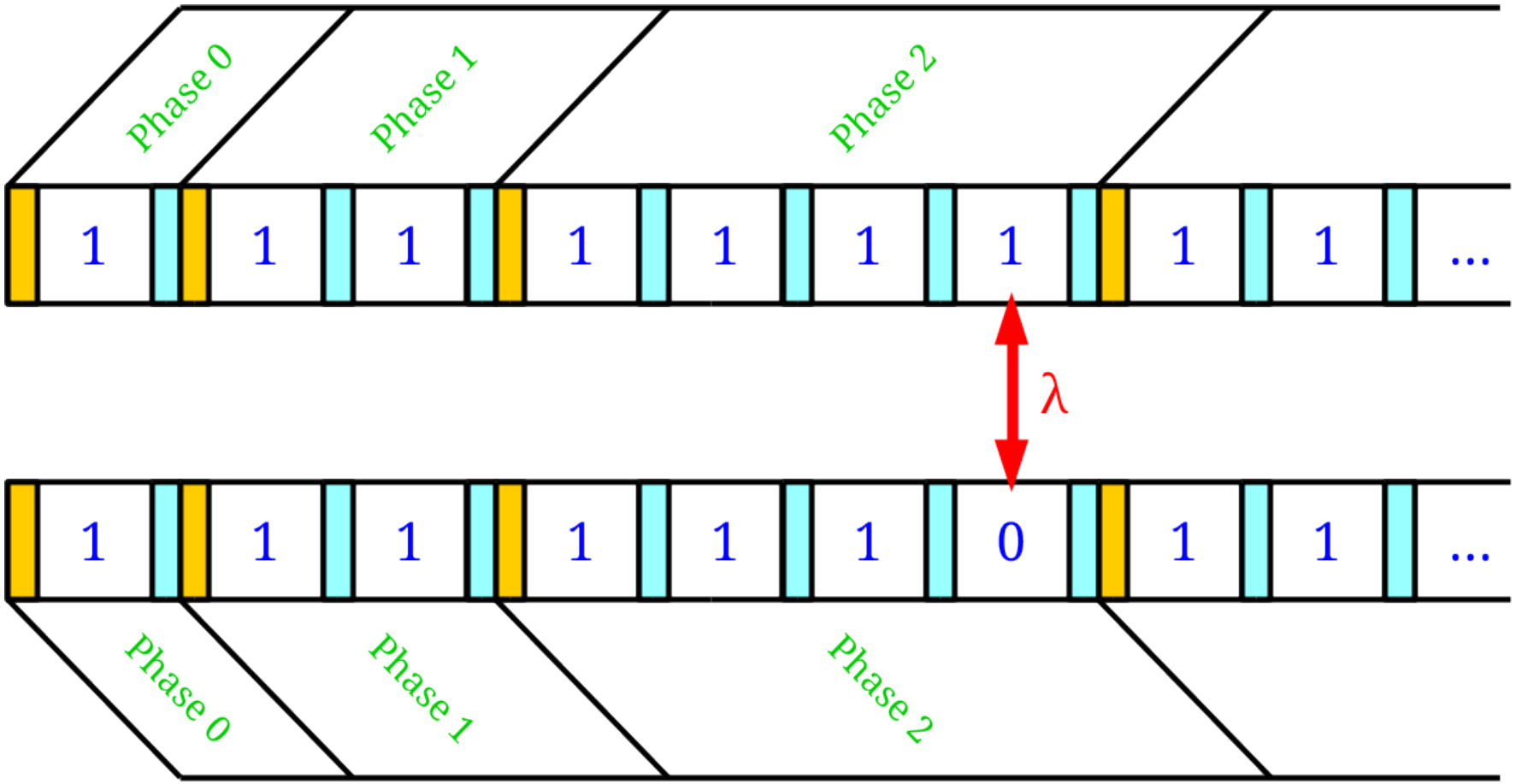
The ahead agent *pushes* the other one:

Either they meet or the later agent makes some progress.

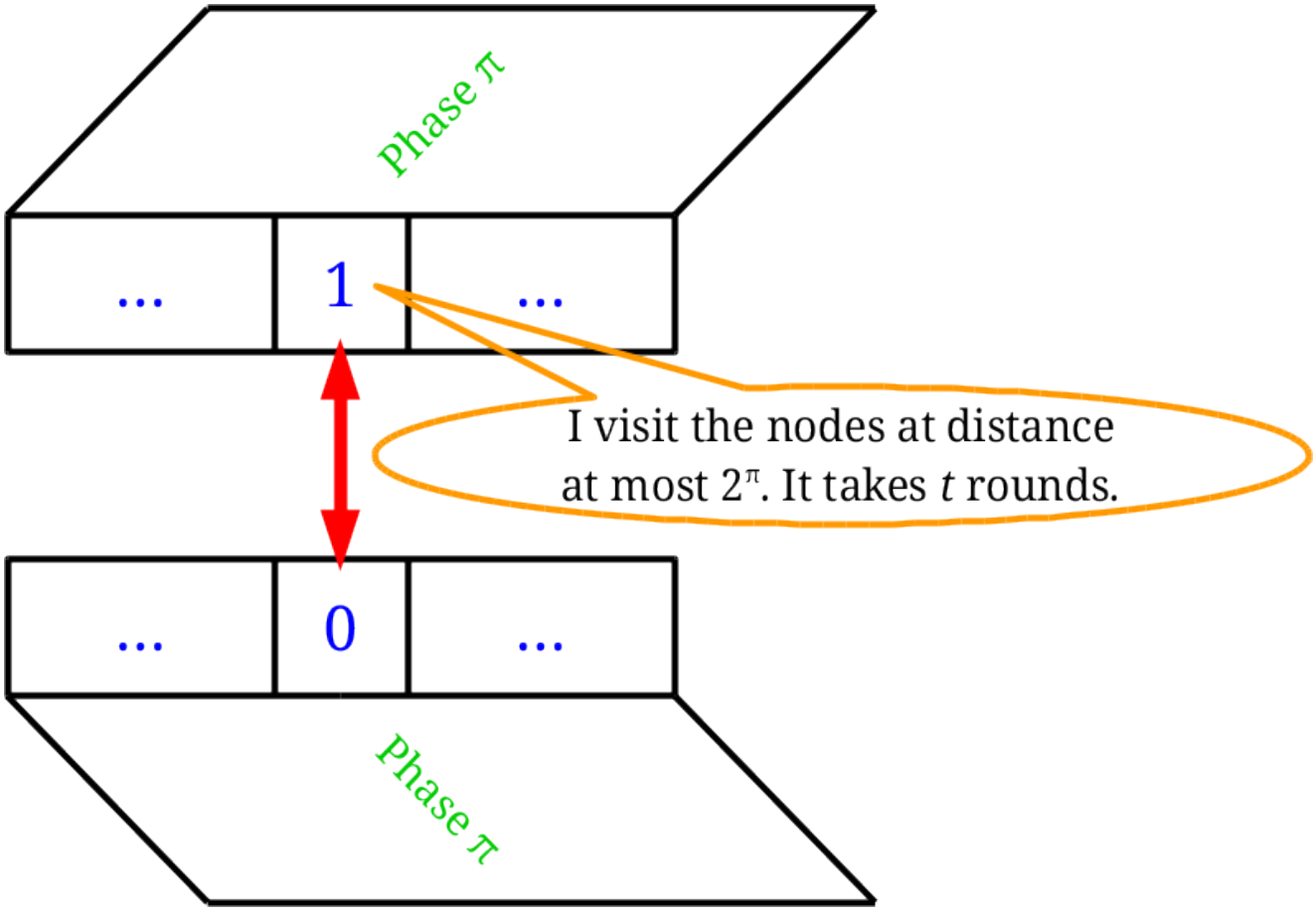
Two kinds:

- First one: between bit readings, pushes the latest bit reading
- Second one: between phases, pushes the previous phases

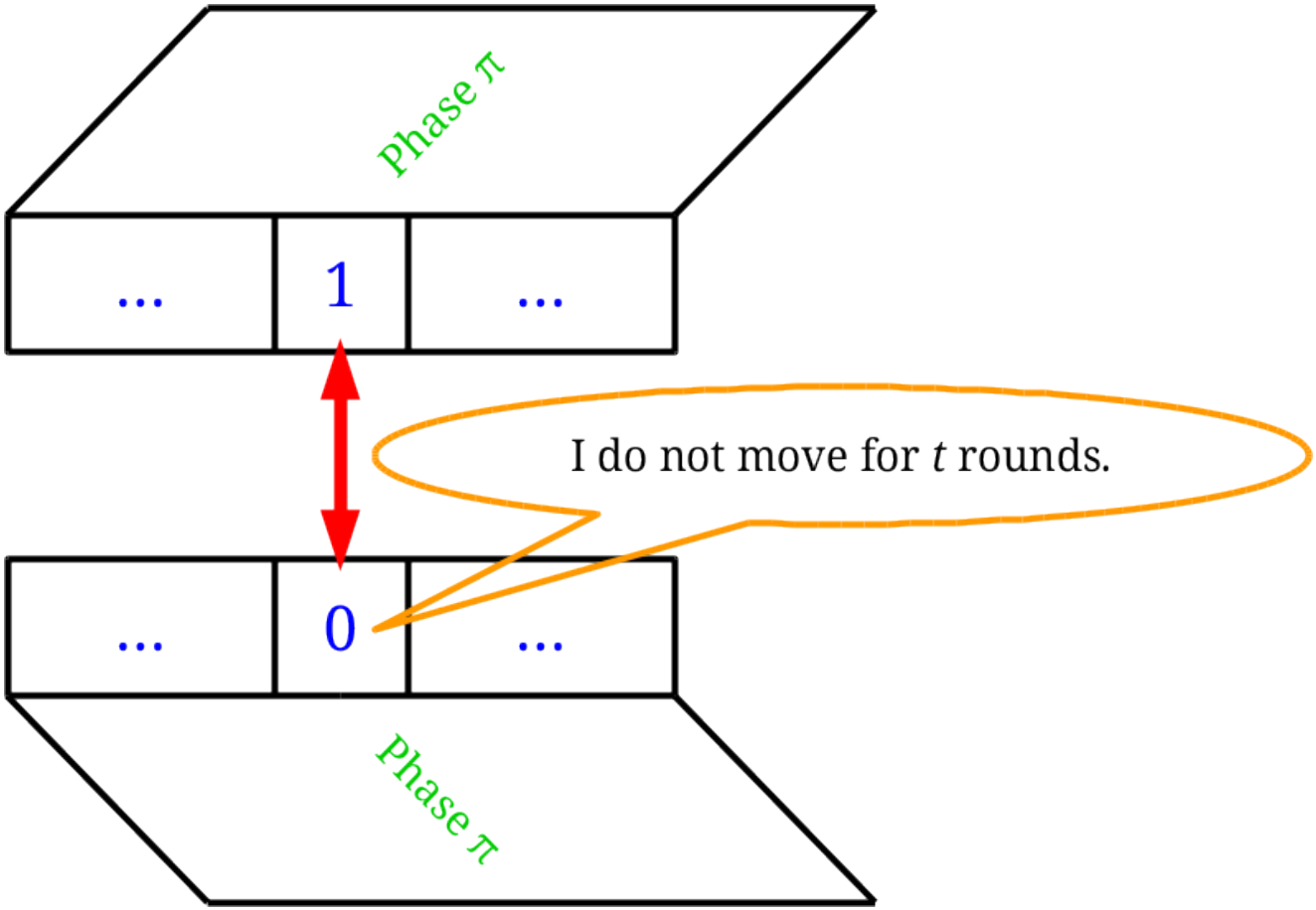
Removing the synchronous setting assumption



Naïve attempt (1)



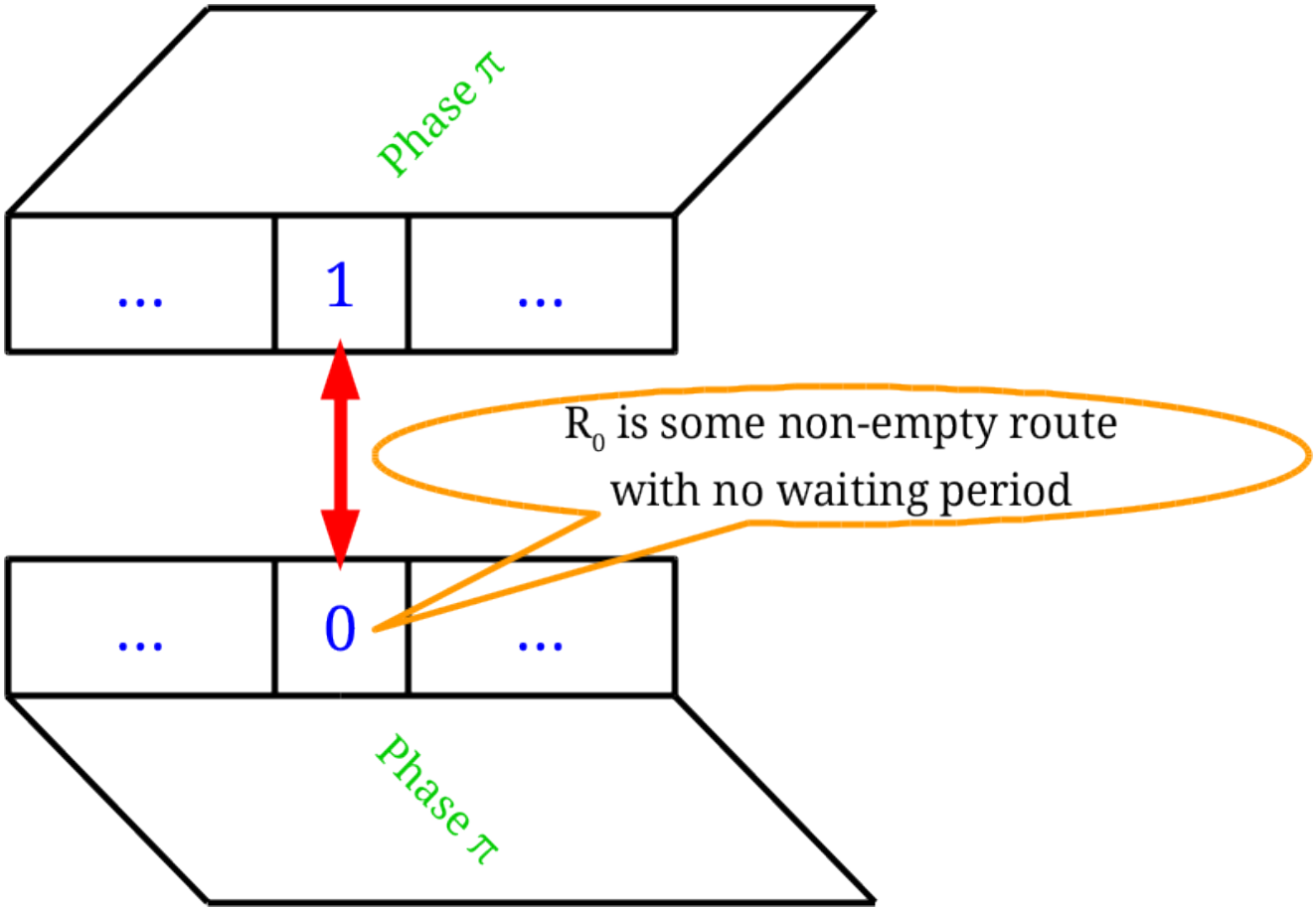
Naïve attempt (1)



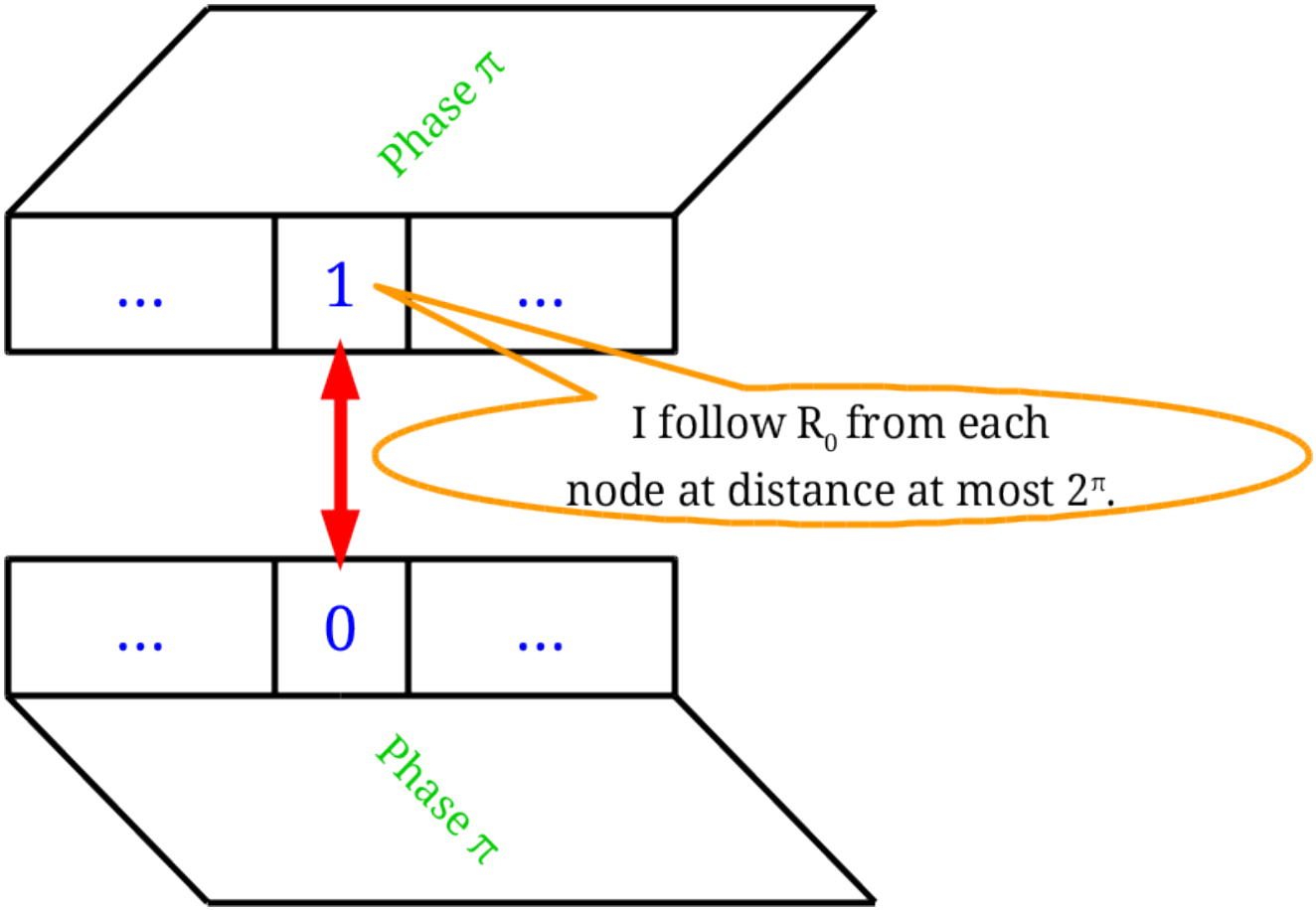
Naïve attempt (1)

Waiting does not work!

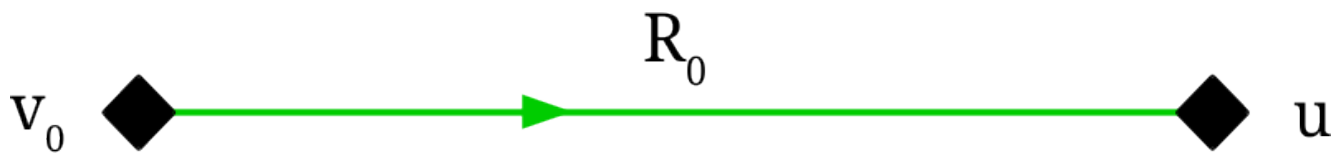
Naïve attempt (1)



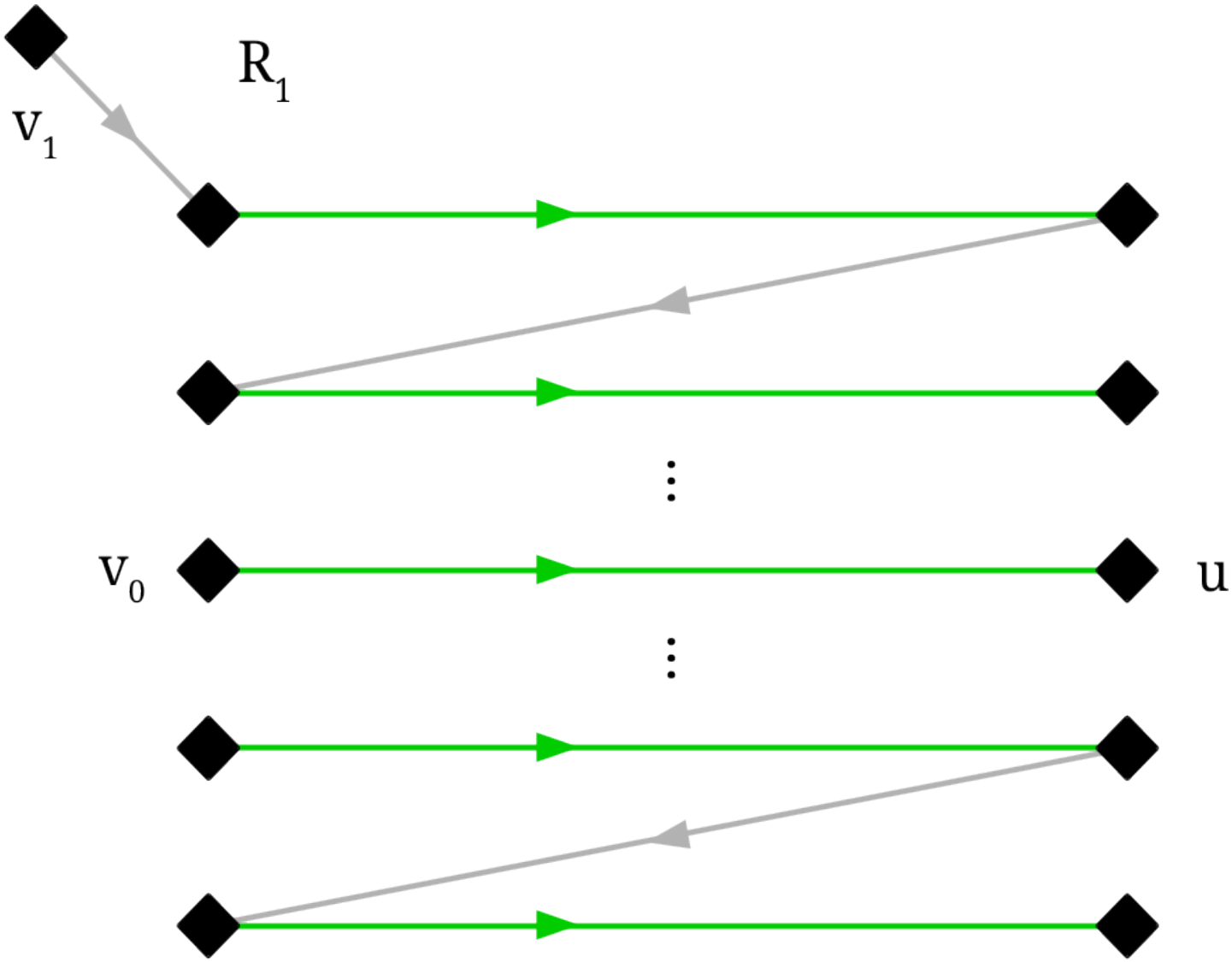
Naïve attempt (1)



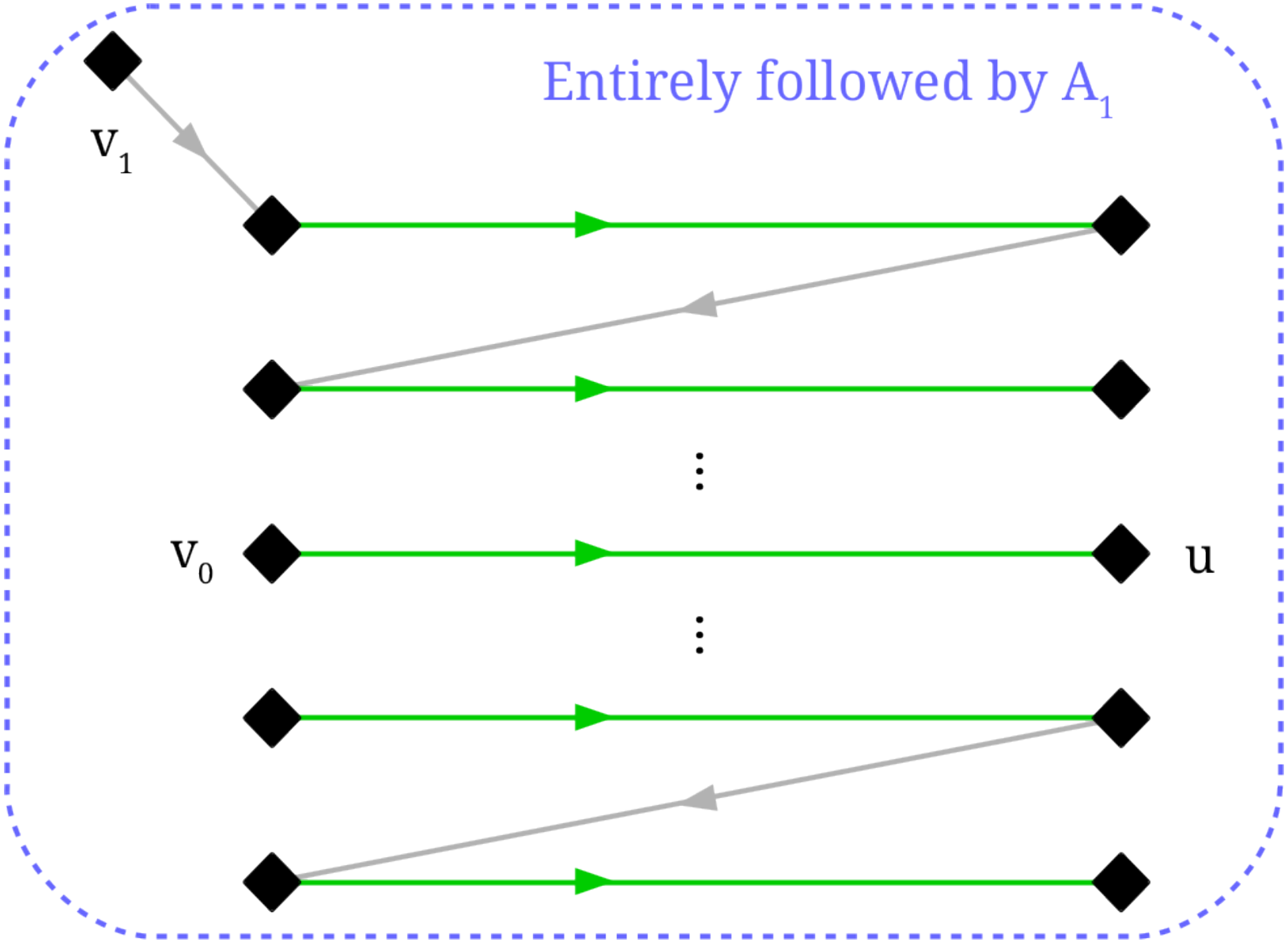
Naïve attempt (1)



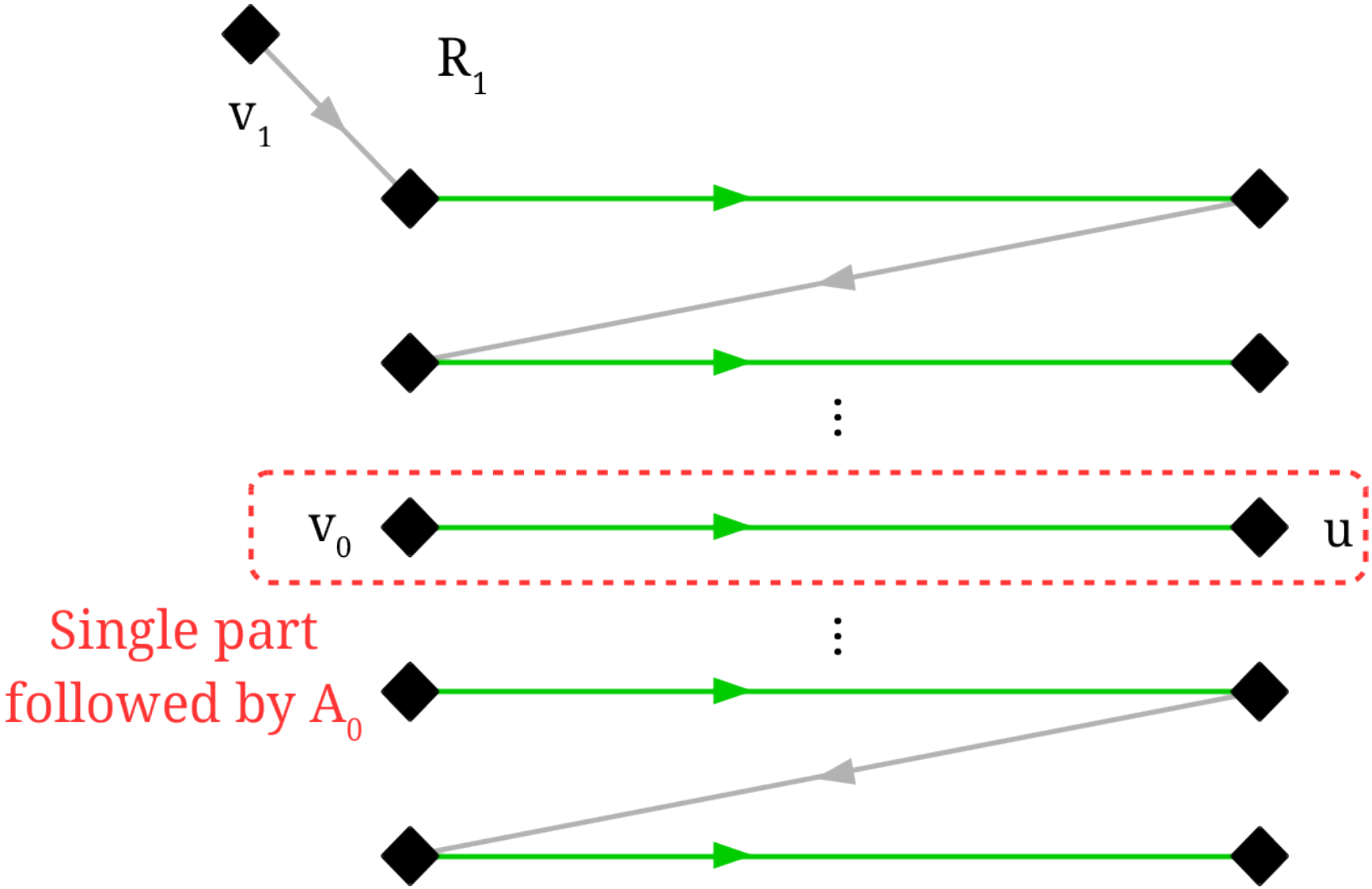
Naïve attempt (1)



Small break (1)



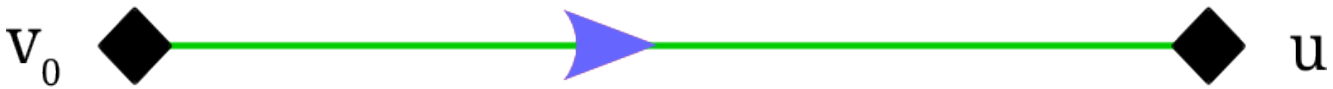
Small break (1)



Naïve attempt (2)

Naïve attempt (2)

Naïve attempt (2)

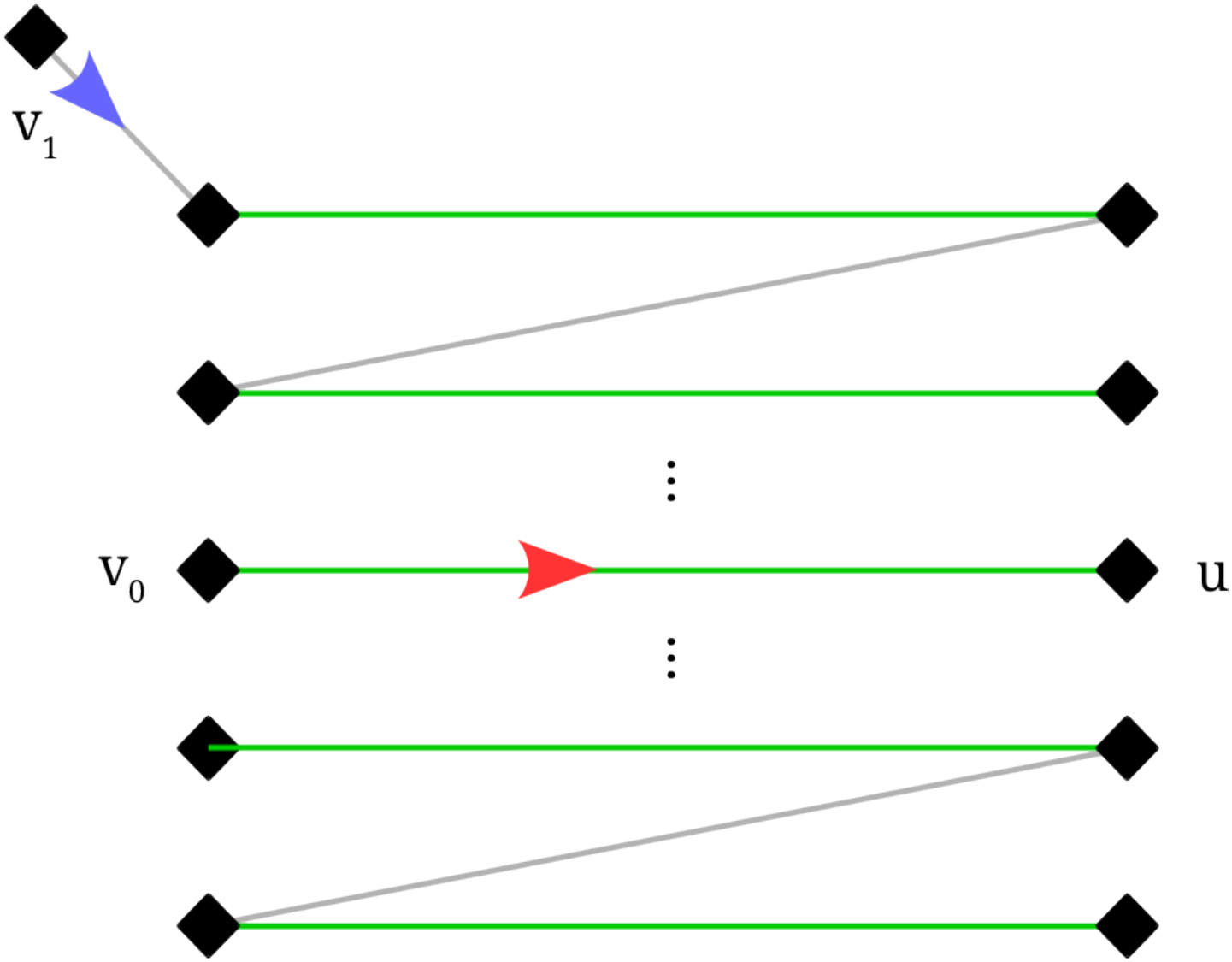


Naïve attempt (2)

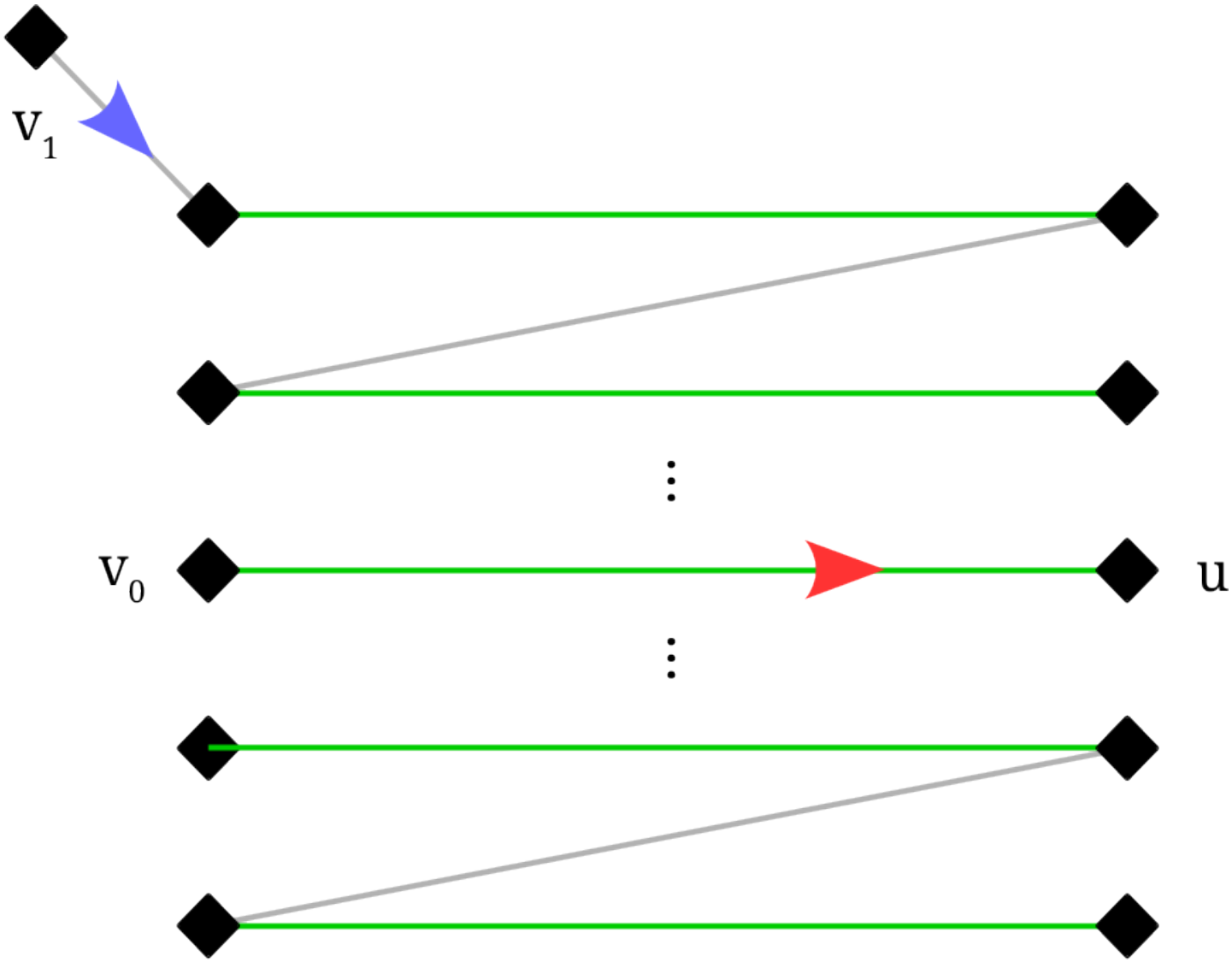
Naïve attempt (2)

Naïve attempt (2)

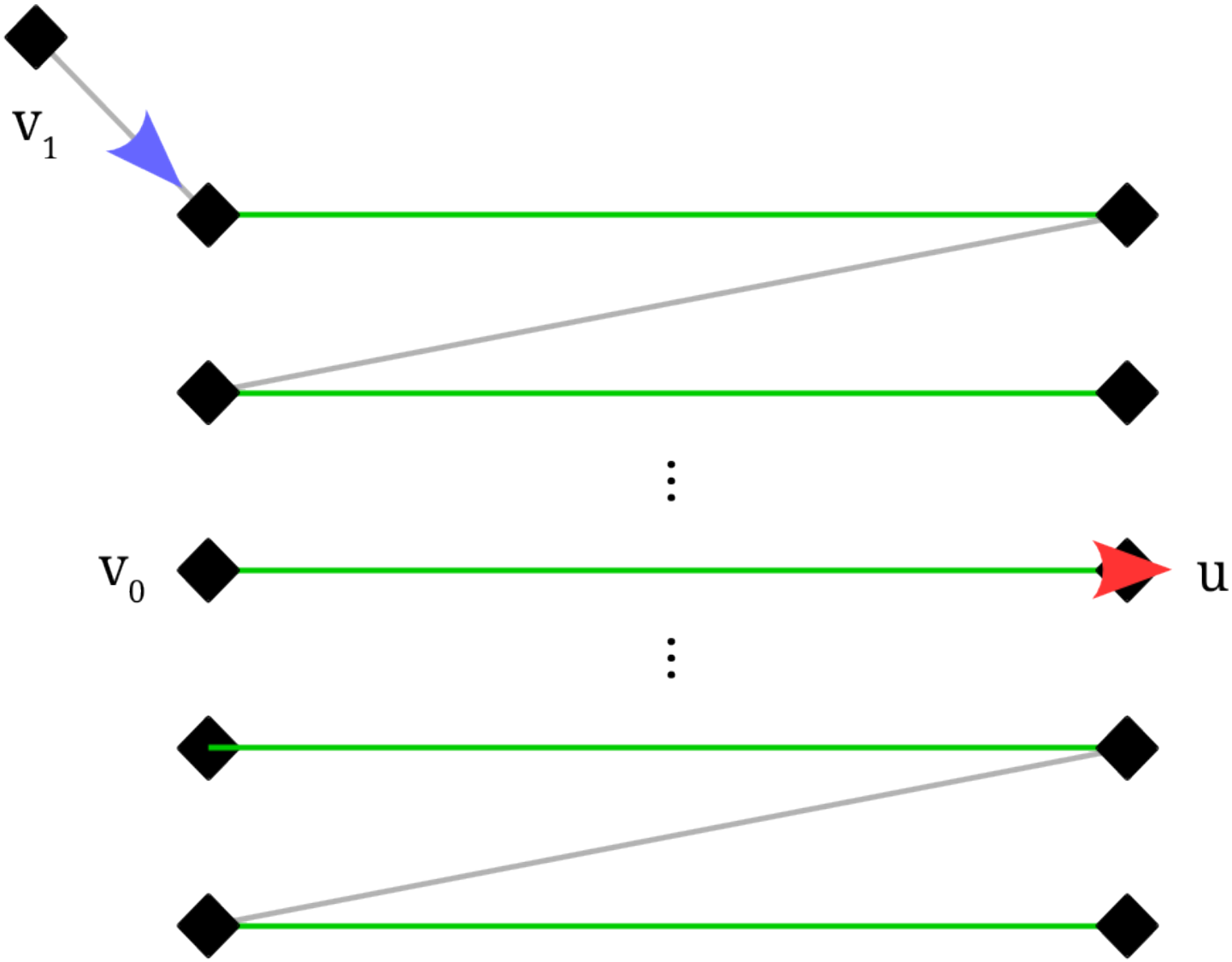
Naïve attempt (2)



Naïve attempt (2)



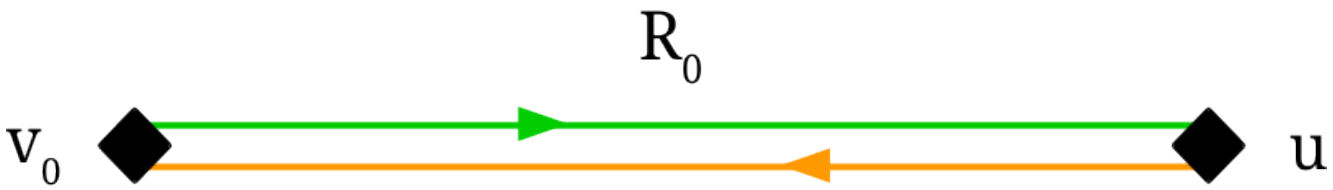
Naïve attempt (2)



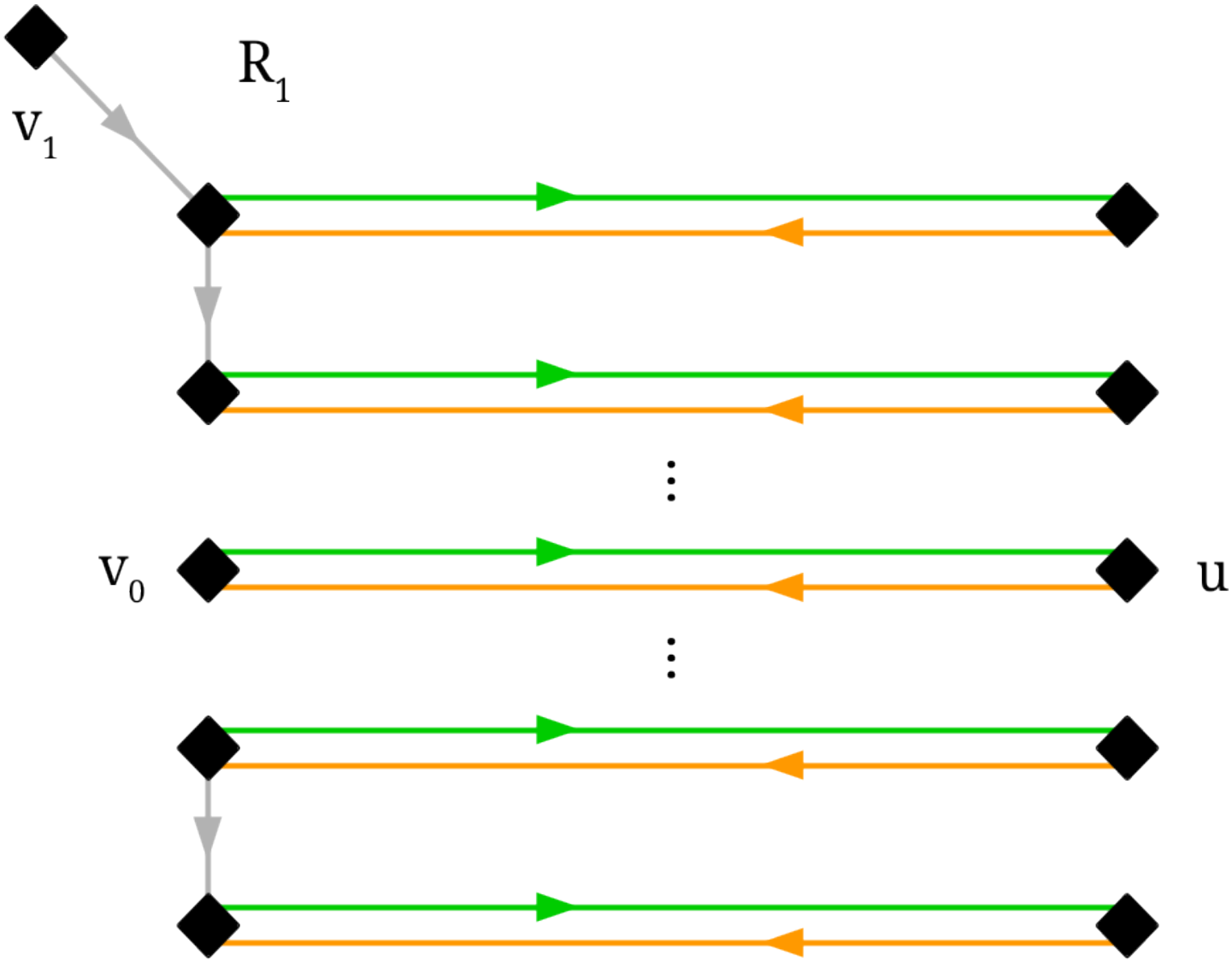
First improvement based on tunnels

Let us use reverse routes and tunnels.

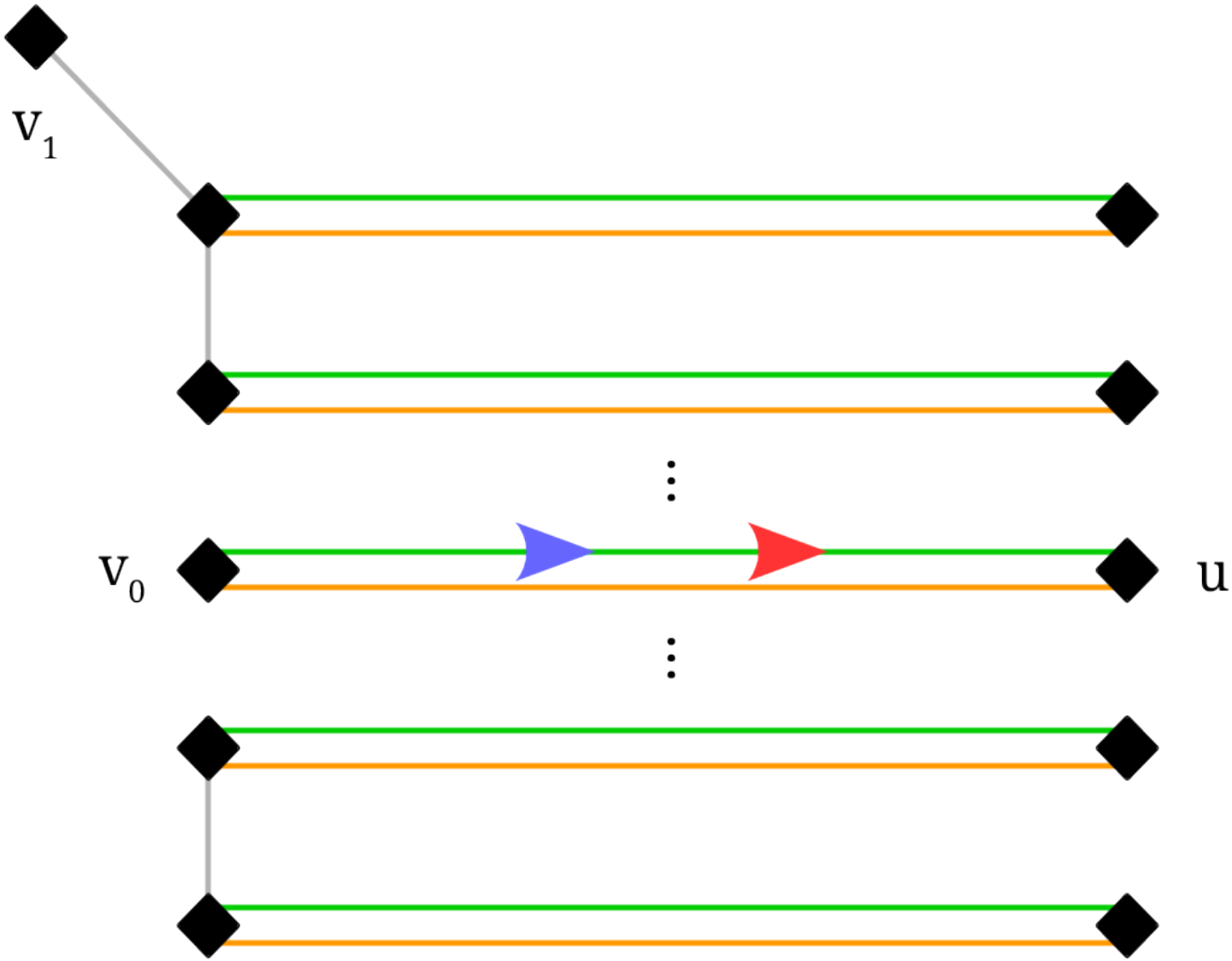
First improvement based on tunnels



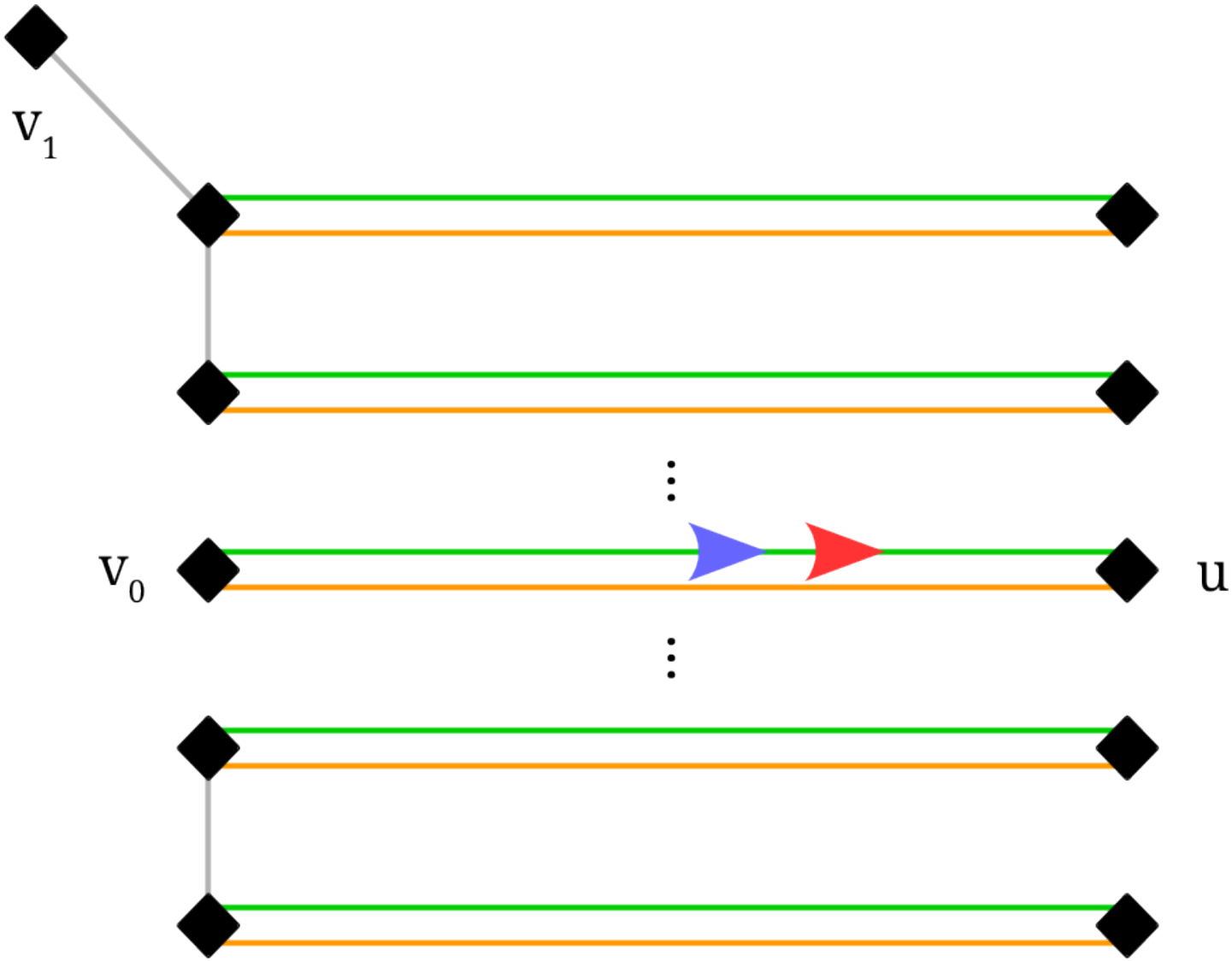
First improvement based on tunnels



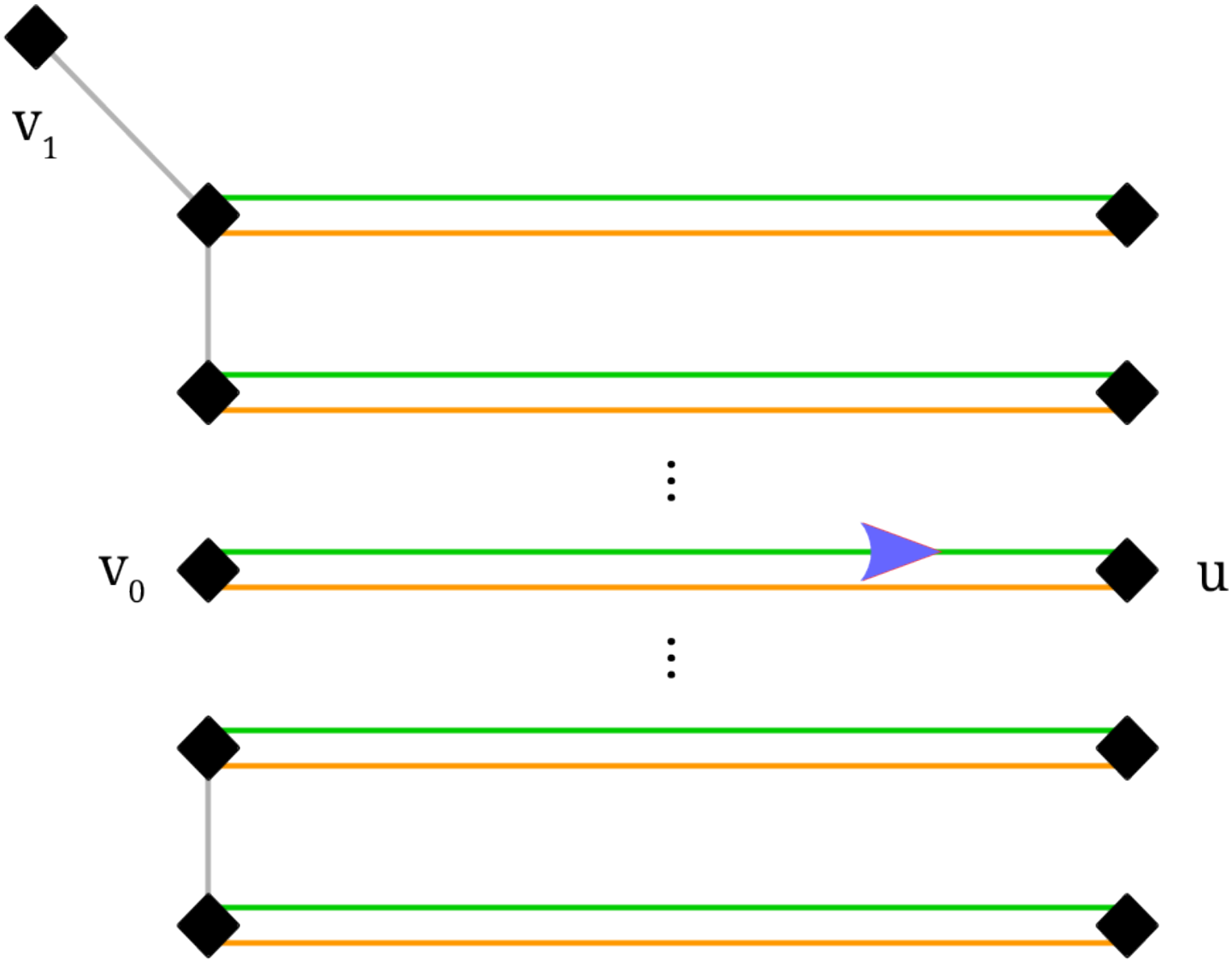
First improvement based on tunnels



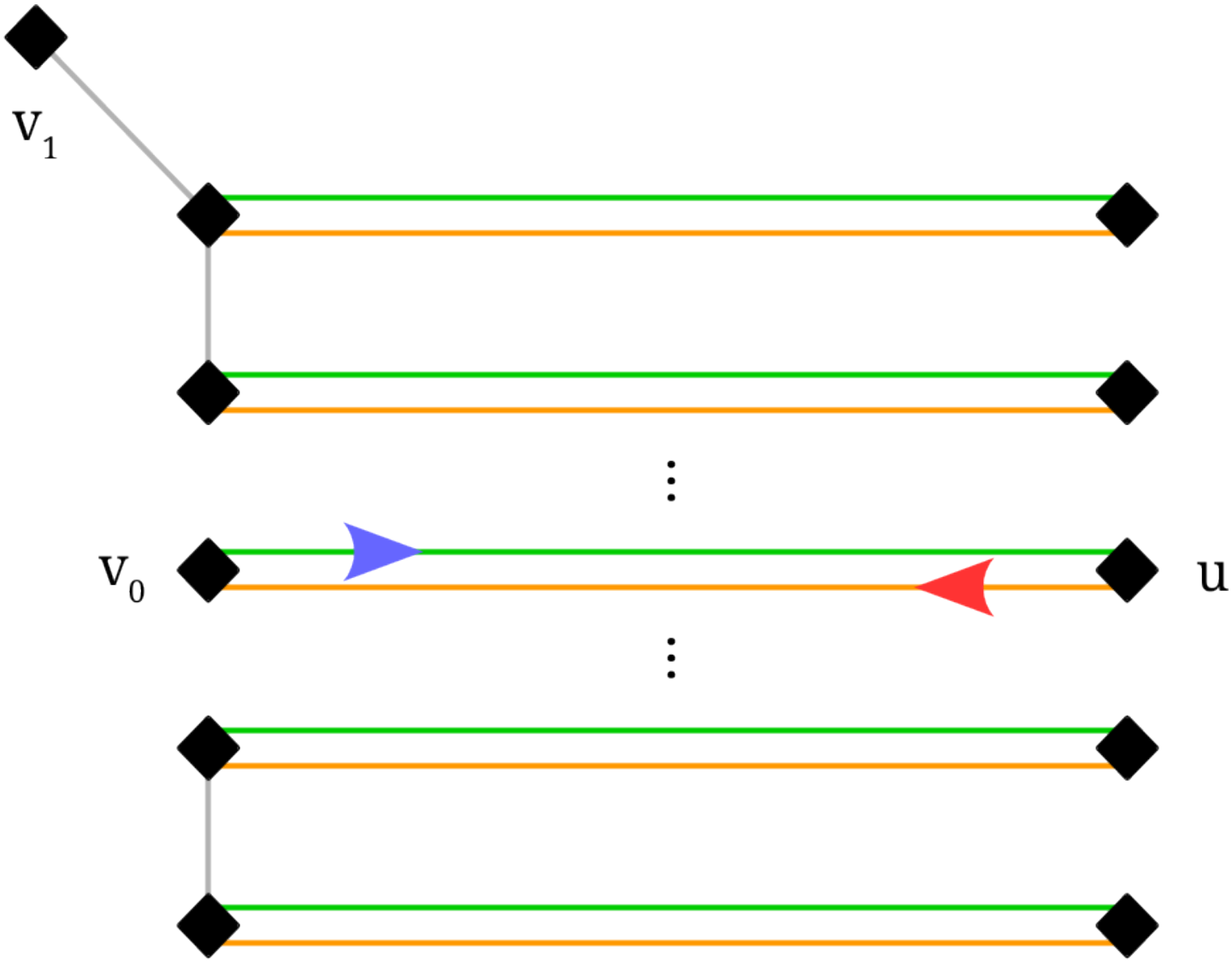
First improvement based on tunnels



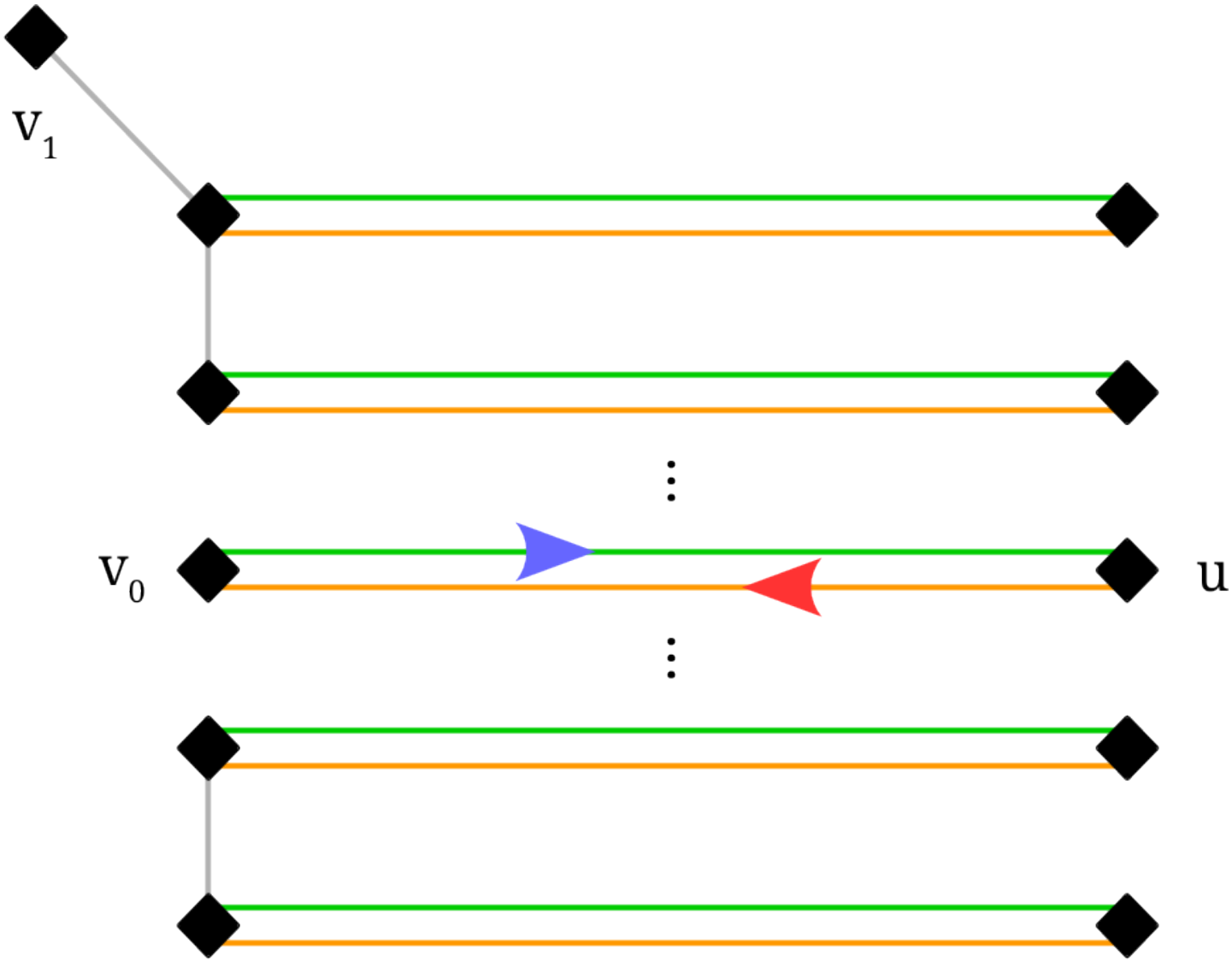
First improvement based on tunnels



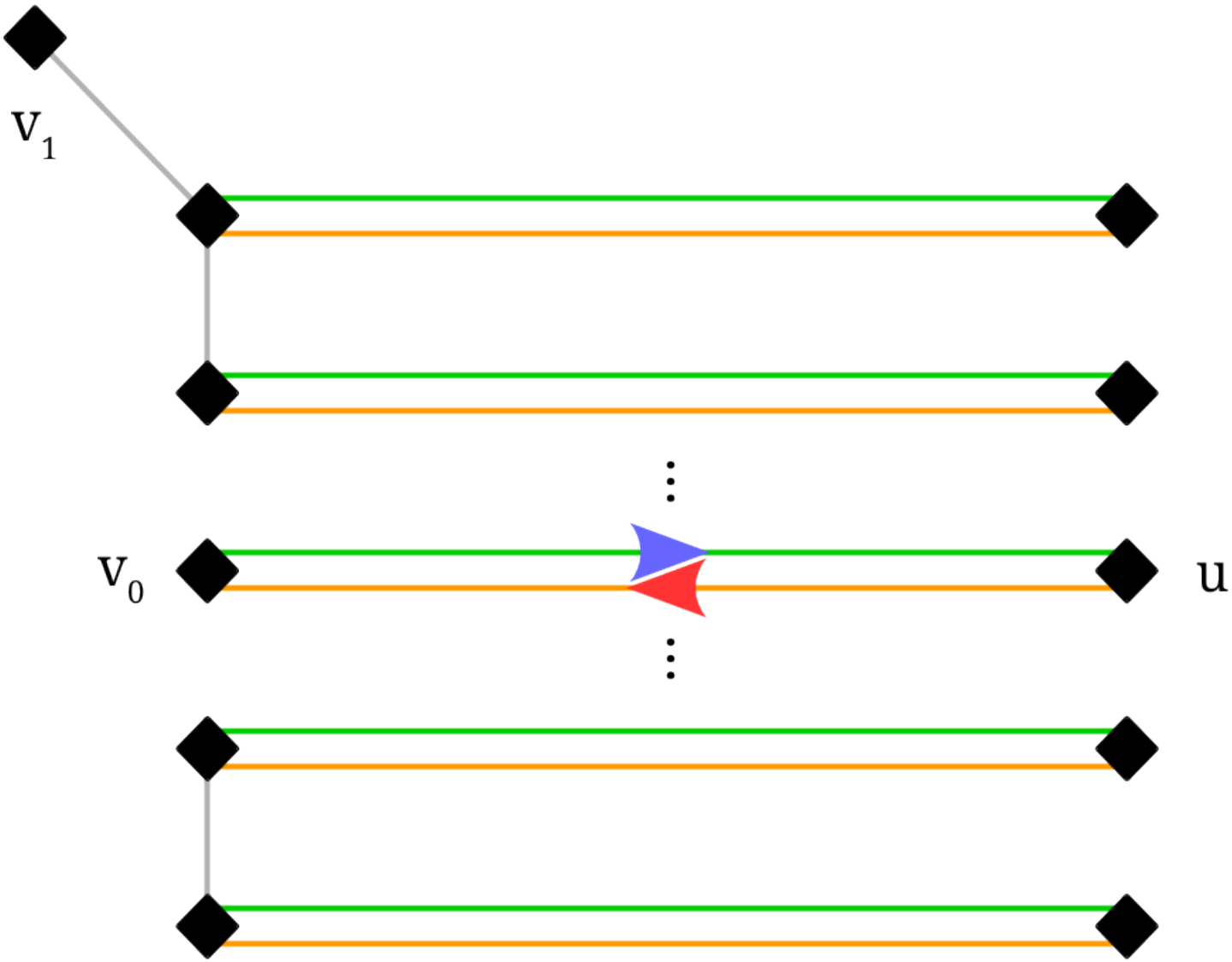
First improvement based on tunnels



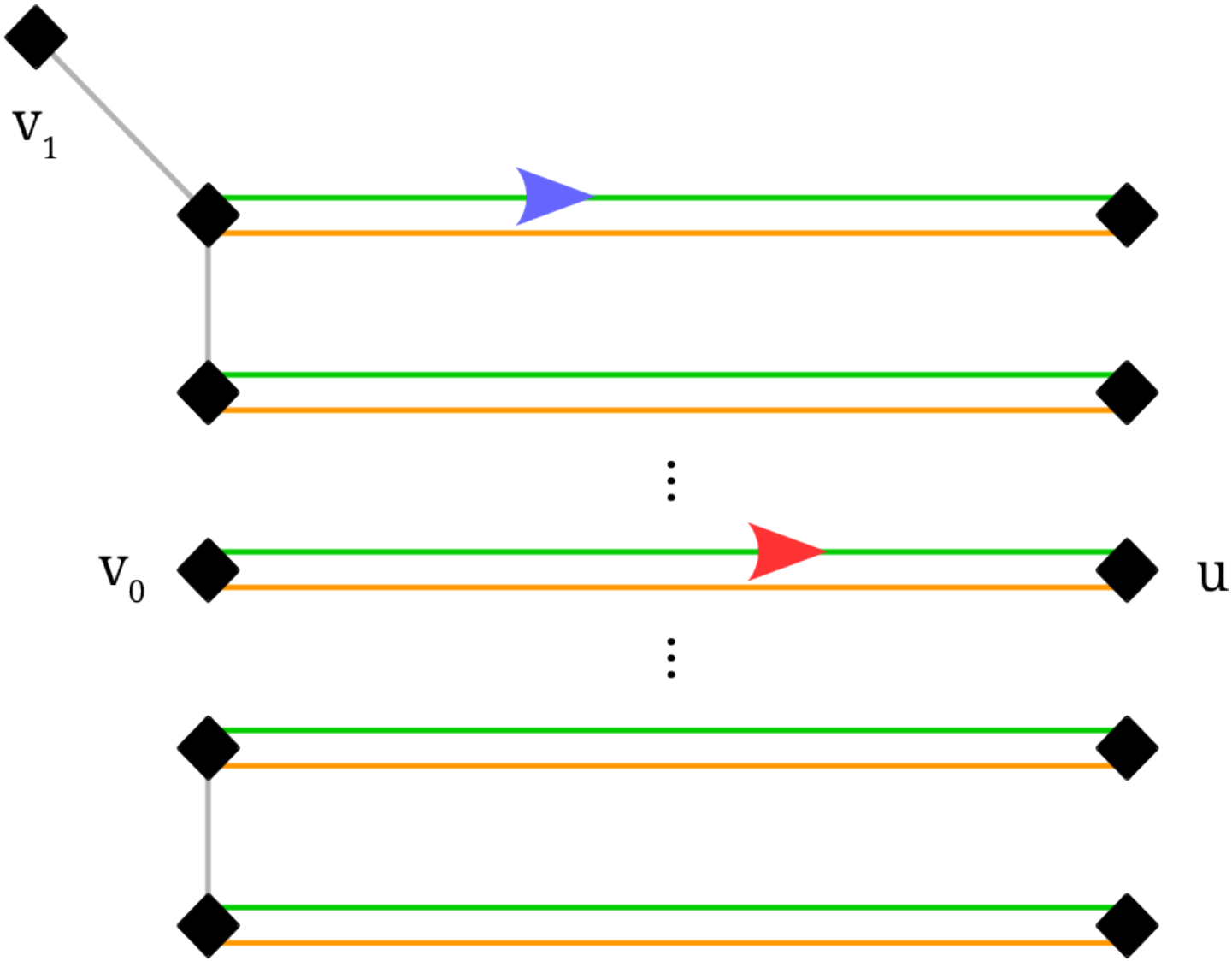
First improvement based on tunnels



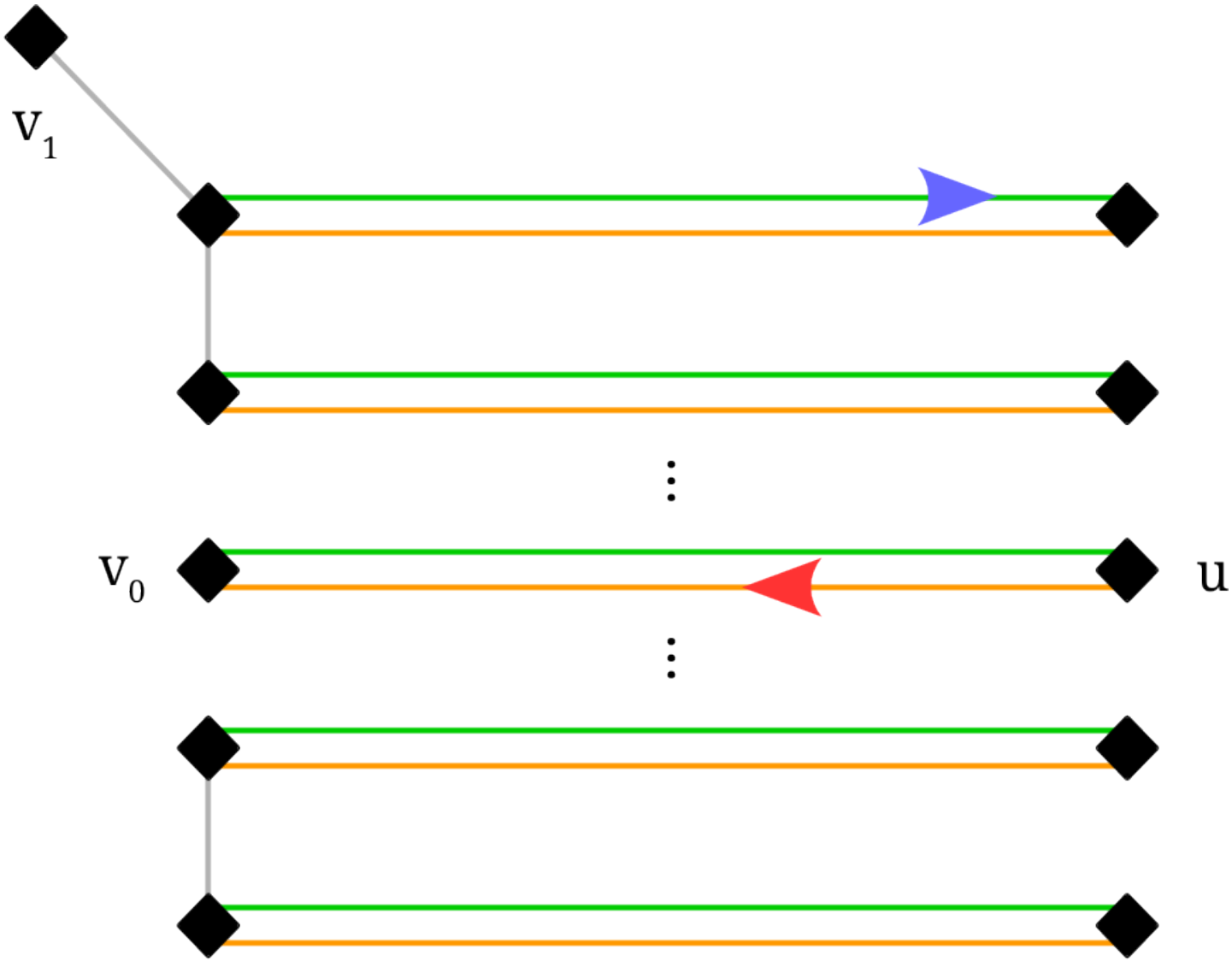
First improvement based on tunnels



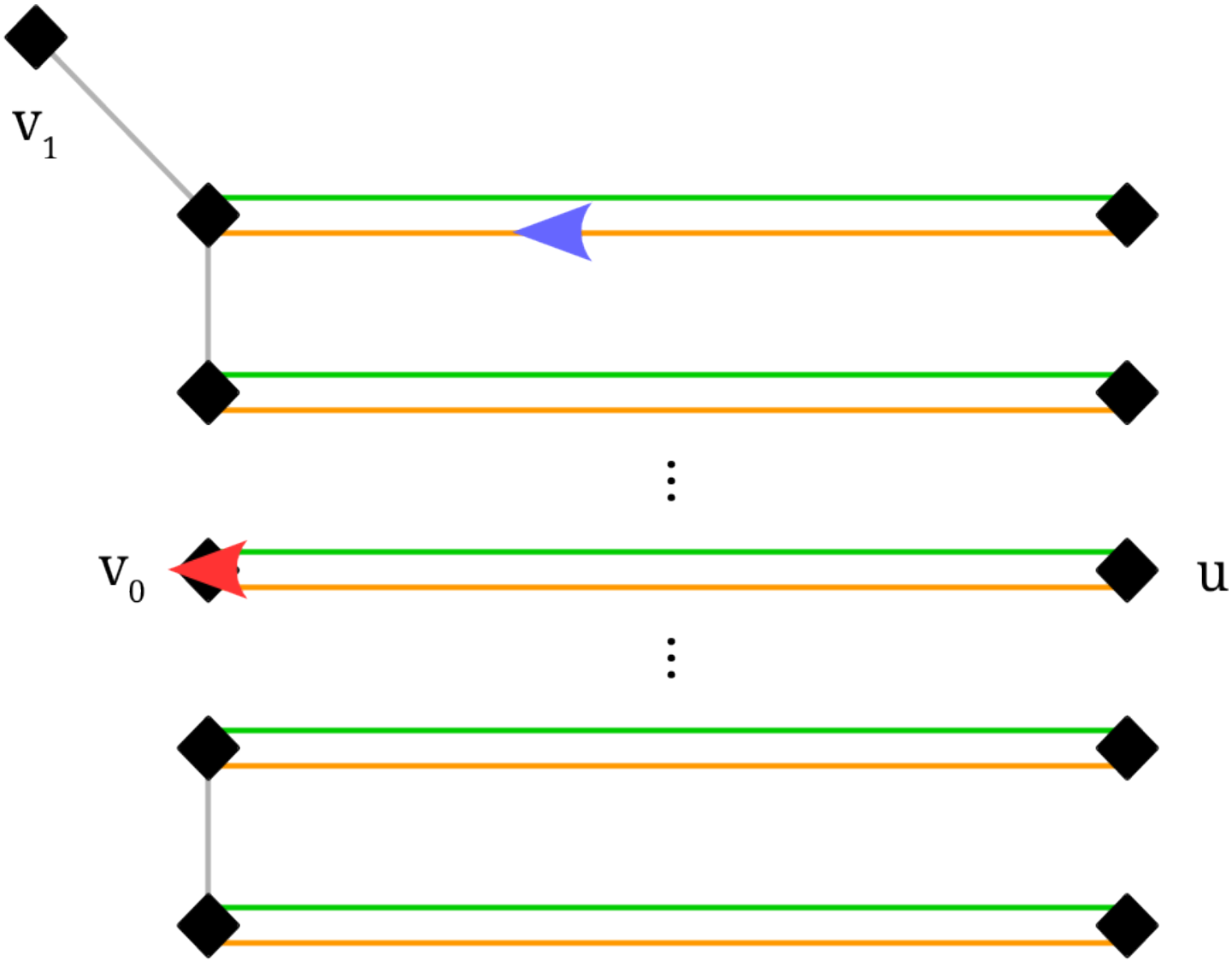
First improvement based on tunnels



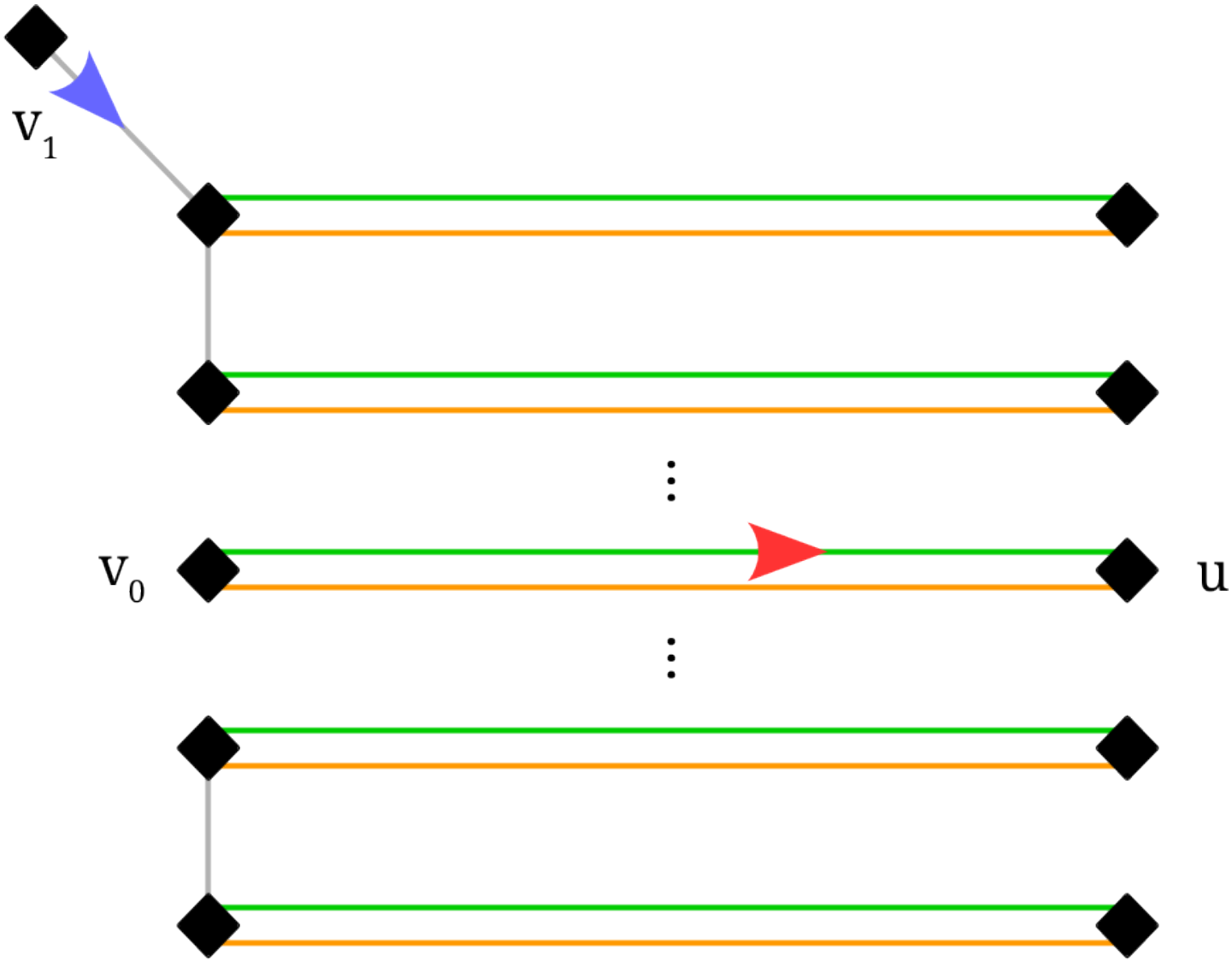
First improvement based on tunnels



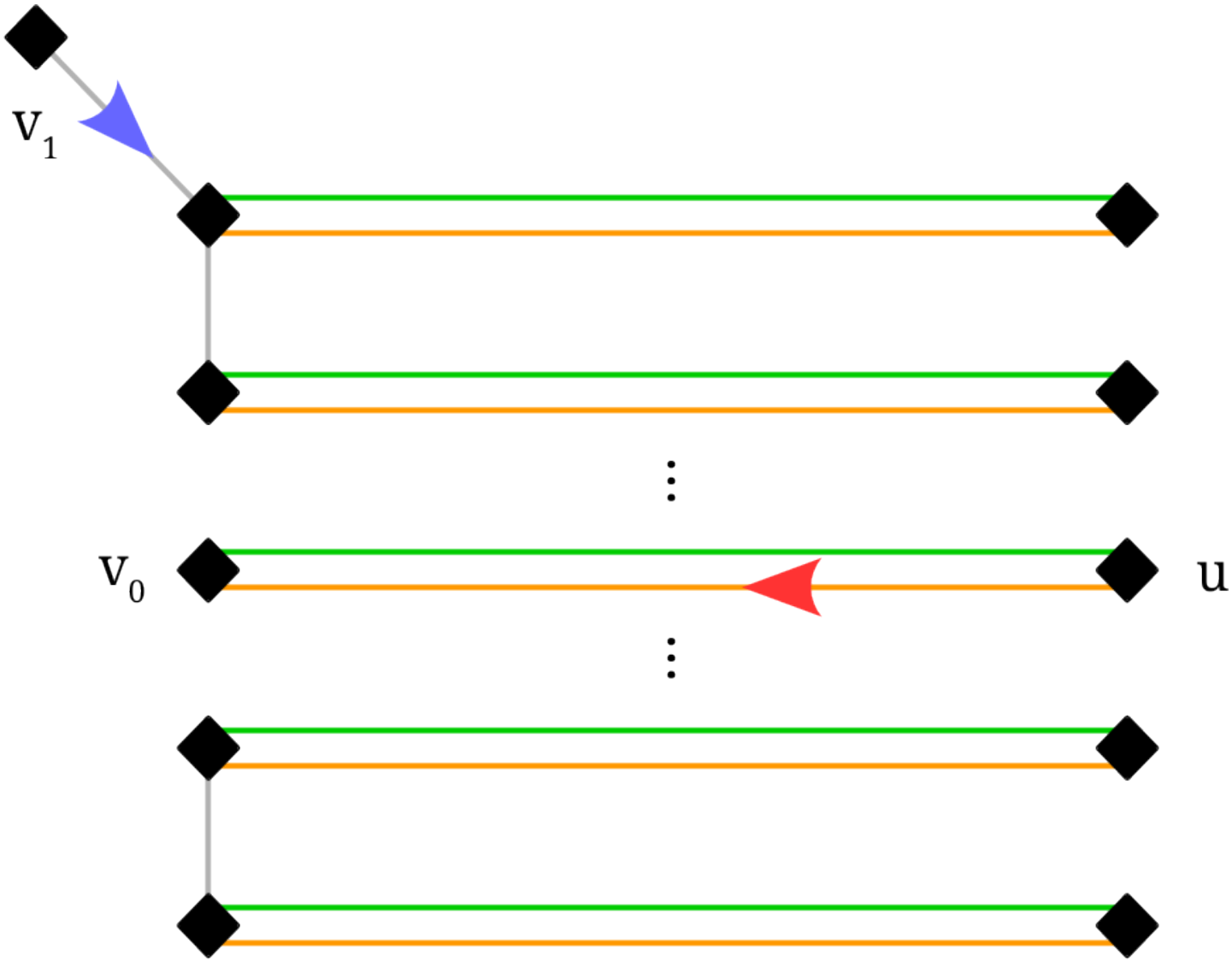
First improvement based on tunnels



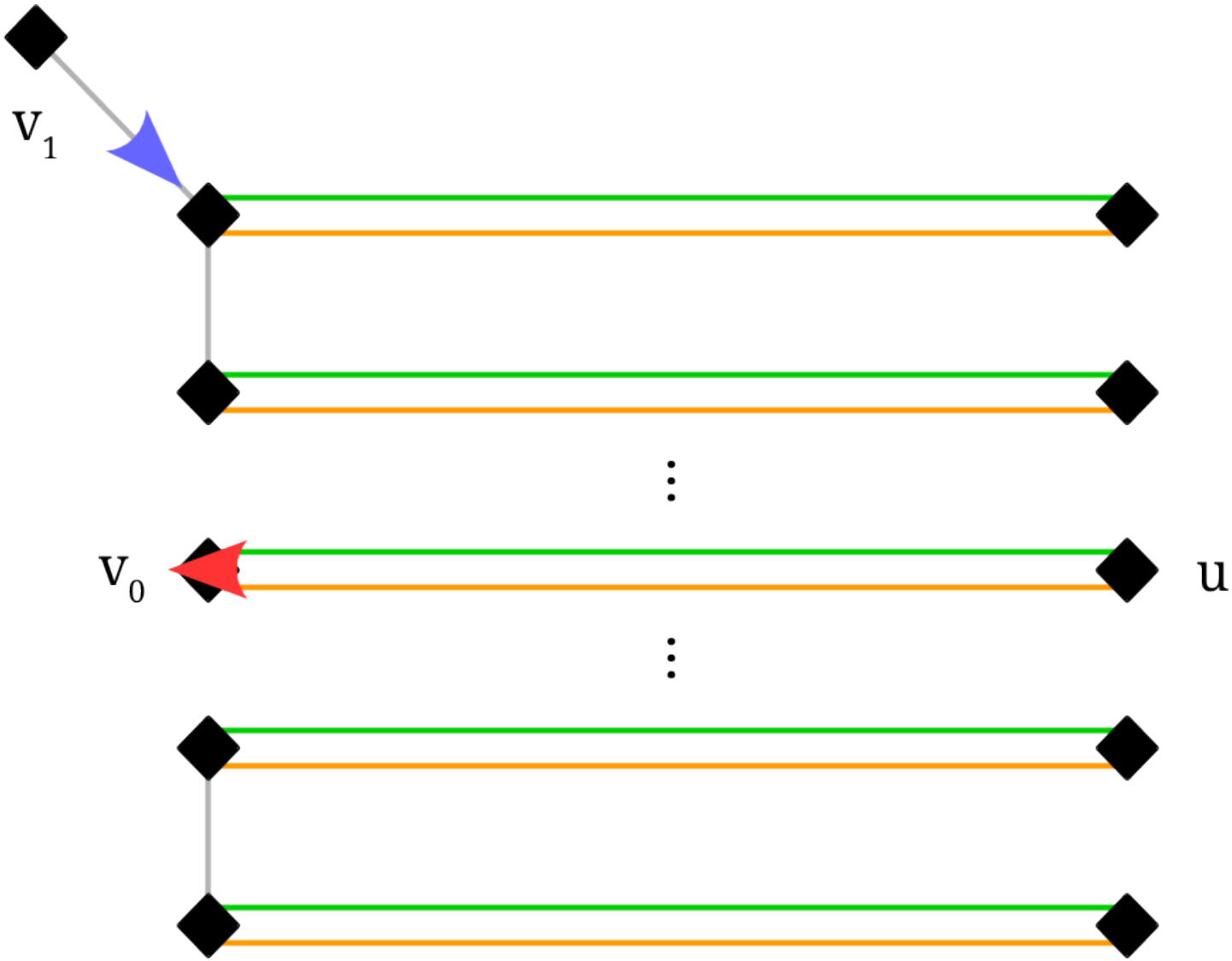
First improvement based on tunnels



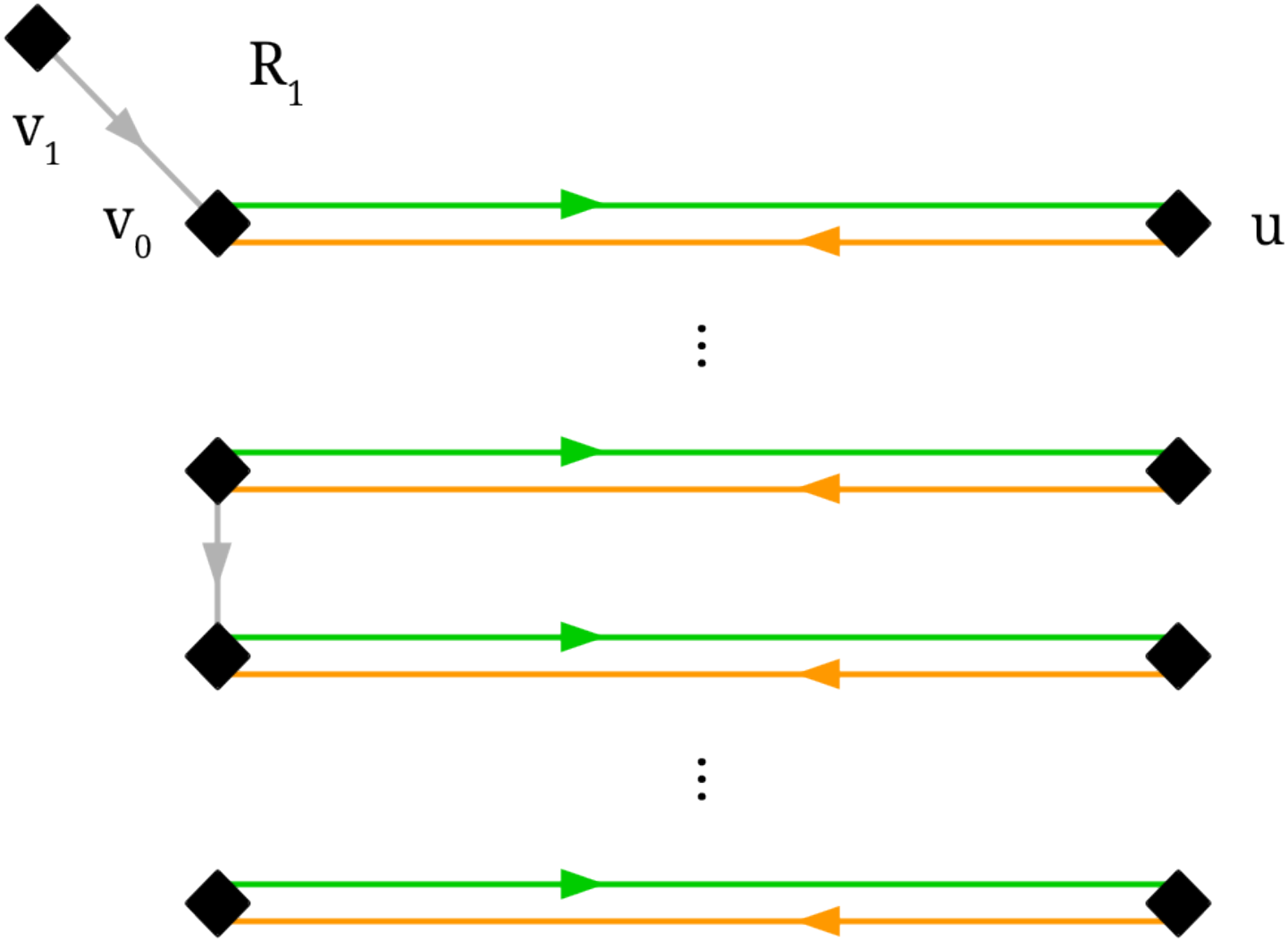
First improvement based on tunnels



First improvement based on tunnels



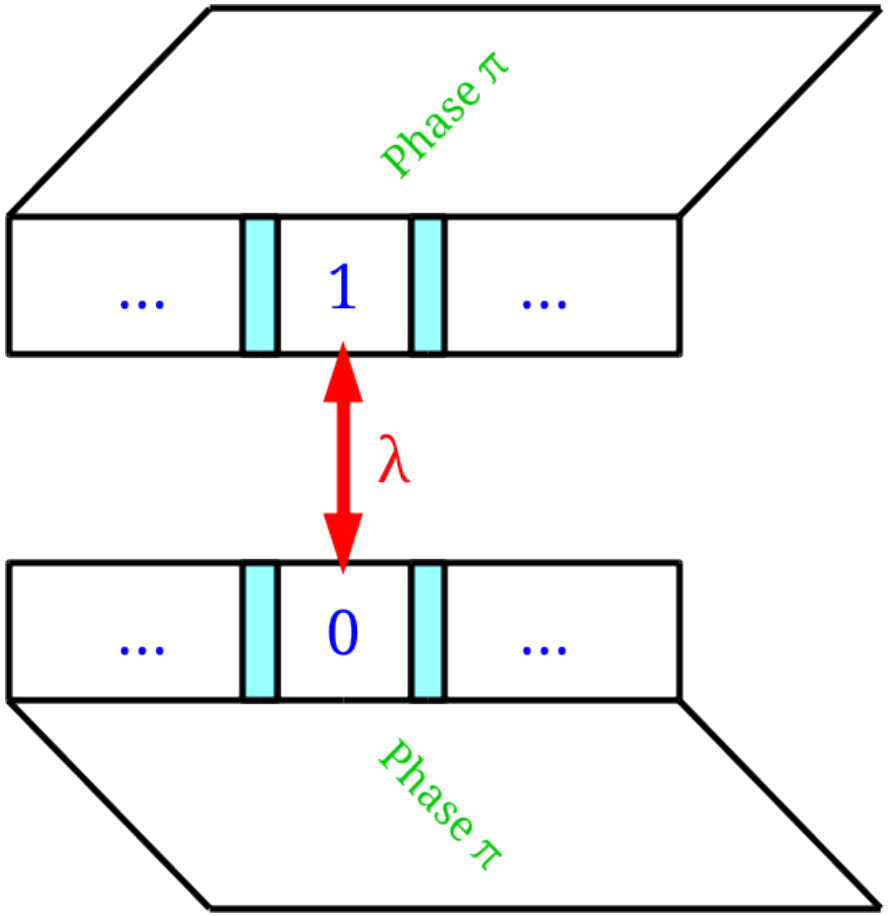
Second improvement: shorten the beginning of R_1



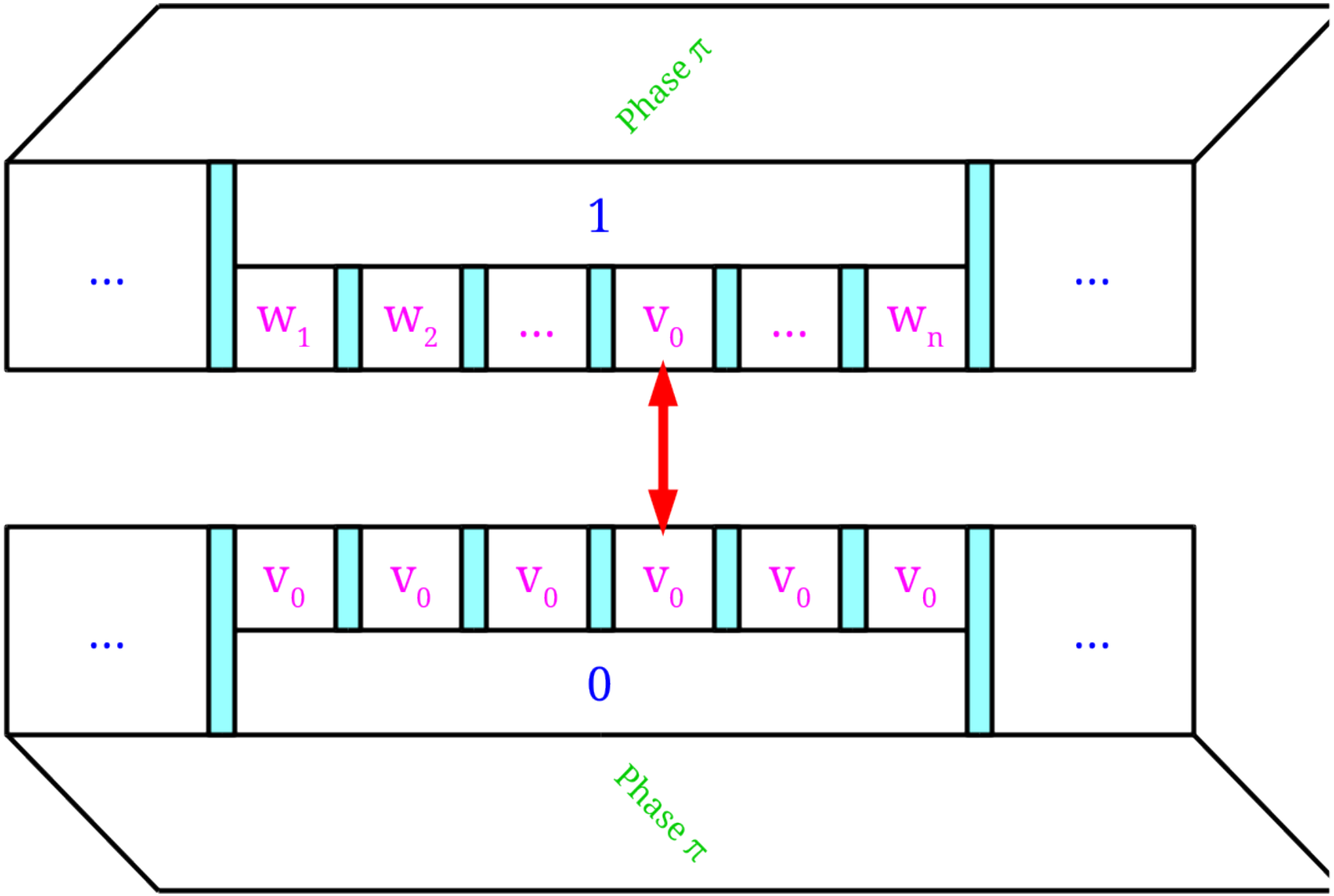
Second improvement: shorten the beginning of R_1

Let us make v_0 the first node from which A_1 follows R_0 during its R_1.
Once again, we use tests separated by the first synchronization mechanism.

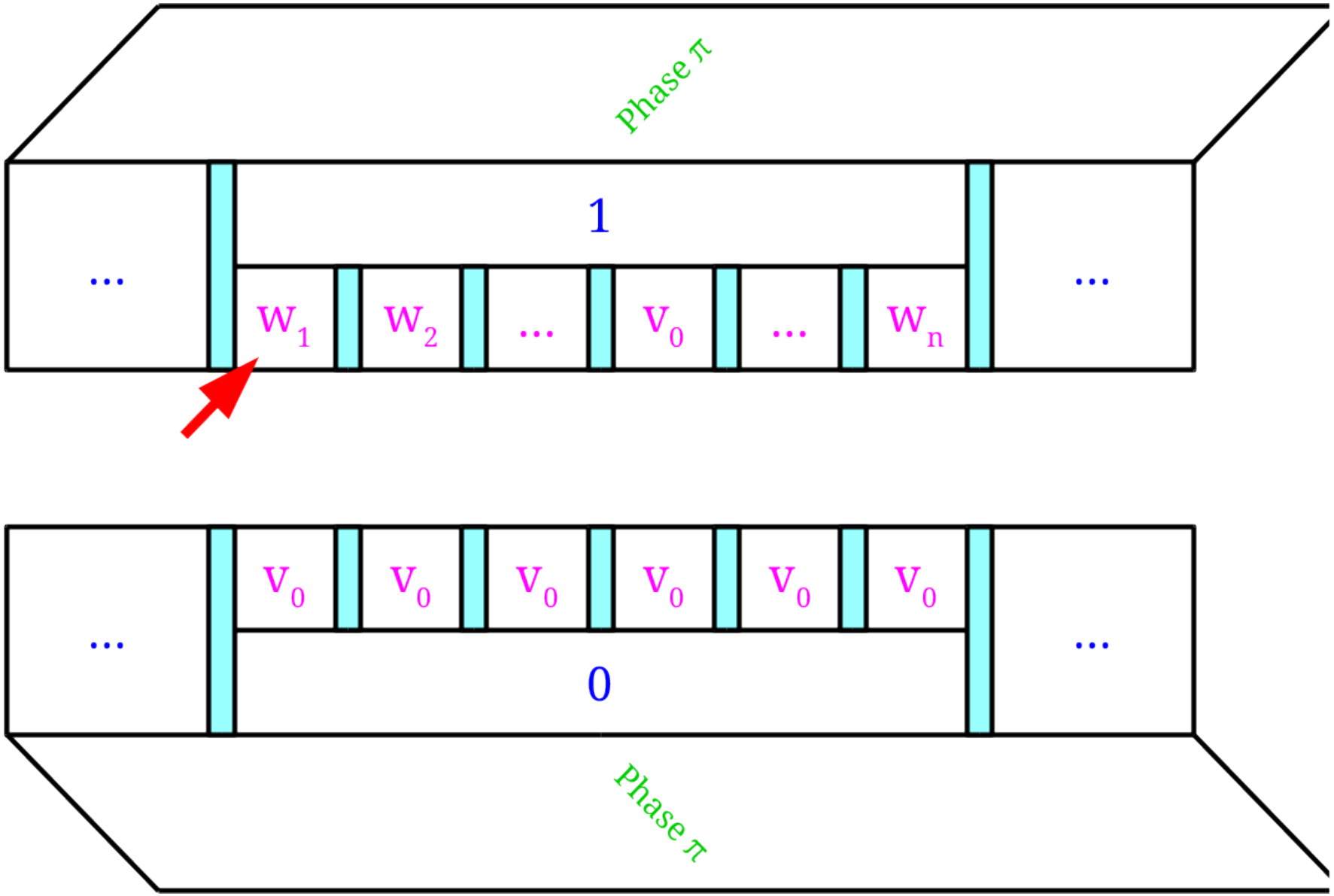
Second improvement: shorten the beginning of R_1



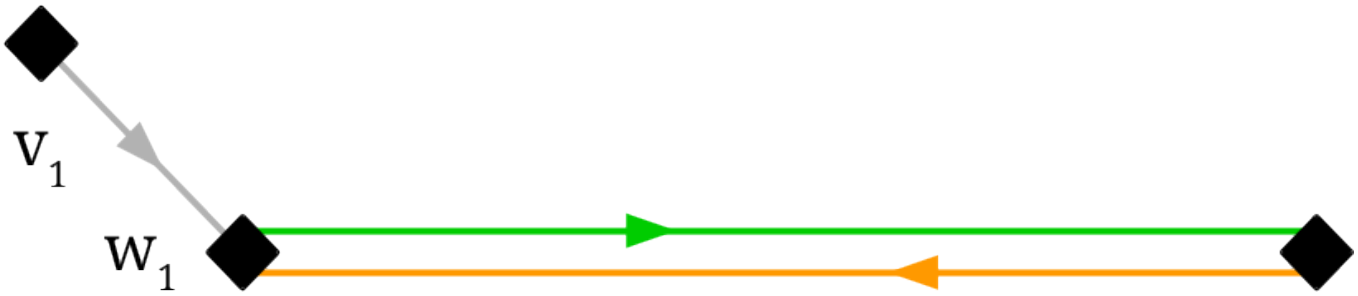
Second improvement: shorten the beginning of R_1



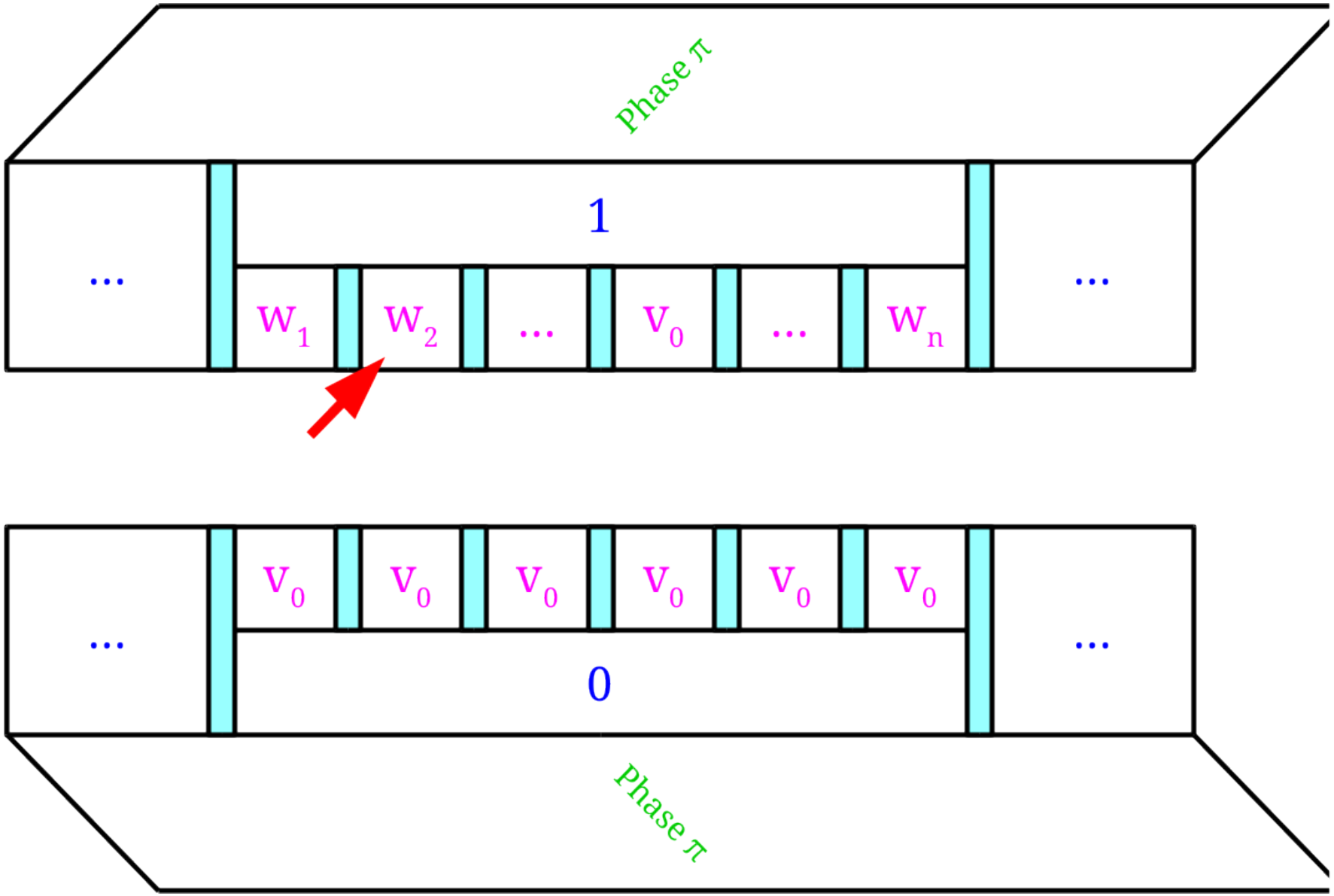
Second improvement: shorten the beginning of R_1



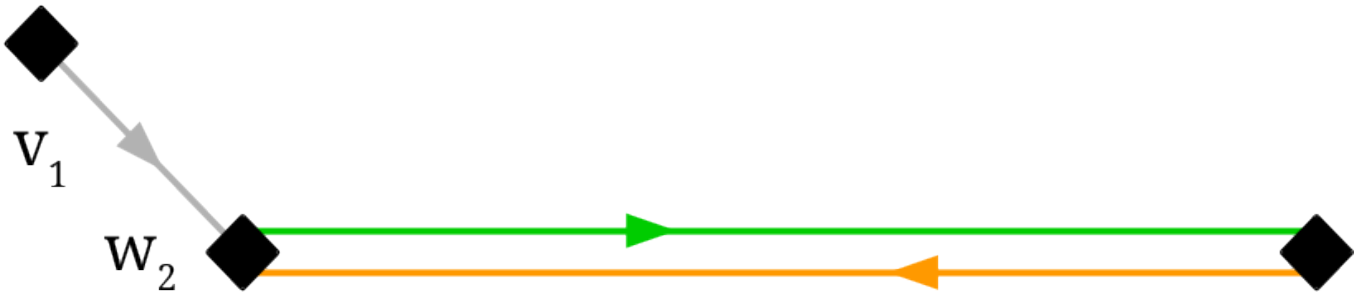
Second improvement: shorten the beginning of R_1



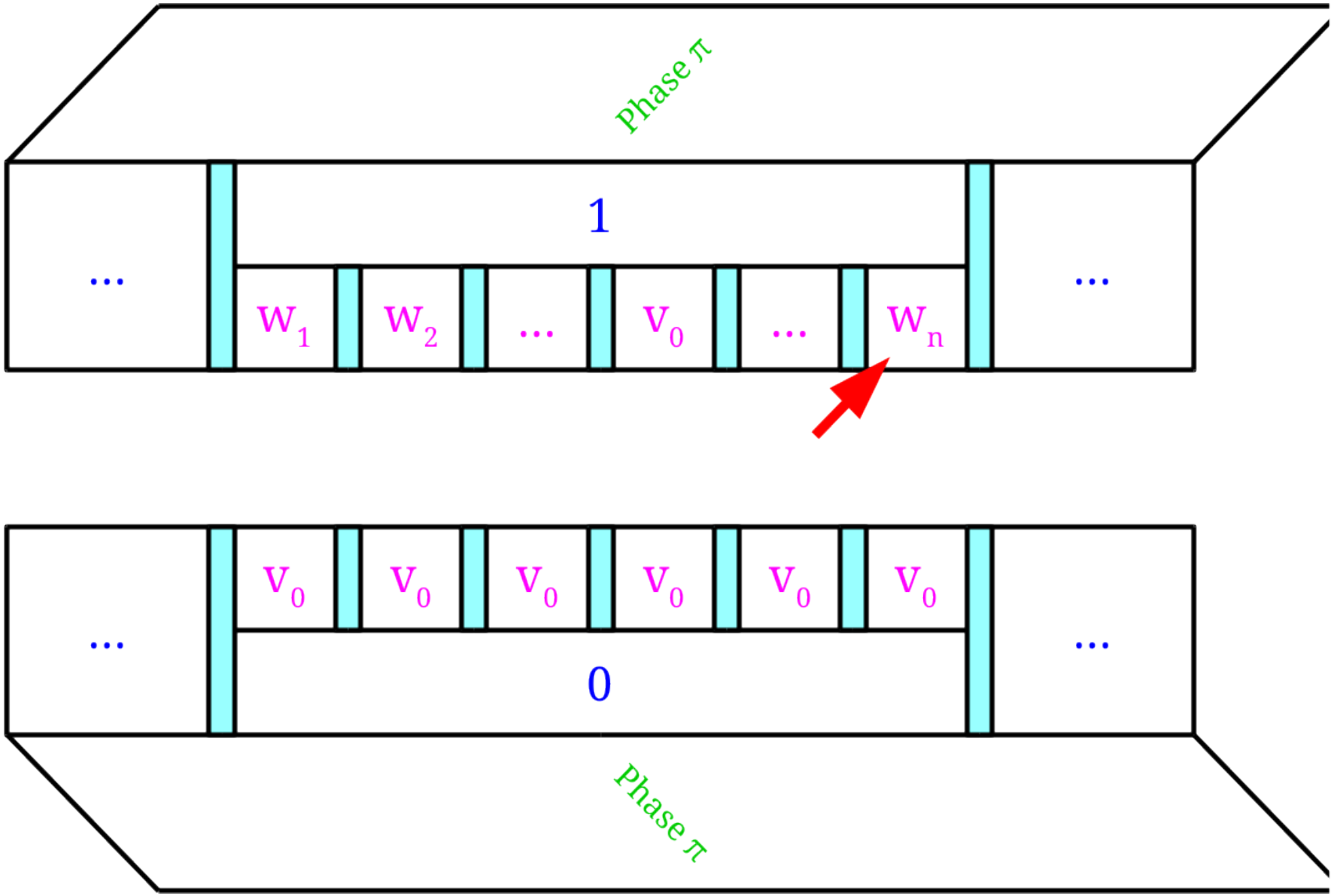
Second improvement: shorten the beginning of R_1



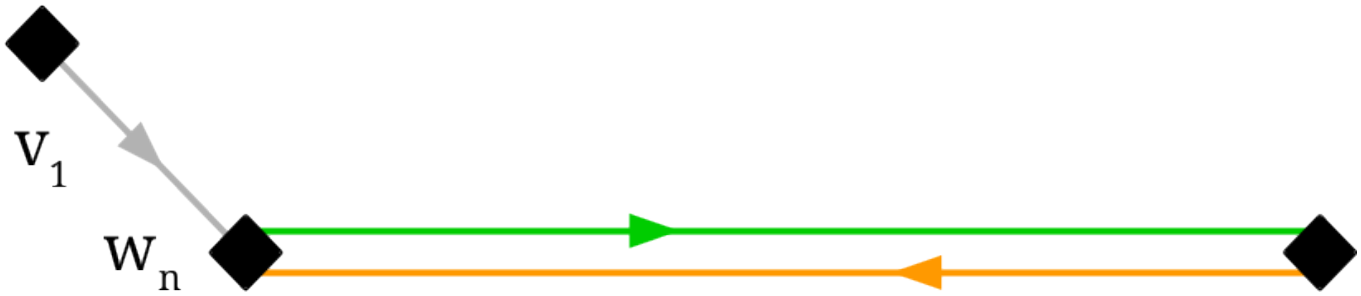
Second improvement: shorten the beginning of R_1



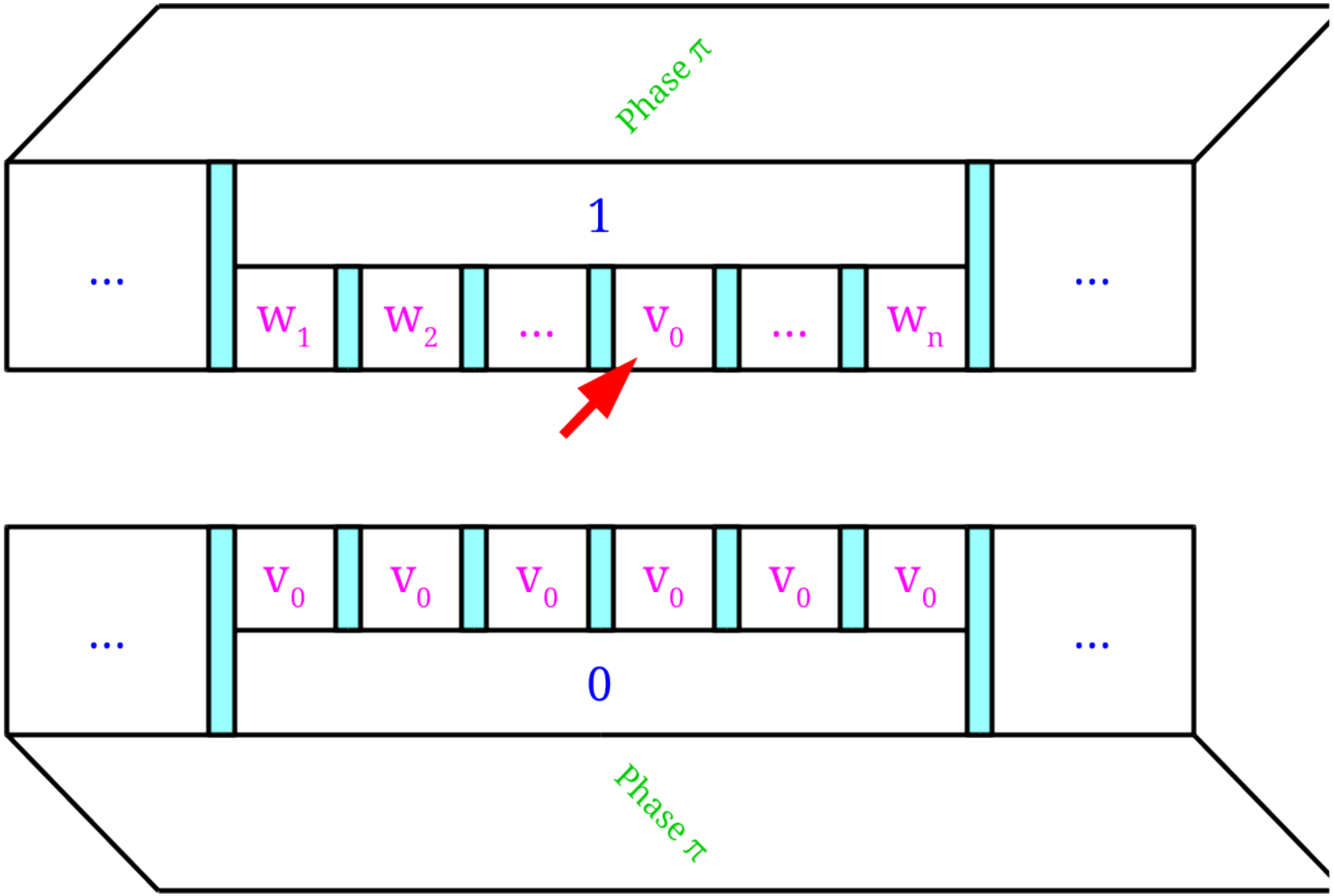
Second improvement: shorten the beginning of R_1



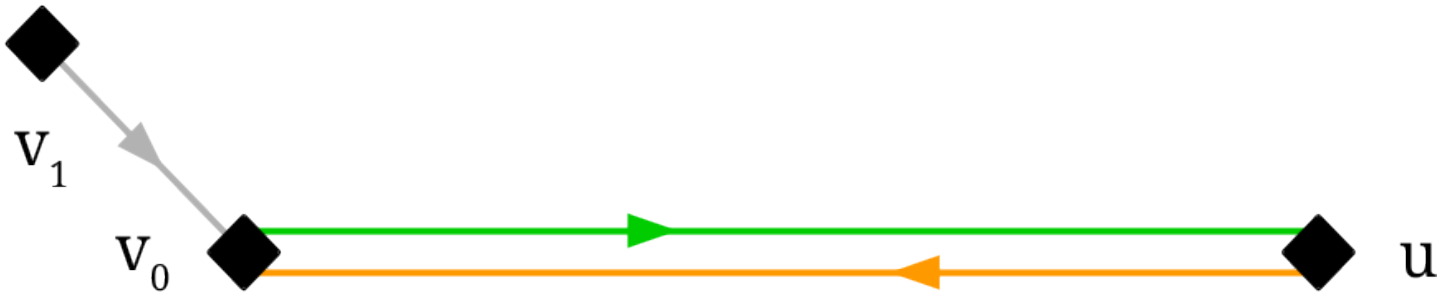
Second improvement: shorten the beginning of R_1



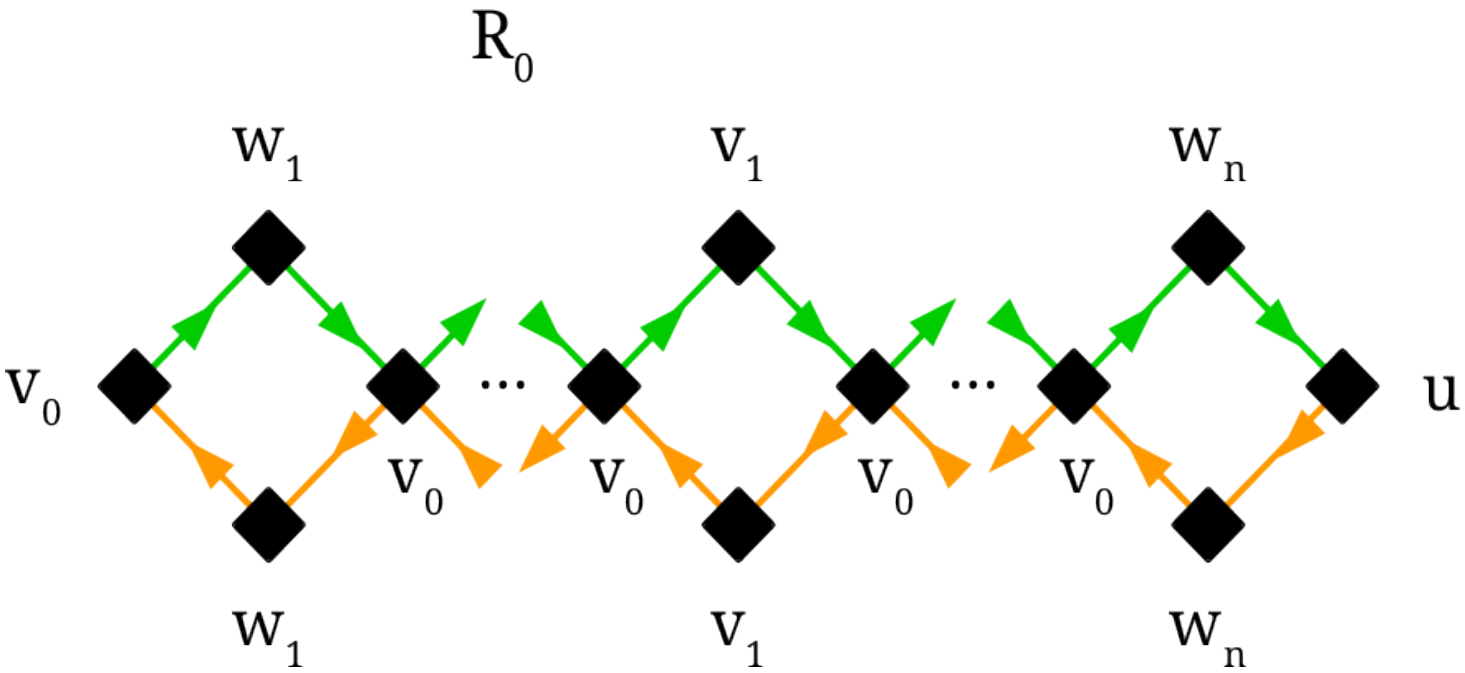
Second improvement: shorten the beginning of R_1



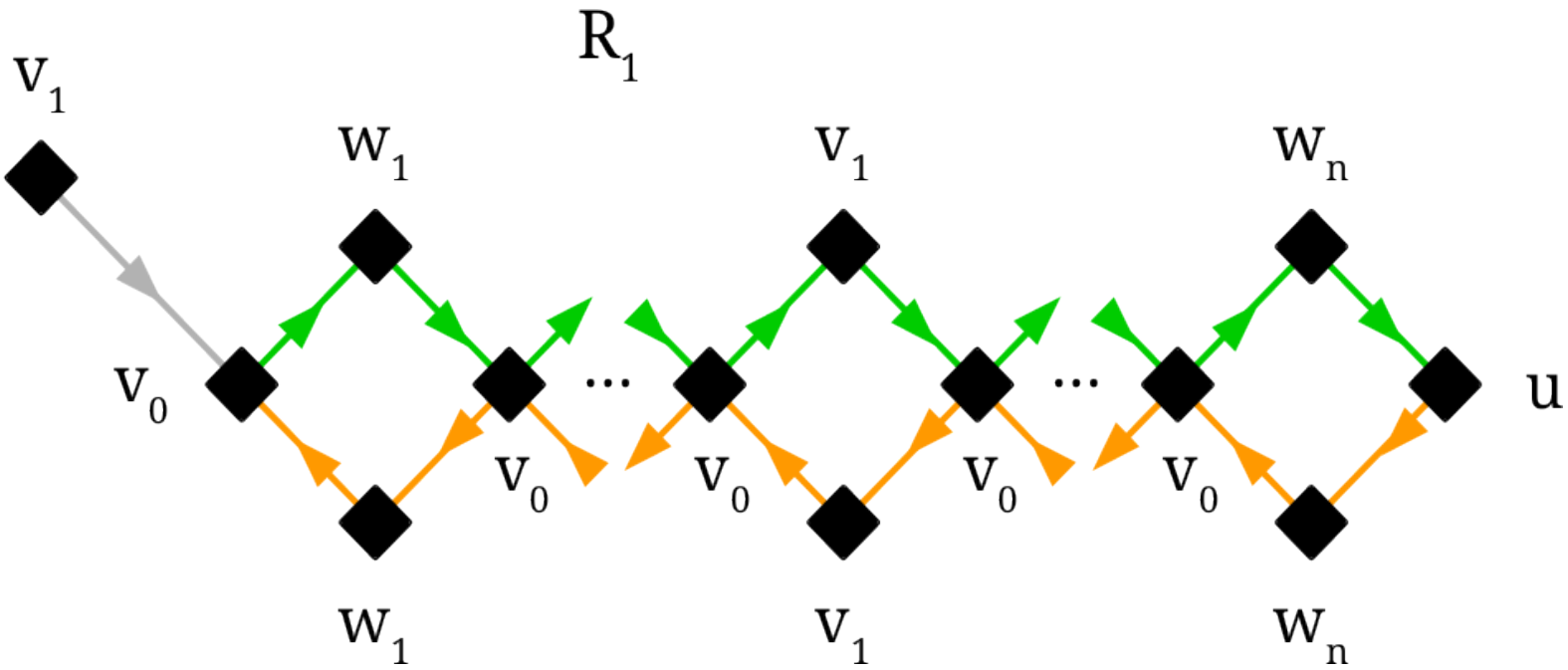
Second improvement: shorten the beginning of R_1



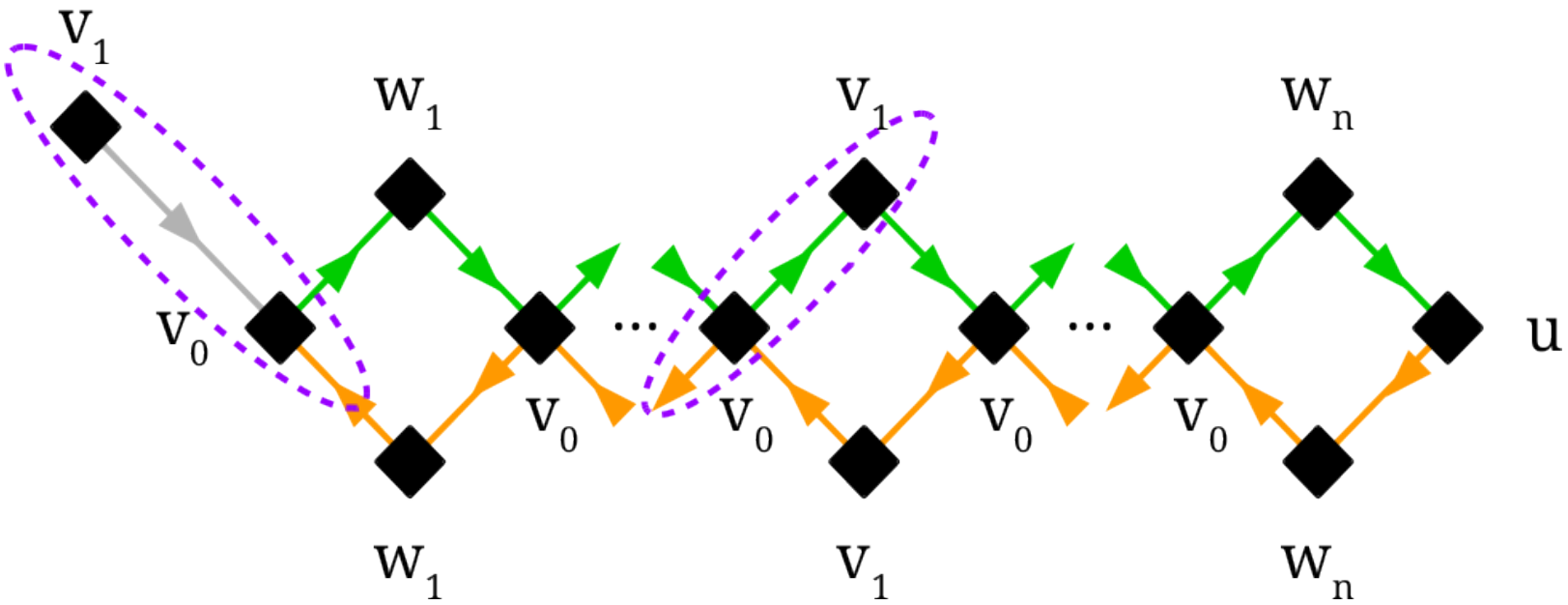
Third improvement once more based on tunnels



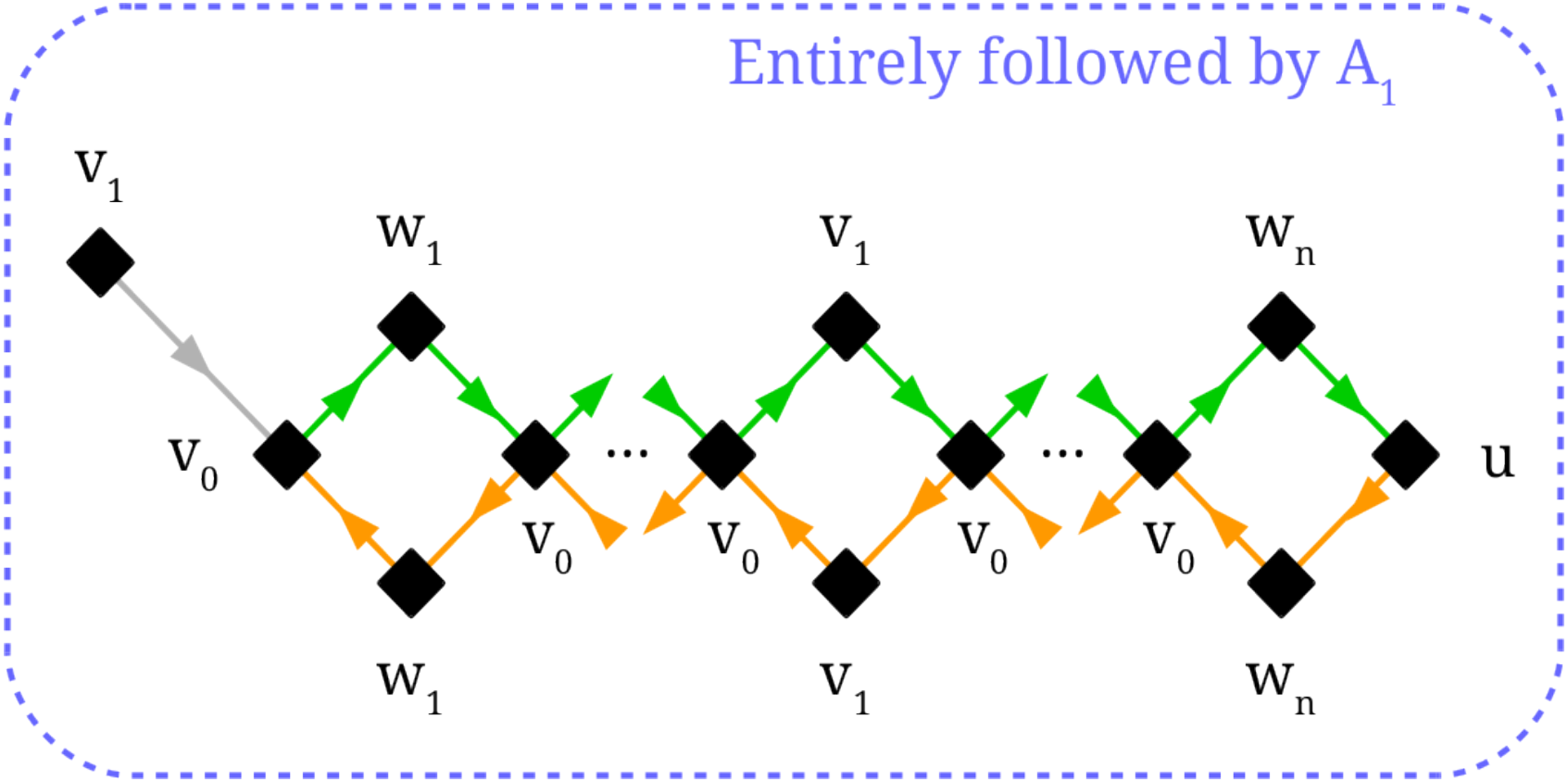
Third improvement once more based on tunnels



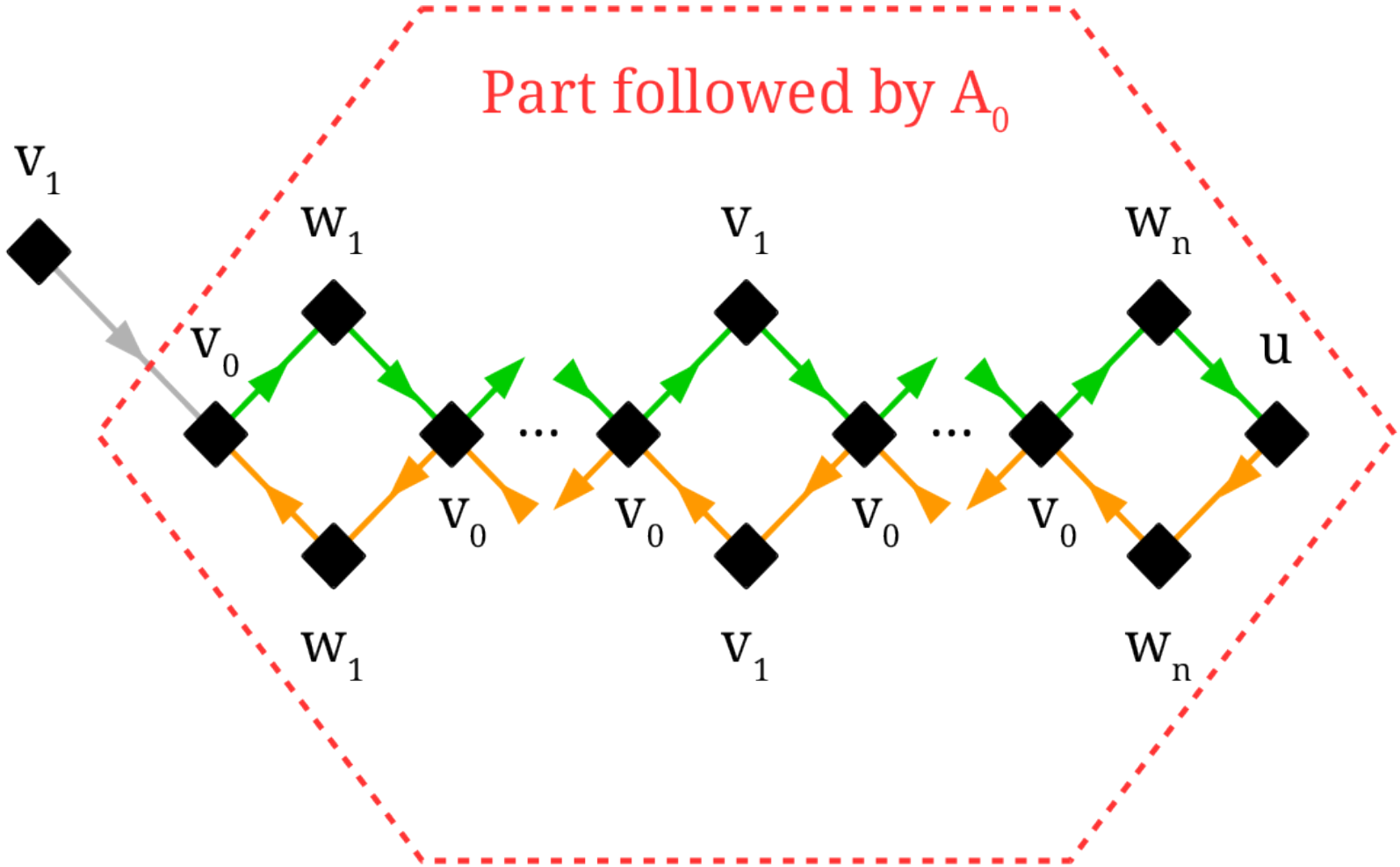
Third improvement once more based on tunnels



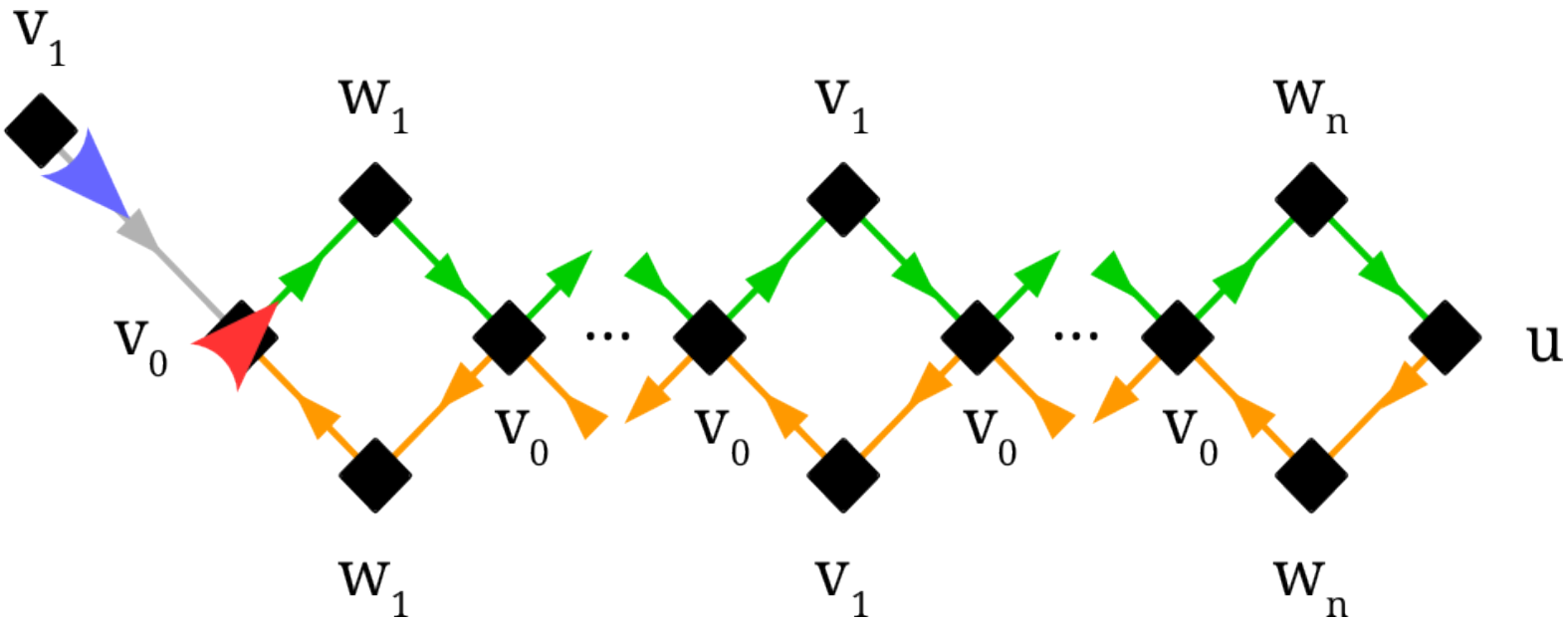
Small break (2)



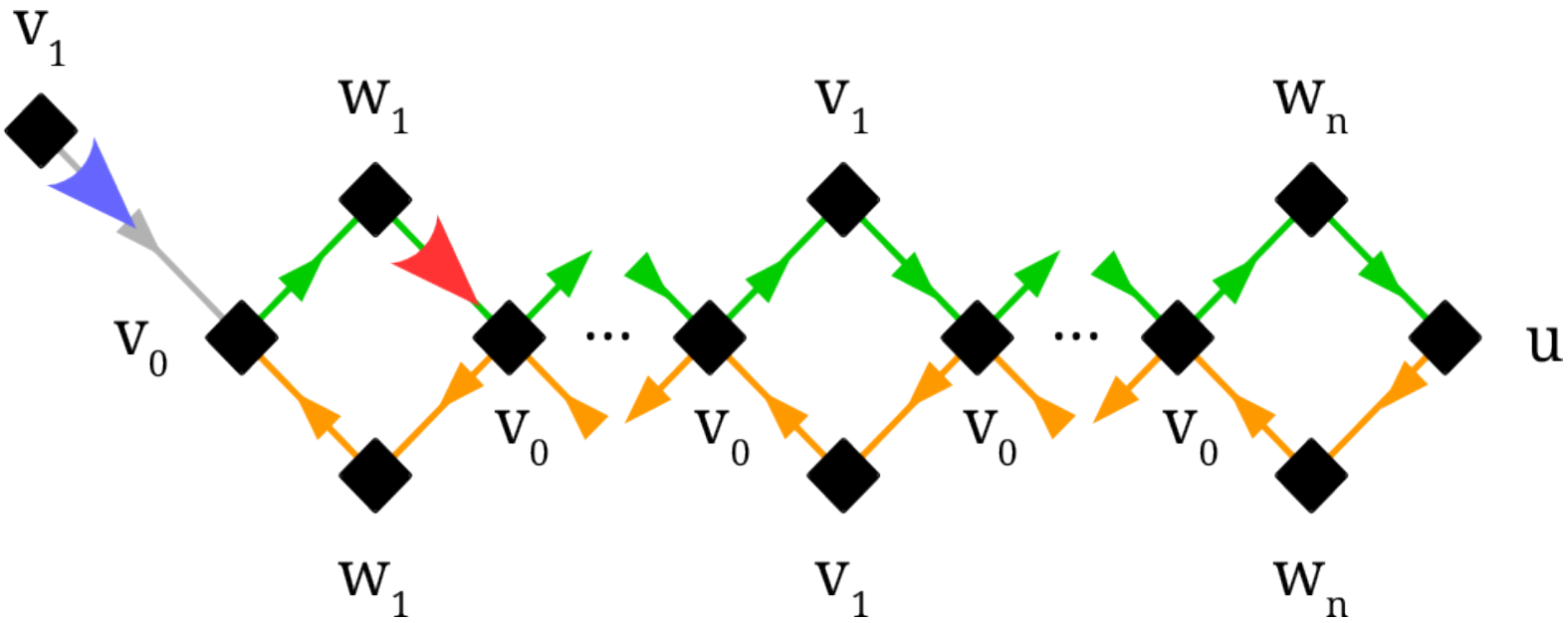
Small break (2)



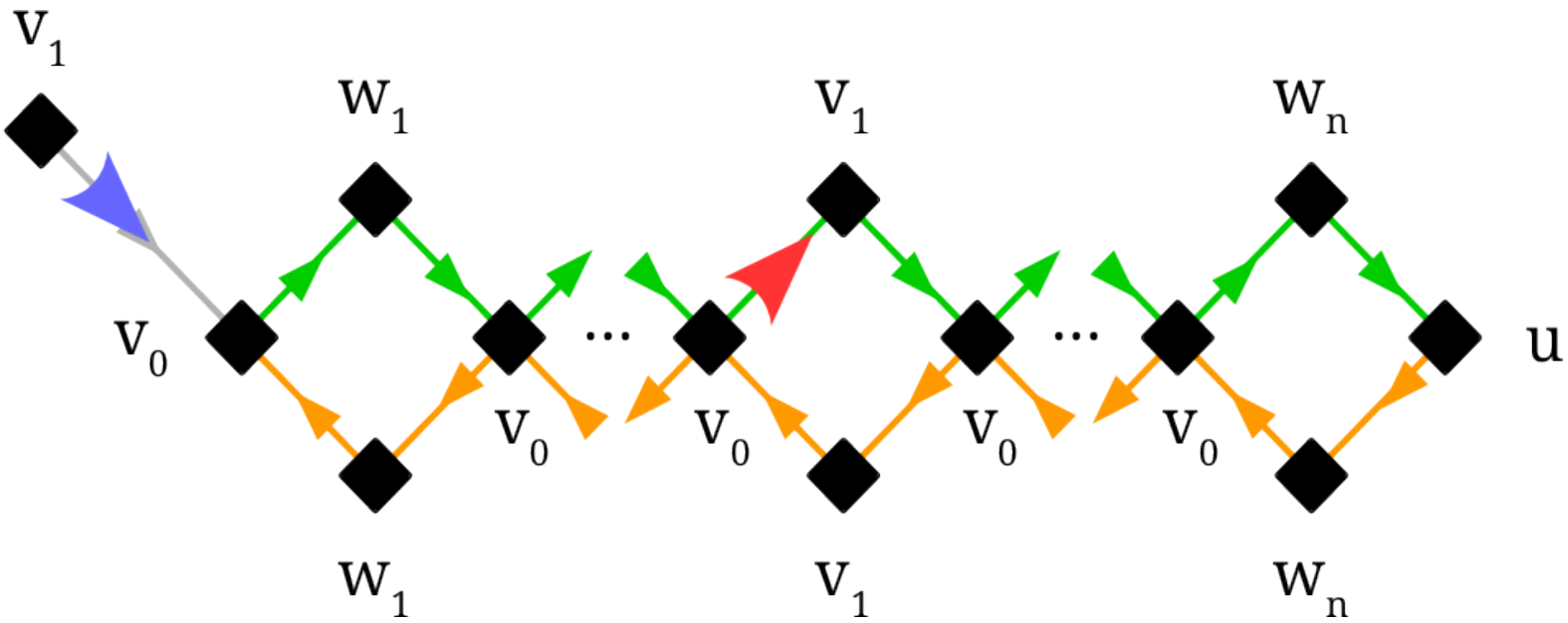
Small break (2)



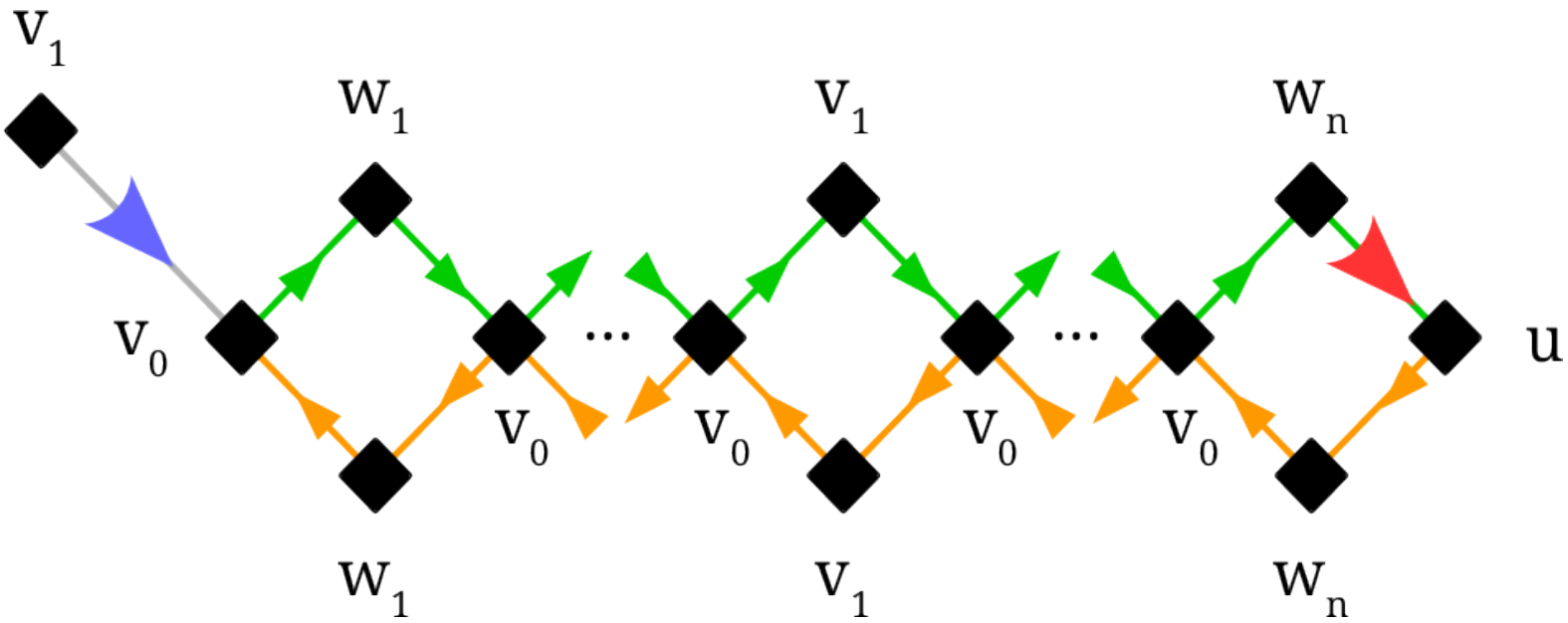
Small break (2)



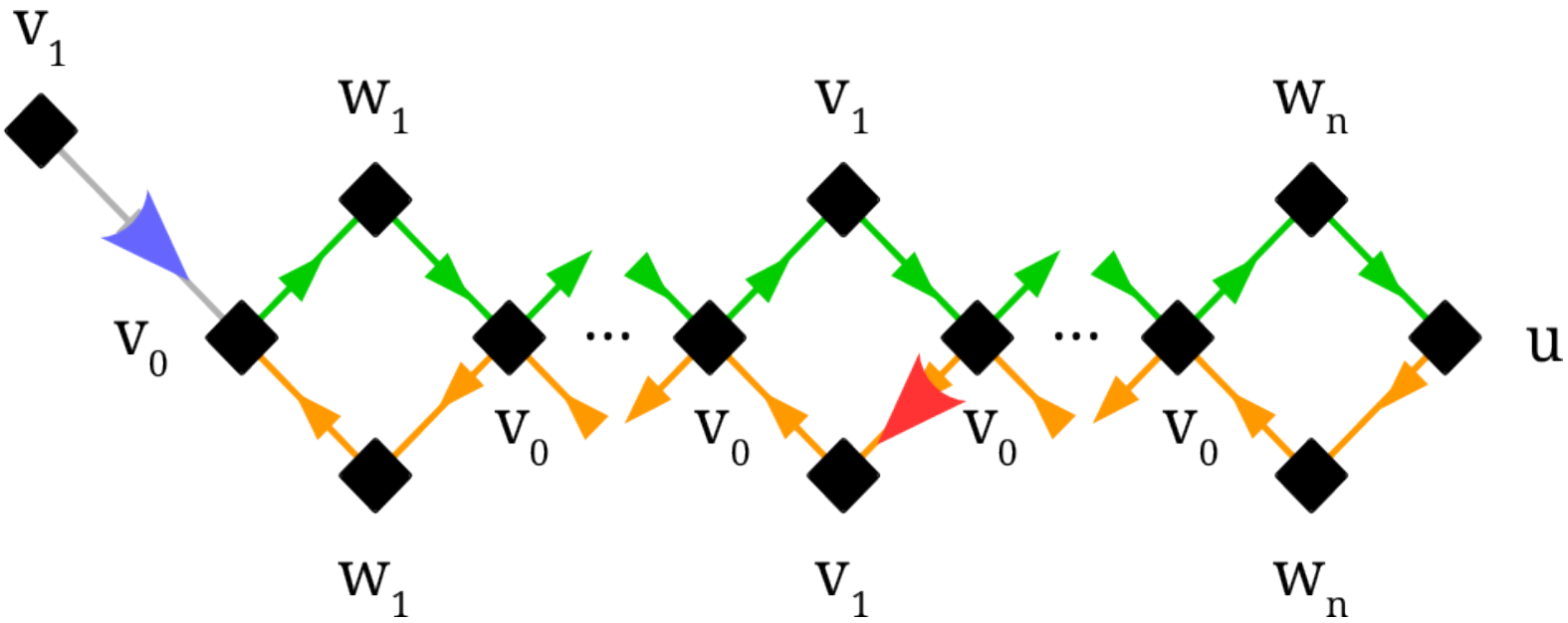
Small break (2)



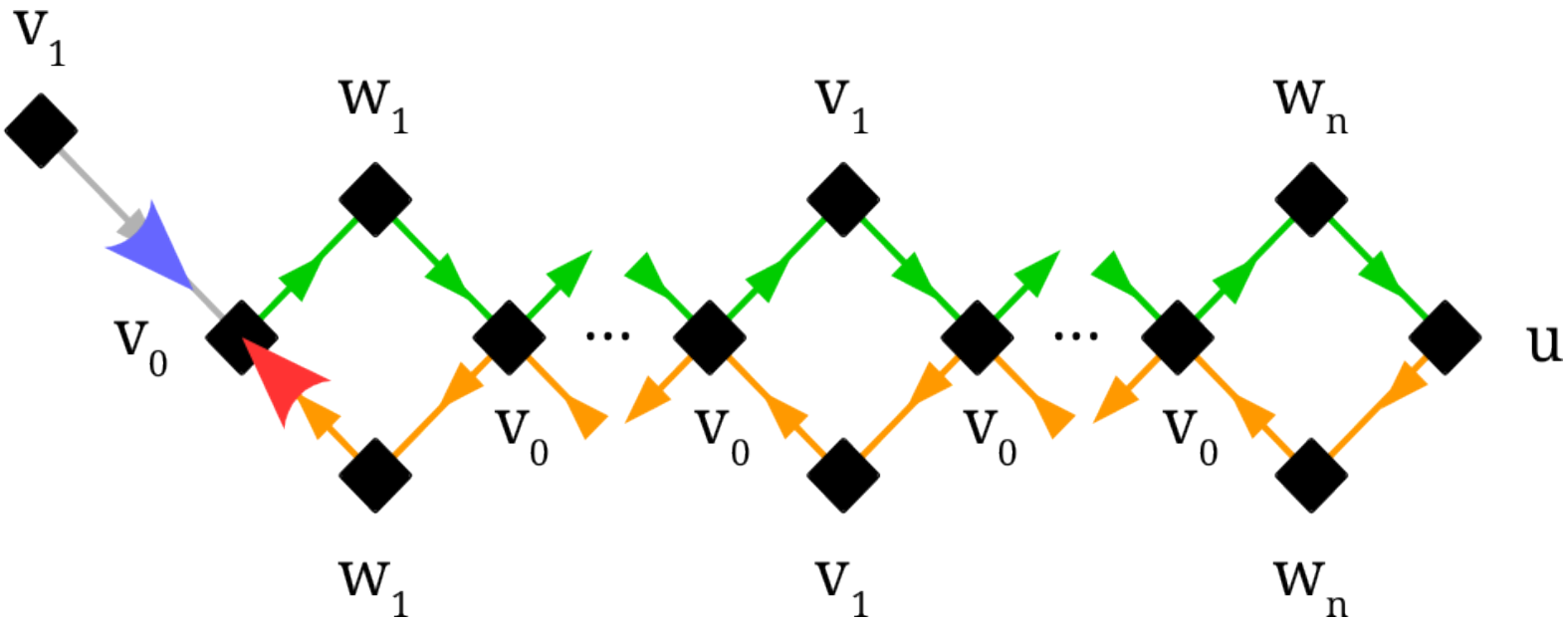
Small break (2)



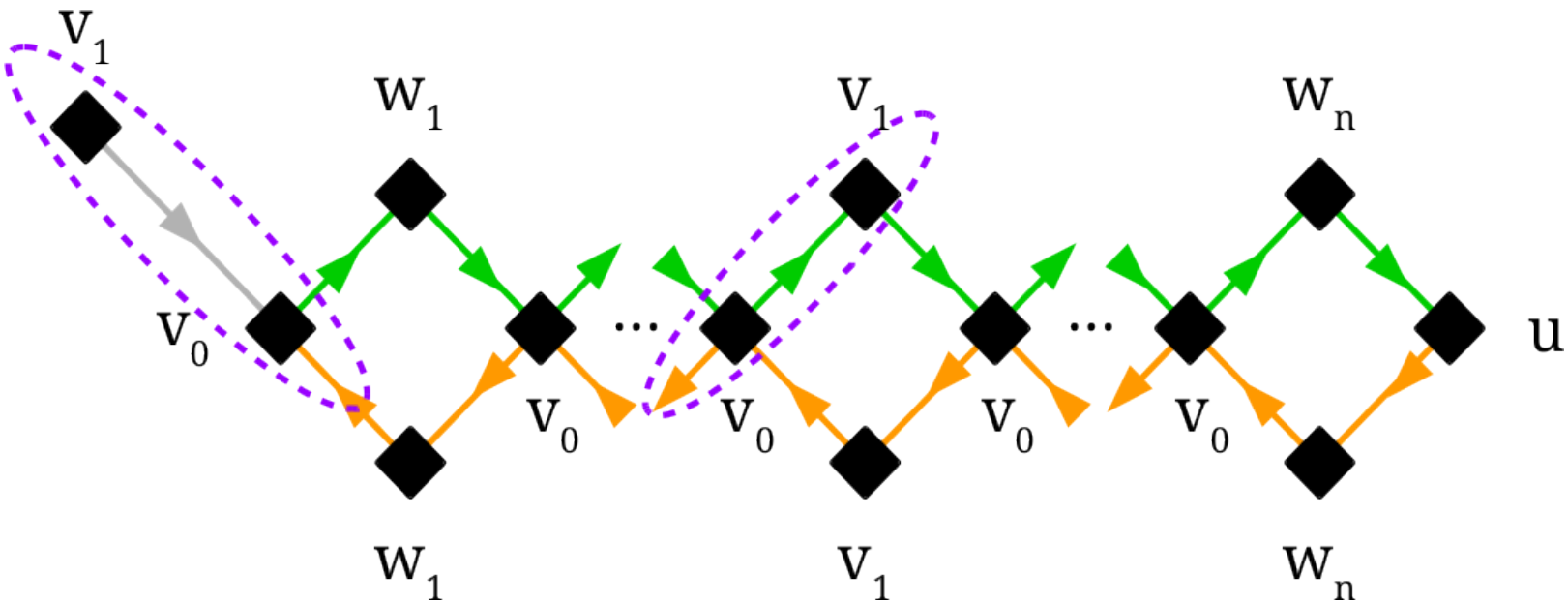
Small break (2)



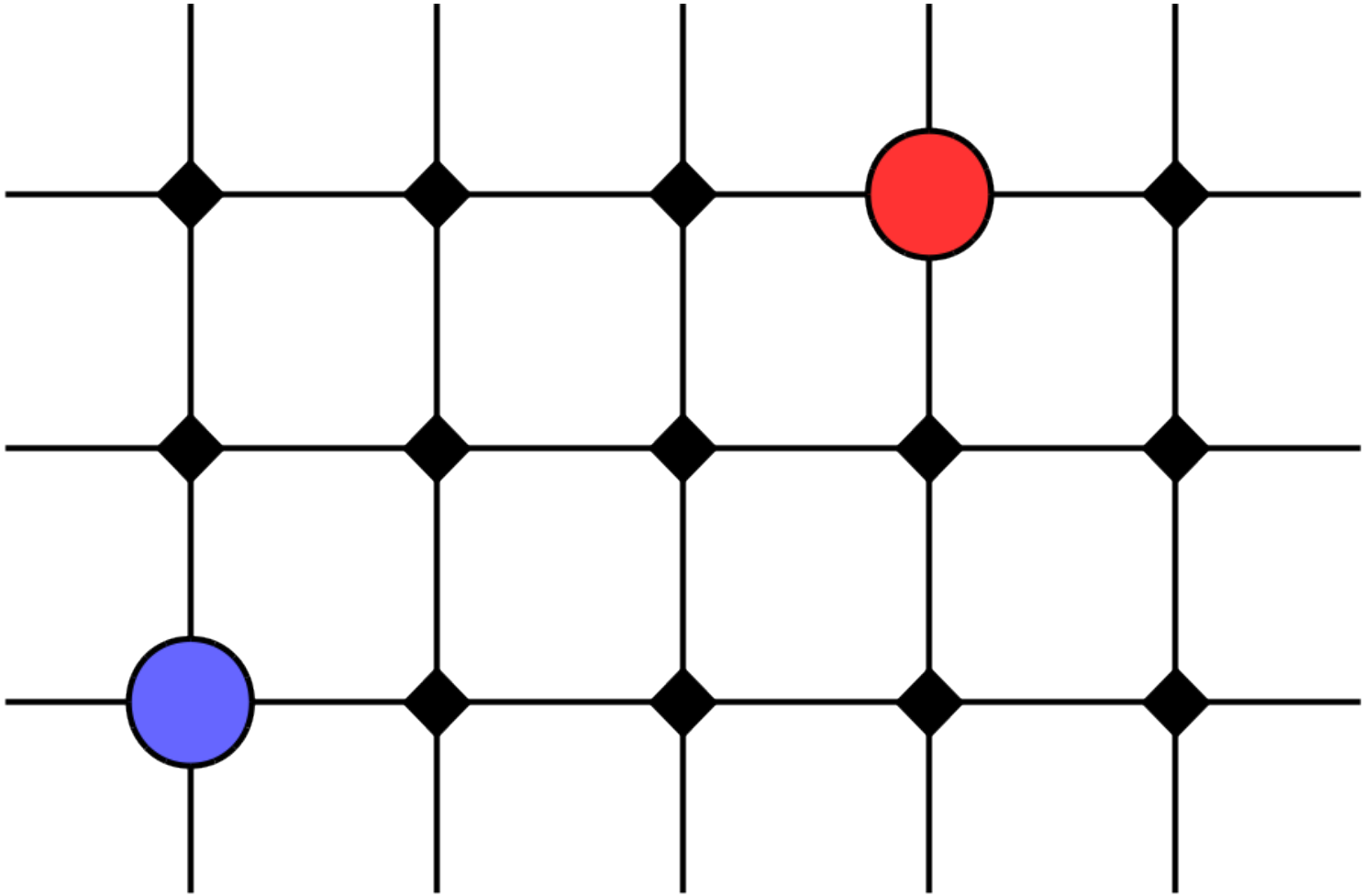
Small break (2)



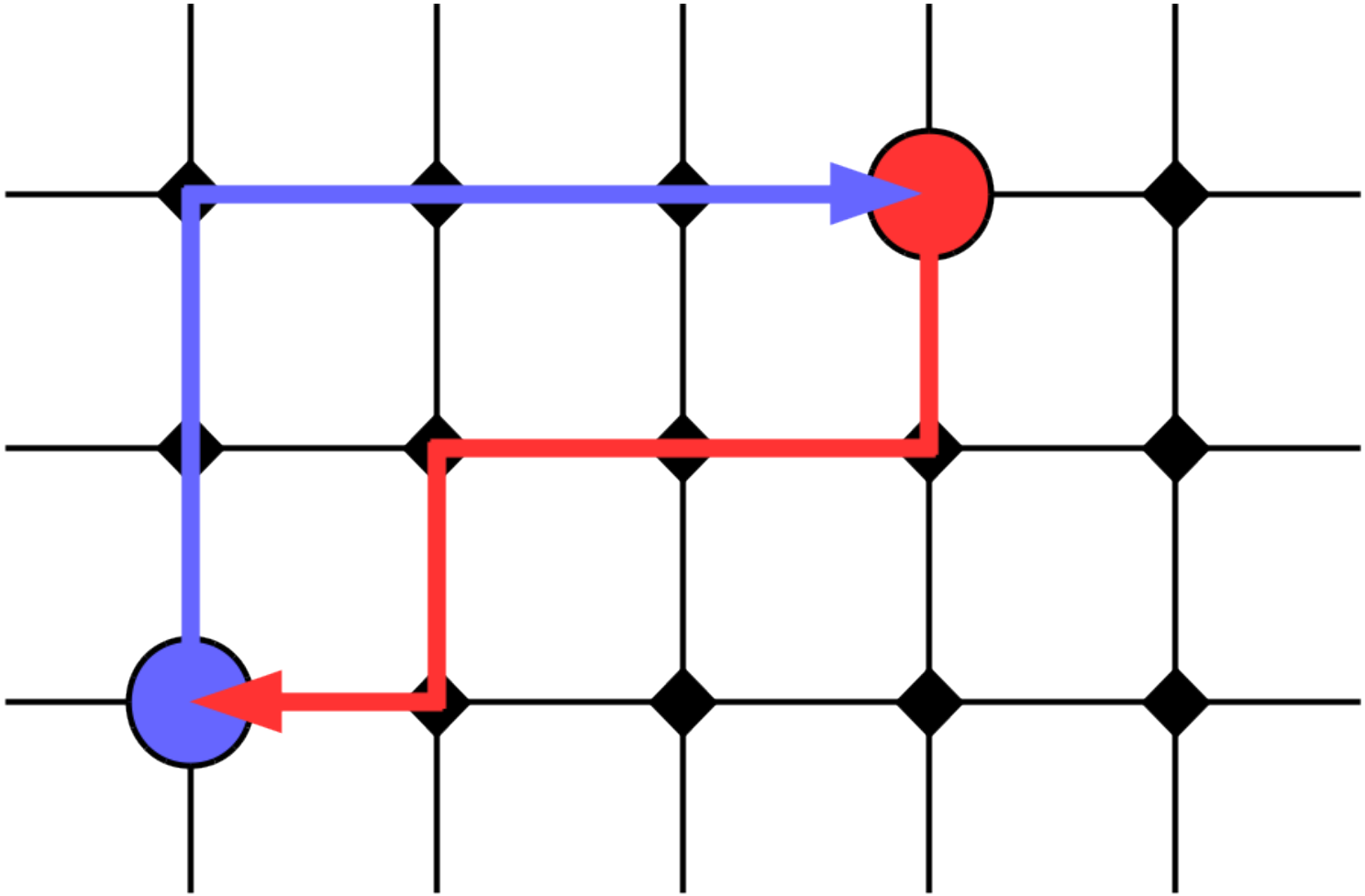
Small break (2)



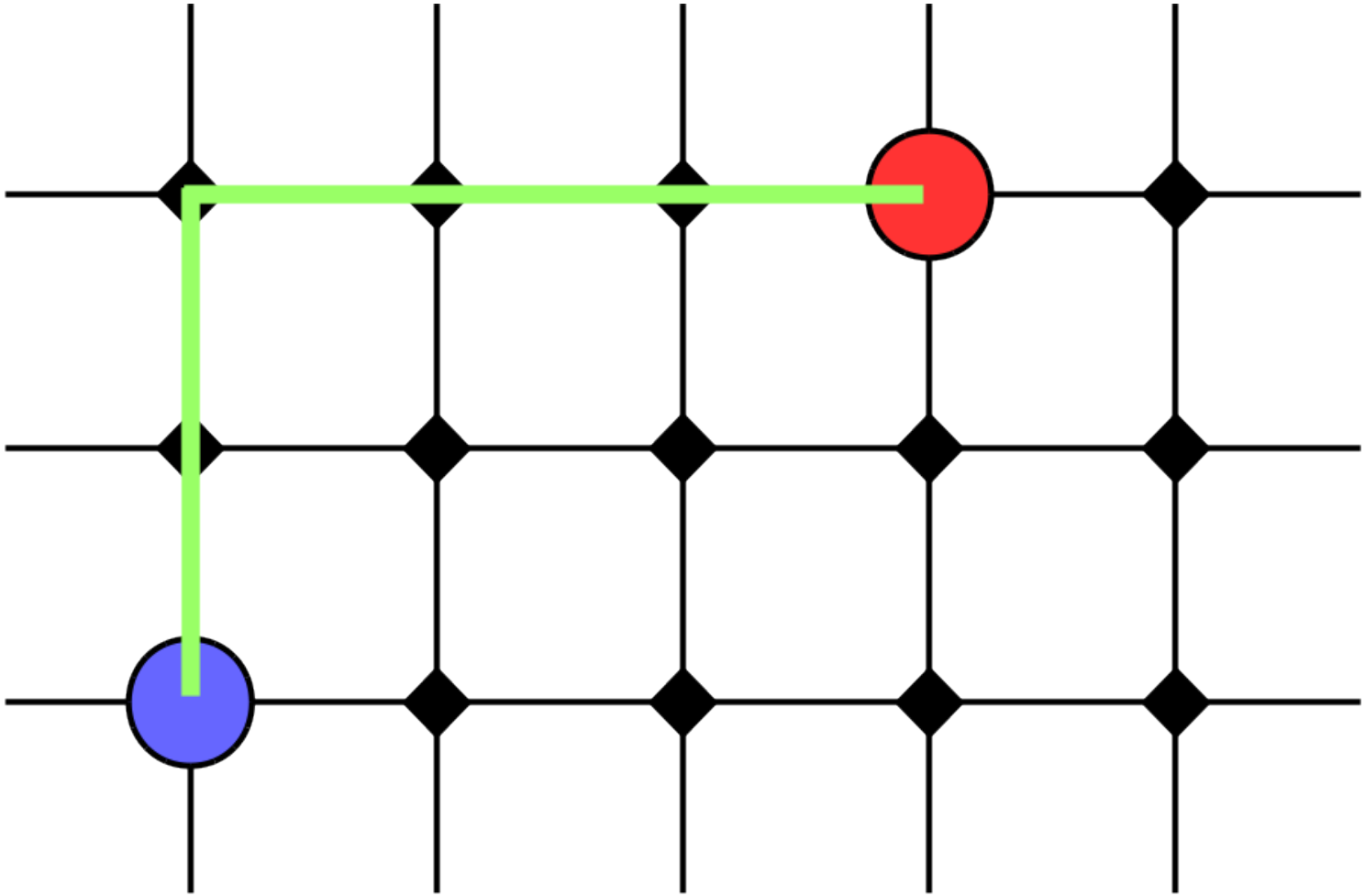
Last improvement: taking a look at the grid



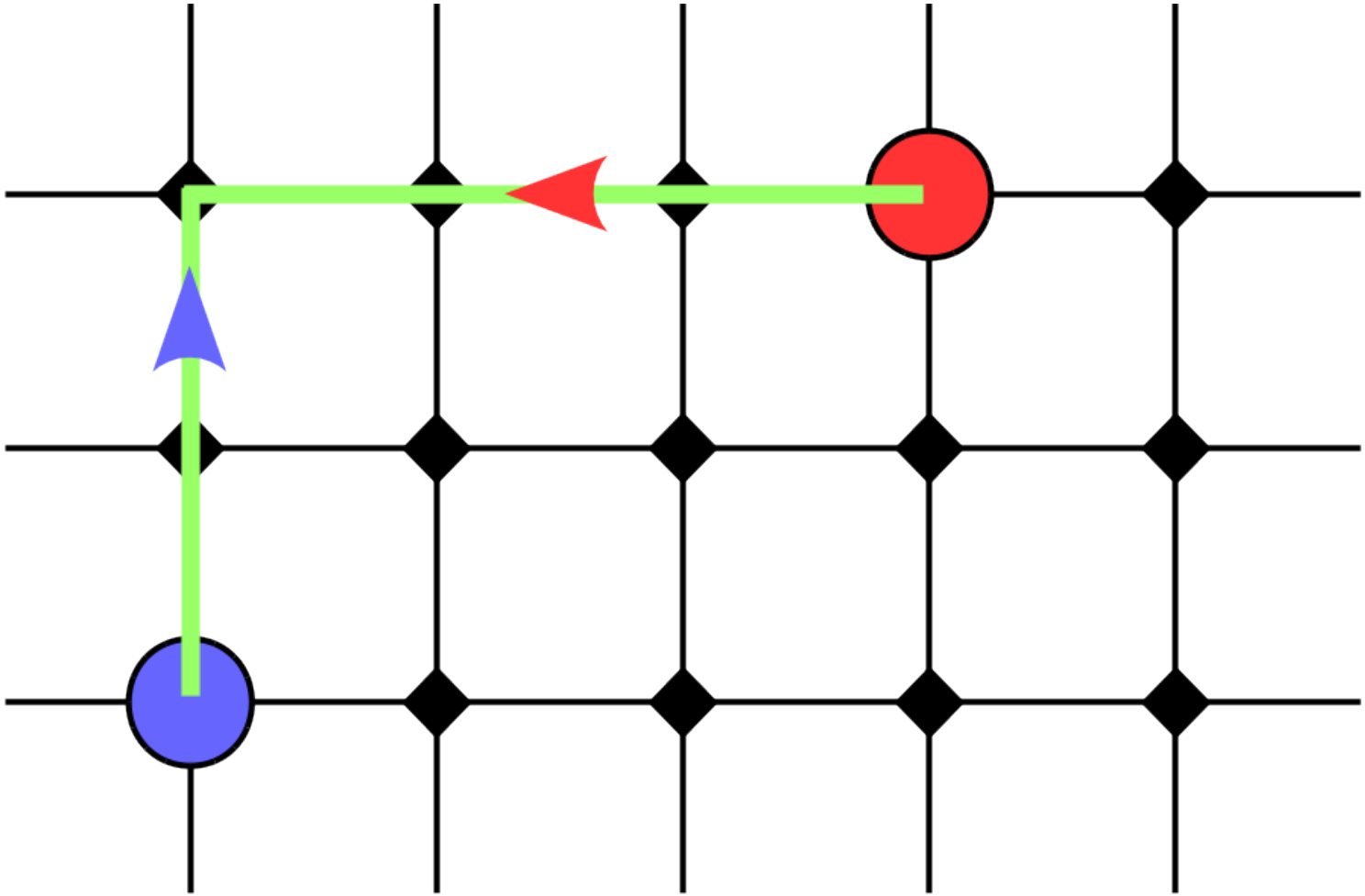
Last improvement: taking a look at the grid



Last improvement: taking a look at the grid



Last improvement: taking a look at the grid



Synchronization mechanisms constraints

We have introduced the main ideas but these are not the actual routes we use.
Unfortunately, due to the lack of time, we cannot explain them in more details.

Conclusion

Our contributions

First deterministic approach algorithm for a fully asynchronous setting whose cost is polynomial in the initial distance and the length of the labels.

Conclusion

Our contributions

First deterministic approach algorithm for a fully asynchronous setting whose cost is polynomial in the initial distance and the length of the labels.

Open question

Impact of the GPS on the cost ?