Распределенные алгоритмы и системы

mk.cs.msu.ru → Лекционные курсы → Распределенные алгоритмы и системы

Блок 24

Примеры волновых алгоритмов:
фазовый алгоритм

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2022/2023, весенний семестр

Напоминание

Волновой алгоритм — это распределённый алгоритм, обладающий тремя свойствами:

1. Завершаемость: каждое вычисление конечно
2. Принятие решения: в каждом вычислении содержится хотя бы одно действие принятия решения
3. Полнота покрытия: в каждом вычислении каждому действию принятия решения причинно-следственно предшествует хотя бы одно действие в каждом узле

Разобранные примеры: кольцевой алгоритм, древесный алгоритм и алгоритм эха — пригодны только для сетей с двусторонними каналами

Для сетей, в которых сообщения по каждому каналу можно отправлять только в одну сторону, требуется предложить другой алгоритм

Ещё немного понятий из теории графов

Конечный орграф (ориентированный граф) — это пара (V, E), где V — конечное множество вершин и $E \subseteq V \times V$ — множество дуг (упорядоченных пар вершин)

Вершина v является началом дуги (v, w) (или, по-другому, $v \to w$), а вершина w — её концом

Путь в орграфе $v_1 \to \cdots \to v_n$ отличается от пути в неорентированном графе $v_1 - \ldots v_n$ только тем, что вместо рёбер используются ду́ги

Расстояние между вершинами в графе — это наименьшее количество дуг в пути, соединяющем эти вершины (если вершины не соединены, то расстояние между ними — ∞)

Диаметр графа — это наибольшее расстояние между его вершинами

Орграф сильно связен, если в нём существует путь из любой вершины в любую другую (то есть если диаметр конечен)

Описание фазового алгоритма

Фазовый алгоритм пригоден для сетей с произвольным сильно связным орграфом топологии $\Gamma = (V, E)$ с хотя бы двумя узлами — далее граф топологии полагается именно таким

Этот алгоритм децентрализован и имеет произвольный набором инициаторов

В узле p используются следующие начальные знания:

- Диаметр $\mathfrak{d}$ орграфа Γ
- Множество входных соседей $\mathfrak{in}_p = \{q \mid q \to p\}$
- Множество выходных соседей $\mathfrak{out}_p = \{q \mid p \to q\}$

Фазовый алгоритм устроен так:

- Инициаторы отправляют фишки всем выходным соседям
- Каждый узел после приёма пересылает фишки: если от каждого входного соседа получено по крайней мере столько же фишек, сколько было разослано выходным соседям, то выходным соседям отправляется ещё по одной фишке
- Когда от каждого входного соседа получено по $\mathfrak{d}$ фишек, узел принимает решение и завершает выполнение

Описание фазового алгоритма

Переменные каждого узла p:

1. $R_p[q] : \{0, 1, \dots, \mathfrak{d}\} = 0$; для каждого $q \in \mathfrak{in}_p$
   - Это число фишек, полученных от входного соседа q
2. $S_p : \{0, 1, \dots, \mathfrak{d}\} = 0$;
   - Это число фишек, отправленных (каждому) выходному соседу

Процедура *Propagate*$_p$ пересылки фишек узлом p:

1. Пока $\min\limits_{q \in \mathfrak{in}(p)} R_p[q] < \mathfrak{d}$:
   - 1.1 $\mathit{receive}(\mathbf{tok}) \leftarrow q_0$ для какого-либо $q_0 \in \mathfrak{in}_p$
   - 1.2 $R_p[q_0] := R_p[q_0] + 1$;
   - 1.3 Если $\min\limits_{q \in \mathfrak{in}_p} R_p[q] \geq S_p$ и $S_p < \mathfrak{d}$:
     - 1.3.1 Для всех $q_0 \in \mathfrak{out}_p$: $\mathit{send}(\mathbf{tok}) \rightarrow q_0$
     - 1.3.2 $S_p[q] := S_p[q] + 1$;

Описание фазового алгоритма

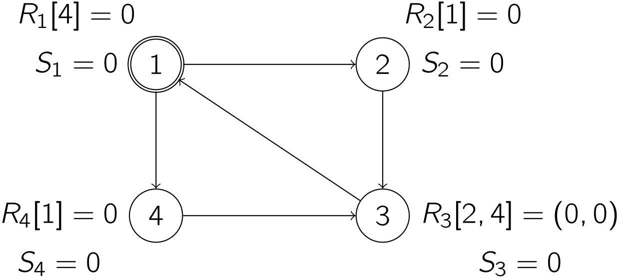
Код последователя p:

1. $\mathit{Propagate}_p$
2. *decide*

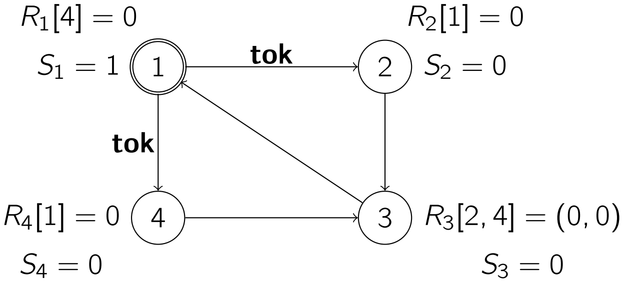
Код инициатора p:

1. Для каждого $q \in \mathfrak{out}_p$: $\mathit{send}(\textbf{tok}) \to q$
2. $S_p[q] := S_p[q] + 1$;
3. $\mathit{Propagate}_p$
4. *decide*

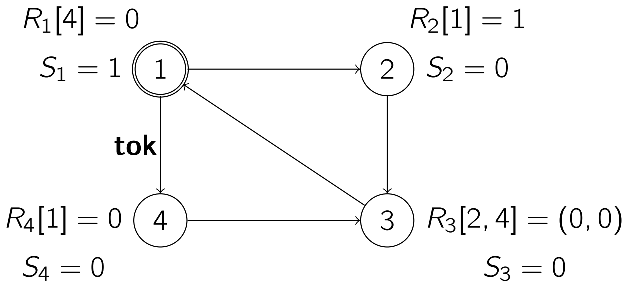
Пример работы фазового алгоритма



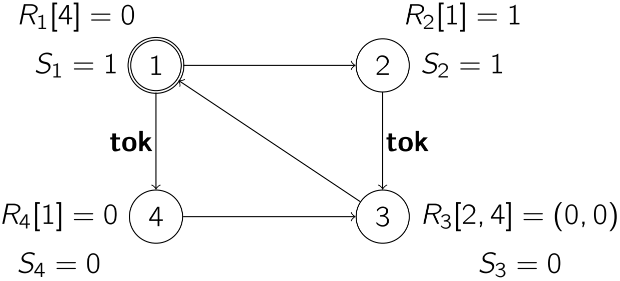
Пример работы фазового алгоритма



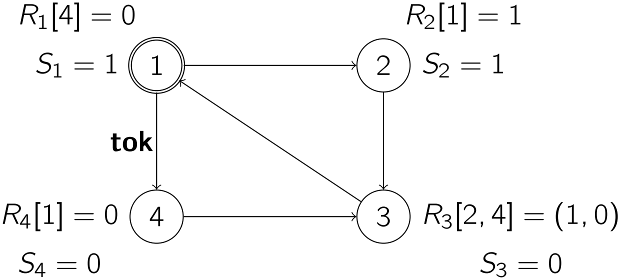
Пример работы фазового алгоритма



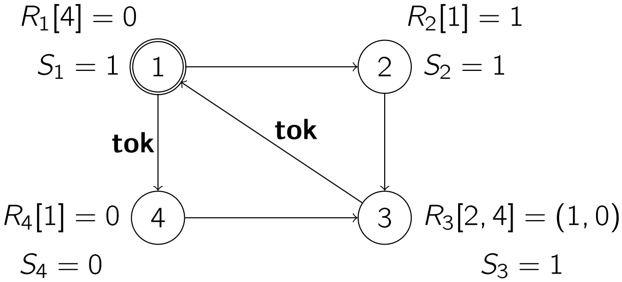
Пример работы фазового алгоритма



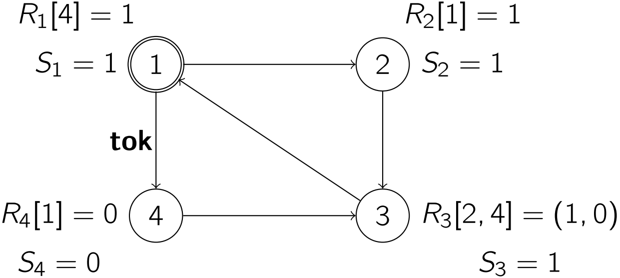
Пример работы фазового алгоритма



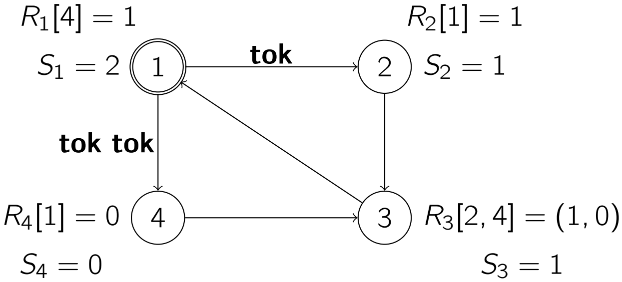
Пример работы фазового алгоритма



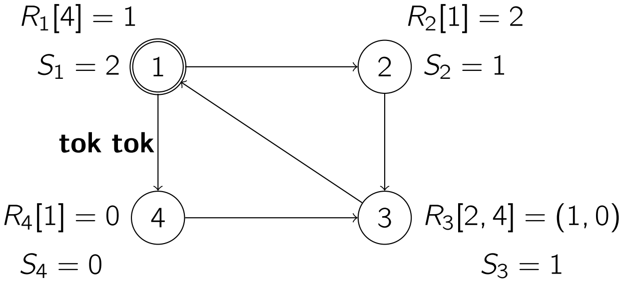
Пример работы фазового алгоритма



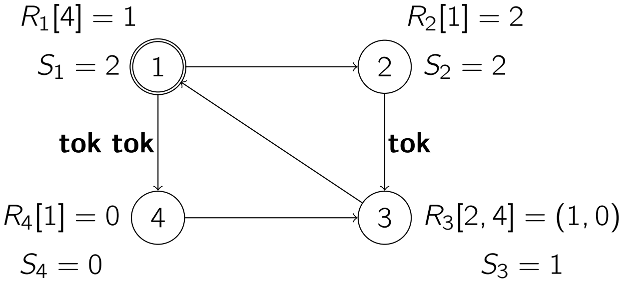
Пример работы фазового алгоритма



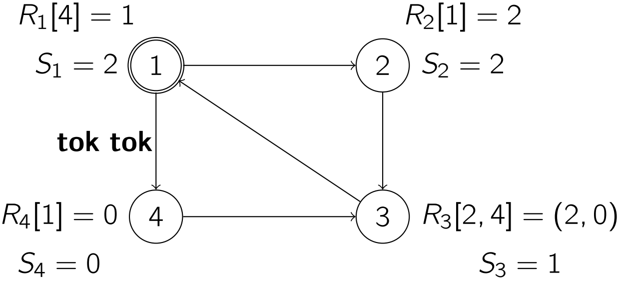
Пример работы фазового алгоритма



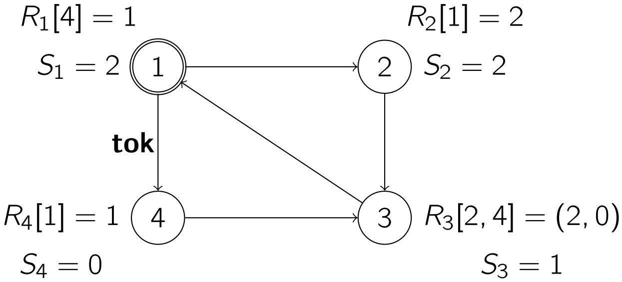
Пример работы фазового алгоритма



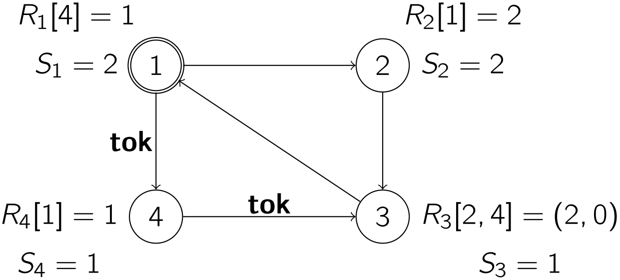
Пример работы фазового алгоритма



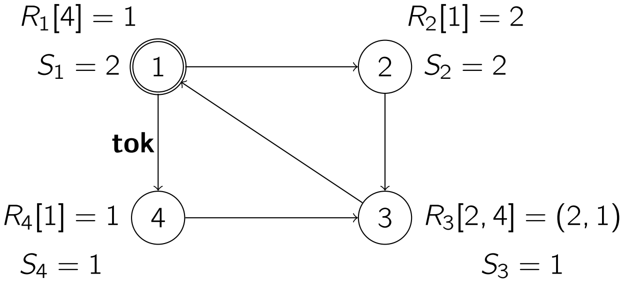
Пример работы фазового алгоритма



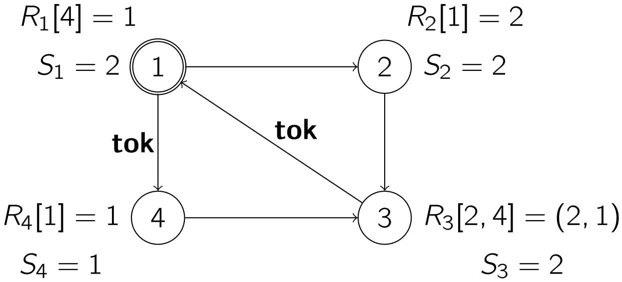
Пример работы фазового алгоритма



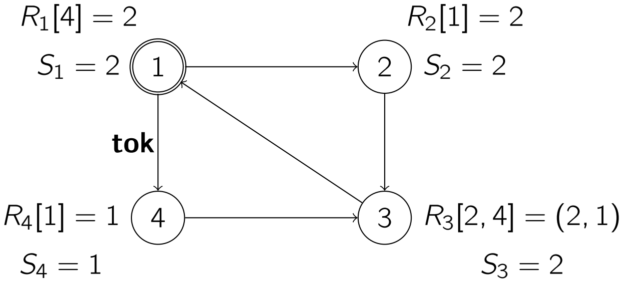
Пример работы фазового алгоритма



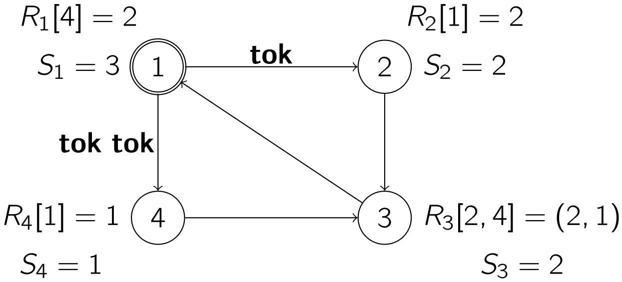
Пример работы фазового алгоритма



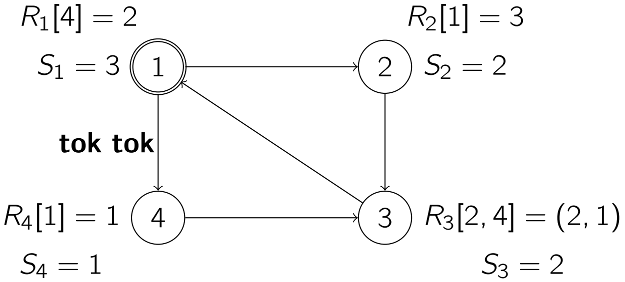
Пример работы фазового алгоритма



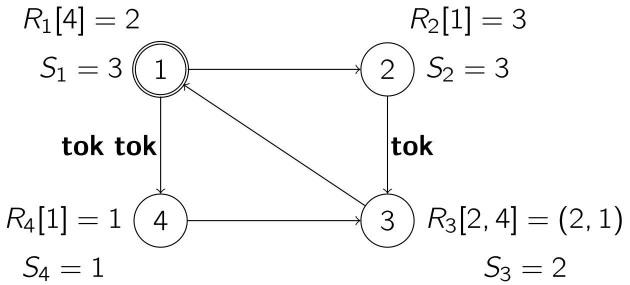
Пример работы фазового алгоритма



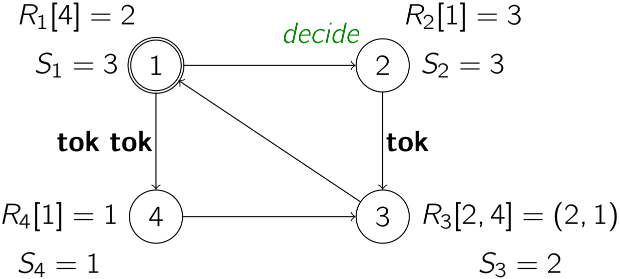
Пример работы фазового алгоритма



Пример работы фазового алгоритма



Пример работы фазового алгоритма



Вспомогательные утверждения

Лемма 1. **В каждом вычислении фазового алгоритма перед каждой итерацией цикла** $Propagate_p$ **для каждого узла** p**, после каждой его итерации и в заключительной конфигурации вычисления верно следующее:**

- $R_p[q]$ **— это количество фишек, принятых узлом** p **от входного соседа** q **и**
- S_p **— это количество фишек, отправленных узлом** p **каждому выходному соседу**

Доказательство.

Следует из устройства алгоритма:

- В начале выполнения $S_p = R_p[q] = 0$
- Отправка фишек узлом p выполняется только одновременно всем выходным соседям и сопровождается присваиванием $S_p := S_p + 1$;
- Приём фишки узлом p от q сопровождается присваиваением $R_p[q] := R_p[q] + 1$; ▼

Вспомогательные утверждения

Лемма 2. В каждом вычислении фазового алгоритма в каждый канал отправляется не более $\mathfrak{d}$ фишек

Доказательство.

Согласно лемме 1 и устройству алгоритма, после каждой отправки сообщений узлом p значение S_p — это количество сообщений, отправленное узлом p каждому выходному соседу

По устройству алгоритма, значение S_p изменяется только присваиванием $S_p := S_p + 1$; и только в двух случаях:

1. В инициаторе в начале выполения
2. В цикле $Propagate_p$, если $S_p < \mathfrak{d}$

Значит, всегда верно $S_p \leq \mathfrak{d}$ ▼

Вспомогательные утверждения

Лемма 3. В каждой заключительной конфигурации каждого вычисления фазового алгоритма все каналы пусты и для любого канала $p \to q$ верно $S_p = R_p[q]$

Доказательство.

По лемме 1, S_p и $R_p[q]$ — количество фишек, соответственно отправленных в канал $p \to q$ и принятых из него

По лемме 2, каждым узлом каждому выходному соседу отправляется не более $\mathfrak{d}$ фишек

По устройству алгоритма, каждым узлом от каждого входного соседа принимаются все отправленные фишки, если их не более $\mathfrak{d}$ ▼

Вспомогательные утверждения

Лемма 4. В каждом вычислении фазового алгоритма каждый узел отправляет хотя бы одну фишку каждому выходному соседу

Доказательство (индукцией по удалённости от p).

База: по устройству алгоритма, каждый инициатор в каждом вычислении отправляет хотя бы одну фишку каждому выходному соседу

Индуктивный переход: положим для произвольно взятого инициатора p, что утверждение верно для каждого узла, находящийся от p на расстоянии не более d; покажем, что утверждение верно для любого узла q, находящегося от p на расстоянии $(d+1)$

Пусть r — предпоследний узел в кратчайшем пути от p к q

По предположению индукции, узел r отправляет q хотя бы одну фишку

По устройству алгоритма, узел q принимает эту фишку

Значит, узел q обязательно принимает хотя бы одну фишку

После приёма первой фишки узлом q верно $\min\limits_{q \in \mathfrak{in}_p} R_p[q] \geq 0 = S_p$ и $S_p = 0 < \mathfrak{d}$, и по устройству алгоритма узел q отправляет по одной фишке каждому выходному соседу после этого приёма ▼

Вспомогательные утверждения

Лемма 5. **В каждом вычислении фазового алгоритма перед каждой итерацией цикла** $\mathit{Propagate}_p$ **для каждого узла** p**, после каждой его итерации и в заключительной конфигурации вычисления верно следующее:**

1. $S_p \geq \min\limits_{q \in \mathfrak{in}_p} R_p[q]$
2. **Если** $\min\limits_{q \in \mathfrak{in}_p} R_p[q] < \mathfrak{d}$ **и верно хотя бы одно из двух: (а)** p **— инициатор; (б)** $\sum\limits_{q \in \mathfrak{in}_p} R_p[q] > \min\limits_{q \in \mathfrak{in}_p} R_p[q]$ **— то** $S_p > \min\limits_{q \in \mathfrak{in}_p} R_p[q]$

Доказательство.

Пункт 1 следует из того, что неравенство верно перед выполнением $\mathit{Propagate}_p$ и поддерживается каждой итерацией этого цикла

Пункт 2 следует из того, что если перед очередной итерацией $\mathit{Propagate}_p$ верно неравенство $S_p > \min\limits_{q \in \mathfrak{in}_p} R_p[q]$ или значение $\min\limits_{q \in \mathfrak{in}_p} R_p[q]$ не изменяется, то строгое неравенство поддерживается сохраняется, если не достигнуты значения $\min\limits_{q \in \mathfrak{in}_p} R_p[q] = S_p = \mathfrak{d}$ ▼

Вспомогательные утверждения

Лемма 6. В каждой заключительной конфигурации γ каждого вычисления фазового алгоритма для любого узла p верно $S_p = \mathfrak{d}$

Доказательство.

По леммам 1 и 2, достаточно показать, что в γ для узла p со значением S_p, наименьшим среди всех узлов, верно $S_p = \mathfrak{d}$

Рассмотрим такой узел p и *предположим от противного*, что $S_p < \mathfrak{d}$

Тогда для любого узла $q \in \mathfrak{in}_p$ в γ верно $S_p \leq S_q$, то есть $S_p \leq \min\limits_{q \in \mathfrak{in}_p} S_q$

По лемме 3, в γ верно $S_q = R_p[q]$, а значит, и $S_p \leq \min\limits_{q \in \mathfrak{in}_p} R_p[q]$

По лемме 5, в γ верно $S_p \geq \min\limits_{q \in \mathfrak{in}_p} R_p[q]$, а значит, $S_p = \min\limits_{q \in \mathfrak{in}_p} R_p[q]$

По лемме 3, существует узел $q \in \mathfrak{in}_p$, такой что $S_q = R_p[q] = S_p$

Вспомогательные утверждения

Лемма 6. **В каждой заключительной конфигурации γ каждого вычисления фазового алгоритма для любого узла p верно $S_p = \mathfrak{D}$**

Доказательство.

Повторяя проведённые рассуждения для q и для других узлов, получим цикл $\mathcal{C} = (p_1 \to \dots \to p_n \to p_1)$, такой что

$$S_{p_1} = R_{p_1}[p_n] = \min_{q \in \mathfrak{in}_{p_1}} R_{p_1}[q] = \dots = S_{p_n} = R_{p_n}[p_{n-1}] = \min_{q \in \mathfrak{in}_{p_n}} R_{p_n}[q] < \mathfrak{D}$$

По лемме 5, в $\mathcal{C}$ нет ни одного инициатора

По сильной связности графа топологии, существует путь из инициатора p в одну из вершин цикла $\mathcal{C}$ (для ясности — p_n), такой что предпоследняя вершина r этого пути не содержится в $\mathcal{C}$

По леммам 3 и 4, верно

$$\sum_{q \in \mathfrak{in}_{p_n}} R_{p_n}[q] \geq R_{p_n}[p_{n-1}] + R_{p_n}[r] > R_{p_n}[p_{n-1}] \geq \min_{q \in \mathfrak{in}_{p_n}} R_{p_n}[q]$$

По лемме 5, тогда верно $S_{p_n} > \min_{q \in \mathfrak{in}_{p_n}} R_{p_n}[q]$, что *противоречит* устройству цикла $\mathcal{C}$ ▼

Теорема о фазовом алгоритме

Теорема. **Для любого сильно связного графа топологии с хотя бы двумя узлами и любого непустого набора инициаторов фазовый алгоритм является волновым**

Доказательство.

Завершаемость

По лемме 2, в каждый канал отправляется не более $\mathfrak{D}$ сообщений

При этом на каждой итерации цикла $Propagate_p$ принимается хотя бы одно сообщение

Следовательно, цикл обязательно завершается за конечное число итераций, и вычисление алгоритма обязательно конечно

Теорема о фазовом алгоритме

Доказательство.

Принятие решения: покажем, что **каждый** узел рано или поздно принимает решение

По лемме 6, в заключительной конфигурации γ для каждого узла p верно $S_p = \mathfrak{d}$

По лемме 3, для любого узла p и любого его входного соседа q в γ верно $R_p[q] = \mathfrak{d}$

По устройству алгоритма, это означает, что цикл $Propagate_p$ завершился и выполнилась следующая за ним команда *decide*

Теорема о фазовом алгоритме

Доказательство.

Полнота покрытия

Рассмотрим узел p, принявший решение в конфигурации γ некоторого вычисления, и произвольный узел q, отличный от p

По сильной связности орграфа топологии и определению диаметра, существует путь из q в p, содержащий не более $\mathfrak{d}$ дуг — для ясности, $q = q_0 \to \cdots \to q_k = p$

По устройству алгоритма и свойствам отношени $\preceq$, верно $\mathfrak{s}^1_{q_0 \to q_1)} \prec \mathfrak{r}^1_{q_0 \to q_1} \prec \mathfrak{s}^2_{q_1 \to q_2} \prec \mathfrak{r}^2_{q_1 \to q_2} \prec \cdots \prec \mathfrak{r}^k_{q_{k-1} \to q_k}$, где $\mathfrak{s}^i_e$ и $\mathfrak{r}^i_e$ — соответственно номер i-й отправки и фишки в канал e и i-го приёма из e в рассматриваемом вычислении

При этом *decide* — последняя команда узла, а значит, если δ — номер действия, отвечающего выполнению *decide* в вычислении, то $\mathfrak{r}^k_{q_{k-1} \to q_k} \prec \delta$

Следовательно, $\mathfrak{s}^1_{q_0 \to q_1)} \prec \delta$, то есть действию принятия решения (любого) узла p обязательно предшествует действие отправки фишки (любого другого) узла q ▼

Заключение

Задача 1. Преобразуйте фазовый алгоритм в INF-алгоритм для вычисления максимума чисел. Каковы коммуникационная и битовая сложности фазового и получившегося алгоритмов? Ответ обосновать

Задача 2. Предположим, что каждая фишка в фазовом алгоритме снабжена уникальным идентификатором, отличающих её от всех других фишек — например, именами узлов на концах канала и порядковым номером отправки в канал. Покажите, что даже в этом случае справедливо соотношение $\mathfrak{s}_e^i \prec \mathfrak{r}_e^i$ для всех e и i

Задача 3. Добавим возможность дублирования сообщений во всех каналах сети. Какое место доказательства корректности фазового алгоритма становится неверным? Как следует изменить фазовый алгоритм, чтобы он остался корректным (ответ обосновать)?