Распределенные алгоритмы и системы

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

Блок 26

Распределённые алгоритмы обхода
Алгоритм Тарри
Классический обход в глубину

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

ВМК МГУ, 2022/2023, весенний семестр

Алгоритмы обхода

Распределённый алгоритм обхода — это централизованный волновой алгоритм, устроенный так:

- Инициатор отправляет фишку обхода, после чего действуют следующие правила
- Каждый узел ожидает приёма фишки обхода, и после приёма либо отправляет одну фишку обхода, либо принимает решение (запрещено одновременно и отправлять, и принимать решение)
- Решение принимает инициатор

Алгоритм Тарри

Алгоритм Тарри — это распределённый алгоритм обхода, подходящий для произвольного неориентированного связного графа топологии $\Gamma = (V, E)$, пересылающий только фишку обхода и устроенный произвольно в рамках двух ограничений (правил):

O1. Фишку нельзя передавать по одному (направленному) каналу дважды

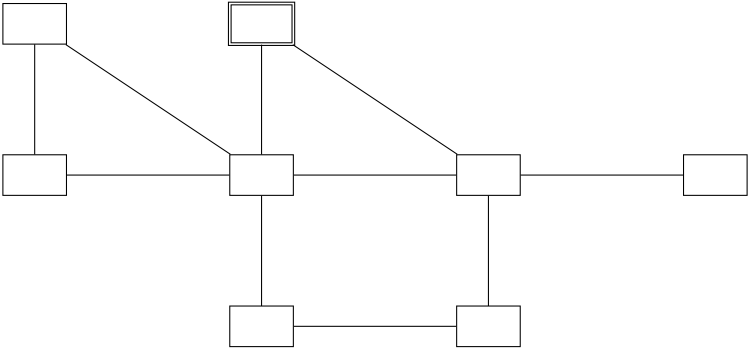
O2. Последователь может передать фишку тому, от кого получил её самым первым действием, только в том случае, если её невозможно передать другим узлам согласно (O1).

Узел, от которого p принял фишку самым первым своим действием, — это родитель узла p

В процессе обхода алгоритм Тарри распределённо вычисляет остовное дерево с обозначенными выше родителями

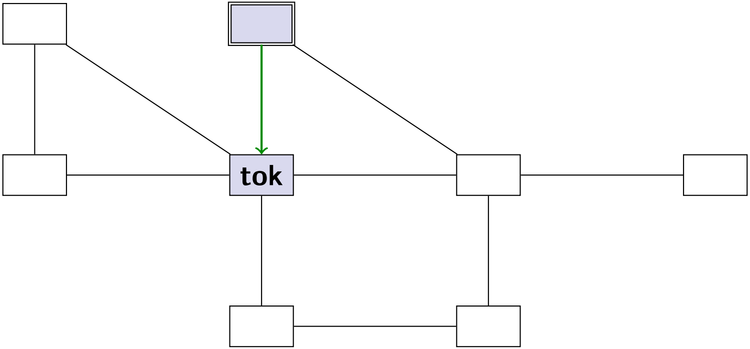
Алгоритм Тарри

Пример выполнения алгоритма Тарри



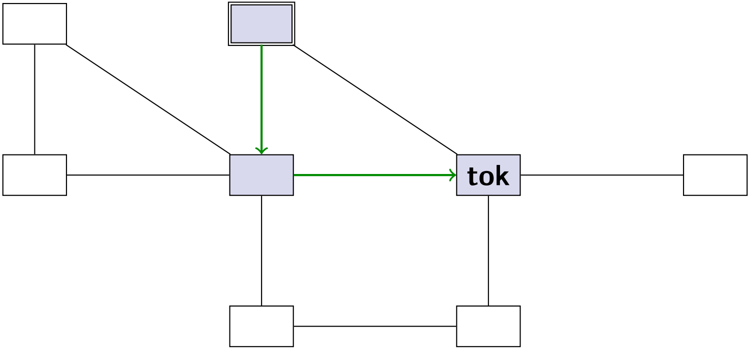
Алгоритм Тарри

Пример выполнения алгоритма Тарри



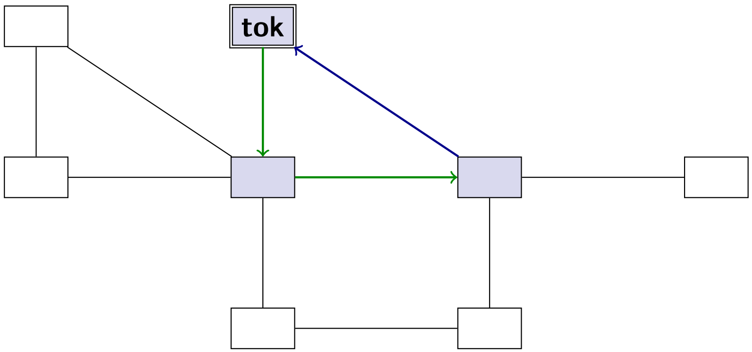
Алгоритм Тарри

Пример выполнения алгоритма Тарри



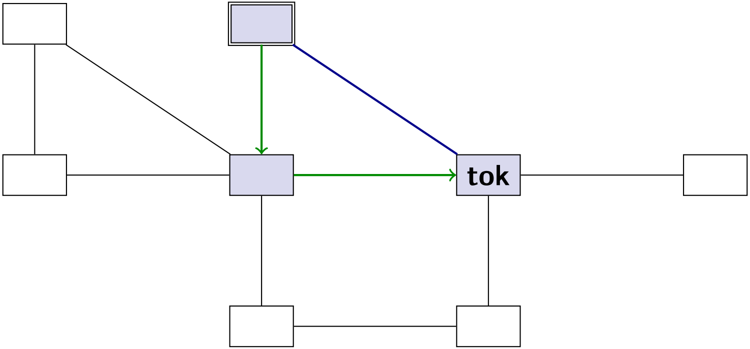
Алгоритм Тарри

Пример выполнения алгоритма Тарри



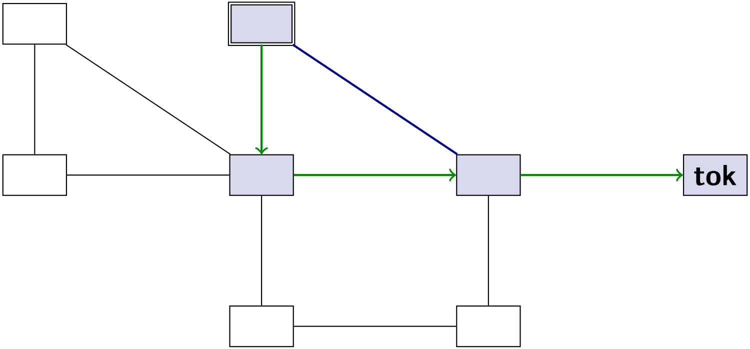
Алгоритм Тарри

Пример выполнения алгоритма Тарри



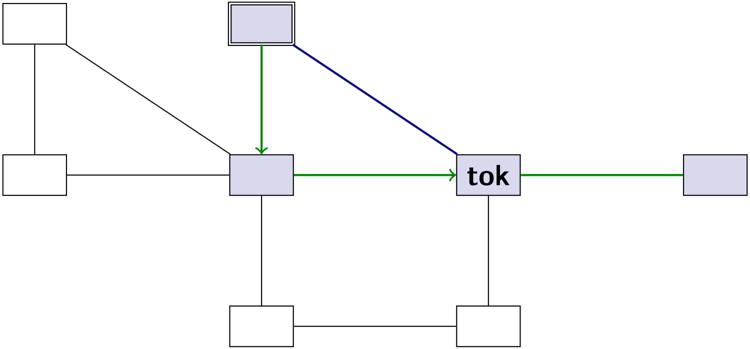
Алгоритм Тарри

Пример выполнения алгоритма Тарри



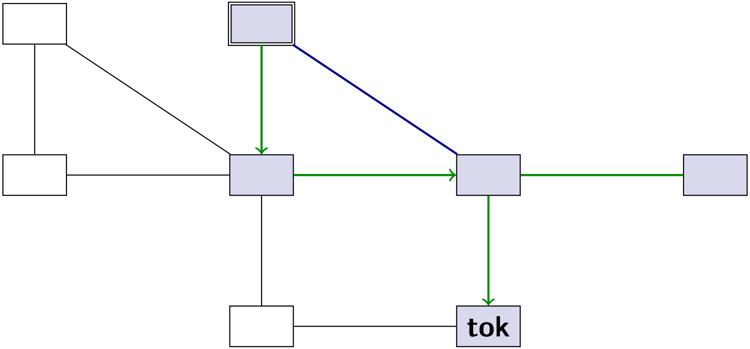
Алгоритм Тарри

Пример выполнения алгоритма Тарри



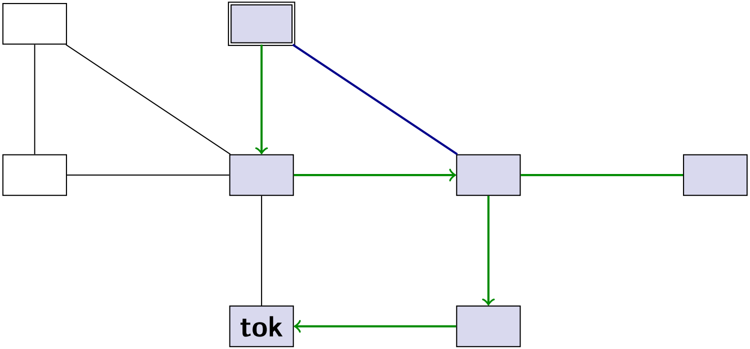
Алгоритм Тарри

Пример выполнения алгоритма Тарри



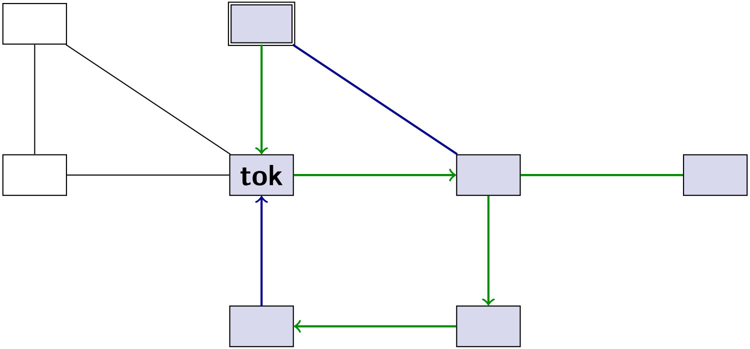
Алгоритм Тарри

Пример выполнения алгоритма Тарри



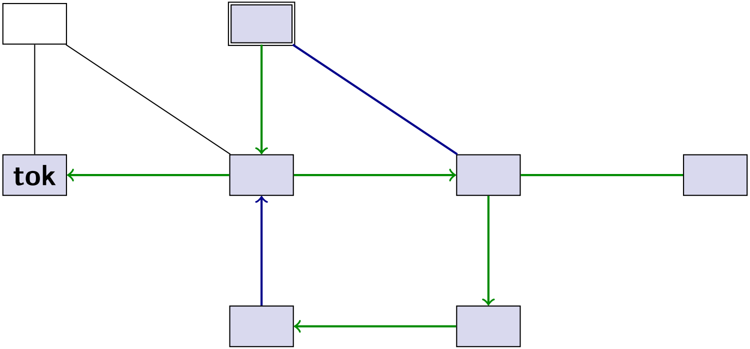
Алгоритм Тарри

Пример выполнения алгоритма Тарри



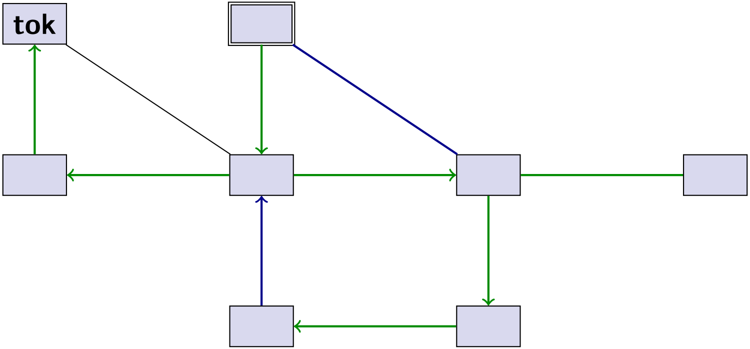
Алгоритм Тарри

Пример выполнения алгоритма Тарри



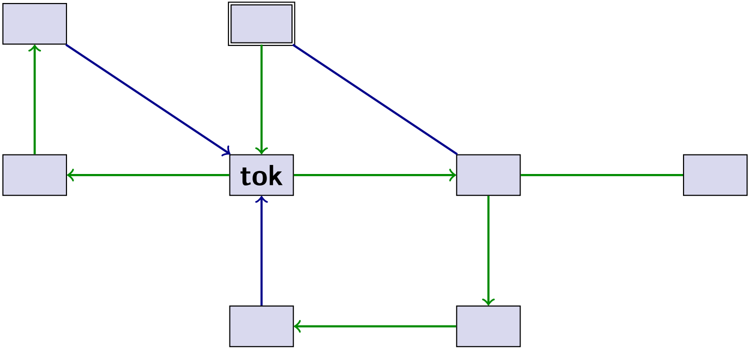
Алгоритм Тарри

Пример выполнения алгоритма Тарри



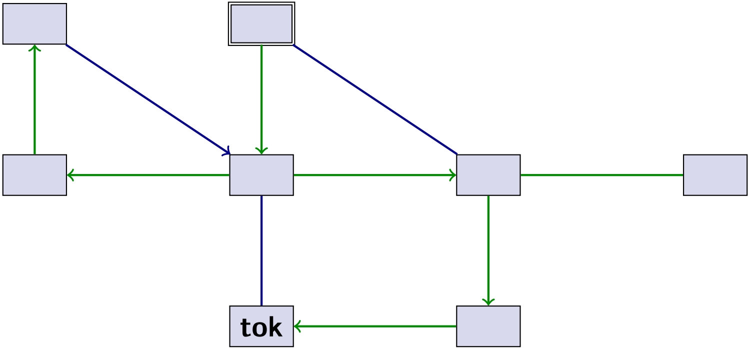
Алгоритм Тарри

Пример выполнения алгоритма Тарри



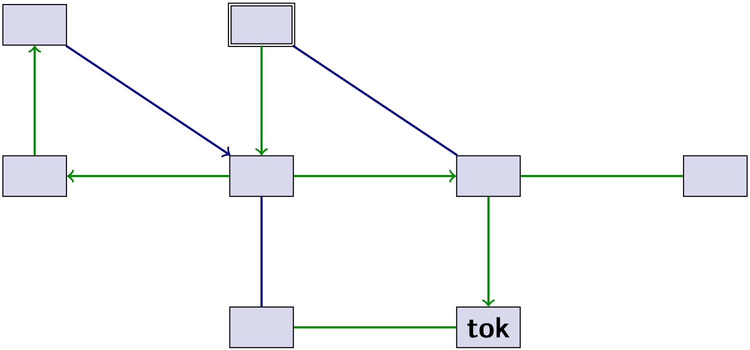
Алгоритм Тарри

Пример выполнения алгоритма Тарри



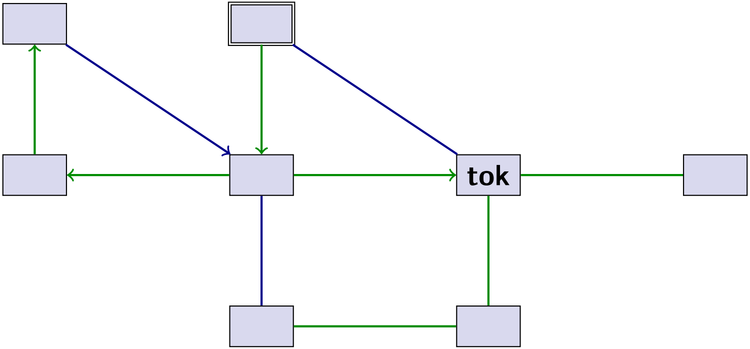
Алгоритм Тарри

Пример выполнения алгоритма Тарри



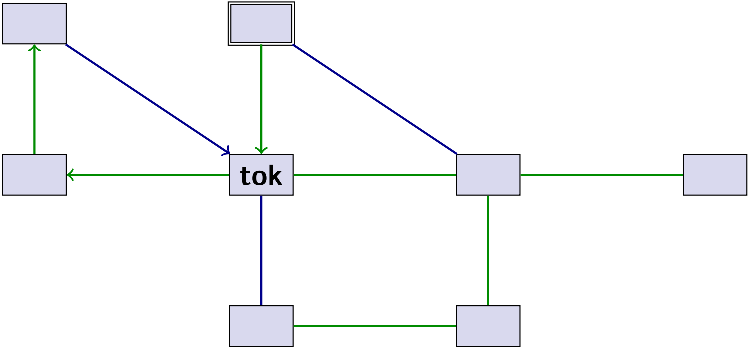
Алгоритм Тарри

Пример выполнения алгоритма Тарри



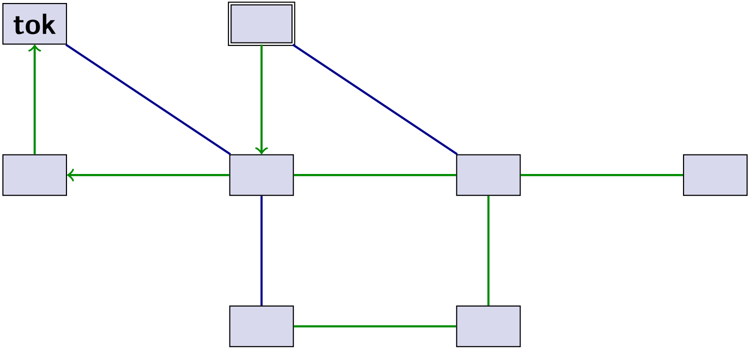
Алгоритм Тарри

Пример выполнения алгоритма Тарри



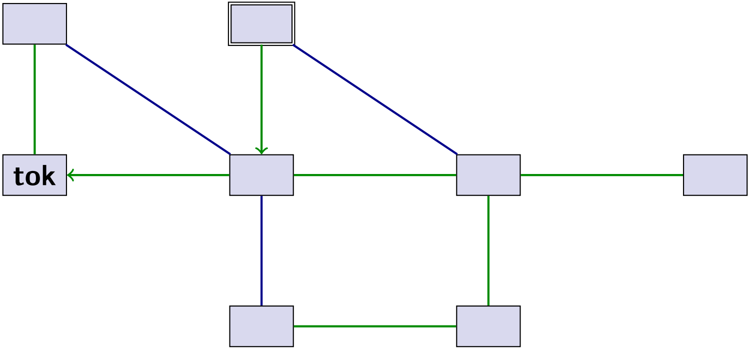
Алгоритм Тарри

Пример выполнения алгоритма Тарри



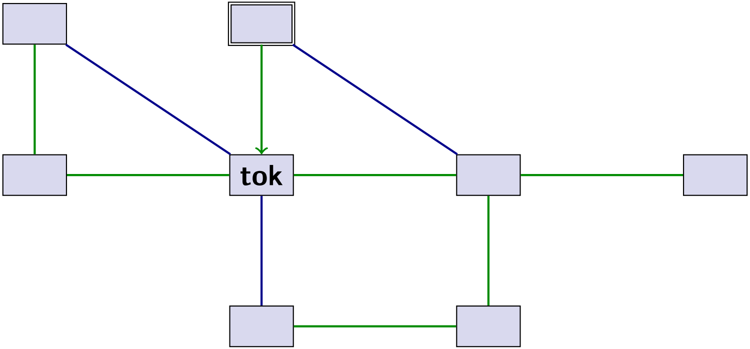
Алгоритм Тарри

Пример выполнения алгоритма Тарри



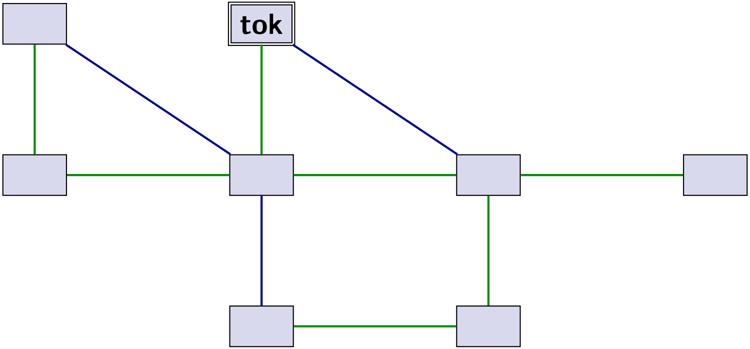
Алгоритм Тарри

Пример выполнения алгоритма Тарри



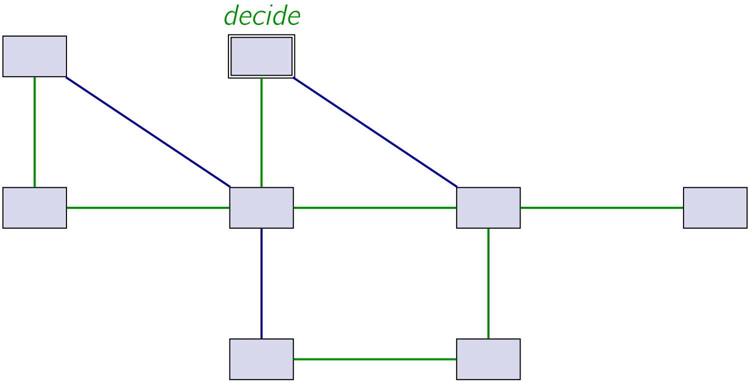
Алгоритм Тарри

Пример выполнения алгоритма Тарри



Алгоритм Тарри

Пример выполнения алгоритма Тарри



Алгоритм Тарри

Переменные в узле p:

- $used_p[q] : bool = \mathbb{f}$; для каждого $q \in Neigh_p$
- $parent_p : V \cup \{\perp\} = \perp$

Процедура Get_p приёма сообщения узлом p с ответными действиями:

1. $receive(\mathbf{tok}) \leftarrow q_0$ для любого $q_0 \in Neigh_p$
2. Если $parent_p = \perp$: $parent_p := q_0$;
3. Если $\underset{q \in Neigh_p}{\&} used_p[q]$:
   3.1 $decide$
4. Иначе, если $\bigvee_{q \in Neigh_p} (q \neq parent_p \,\&\, \neg used_p[q])$:
   4.1 Выбрать $q \in Neigh_p \setminus \{parent_p\}$, такой что $\neg used_p[q]$
   4.2 $used_p[q] := \mathbb{t}$;
   4.3 $send(\mathbf{tok}) \rightarrow q$
5. Иначе:
   5.1 $used_p[parent_p] := \mathbb{t}$;
   5.2 $send(\mathbf{tok}) \rightarrow parent_p$

Алгоритм Тарри

Код последователя p:

1. В бесконечном цикле:
   1.1 Get_p

Код инициатора p:

1. $parent_p := p$;
2. Выбрать $q \in Neigh_p$
3. $used_p[q] := \mathbb{t}$;
4. $send(\mathbf{tok}) \to q$
5. В бесконечном цикле:
   5.1 Get_p

Алгоритм Тарри

Теорема. **Для любого связного графа топологии алгоритм Тарри — это распределённый алгоритм обхода**

Доказательство.

Завершаемость

Фишка передаётся не более одного раза по одному каналу (по одному разу в каждую сторону)

Следовательно, всего при работе алгоритма передаётся не более $2|E|$ фишек, и вычисление алгоритма конечно

Принятие решения

Каждый узел отправляет фишку каждому соседу не более одного раза

Значит, каждый узел принимает фишку от каждого соседка не более одного раза

Если последователь получил фишку и после этого ещё не отправил, то фишка получена на один раз больше, чем отправлена — то есть не все каналы использованы, и фишка отправляется в неиспользованный канал

Следовательно, принять решение, получив фишку, может только инициатор

Алгоритм Тарри

Доказательство. *Полнота покрытия*

Наблюдение 1: до принятия решения через каждый канал, инцидентный инициатору, фишка отправлялась по одному разу в каждую сторону

Чтобы инициатор принял решение, требуется перед этим отправить фишку по очереди во все инцидентные каналы

Отправка и приём фишки чередуются, начиная с отправки

Следовательно, перед принятием решения фишка получена инициатором по одному разу из каждого канала

Алгоритм Тарри

Доказательство. *Полнота покрытия*

Наблюдение 2: если последователь p хотя бы раз принимает фишку, то фишка передаётся по всем каналам, инцидентным p, по одному разу в каждом направлении (до принятия решения)

Предположим от противного, что это не так: существуют последователь p и инцидентный ему канал, в который фишка отправляется менее двух раз

Рассмотрим такого последователя p, принявшего первую фишку раньше всех в вычислении

По выбору p, значение $parent_p$ определено и в каждый канал, инцидентный $parent_p$, фишка передавалась по разу в каждую сторону

В частности, это означает, что p отправляет фишку узлу $parent_p$

Согласно ограничениям алгоритма, такое возможно только в том случае, если перед этим фишка была отправлена в каждый другой канал и принята из каждого другого канала

Следовательно, фишка в совокупности пересылается по одному разу через каждый канал, инцидентный p, в каждую сторону (*противоречие*)

Алгоритм Тарри

Доказательство. *Полнота покрытия*

Наблюдение 3: до принятия решения каждый последователь хотя бы раз принял фишку

Это следует из того, наблюдений 1 и 2: из них в совокупности следует, что фишка отправляется в каждый канал по одному разу в каждом направлении, а значит, и принимается хотя бы раз из каждого канала в каждом направлении

При этом, по устройству алгоритма, каждое действие, выполняющееся хронологически раньше действия инициатора, предшествует ему согласно порядку $\preceq$ ▼

Обход в глубину

Пусть заданы граф $\Gamma = (V, E)$ и его остовное дерево $T = (V, E_T)$ с выделенным корнем v (инициатором)

Тогда стягивающее ребро — это ребро Γ, не входящее в T

Записью $T[p]$ для узла p будем обозначать множество вершин поддерева, произрастающего из p (включая p)

Запиью $Anc[p]$ будем обозначать все вершины на пути из корня в p (не включая p)

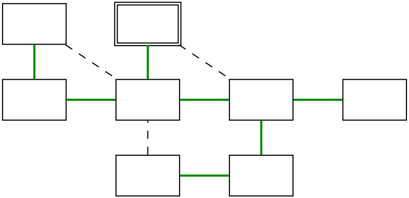
Заметим, что $q \in T[p] \Leftrightarrow p \in Anc[q]$

Дерево поиска в глубину графа Γ — это остовное дерево T, такое что для каждого стягивающего ребра $p - q$ верно $q \in T[p] \vee q \in Anc[p]$

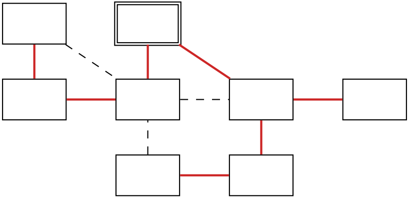
То есть в дереве поиска в глубину любые две несмежные вершины связаны отношением потомства

Обход в глубину

Пример: зелёным цветом выделены рёбра дерева обхода в глубину, а остальные рёбра (пунктирные) — стягивающие



Пример: красным цветом выделены рёбра остовного дерева, не являющегося деревом обхода в глубину



Обход в глубину

Задача 1. Приведите пример вычисления алгоритма Тарри, по завершении которого рёбрами $(p, parent_p)$ задаётся остовное дерево, не являющееся деревом обхода в глубину

Обход в глубину

Классический распределённый обход в глубину получается из алгоритма Тарри добавлением третьего ограничения:

О3. Если это не противоречит (О1) и (О2), то после получения фишки она отправляется в тот же канал, из которого была получена

Корректность обхода в глубину (то, что это действительно распределённый алгоритм обхода) следует из корректности алгоритма Тарри

Задача 2. Докажите, что классическим распределённым обходом в глубину действительно строится дерево обхода в глубину