

Распределённые алгоритмы

mk.cs.msu.ru → Лекционные курсы → Распределённые алгоритмы

Блок 29

Алгоритм Тарри

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

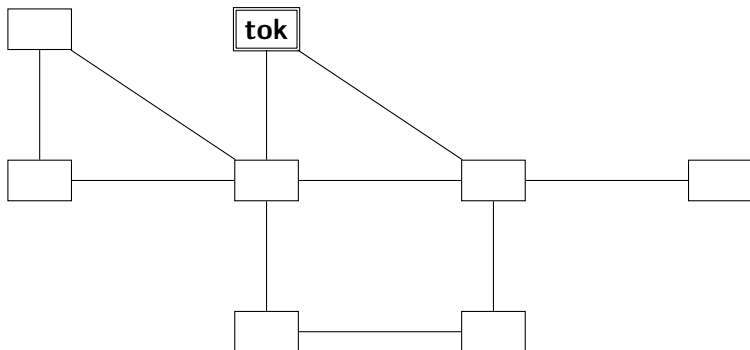
Алгоритм Тарри (далее — \mathfrak{T}) — это РАО, предназначенный для произвольного неориентированного связного графа топологии, не использующий сообщения кроме фишек и устроенный произвольно в рамках следующих ограничений (правил):

01. Фишку нельзя отправить одному соседу дважды
02. Последователь не может отправить фишку тому, от кого её принял первым действием, пока не отправит её остальным соседям
03. Узел принимает решение тогда и только тогда, когда не может отправить фишку соседу

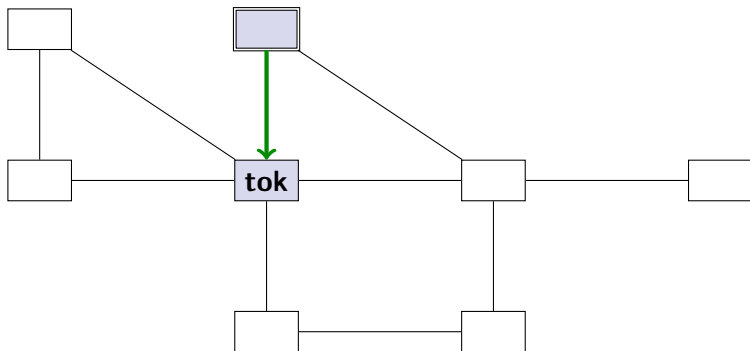
Узел, от которого p принял фишку самым первым своим действием, будем называть **родителем** узла p (у инициатора нет родителя)

Согласно **одному из утверждений в блоке 24**, если признать, что алгоритм Тарри — волновой, то граф, составленный из всех вершин и дуг (p, q) , где p — последователь и q — его родитель, является **остовным деревом** сети

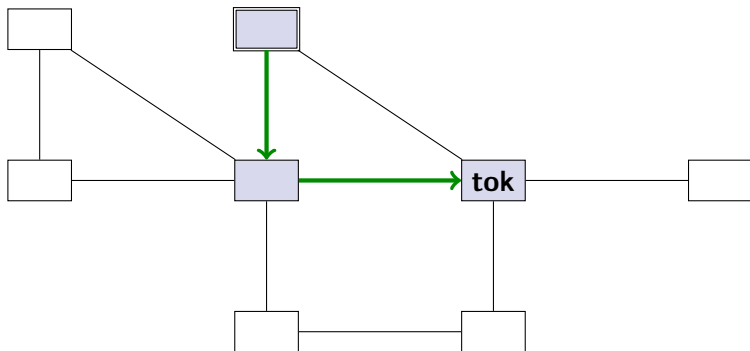
Пример выполнения ξ



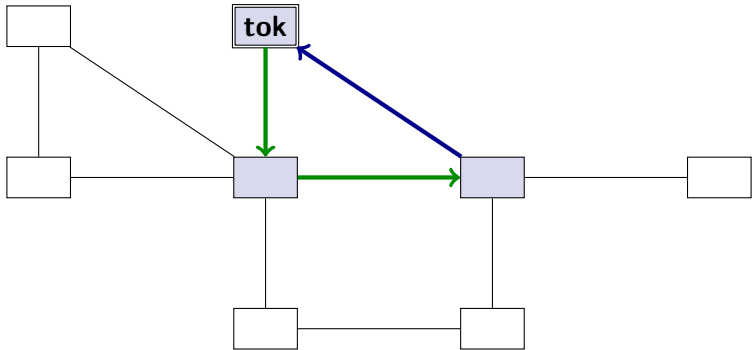
Пример выполнения ξ



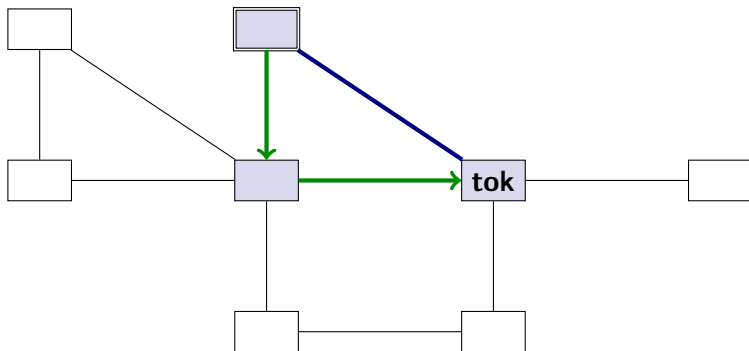
Пример выполнения ξ



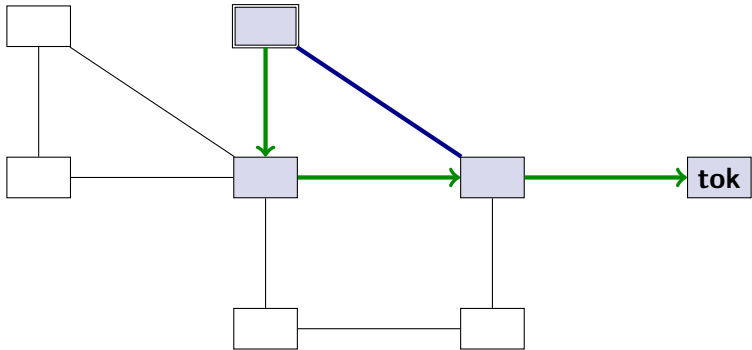
Пример выполнения Σ



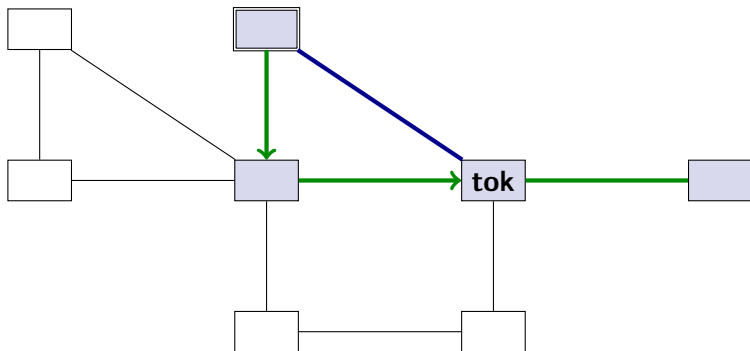
Пример выполнения Σ



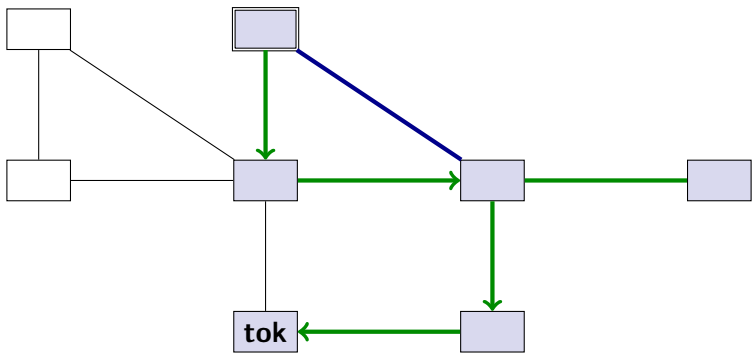
Пример выполнения Σ



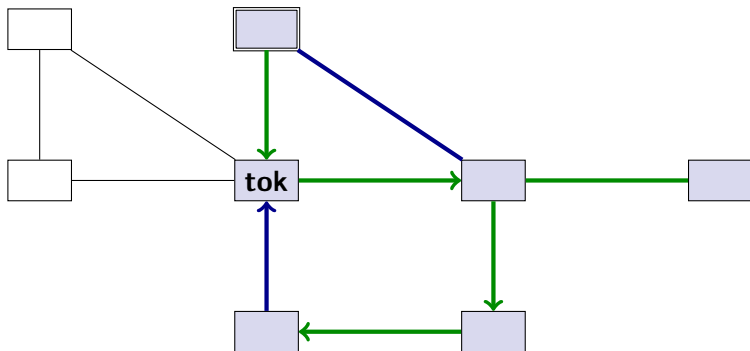
Пример выполнения Σ



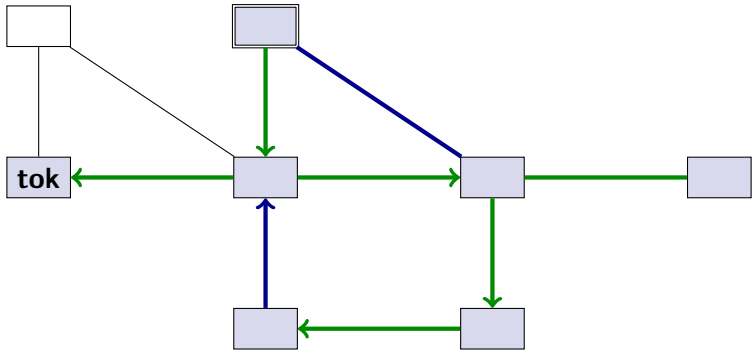
Пример выполнения Σ



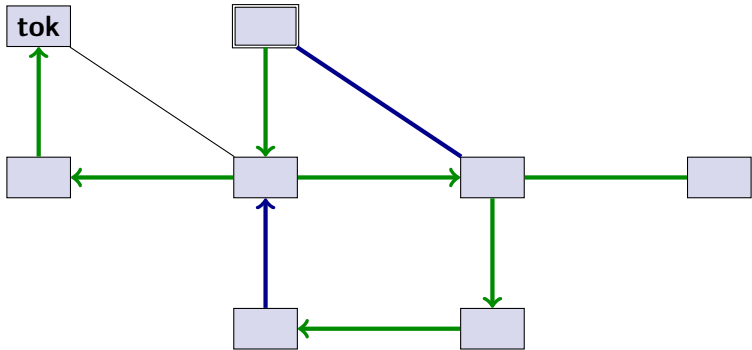
Пример выполнения ξ



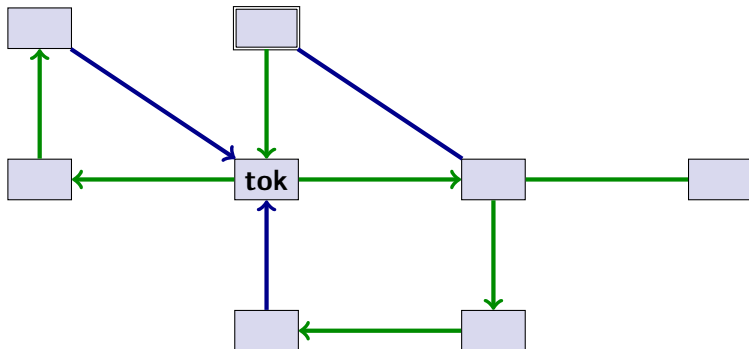
Пример выполнения ξ



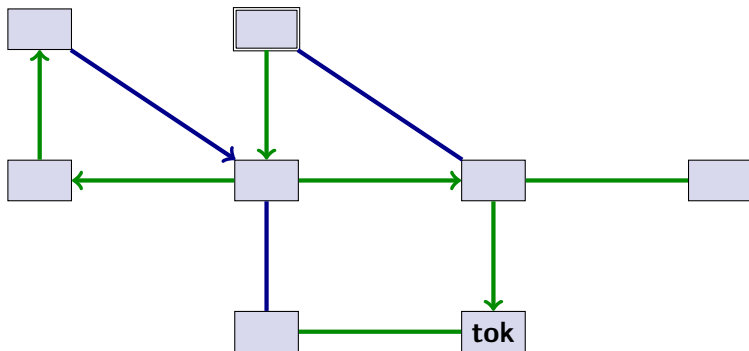
Пример выполнения ξ



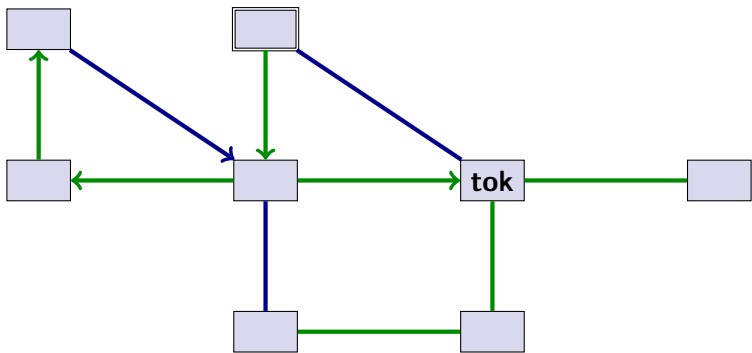
Пример выполнения ξ



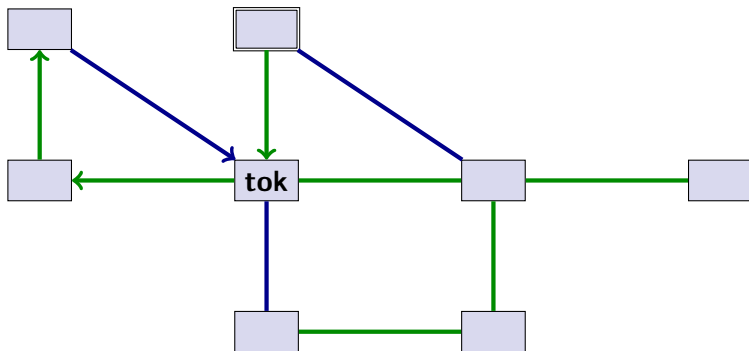
Пример выполнения ξ



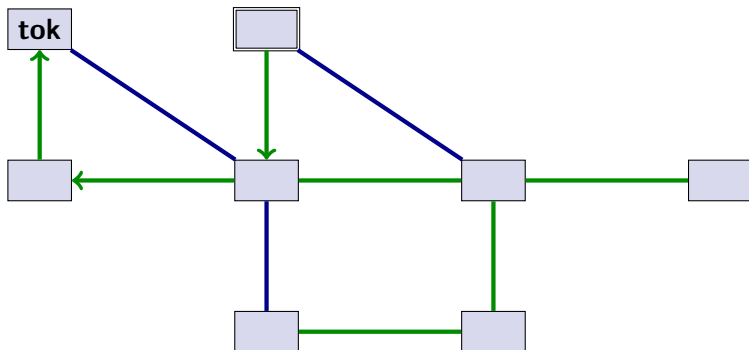
Пример выполнения ξ



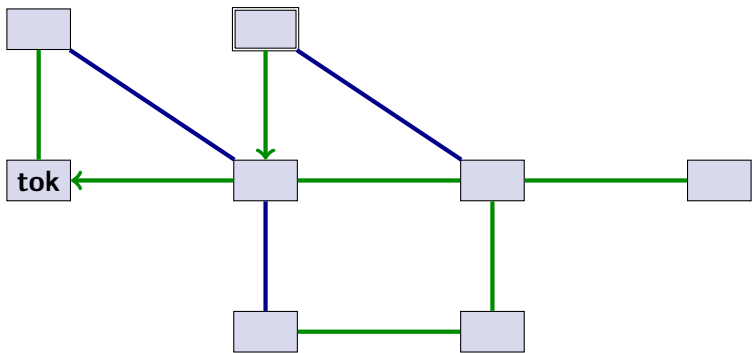
Пример выполнения ξ



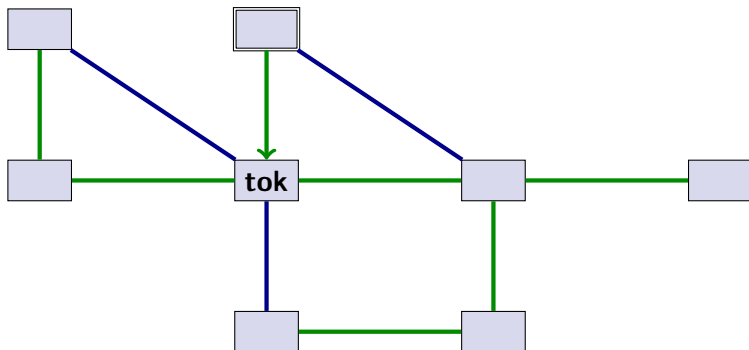
Пример выполнения ξ



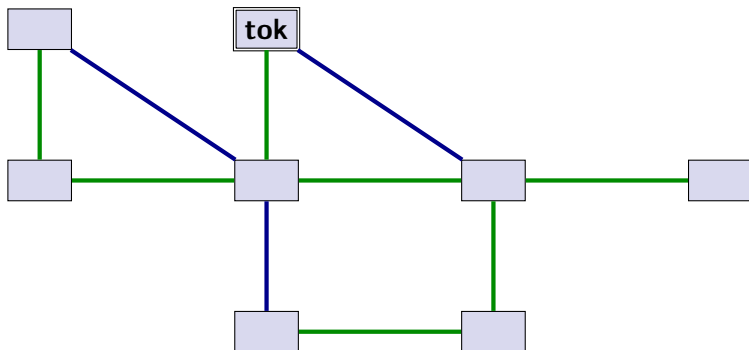
Пример выполнения ξ



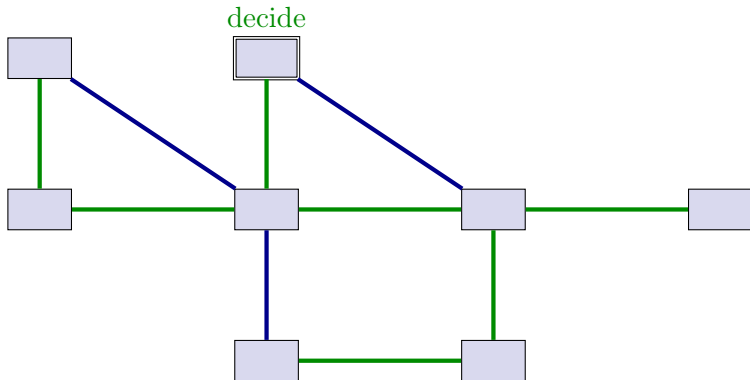
Пример выполнения ξ



Пример выполнения ξ



Пример выполнения Σ



Переменные в узле p :

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

Процедура Get_p приёма и обработки сообщения узлом p :

1. receive $_{q_0}(\mathbf{tok})$ для произвольного $q_0 \in Neigh_p$
2. Если $parent_p = \perp$, то: $parent_p := q_0$;
3. Если $(\forall q \in Neigh_p : used_p[q])$, то: *decide*
4. Если не (3) и $(\exists q \in Neigh_p \setminus \{parent_p\} : \neg used_p[q])$, то:
 - 4.1 Произвольно выбрать q такого вида, как написано в (4)
 - 4.2 $used_p[q] := \mathbb{t}$;
 - 4.3 send $_q(\mathbf{tok})$
5. Если не (3) и не (4):
 - 5.1 $used_p[parent_p] := \mathbb{t}$;
 - 5.2 send $_{parent_p}(\mathbf{tok})$

Далее (4.1)–(4.3) считаются одним действием (4.X), а (5.1)–(5.2) — одним действием (5.X)

Код последователя p :

1. В бесконечном цикле:
 - 1.1 Get_p

Код инициатора p :

1. $parent_p := p$;
2. Произвольно выбрать $q \in Neigh_p$
3. $used_p[q] := \text{t}$;
4. $send_q(\mathbf{tok})$
5. В бесконечном цикле:
 - 5.1 Get_p

Утверждение (смысл $used_p$). В любой конфигурации γ любого вычисления \mathcal{E} р.а. \mathfrak{T} верно:

$$used_p[q] = \mathfrak{t} \text{ в } \gamma$$

\Leftrightarrow

узел p отправил хотя бы одну фишку узлу q в \mathcal{E} до γ

Утверждение (уникальность отправки). В любом вычислении р.а. \mathfrak{T} для любых узла p и его соседа q узел p

- ▶ отправляет фишку узлу q не более одного раза и
- ▶ принимает фишку от узла q не более одного раза

Утверждение (завершаемость). Все вычисления р.а. \mathfrak{T} конечны

Утверждение (баланс фишек). В любой конфигурации любого вычисления р.а. \mathcal{T} для любого узла p непосредственно перед началом и непосредственно после завершения выполнения действий Get_p для количества фишек, отправленных узлом p (S_p) и принятых им (R_p), верно следующее:

- ▶ Если p — последователь, то $S_p = R_p$
- ▶ Если p — инициатор, не принявший решение, то $S_p = R_p + 1$

Утверждение (возможность переправки последователем). В любом вычислении р.а. \mathcal{T} любой последователь p непосредственно после выполнения приёма (1) в Get_p имеет значение $used_p[q] = \mathbb{f}$ хотя бы для одного соседа q

Утверждение (принятие решения инициатором). В любом вычислении р.а. \mathcal{T} рано или поздно принимается решение, и оно принимается только инициатором

Утверждение (покрытие соседей). В любом вычислении р.а. Σ ко времени принятия решения инициатор и каждый последователь, принимающий хотя бы одну фишку,

- ▶ отправляет каждому соседу ровно одну фишку и
- ▶ принимает от каждого соседа ровно одну фишку

Утверждение (вовлечение всех узлов). В любом вычислении р.а. Σ ко времени принятия решения каждый последователь принимает хотя бы одну фишку

Утверждение (линейная причинность). В любом вычислении р.а. Σ любое действие, выполняющееся хронологически позже, является следствием любого действия, выполняющегося хронологически раньше

Утверждение (полнота покрытия). В любом вычислении р.а. Σ принятие решения является следствием хотя бы одного действия в каждом узле

Д.з. 1. Доказать по одному утверждению с каждого из слайдов 6–8

Теорема. Алгоритм Тарри — это распределённый алгоритм обхода