

Распределённые алгоритмы

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

Блок 31

Алгоритм Авербаха

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2025, февраль–май

В классическом алгоритме распределённого обхода в глубину:

- ▶ Согласно **покрытию соседей** и **вовлечению всех узлов** в алгоритме Тарри, коммуникационная сложность — это дважды число каналов в топологии
- ▶ Согласно **линейной причинности** алгоритма Тарри, **сложность по времени** равна коммуникационной

В **алгоритме Авербаха** (\mathcal{A}) за счёт обмена вспомогательными сообщениями

- ▶ увеличивается коммуникационная сложность, но при этом
- ▶ уменьшается время работы алгоритма

Обмен вспомогательными сообщениями в \mathcal{A} устроен так: когда становится посещённым (принимает первую фишку), узел

- ▶ отправляет сообщение **vis** всем соседям и
- ▶ принимает от них всех отклик **ack**

Каждый узел обновляет свои знания о посещённых соседях согласно сообщениям **vis** и не передаёт фишку посещённым узлам

Переменные узла p :

- ▶ $used_p[q] : \text{bool}/\text{f}$ для каждого $q \in \text{Neigh}_p$
- ▶ $parent_p : \text{Neigh}_p \cup \{\perp\}/\perp$

Процедура оповещения соседей $Vis_p(q_0)$ узлом p :

1. $parent_p := q_0;$
2. Для всех $q \in \text{Neigh}_p \setminus \{parent_p\}$: $\text{send}_q(\text{vis})$
3. Для всех $q \in \text{Neigh}_p \setminus \{parent_p\}$: $\text{receive}_q(\underline{\text{ack}})$

Процедура Get_p приёма и обработки фишки:

1. $\text{receive}_{q_0}(\text{tok})$ для произвольного $q_0 \in \text{Neigh}_p$
2. Если $\text{parent}_p = \perp$: $\text{Vis}_p(q_0)$
3. Если p — инициатор и $(\forall q \in \text{Neigh}_p : \text{used}_p[q])$, то: decide
4. Если не (3) и $(\exists q \in \text{Neigh}_p \setminus \{\text{parent}_p\} : \neg \text{used}_p[q])$, то:
 - 4.1 Если $\text{parent}_p \neq q_0$ и $\neg \text{used}_p[q_0]$: $q := q_0$;
Иначе: выбрать $q \in \text{Neigh}_p \setminus \{\text{parent}_p\}$, такой что $\neg \text{used}_p[q]$
 - 4.2 $\text{used}_p[q] := \text{t}$;
 - 4.3 $\text{send}_q(\text{tok})$
5. Если не (3) и не (4):
 - ▶ $\text{used}_p[\text{parent}_p] := \text{t}$;
 - ▶ $\text{send}_{\text{parent}_p}(\text{tok})$

Код последователя p : в бесконечном цикле недетерминированно выбирается и выполняется одна из двух процедур:

1. Если в коммуникационной подсистеме есть хотя бы одна фишка, адресованная p :
 Get_p
2. Если в коммуникационной подсистеме содержится хотя бы одно сообщение **vis**, адресованное p :
 - 2.1 $receive_q(\text{vis})$ для произвольного $q \in Neigh_p$
 - 2.2 $used_p[q] := \text{t};$
 - 2.3 $send_q(\text{ack})$

Код инициатора p :

1. $Vis_p(p)$
2. Произвольно выбрать $q \in Neigh_p$
3. $used_p[q] := \text{t};$
4. $send_q(\text{tok})$
5. Выполнить код последователя

Теорема (Д.з. 1, трудное). Алгоритм Авербаха — это распределённый алгоритм обхода, строящий дерево обхода в глубину

Теорема (Д.з. 2).

Алгоритм Авербаха имеет коммуникационную сложность $4|E|$ и сложность по времени $(4|V| - 2)$