Распределенные алгоритмы

ЛЕКТОР: В.А. Захаров

Лекция 12.

Отказоустойчивые алгоритмы.

Модели неисправностей.

Задачи принятия решения.

Задача консенсуса.

Неразрешимые случаи задачи консенсуса.

Разрешимые случаи задачи консенсуса.

Задача консенсуса для синхронных систем.

Отказоустойчивые алгоритмы.

С увеличением числа компонентов распределенной системы увеличивается и вероятность того, что некоторые из этих компонентов будут выходить из строя при выполнении распределенного алгоритма.

Компьютеры становятся все более надежными, и вероятность возникновения ошибок в каждой отдельной вычислительной машине непрерывно снижается.

Но вероятность неисправности, которая может произойти в какой-нибудь части распределенной системы, возрастает с увеличением числа компонентов системы.

Во избежание необходимости перезапускать алгоритм всякий раз, когда случается сбой в работе одного из компонентов системы, алгоритмы должны быть устроены так, чтобы они могли правильным образом справляться с неполадками.

Отказоустойчивые алгоритмы.

Выделяют два подхода к построению отказоустойчивых систем.

1. В робастных алгоритмах любой процесс совершает каждый шаг вычисления достаточно осторожно, оценивая достоверность данных, полученных от других процессов. Поэтому, несмотря на сбои в системе, исправные процессы совершают только правильные действия.

2. В стабилизирующихся алгоритмах исправные процессы могут выходить из строя, но при этом гарантируется, что правильное выполнение алгоритма будет восстановлено из любой конфигурации, как только правильная работа процессов возобновится.

Отказоустойчивые алгоритмы.

Стабилизирующиеся алгоритмы служат для защиты от кратковременных неисправностей . Примером может служить система управления космическим кораблем, которая подверглась воздействию интенсивного потока космических лучей. Как только нормальные условия восстанавливаются, процессы возобновляют свою работу в соответствии с их программами, хотя из-за кратковременного сбоя в работе глобальное состояние системы может быть произвольным. Стабилизирующие свойства призваны гарантировать постепенное восстановление требуемого поведения.

Робастные алгоритмы предназначены для защиты от устойчивых неисправностей . Процессы, оставшиеся исправными, поддерживают правильное (хотя и менее эффективное) поведение системы, пока проводится ее ремонт и реконфигурация. Таким образом, робастные алгоритмы нужно использовать тогда, когда недопустимо даже кратковременное прерывание обслуживания.

Модели неисправностей

Основные допущения.

1. Только процессы могут выходить из строя, а каналы остаются исправными. Неисправность канала связи можно промоделировать, если объявить неисправным один из процессов, инцидентных этому каналу (так называемая ошибка пропуска).
2. Если один исправный процесс отправляет сообщение другому исправному процессу, то это сообщение обязательно будет доставлено адресату спустя конечный промежуток времени.
3. Каждый процесс может отправить сообщение любому другому процессу.

Модели неисправностей

1. Изначально бездействующие процессы. Процесс называется изначально бездействующим, если его локальный алгоритм не совершает ни одного действия.
2. Модель выхода процесса из строя. Процесс выходит из строя, если до некоторого момента времени его локальный алгоритм работает правильно, а затем не выполняет ни одного действия.
3. Византийское (предательское) поведение. Процесс называется византийским (или предательским), если он выполняет сколь угодно много действий, которые не предписаны его локальным алгоритмом. В частности, византийский процесс может отправлять сообщения с произвольным содержанием.

Требования корректности для робастных алгоритмов выдвигаются только в отношении локальных состояний (или выходных данных) исправных процессов.

Задачи принятия решения

При изучении робастных алгоритмов основное внимание уделяется исследованию проблем принятия решения , в которых требуется, чтобы каждый исправный процесс в качестве результата своей работы осуществлял необратимую запись некоторого значения, определяющего решение, принятое процессом.

Требуемое решение обычно находится в очень простой зависимости от исходных данных, которые получают процессы.

К алгоритмам принятия решения обычно предъявляются следующие требования:

завершаемость ,

непротиворечивость ,

невырожденность .

Задачи принятия решения

Общие требования к алгоритмам.

1. Требование завершаемости. Все исправно работающие процессы должны завершить свою работу, т. е. рано или поздно осуществить запись принятого решения в качестве выходных данных.
2. Требование непротиворечивости. Должны быть выполнены некоторые отношения между решениями, принятыми разными процессами. В простейшем случае требуется, чтобы все решения были одинаковыми; в таком случае говорят о проблеме консенсуса .
3. Требование невырожденности. Различные выполнения алгоритма могут приводить по крайней мере к двум существенно отличным друг от друга ответам (например, в случае консенсуса алгоритм должен иметь выполнения, которые приводят к записи «0», и выполнения, которые приводят к записи «1»

Задачи принятия решения

1) **Согласование транзакций в базах данных.**
   В распределенных базах данных транзакция, в которой участвуют несколько сайтов, может быть выполнена либо **во всех** сайтах, либо **ни в одном** из них.
   Поэтому после объявления о преобразовании данных в этих сайтах каждый сайт должен определить, может ли это преобразование быть осуществлено локально, и принять решение либо об осуществлении преобразования, либо об отказе от него.
   Таким образом, все исправно работающие сайты должны согласовать вопрос о том, следует ли им всем совершить преобразование или отказаться от него.

Задачи принятия решения

2) **Согласование вычисленных данных.**

В некоторых системах для повышения надежности вычисления дублируются.

Влияние дефектного сенсора, отправляющего разные значения разным процессорам, должно быть устранено за счет выполнения алгоритма достижения консенсуса между процессорами.

Входными данными для каждого процессора служат значения, поступающие от сенсора, и каждый исправный процессор принимает решение на основе одних и тех же данных, которые будут использованы в последующем вычислении.

В качестве выходного результата выступало то значение, которое поступает на большее число процессоров, или, в крайнем случае, которое возникает хотя бы на одном входе.

Задачи принятия решения

3) **Избрание лидера.**
   В задаче избрания лидера требуется, чтобы один из процессов принял решение стать лидером, а остальные (исправные) процессы пришли к решению отказаться от лидерства.
   Эта задача становится нетривиальной, если нескольлко разных процессов имеют потенциальную возможность стать лидером.

Задача консенсуса

Процесс p имеет входную переменную x_p , которая доступна только для считывания, и выходной регистр y_p , который доступен только для однократной записи и в котором вначале содержится значение $\perp$.

Для каждого процесса исходная конфигурация полностью определяется значением переменной x_p .

Процесс p может принять решение , выработав одно из двух значений 0 или 1 и записав его в регистр y_p .

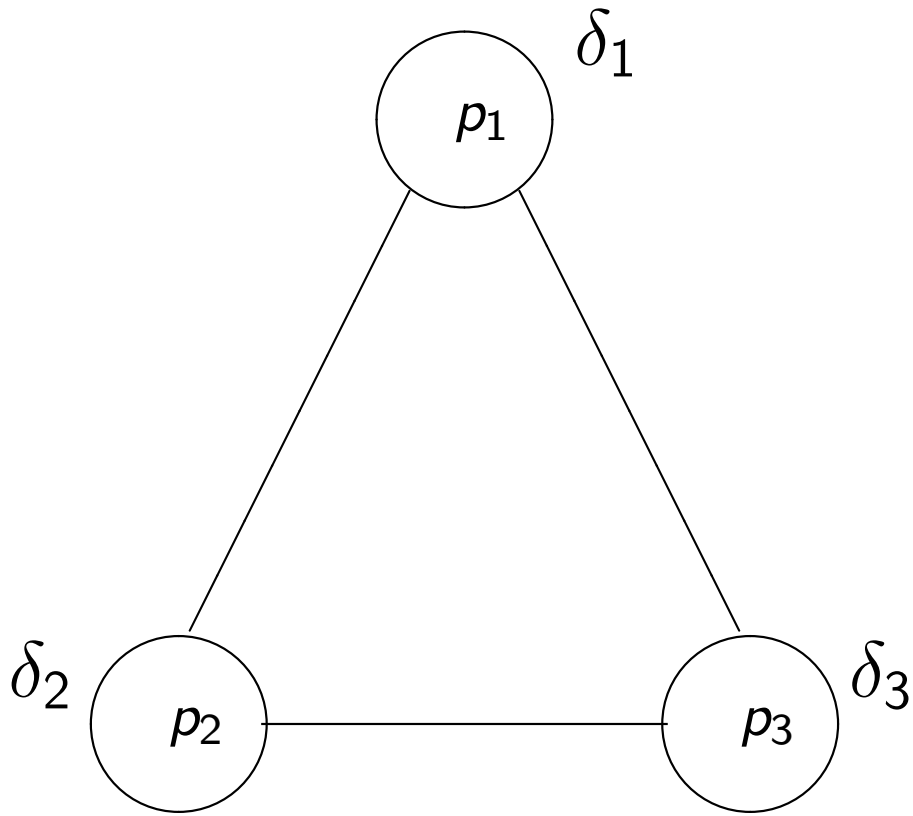
Предполагается, что в исправно работающем процессе происходит бесконечно много событий на протяжении выполнения (справедливые выполнения).

Задача консенсуса

Алгоритмом консенсуса, робастным относительно t неисправностей , называется алгоритм, который удовлетворяет следующим трем требованиям.

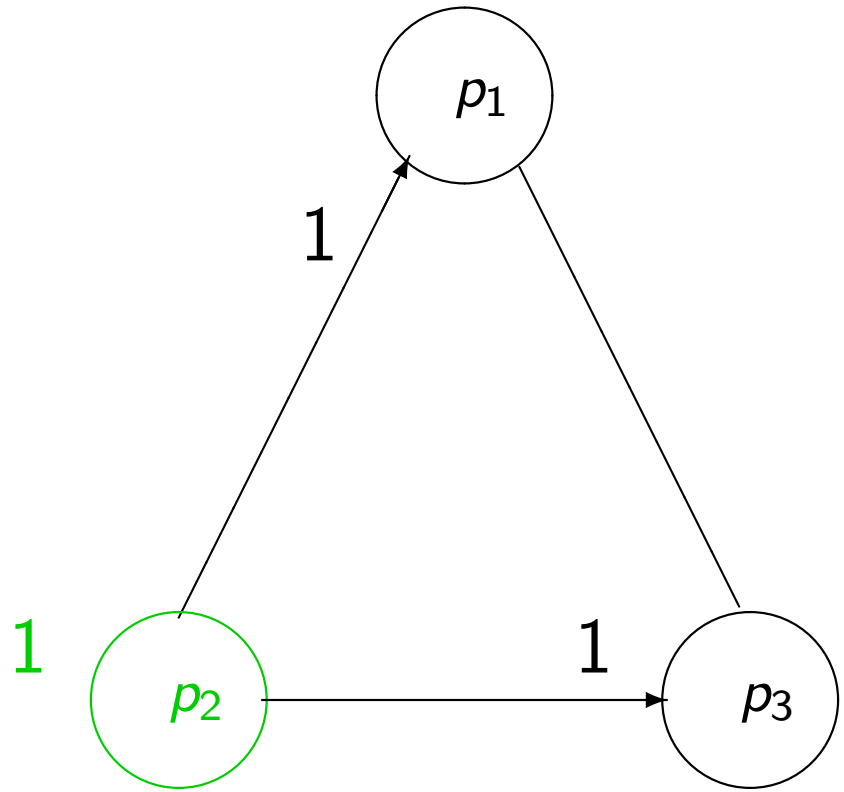
1. **Завершаемость.** В ходе каждого справедливого выполнения с t неисправными процессами все исправно работающие процессы принимают решение.
2. **Непротиворечивость.** Если в достижимой конфигурации для исправных процессов p и q выполняются неравенства $y_p \neq \perp$ и $y_q \neq \perp$, то $y_p = y_q$.
3. **Невырожденность.** Для $v = 0$ и для $v = 1$ существуют достижимые конфигурации, в которых для одного из исправных процессов p выполняется равенство $y_p = v$.

Задача консенсуса



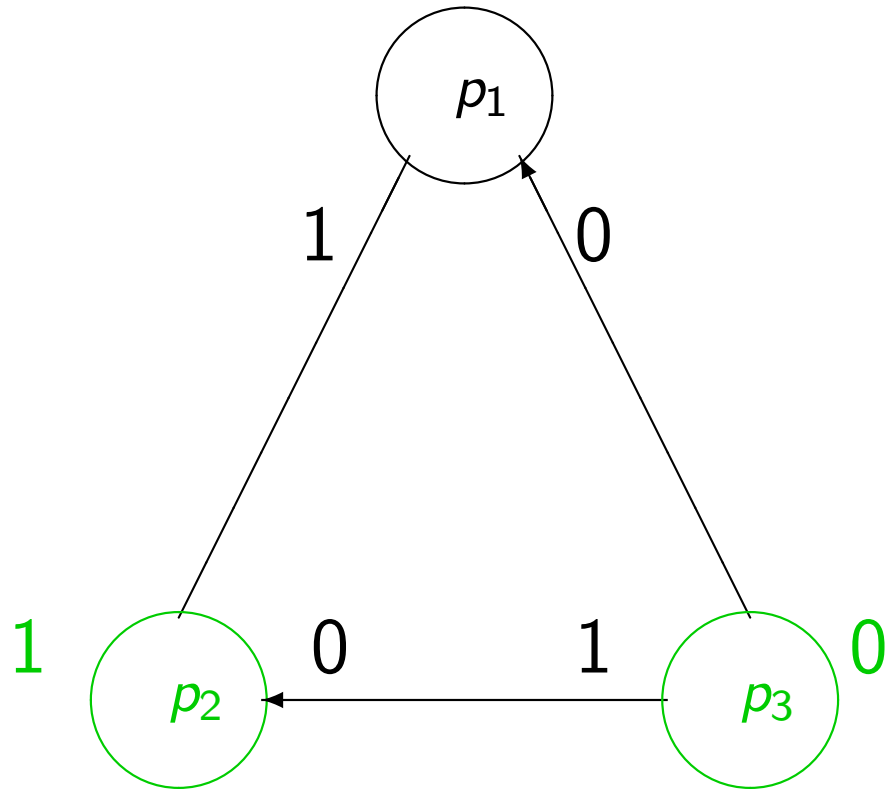
Задача консенсуса

Византийская неисправность



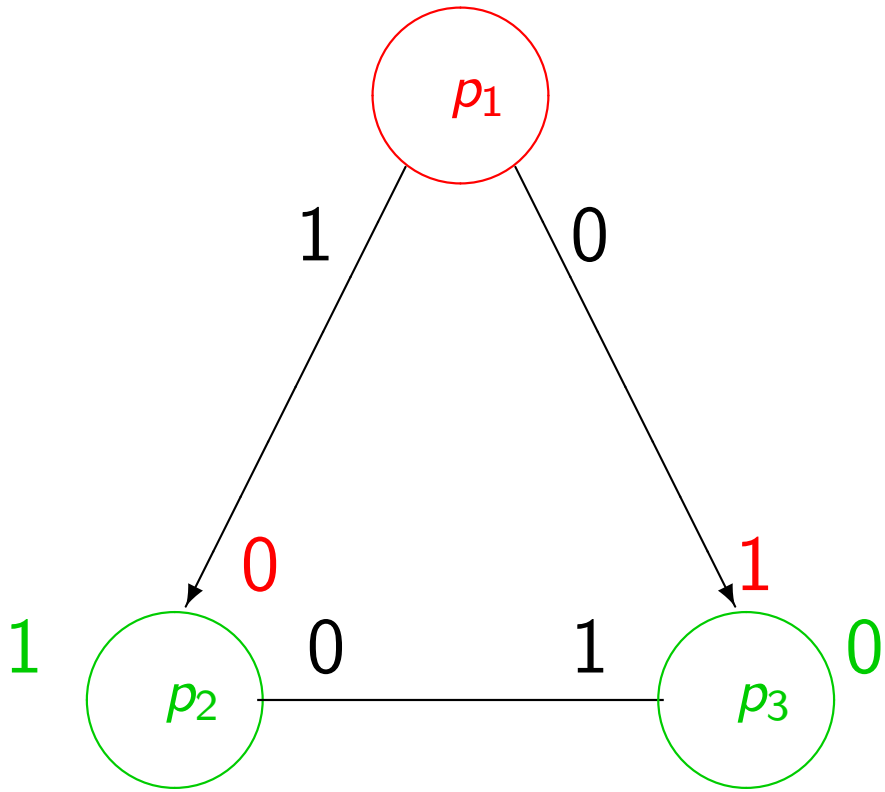
Задача консенсуса

Византийская неисправность



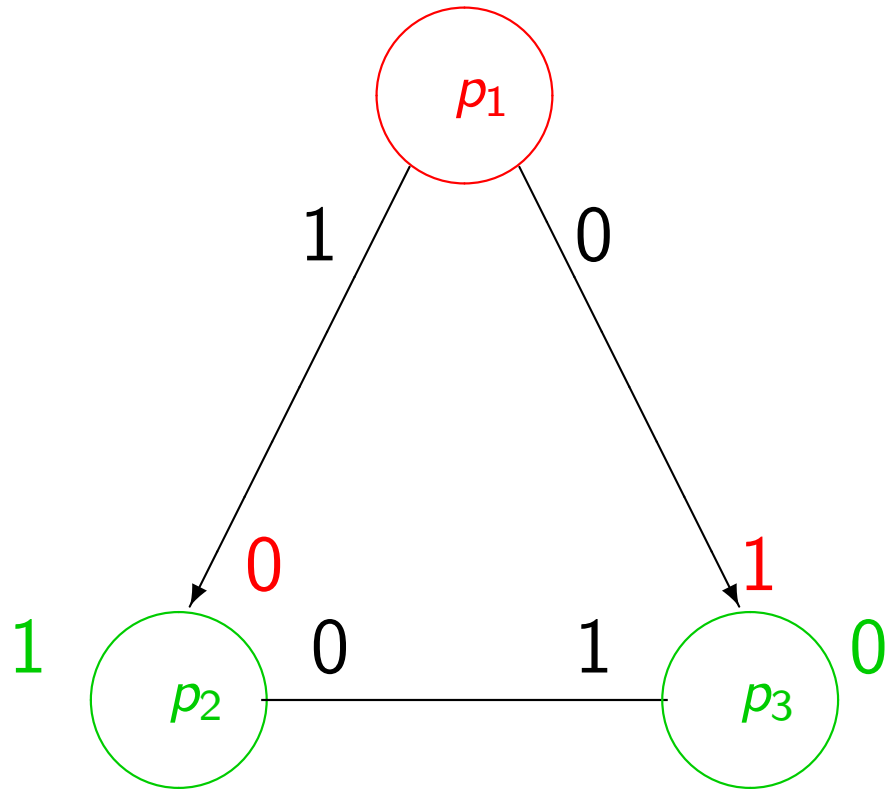
Задача консенсуса

Византийская неисправность



Задача консенсуса

Византийская неисправность



Как видите согласованное решение принять затруднительно!

Невозможность консенсуса

Теорема 1.

Не существует алгоритмов достижения консенсуса, которые были бы одновременно

— асинхронными,

— детерминированными и

— робастными

относительно выхода из строя **одного** процесса.

Невозможность консенсуса

Последовательность событий $\sigma = (e_1, \ldots, e_k)$ применима к конфигурации γ , если e_1 допустимо в γ , e_2 допустимо в $e_1(\gamma)$ и. т. д.

Если в результате будет достигнута конфигурация δ , то будем использовать запись $\gamma \rightsquigarrow^{\sigma} \delta$ или $\sigma(\gamma) = \delta$.

Если $S \subseteq \mathbb{P}$ и σ состоит только из тех событий, которые происходят в процессах множества S , то будем записывать $\gamma \rightsquigarrow_S \delta$.

Невозможность консенсуса

Последовательность событий $\sigma = (e_1, \ldots, e_k)$ применима к конфигурации γ , если e_1 допустимо в γ , e_2 допустимо в $e_1(\gamma)$ и. т. д.
Если в результате будет достигнута конфигурация δ , то будем использовать запись $\gamma \rightsquigarrow^{\sigma} \delta$ или $\sigma(\gamma) = \delta$.
Если $S \subseteq \mathbb{P}$ и σ состоит только из тех событий, которые происходят в процессах множества S , то будем записывать $\gamma \rightsquigarrow_S \delta$.

Утверждение 1.

Пусть последовательности σ_1 и σ_2 применимы к конфигурации γ и ни один процесс не участвует одновременно в σ_1 и σ_2 .
Тогда σ_2 применима к $\sigma_1(\gamma)$, а σ_1 применима к $\sigma_2(\gamma)$ и при этом $\sigma_2(\sigma_1(\gamma)) = \sigma_1(\sigma_2(\gamma))$.

Невозможность консенсуса

Последовательность событий $\sigma = (e_1, \ldots, e_k)$ применима к конфигурации γ , если e_1 допустимо в γ , e_2 допустимо в $e_1(\gamma)$ и. т. д.
Если в результате будет достигнута конфигурация δ , то будем использовать запись $\gamma \rightsquigarrow^{\sigma} \delta$ или $\sigma(\gamma) = \delta$.
Если $S \subseteq \mathbb{P}$ и σ состоит только из тех событий, которые происходят в процессах множества S , то будем записывать $\gamma \rightsquigarrow_S \delta$.

Утверждение 1.

Пусть последовательности σ_1 и σ_2 применимы к конфигурации γ и ни один процесс не участвует одновременно в σ_1 и σ_2 .
Тогда σ_2 применима к $\sigma_1(\gamma)$, а σ_1 применима к $\sigma_2(\gamma)$ и при этом $\sigma_2(\sigma_1(\gamma)) = \sigma_1(\sigma_2(\gamma))$.

Доказательство. Очевидно.

Невозможность консенсуса

В случае $v = 0, 1$ конфигурация γ называется конфигурацией принятия v -решения , если для какого-то процесса p верно равенство $y_p = v$

Невозможность консенсуса

В случае $v = 0, 1$ конфигурация γ называется конфигурацией принятия v -решения , если для какого-то процесса p верно равенство $y_p = v$

Конфигурация γ называется v -валентной , если во всех конфигурациях принятия решения, которые достижимы из γ , принимается решение v .

Невозможность консенсуса

В случае $v = 0, 1$ конфигурация γ называется конфигурацией принятия v -решения , если для какого-то процесса p верно равенство $y_p = v$

Конфигурация γ называется v -валентной , если во всех конфигурациях принятия решения, которые достижимы из γ , принимается решение v .

Конфигурация называется бивалентной , если из нее достижимы как конфигурации принятия 0 -решения, так и конфигурации принятия 1 -решения.

Невозможность консенсуса

В случае $v = 0, 1$ конфигурация γ называется конфигурацией принятия v -решения , если для какого-то процесса p верно равенство $y_p = v$

Конфигурация γ называется v -валентной , если во всех конфигурациях принятия решения, которые достижимы из γ , принимается решение v .

Конфигурация называется бивалентной , если из нее достижимы как конфигурации принятия 0 -решения, так и конфигурации принятия 1 -решения.

Утверждение 2.

Для каждой достижимой конфигурации t -робастного алгоритма для каждого подмножества S , содержащего не менее $N - t$ процессов, существует такая конфигурация принятия решения δ , для которой выполняется соотношение $\gamma \rightsquigarrow_S \delta$.

Невозможность консенсуса

В случае $v = 0, 1$ конфигурация γ называется конфигурацией принятия v -решения , если для какого-то процесса p верно равенство $y_p = v$

Конфигурация γ называется v -валентной , если во всех конфигурациях принятия решения, которые достижимы из γ , принимается решение v .

Конфигурация называется бивалентной , если из нее достижимы как конфигурации принятия 0 -решения, так и конфигурации принятия 1 -решения.

Утверждение 2.

Для каждой достижимой конфигурации t -робастного алгоритма для каждого подмножества S , содержащего не менее $N - t$ процессов, существует такая конфигурация принятия решения δ , для которой выполняется соотношение $\gamma \rightsquigarrow_S \delta$.

Доказательство.

Следствие требования завершаемости

Невозможность консенсуса

Конфигурация γ , возникающая при выполнении t -робастного протокола, называется разветвлением , если существуют множество T , состоящее не более чем из t процессов, а также такие конфигурации γ_0 и γ_1 , что $\gamma \rightsquigarrow_T \gamma_0$, $\gamma \rightsquigarrow_T \gamma_1$ и при этом γ_v является v -валентной конфигурацией.

Достижимость разветвления означает, что тип решения, которое еще предстоит достичь, зависит только от тех процессов, которые вполне способны выйти из строя.

Невозможность консенсуса

Конфигурация γ , возникающая при выполнении t -робастного протокола, называется разветвлением , если существуют множество T , состоящее не более чем из t процессов, а также такие конфигурации γ_0 и γ_1 , что $\gamma \rightsquigarrow_T \gamma_0$, $\gamma \rightsquigarrow_T \gamma_1$ и при этом γ_v является v -валентной конфигурацией.

Достижимость разветвления означает, что тип решения, которое еще предстоит достичь, зависит только от тех процессов, которые вполне способны выйти из строя.

Так может ли робастный алгоритм доверить выбор типа решения ненадежным процессам?

Невозможность консенсуса

Утверждение 3.

При выполнении t -робастного алгоритма не существует достижимого разветвления.

Невозможность консенсуса

Утверждение 3.

При выполнении t -робастного алгоритма не существует достижимого разветвления.

Доказательство.

Рассмотрим произвольную достижимую конфигурацию γ и множества процессов T , $|T| \leq t$, и $S = \mathbb{P} \setminus T$.
Т.к. $|S| \geq N - t$, по Утверждению 2 есть такая конфигурация принятия решения δ , что $\gamma \rightsquigarrow_S \delta$. Будем считать, что в ней принимается 0 -решение.

Невозможность консенсуса

Утверждение 3.

При выполнении t -робастного алгоритма не существует достижимого разветвления.

Доказательство.

Рассмотрим произвольную достижимую конфигурацию γ и множества процессов T , $|T| \leq t$, и $S = \mathbb{P} \setminus T$.
Т.к. $|S| \geq N - t$, по Утверждению 2 есть такая конфигурация принятия решения δ , что $\gamma \rightsquigarrow_S \delta$. Будем считать, что в ней принимается 0 -решение.
Покажем, что ни для одной 1 -валентной конфигурации γ' не выполняется $\gamma \rightsquigarrow_T \gamma'$.

Невозможность консенсуса

Утверждение 3.

При выполнении t -робастного алгоритма не существует достижимого разветвления.

Доказательство.

Рассмотрим произвольную достижимую конфигурацию γ и множества процессов T , $|T| \leq t$, и $S = \mathbb{P} \setminus T$.
Т.к. $|S| \geq N - t$, по Утверждению 2 есть такая конфигурация принятия решения δ , что $\gamma \rightsquigarrow_S \delta$. Будем считать, что в ней принимается 0 -решение.
Покажем, что ни для одной 1 -валентной конфигурации γ' не выполняется $\gamma \rightsquigarrow_T \gamma'$. Допустим противное: $\gamma \rightsquigarrow_T \gamma'$.

Невозможность консенсуса

Утверждение 3.

При выполнении t -робастного алгоритма не существует достижимого разветвления.

Доказательство.

Рассмотрим произвольную достижимую конфигурацию γ и множества процессов T , $|T| \leq t$, и $S = \mathbb{P} \setminus T$.
Т.к. $|S| \geq N - t$, по Утверждению 2 есть такая конфигурация принятия решения δ , что $\gamma \rightsquigarrow_S \delta$. Будем считать, что в ней принимается 0 -решение.
Покажем, что ни для одной 1 -валентной конфигурации γ' не выполняется $\gamma \rightsquigarrow_T \gamma'$. Допустим противное: $\gamma \rightsquigarrow_T \gamma'$.
Т.к. шаги выполнения процессов из T и из S можно переставлять, найдется такая конфигурация δ' , которая достижима как из δ , так и из γ' .

Невозможность консенсуса

Утверждение 3.

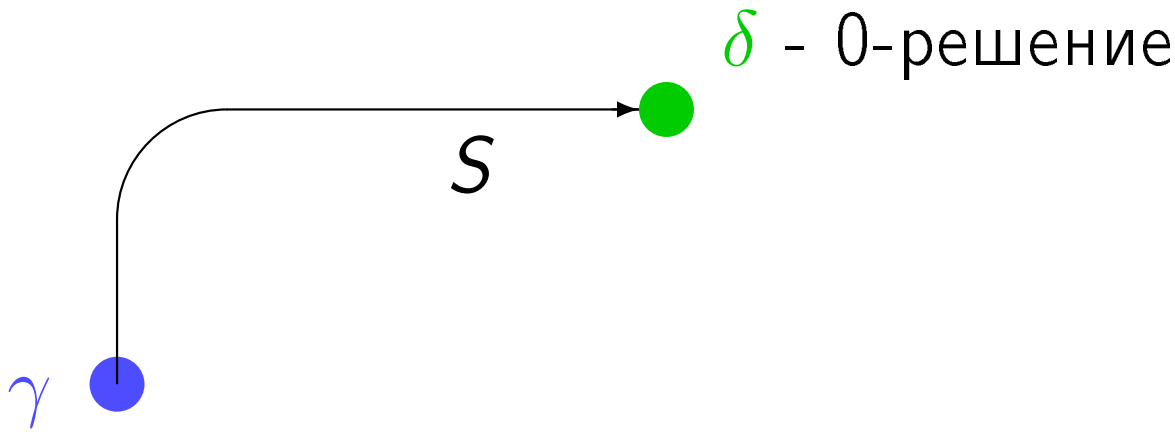
При выполнении t -робастного алгоритма не существует достижимого разветвления.

Доказательство.

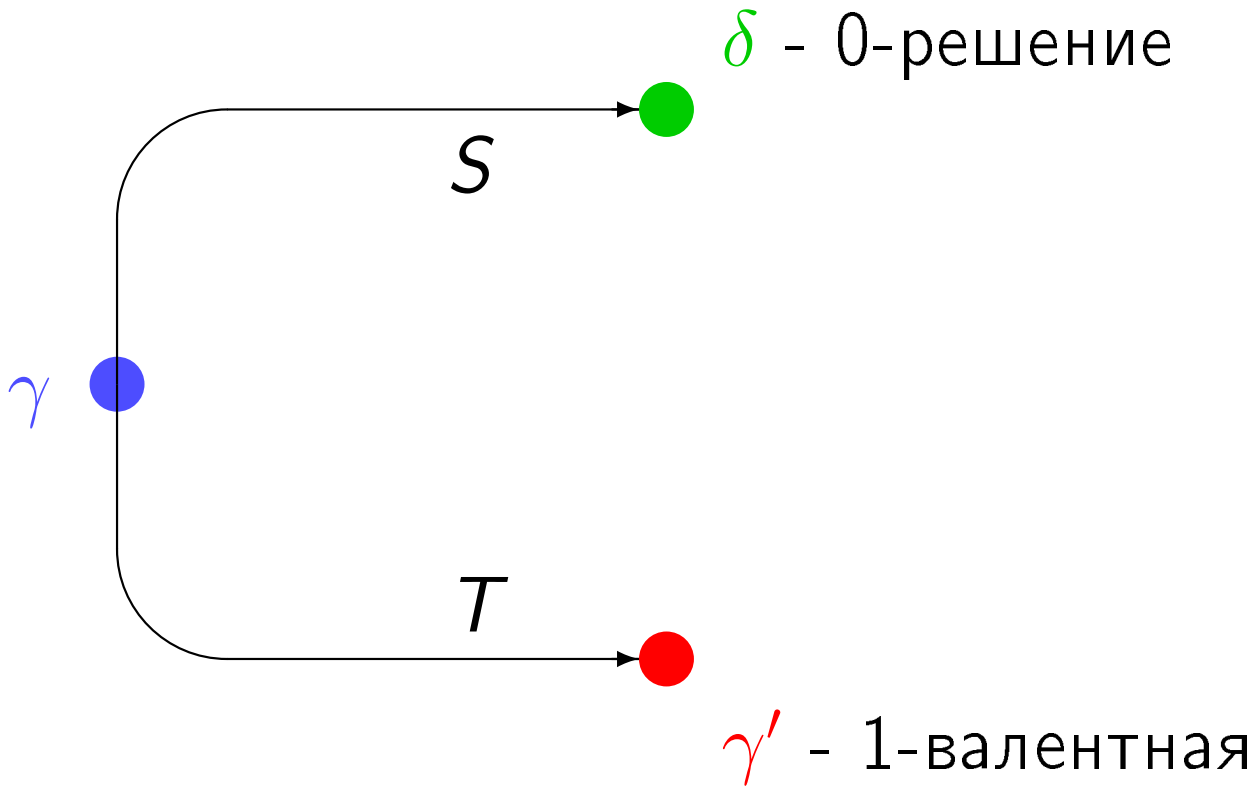
Рассмотрим произвольную достижимую конфигурацию γ и множества процессов T , $|T| \leq t$, и $S = \mathbb{P} \setminus T$.
Т.к. $|S| \geq N - t$, по Утверждению 2 есть такая конфигурация принятия решения δ , что $\gamma \rightsquigarrow_S \delta$. Будем считать, что в ней принимается 0 -решение.
Покажем, что ни для одной 1 -валентной конфигурации γ' не выполняется $\gamma \rightsquigarrow_T \gamma'$. Допустим противное: $\gamma \rightsquigarrow_T \gamma'$.
Т.к. шаги выполнения процессов из T и из S можно переставлять, найдется такая конфигурация δ' , которая достижима как из δ , так и из γ' .
Поскольку δ — это конфигурация принятия 0 -решения, такой же должна быть и δ' , и это означает, что γ' **не является** 1 -валентной.

Невозможность консенсуса

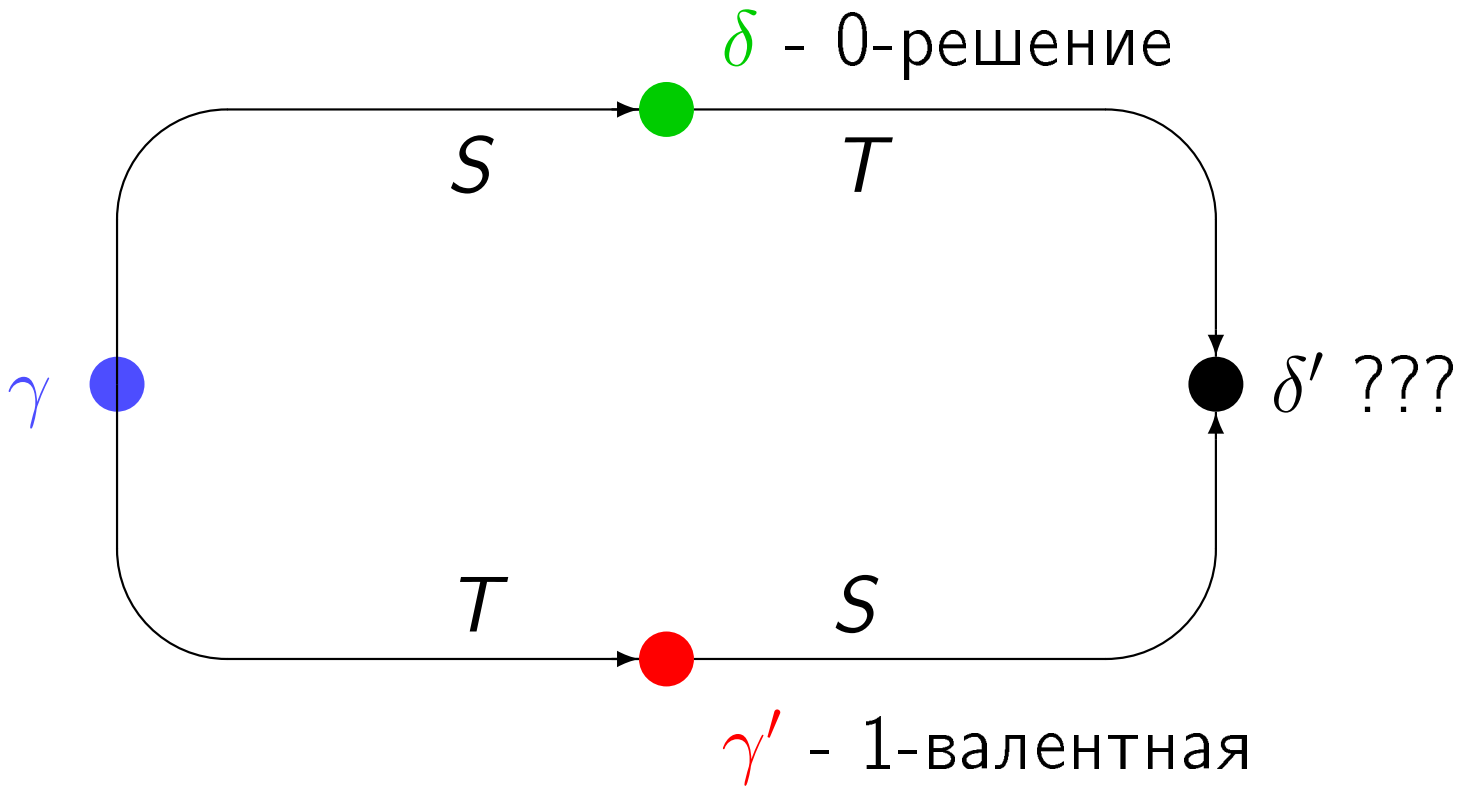
Невозможность консенсуса



Невозможность консенсуса



Невозможность консенсуса



Невозможность консенсуса

Утверждение 4.

Для всякого алгоритма консенсуса A , робастного по отношению к выходу из стороя одного процесса, существует бивалентная начальная конфигурация.

Невозможность консенсуса

Утверждение 4.

Для всякого алгоритма консенсуса A , робастного по отношению к выходу из стороя одного процесса, существует бивалентная начальная конфигурация.

Доказательство.

A невырожденный, и, значит, он имеет такие начальные конфигурации δ_0 и δ_1 , что из δ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения.

Невозможность консенсуса

Утверждение 4.

Для всякого алгоритма консенсуса A , робастного по отношению к выходу из стороя одного процесса, существует бивалентная начальная конфигурация.

Доказательство.

A невырожденный, и, значит, он имеет такие начальные конфигурации δ_0 и δ_1 , что из δ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения.

Если $\delta_0 = \delta_1$, то O'K.

Невозможность консенсуса

Утверждение 4.

Для всякого алгоритма консенсуса A , робастного по отношению к выходу из стороя одного процесса, существует бивалентная начальная конфигурация.

Доказательство.

A невырожденный, и, значит, он имеет такие начальные конфигурации δ_0 и δ_1 , что из δ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения.

Если $\delta_0 = \delta_1$, то O'K.

Иначе есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

$x_1, x_2, \ldots, x_n$

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

$x_1, x_2, \ldots, x_n$

$0,\ 0,\ \ldots,\ 0$ 0-решение

$1,\ 1,\ \ldots,\ 1$ 1-решение

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

$x_1, x_2, \ldots, x_n$

$0, 0, \ldots, 0$ 0-решение

$0, 0, \ldots, 1$

$1, 1, \ldots, 1$ 1-решение

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

$x_1, x_2, \ldots, x_n$	
$0, 0, \ldots, 0$	0-решение
$0, 0, \ldots, 1$	0-решение
$1, 1, \ldots, 1$	1-решение

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

$x_1, x_2, \ldots, x_n$	
$0, 0, \ldots, 0$	0-решение
$0, 0, \ldots, 1$	0-решение

$1, 1, \ldots, 0$	
$1, 1, \ldots, 1$	1-решение

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

$x_1, x_2, \ldots, x_n$	
$0, 0, \ldots, 0$	0-решение
$0, 0, \ldots, 1$	0-решение

$1, 1, \ldots, 0$	1-решение
$1, 1, \ldots, 1$	1-решение

Невозможность консенсуса

Есть такие начальные конфигурации γ_0 и γ_1 , что из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения и при этом γ_0 и γ_1 отличаются входными данными только одного процесса p .

$x_1, x_2, \ldots, x_n$

$0,\ 0,\ \ldots,\ 0$ 0-решение

$0,\ 0,\ \ldots,\ 1$ 0-решение

$\sigma_1, \sigma_2, \ldots, 0, \ldots, \sigma_n$ 0-решение

$\sigma_1, \sigma_2, \ldots, 1, \ldots, \sigma_n$ 1-решение

$1,\ 1,\ \ldots,\ 0$ 1-решение

$1,\ 1,\ \ldots,\ 1$ 1-решение

Невозможность консенсуса

Доказательство.

Итак, γ_0 и γ_1 отличаются только входными данными процесса p , и из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения.

Невозможность консенсуса

Доказательство.

Итак, γ_0 и γ_1 отличаются только входными данными процесса p , и из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения.

Рассмотрим справедливое выполнение, начинающееся из γ_0 , в котором p не выполняет ни одного шага (вышел из строя). В ходе него будет достигнута конфигурация принятия решения β .

Невозможность консенсуса

Доказательство.

Итак, γ_0 и γ_1 отличаются только входными данными процесса p , и из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения.

Рассмотрим справедливое выполнение, начинающееся из γ_0 , в котором p не выполняет ни одного шага (вышел из строя). В ходе него будет достигнута конфигурация принятия решения β . Если β является конфигурацией принятия 1 -решения, то γ_0 — бивалентная конфигурация.

Невозможность консенсуса

Доказательство.

Итак, γ_0 и γ_1 отличаются только входными данными процесса p , и из γ_v , $v = 0, 1,$ достижима некоторая конфигурация принятия v -решения.

Рассмотрим справедливое выполнение, начинающееся из γ_0 , в котором p не выполняет ни одного шага (вышел из строя). В ходе него будет достигнута конфигурация принятия решения β . Если β является конфигурацией принятия 1 -решения, то γ_0 — бивалентная конфигурация.

Если же β является конфигурацией принятия 0 -решения, то заметим, что γ_1 отличается от γ_0 только данными процесса p , а он не участвует в выполнении.

Следовательно, β достижима из γ_1 , т.е. γ_1 — бивалентная конфигурация (точнее, из γ_1 можно достичь такой конфигурации β' , отличная от β только состоянием процесса p , и поэтому β' является конфигурацией принятия 0 -решения.)

□

Невозможность консенсуса

Утверждение 5.

Пусть γ — достижимая бивалентная конфигурация и в ней процесс p может совершить шаг s .
Тогда есть такая последовательность событий σ , что шаг s может быть применен к конфигурации $\sigma(\gamma)$ и при этом $s(\sigma(\gamma))$ будет бивалентной конфигурацией.

Невозможность консенсуса

Утверждение 5.

Пусть γ — достижимая бивалентная конфигурация и в ней процесс p может совершить шаг s .
Тогда есть такая последовательность событий σ , что шаг s может быть применен к конфигурации $\sigma(\gamma)$ и при этом $s(\sigma(\gamma))$ будет бивалентной конфигурацией.

Если это утверждение верно, то мы можем заставить **каждый** активный процесс совершать свой шаг, но при этом вся система так и не выберет никакого решения.

Невозможность консенсуса

Доказательство.

Пусть $C = \{\sigma(\gamma) : s \text{ не встречается в } \sigma\}$. Заметим, что шаг s применим к любой конфигурации из C .

Невозможность консенсуса

Доказательство.

Пусть $C = \{\sigma(\gamma) : s \text{ не встречается в } \sigma\}$. Заметим, что шаг s применим к любой конфигурации из C .

1) Покажем, что в C есть такие конфигурации α_0 и α_1 , что какая-нибудь конфигурация принятия v -решения достижима из $s(\alpha_v)$.

Невозможность консенсуса

Доказательство.

Пусть $C = \{\sigma(\gamma) : s \text{ не встречается в } \sigma\}$. Заметим, что шаг s применим к любой конфигурации из C .

1) Покажем, что в C есть такие конфигурации α_0 и α_1 , что какая-нибудь конфигурация принятия v -решения достижима из $s(\alpha_v)$.

В силу бивалентности γ некоторые конфигурации β_v принятия v -решения, $v = 0, 1$, достижимы из γ .

Если $\beta_v \in C$, то $s(\beta_v)$ является конфигурацией принятия v -решения. Поэтому $\alpha_v = \beta_v$.

Если $\beta_v \notin C$, то в качестве α_v можно взять ту конфигурацию, в которой впервые выполнился шаг s :

$$\gamma \rightsquigarrow_{\sigma'} \alpha_v \rightsquigarrow_s s(\alpha_v) \rightsquigarrow_{\sigma''} \beta_v$$

Невозможность консенсуса

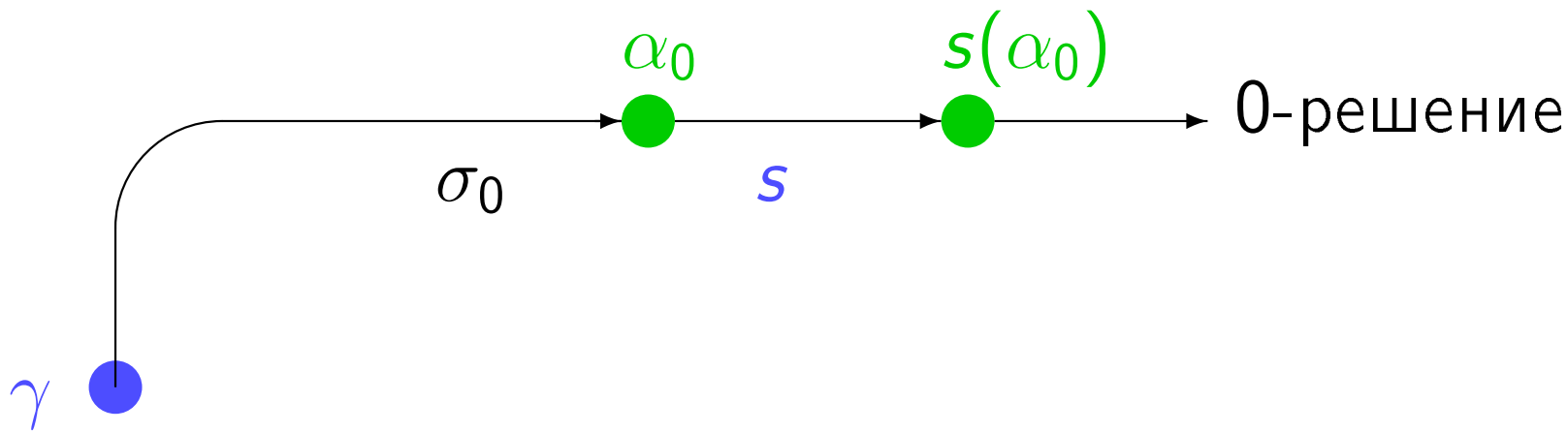
Покажем, что в C есть такие конфигурации α_0 и α_1 , что какая-нибудь конфигурация принятия v -решения достижима из $s(\alpha_v)$.

Невозможность консенсуса

Покажем, что в C есть такие конфигурации α_0 и α_1 , что какая-нибудь конфигурация принятия v -решения достижима из $s(\alpha_v)$.

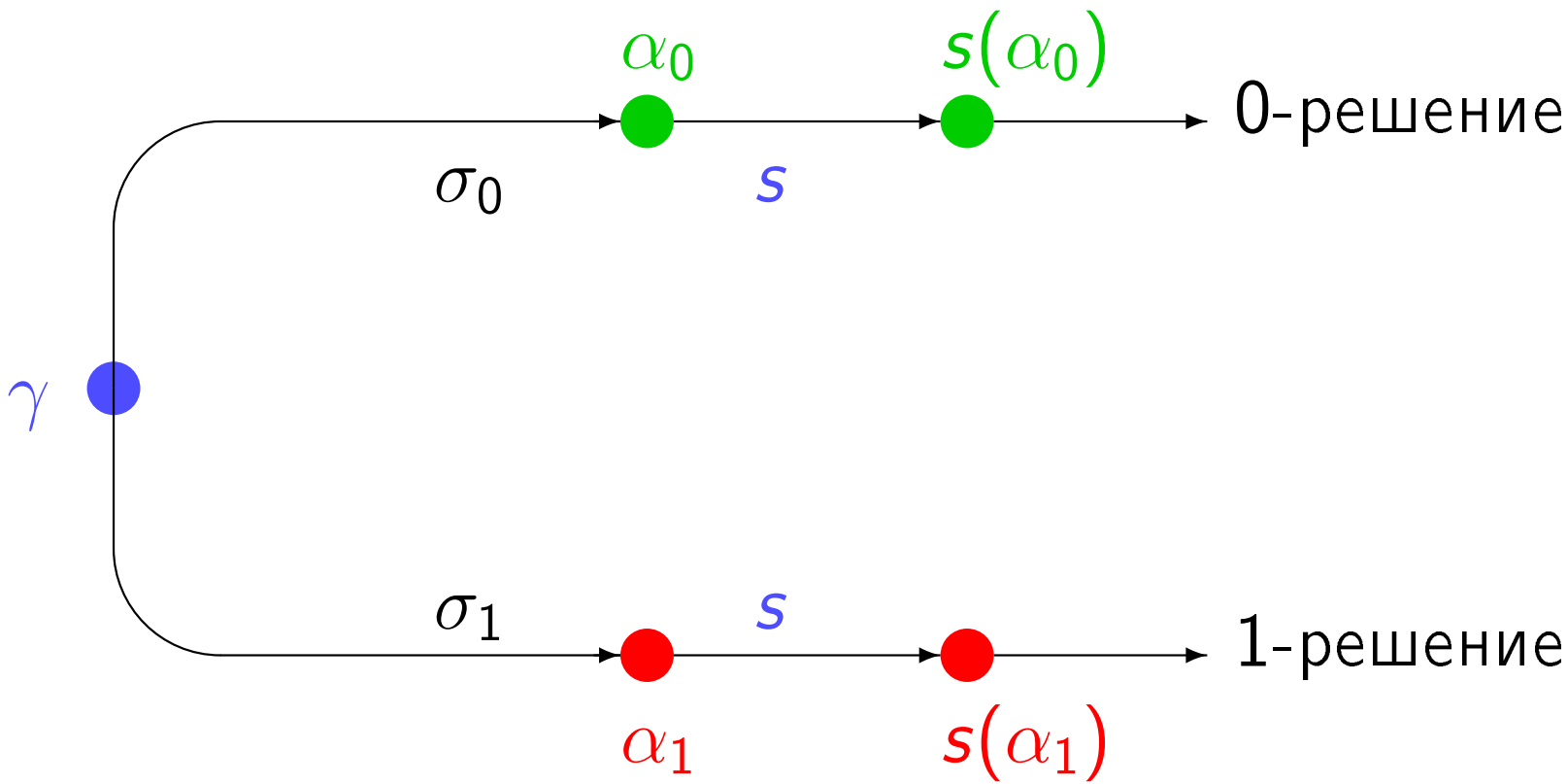
Невозможность консенсуса

Покажем, что в C есть такие конфигурации α_0 и α_1 , что какая-нибудь конфигурация принятия v -решения достижима из $s(\alpha_v)$.



Невозможность консенсуса

Покажем, что в C есть такие конфигурации α_0 и α_1 , что какая-нибудь конфигурация принятия v -решения достижима из $s(\alpha_v)$.



Невозможность консенсуса

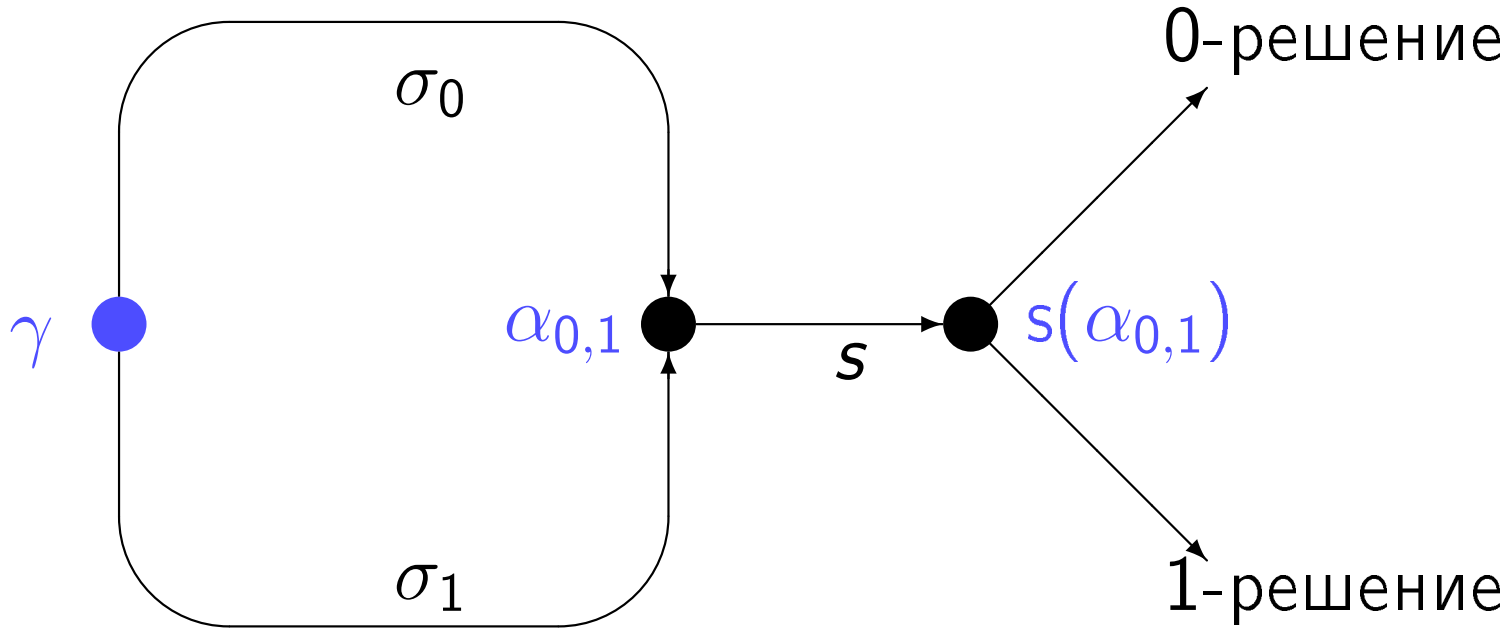
Доказательство.

Если $\alpha_0 = \alpha_1$, то $s(\alpha_0)$ — это и есть требуемая бивалентная конфигурация.

Невозможность консенсуса

Доказательство.

Если $\alpha_0 = \alpha_1$, то $s(\alpha_0)$ — это и есть требуемая бивалентная конфигурация.



Невозможность консенсуса

Доказательство.

Допустим, что $\alpha_0 \neq \alpha_1$. Тогда рассмотрим все те конфигурации, через которые проходят пути $\gamma \rightsquigarrow \alpha_0$ и $\gamma \rightsquigarrow \alpha_1$. Две конфигурации на этих путях назовем соседними , если одна из них достигается из другой за один шаг.

Невозможность консенсуса

Доказательство.

Допустим, что $\alpha_0 \neq \alpha_1$. Тогда рассмотрим все те конфигурации, через которые проходят пути $\gamma \rightsquigarrow \alpha_0$ и $\gamma \rightsquigarrow \alpha_1$. Две конфигурации на этих путях назовем соседними , если одна из них достигается из другой за один шаг.

Т.к. некоторая конфигурация принятия 0 -решения достижима из $s(\alpha_0)$ и некоторая конфигурация принятия 1 -решения достижима из $s(\alpha_1)$, верно хотя бы одно из двух:

Невозможность консенсуса

Доказательство.

Допустим, что $\alpha_0 \neq \alpha_1$. Тогда рассмотрим все те конфигурации, через которые проходят пути $\gamma \rightsquigarrow \alpha_0$ и $\gamma \rightsquigarrow \alpha_1$. Две конфигурации на этих путях назовем соседними , если одна из них достигается из другой за один шаг.

Т.к. некоторая конфигурация принятия 0 -решения достижима из $s(\alpha_0)$ и некоторая конфигурация принятия 1 -решения достижима из $s(\alpha_1)$, верно хотя бы одно из двух:

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,

Невозможность консенсуса

Доказательство.

Допустим, что $\alpha_0 \neq \alpha_1$. Тогда рассмотрим все те конфигурации, через которые проходят пути $\gamma \rightsquigarrow \alpha_0$ и $\gamma \rightsquigarrow \alpha_1$. Две конфигурации на этих путях назовем соседними , если одна из них достигается из другой за один шаг.

Т.к. некоторая конфигурация принятия 0 -решения достижима из $s(\alpha_0)$ и некоторая конфигурация принятия 1 -решения достижима из $s(\alpha_1)$, верно хотя бы одно из двух:

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.

Невозможность консенсуса

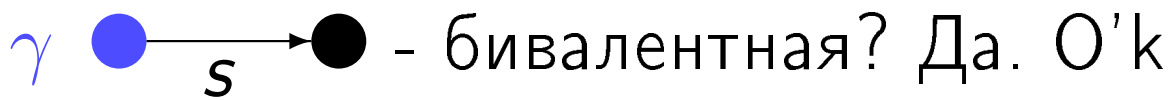
1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,

Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.

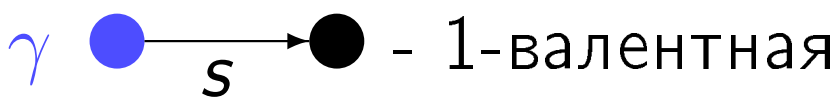
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



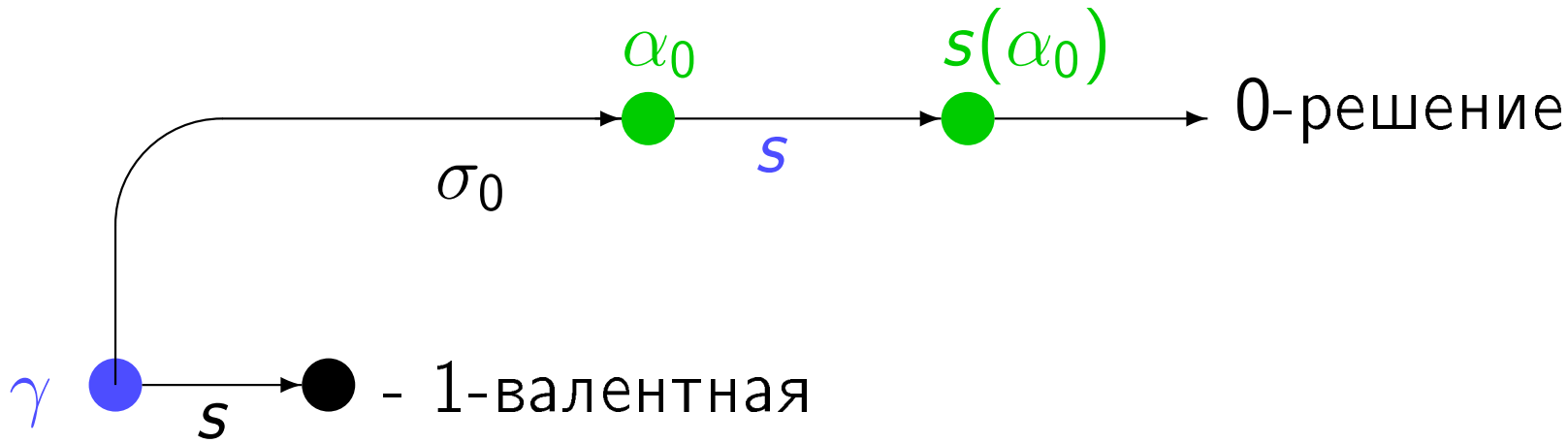
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



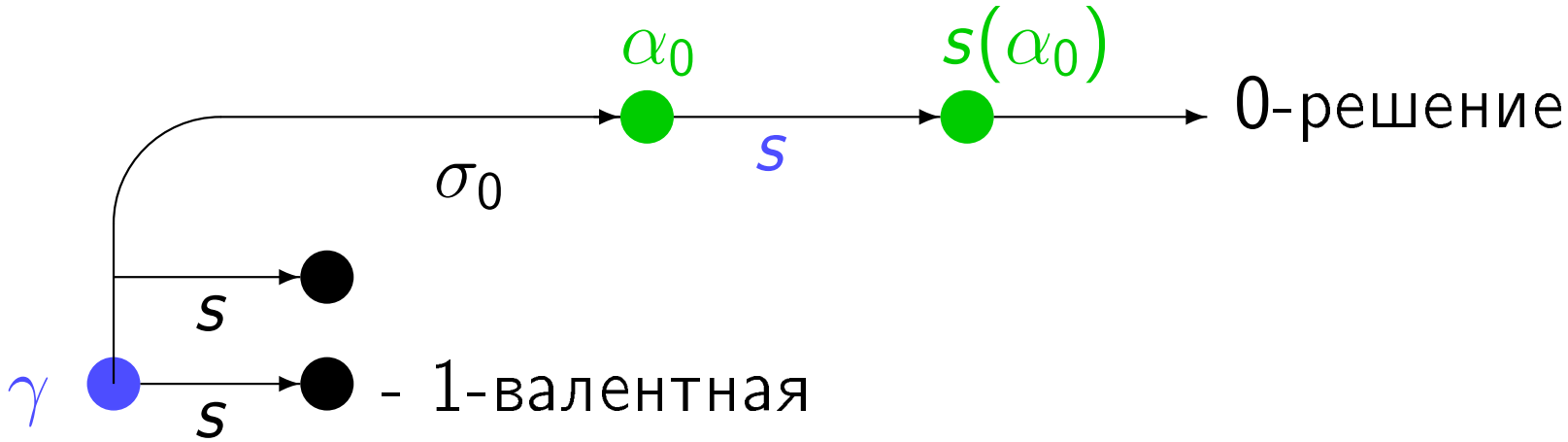
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



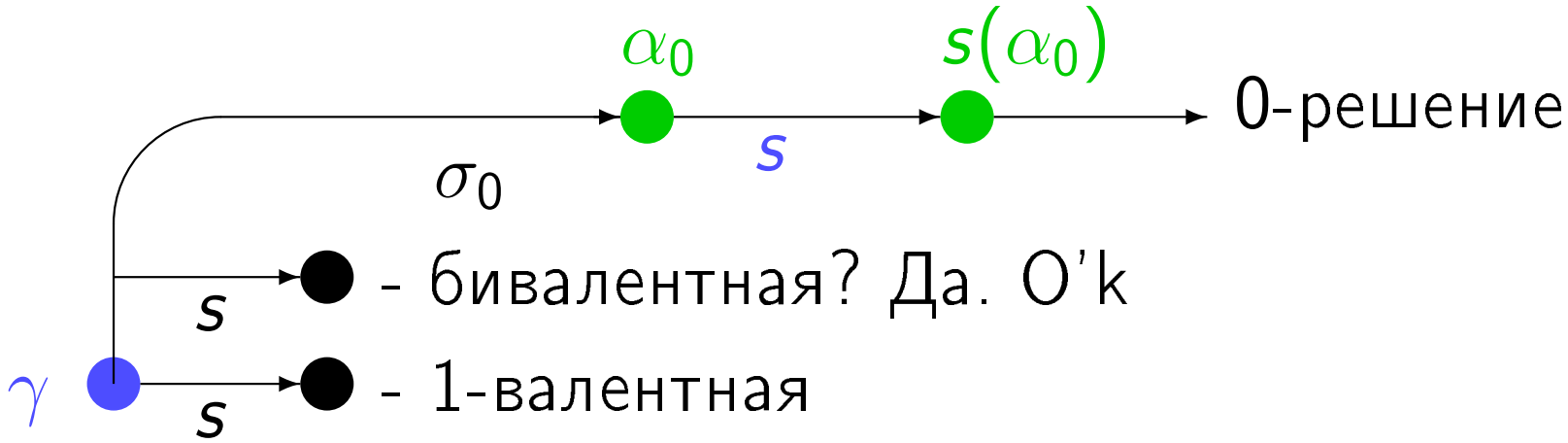
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



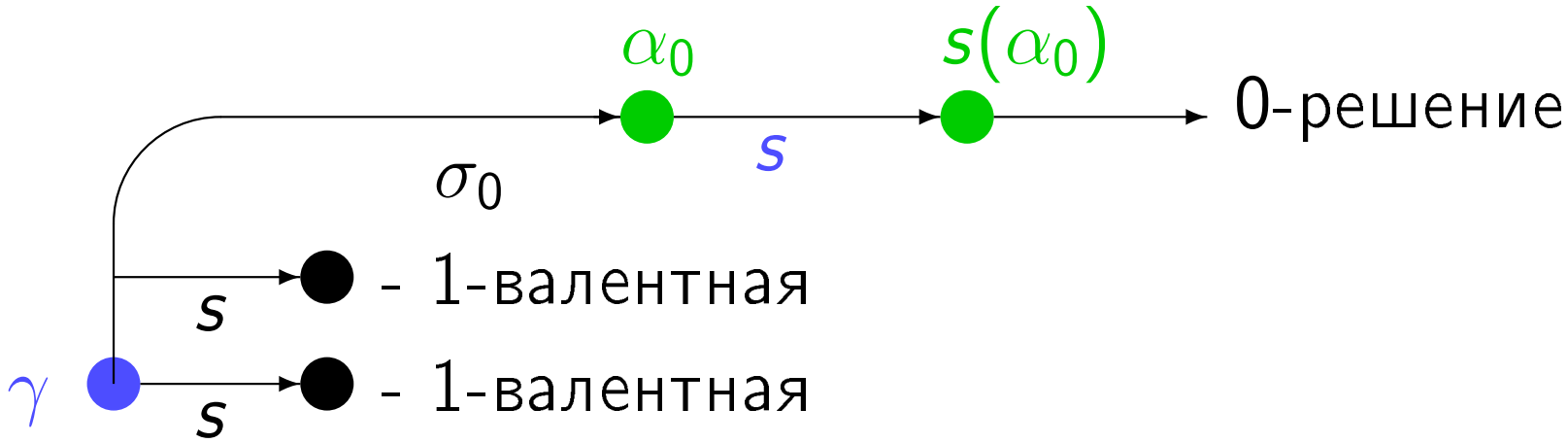
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



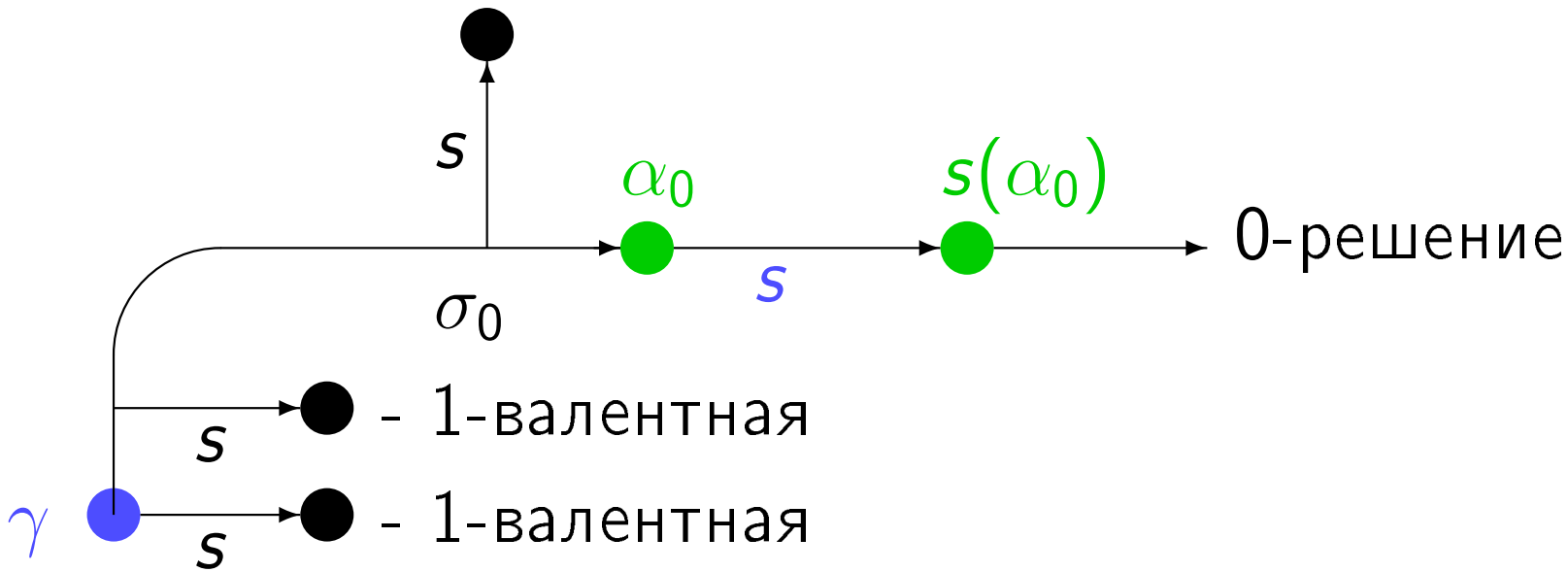
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



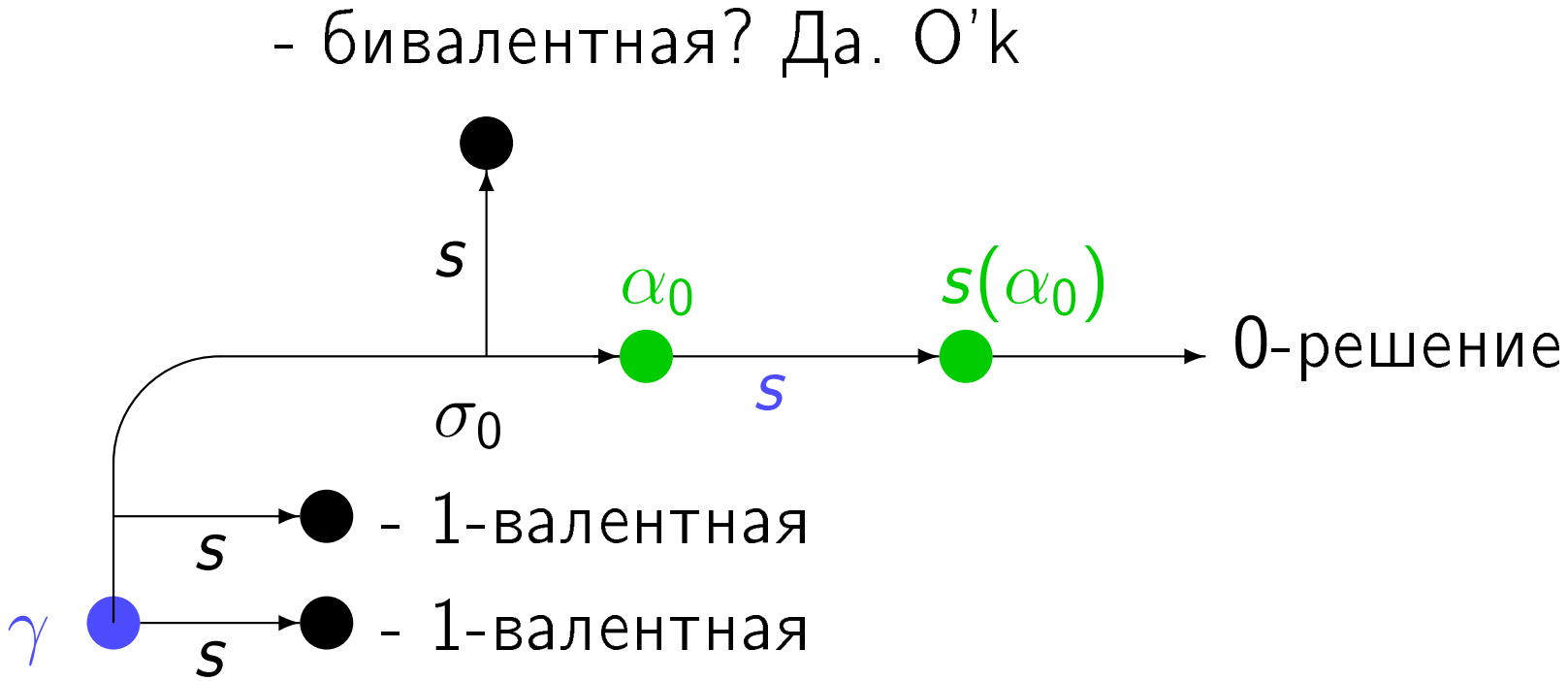
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



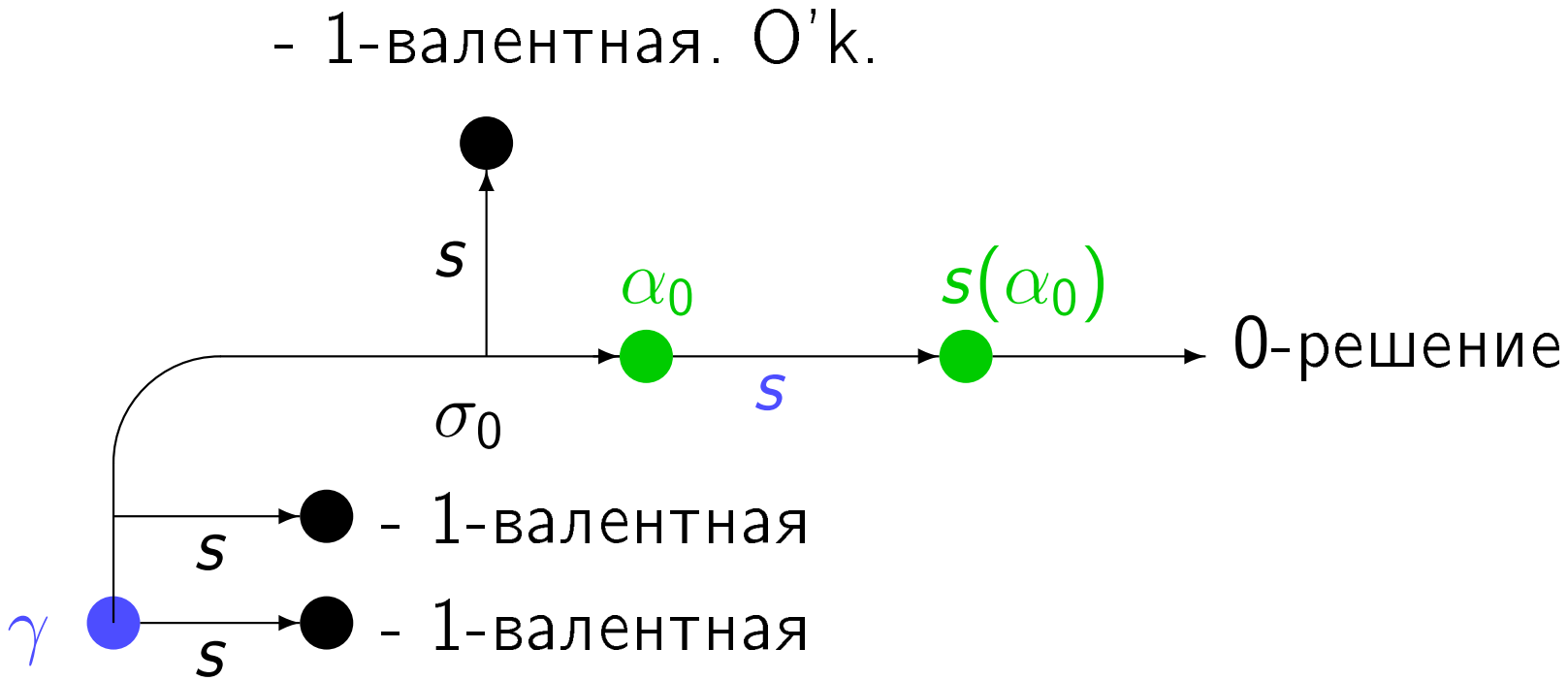
Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



Невозможность консенсуса

1) либо на на одном из этих путей лежит такая конфигурация γ' , что $s(\gamma')$ является бивалентной конфигурацией,
2) либо существуют такие соседние конфигурации γ_0 и γ_1 , что $s(\gamma_0)$ является конфигурацией принятия 0 -решения $s(\gamma_1)$ является конфигурацией принятия 1 -решения.



Невозможность консенсуса

Доказательство.

Первая альтернатива

на одном из этих путей лежит такая конфигурация γ' *, что* $s(\gamma')$ *является бивалентной конфигурацией*

— это как раз то, что требуется в доказываемом утверждении.

Невозможность консенсуса

Доказательство.

Первая альтернатива

на одном из этих путей лежит такая конфигурация γ' *, что* $s(\gamma')$ *является бивалентной конфигурацией*

— это как раз то, что требуется в доказываемом утверждении.

Покажем, что предположение об осуществимости второй альтернативы

существуют такие соседние конфигурации γ_0 *и* γ_1 *, что* $s(\gamma_0)$ *является конфигурацией принятия* 0 *-решения* $s(\gamma_1)$ *является конфигурацией принятия* 1 *-решения*

приводит к выводу о том, что одна из конфигурацией γ_0 или γ_1 будет являться разветвлением, что противоречит ранее доказанному Утверждению 3 о недостижимости разветвления.

Невозможность консенсуса

Доказательство.

Допустим, не ограничивая общности, что $\gamma_1 = e(\gamma_0)$ для некоторого события e в процессе q .

Невозможность консенсуса

Доказательство.

Допустим, не ограничивая общности, что $\gamma_1 = e(\gamma_0)$ для некоторого события e в процессе q .

Тогда $s(e(\gamma_0)) = s(\gamma_1)$, и поэтому $s(e(\gamma_0))$ является 1-валентной конфигурацией.

Невозможность консенсуса

Доказательство.

Допустим, не ограничивая общности, что $\gamma_1 = e(\gamma_0)$ для некоторого события e в процессе q .

Тогда $s(e(\gamma_0)) = s(\gamma_1)$, и поэтому $s(e(\gamma_0))$ является 1-валентной конфигурацией.
Если бы шаги e и s выполнялись бы разными процессами, то $s(e(\gamma_0)) = e(s(\gamma_0))$.
Но конфигурация $e(s(\gamma_0))$ не может быть 1 -валентной, поскольку $s(\gamma_0)$ уже является 0 -валентной конфигурацией.

Невозможность консенсуса

Доказательство.

Допустим, не ограничивая общности, что $\gamma_1 = e(\gamma_0)$ для некоторого события e в процессе q .

Тогда $s(e(\gamma_0)) = s(\gamma_1)$, и поэтому $s(e(\gamma_0))$ является 1-валентной конфигурацией.
Если бы шаги e и s выполнялись бы разными процессами, то $s(e(\gamma_0)) = e(s(\gamma_0))$.
Но конфигурация $e(s(\gamma_0))$ не может быть 1 -валентной, поскольку $s(\gamma_0)$ уже является 0 -валентной конфигурацией.

Таким образом, e и s происходят в одном процессе, т.е. $p = q$.

Невозможность консенсуса

Доказательство.

Допустим, не ограничивая общности, что $\gamma_1 = e(\gamma_0)$ для некоторого события e в процессе q .

Тогда $s(e(\gamma_0)) = s(\gamma_1)$, и поэтому $s(e(\gamma_0))$ является 1-валентной конфигурацией.
Если бы шаги e и s выполнялись бы разными процессами, то $s(e(\gamma_0)) = e(s(\gamma_0))$.
Но конфигурация $e(s(\gamma_0))$ не может быть 1 -валентной, поскольку $s(\gamma_0)$ уже является 0 -валентной конфигурацией.

Таким образом, e и s происходят в одном процессе, т.е. $p = q$.

Но тогда действиями одного процесса p из γ_0 достижимы 0 -валентная конфигурация $\gamma_0 \rightsquigarrow_p s(\gamma_0)$ и 1 -валентная конфигурация $\gamma_0 \rightsquigarrow_p s(e(\gamma_0))$,
т.е. конфигурация γ_0 — достижимое разветвление. $\square$

Невозможность консенсуса

Теорема 1. [Фишера—Линч—Патерсона]

Не существует алгоритмов достижения консенсуса, которые были бы одновременно

асинхронными, детерминированными и робастными

относительно выхода из строя **одного** процесса.

Невозможность консенсуса

Теорема 1. [Фишера—Линч—Патерсона]

Не существует алгоритмов достижения консенсуса, которые были бы одновременно

асинхронными, детерминированными и робастными

относительно выхода из строя **одного** процесса.

Доказательство.

Допустим, что такой алгоритм есть!

Невозможность консенсуса

Теорема 1. [Фишера—Линч—Патерсона]

Не существует алгоритмов достижения консенсуса, которые были бы одновременно

асинхронными, детерминированными и робастными

относительно выхода из строя **одного** процесса.

Доказательство.

Допустим, что такой алгоритм есть!

По утверждению 4 одна из его начальных конфигураций γ_0 является бивалентной.

Невозможность консенсуса

Теорема 1. [Фишера—Линч—Патерсона]

Не существует алгоритмов достижения консенсуса, которые были бы одновременно

асинхронными, детерминированными и робастными

относительно выхода из строя **одного** процесса.

Доказательство.

Допустим, что такой алгоритм есть!

По утверждению 4 одна из его начальных конфигураций γ_0 является бивалентной.

По утверждению 5, если конфигурация γ_i бивалентна, и процесс p активен, то $\gamma_i \rightsquigarrow \gamma' \rightsquigarrow_p \gamma_{i+1}$, причем γ_{i+1} — бивалентная конфигурация.

Невозможность консенсуса

Теорема 1. [Фишера—Линч—Патерсона]

Не существует алгоритмов достижения консенсуса, которые были бы одновременно

асинхронными, детерминированными и робастными

относительно выхода из строя **одного** процесса.

Доказательство.

Допустим, что такой алгоритм есть!

По утверждению 4 одна из его начальных конфигураций γ_0 является бивалентной.

По утверждению 5, если конфигурация γ_i бивалентна, и процесс p активен, то $\gamma_i \rightsquigarrow \gamma' \rightsquigarrow_p \gamma_{i+1}$, причем γ_{i+1} — бивалентная конфигурация.

Значит, алгоритм имеет бесконечное справедливое вычисление, не приводящее ни к одному решению. $\square$

Возможность консенсуса

Обсуждение.

Этот результат очень чувствителен к ослаблению любого из требований.

Возможность консенсуса

Обсуждение.

Этот результат очень чувствителен к ослаблению любого из требований.

1. Ослабленная модель отказа. В модели неисправностей изначально бездействующих процессов существуют детерминированные асинхронные алгоритмы консенсуса.

Возможность консенсуса

Обсуждение.

Этот результат очень чувствителен к ослаблению любого из требований.

1. Ослабленная модель отказа. В модели неисправностей изначально бездействующих процессов существуют детерминированные асинхронные алгоритмы консенсуса.
2. Рандомизация. В византийской модели неисправностей существуют рандомизованные асинхронные алгоритмы достижения консенсуса с вероятностью, стремящейся к 1.

Возможность консенсуса

Обсуждение.

Этот результат очень чувствителен к ослаблению любого из требований.

1. Ослабленная модель отказа. В модели неисправностей изначально бездействующих процессов существуют детерминированные асинхронные алгоритмы консенсуса.
2. Рандомизация. В византийской модели неисправностей существуют рандомизованные асинхронные алгоритмы достижения консенсуса с вероятностью, стремящейся к 1.
3. Ослабленное условие завершаемости. В византийской модели неисправностей существует асинхронный детерминированный алгоритм широковещательной рассылки информации.

Возможность консенсуса

Обсуждение.

Этот результат очень чувствителен к ослаблению любого из требований.

1. Ослабленная модель отказа. В модели неисправностей изначально бездействующих процессов существуют детерминированные асинхронные алгоритмы консенсуса.
2. Рандомизация. В византийской модели неисправностей существуют рандомизованные асинхронные алгоритмы достижения консенсуса с вероятностью, стремящейся к 1.
3. Ослабленное условие завершаемости. В византийской модели неисправностей существует асинхронный детерминированный алгоритм широковещательной рассылки информации.
4. Синхронность. В византийской модели неисправностей существуют детерминированные синхронные алгоритмы консенсуса.

Возможность консенсуса

Теорема 2.

Существует асинхронный детерминированный робастный относительно t изначально бездействующих процессов алгоритм консенсуса для любого $t < N/2$.

Возможность консенсуса

Теорема 2.

Существует асинхронный детерминированный робастный относительно t изначально бездействующих процессов алгоритм консенсуса для любого $t < N/2$.

Идея алгоритма.

Возможность консенсуса

Теорема 2.

Существует асинхронный детерминированный робастный относительно t изначально бездействующих процессов алгоритм консенсуса для любого $t < N/2$.

Идея алгоритма.

1. Исправный процесс извещает всех о своей исправности и дожидается таких же известий от $\lceil N + 1/2 \rceil$ соседей.

Возможность консенсуса

Теорема 2.

Существует асинхронный детерминированный робастный относительно t изначально бездействующих процессов алгоритм консенсуса для любого $t < N/2$.

Идея алгоритма.

1. Исправный процесс извещает всех о своей исправности и дожидается таких же известий от $\lceil N + 1/2 \rceil$ соседей.
2. Исправный процесс извещает всех о том, какие исправные процессы ему известны, и дожидается таких же известий от всех известных ему исправных процессов.

Возможность консенсуса

Теорема 2.

Существует асинхронный детерминированный робастный относительно t изначально бездействующих процессов алгоритм консенсуса для любого $t < N/2$.

Идея алгоритма.

1. Исправный процесс извещает всех о своей исправности и дожидается таких же известий от $\lceil N + 1/2 \rceil$ соседей.
2. Исправный процесс извещает всех о том, какие исправные процессы ему известны, и дожидается таких же известий от всех известных ему исправных процессов.
3. В результате образуется большая клика («клубок») исправных процессов, осведомленных об исправности друг друга, а все прочие процессы либо оказываются неисправными, либо выходят из игры.

Возможность консенсуса

Теорема 2.

Существует асинхронный детерминированный робастный относительно t изначально бездействующих процессов алгоритм консенсуса для любого $t < N/2$.

Идея алгоритма.

1. Исправный процесс извещает всех о своей исправности и дожидается таких же известий от $\lceil N + 1/2 \rceil$ соседей.
2. Исправный процесс извещает всех о том, какие исправные процессы ему известны, и дожидается таких же известий от всех известных ему исправных процессов.
3. В результате образуется большая клика («клубок») исправных процессов, осведомленных об исправности друг друга, а все прочие процессы либо оказываются неисправными, либо выходят из игры.
4. В клике выбирается лидер («вожак»), который оповещает всех о своем решении.

Возможность консенсуса

var $Succ_p$, $Alive_p$, $Rcvd_p$: множества процессов **init** $\varnothing$;

begin shout $\langle$**name**, $p\rangle$;
(* то есть: **forall** $q \in \mathbb{P}$ **do** send $\langle$**name**, $p\rangle$ to q *)
while $\#Succ_p < \lceil (N+1)/2 \rceil$
do begin receive $\langle$**name**, $q\rangle$; $Succ_p := Succ_p \cup \{q\}$ **end**;
shout $\langle$**pre**, p, $Succ_p\rangle$;
$Alive_p := Succ_p$;
while $Alive_p \not\subseteq Rcvd_p$
do begin receive $\langle$**pre**, q, $Succ\rangle$;
$Alive_p := Alive_p \cup Succ \cup \{q\}$;
$Rcvd_p := Rcvd_p \cup \{q\}$
end;
end

Алгоритм вычисления «клубка» процессов, взаимно осведомленных об исправности друг друга.

Невозможность консенсуса

Теорема 3.

Не существует вероятностных протоколов консенсуса, которые были бы робастными относительно выхода из строя t процессов, где $t \geq N/2$.

Невозможность консенсуса

Теорема 3.

Не существует вероятностных протоколов консенсуса, которые были бы робастными относительно выхода из строя t процессов, где $t \geq N/2$.

Теорема 4.

Не существует робастных вероятностных протоколов консенсуса в византийской модели с t неисправными процессами в случае, когда $t \geq N/3$.

Невозможность консенсуса

Доказательство теоремы 3.

Невозможность консенсуса

Доказательство теоремы 3.

Предположим противное: такой протокол *Prot* существует.

Невозможность консенсуса

Доказательство теоремы 3.

Предположим противное: такой протокол *Prot* существует.

Для подмножества процессов S конфигурацию γ назывем S-бивалентной , если, выполняя только действия процессов семейства S , из γ можно достичь как конфигурации принятия 0 -решения, так и конфигурации принятия 1 -решения.

Невозможность консенсуса

Доказательство теоремы 3.

Предположим противное: такой протокол *Prot* существует.

Для подмножества процессов S конфигурацию γ назывем S-бивалентной , если, выполняя только действия процессов семейства S , из γ можно достичь как конфигурации принятия 0 -решения, так и конфигурации принятия 1 -решения.

Конфигурация γ называется S-0 -валентной, если, выполняя только действия процессов семейства S , из γ можно достичь конфигурации принятия 0-решения , но нельзя достичь конфигурации принятия 1 -решения.

Аналогичным образом определяется S-1 -валентная конфигурация.

Невозможность консенсуса

Доказательство теоремы 3.

Предположим противное: такой протокол $Prot$ существует.

Для подмножества процессов S конфигурацию γ назывем S-бивалентной , если, выполняя только действия процессов семейства S , из γ можно достичь как конфигурации принятия 0 -решения, так и конфигурации принятия 1 -решения.

Конфигурация γ называется S-0 -валентной, если, выполняя только действия процессов семейства S , из γ можно достичь конфигурации принятия 0-решения , но нельзя достичь конфигурации принятия 1 -решения.

Аналогичным образом определяется S-1 -валентная конфигурация.

Разделим все процессы на две группы S и T размера $\lfloor N/2 \rfloor$ и $\lceil N/2 \rceil$ соответственно.

Невозможность консенсуса

Утверждение 6.

Какова бы ни была достижимая конфигурация γ , она либо одновременно является S-0 -валентной и T-0 -валентной, либо является одновременно S-1 -валентной и T-1 -валентной.

Невозможность консенсуса

Утверждение 6.

Какова бы ни была достижимая конфигурация γ , она либо одновременно является S-0 -валентной и T-0 -валентной, либо является одновременно S-1 -валентной и T-1 -валентной.

Доказательство.

Высокая стойкость протокола *Prot* приводит к тому, что оба семейства процессов S и T могут принимать решение независимо друг от друга.
Если бы эти решения были **разными**, то, взяв комбинацию используемых расписаний, можно было бы достичь противоречивой конфигурации.

Невозможность консенсуса

Утверждение 7.

В протоколе *Prot* нельзя достичь бивалентной конфигурации.

Невозможность консенсуса

Утверждение 7.

В протоколе $Prot$ нельзя достичь бивалентной конфигурации.

Доказательство утверждения 7.

Предположим, что бивалентная конфигурация γ достижима.

Согласно Утверждению 6, для некоторого v она является S-1 -валентной и T-1 -валентной.

Предположим, не ограничивая общности, что γ является S-1 -валентной и T-1 -валентной конфигурацией.

Однако, коль скоро конфигурация γ является бивалентной, конфигурация δ_0 принятия 0 -решения может быть достигнута из конфигурации γ (разумеется, при взаимодействии процессов, относящихся к разным группам S и T).

Невозможность консенсуса

Доказательство утверждения 7.

В последовательности конфигураций выполнения, состоящей из всех конфигураций, следующих за γ и предшествующих δ_0 , можно выделить две последовательно идущие конфигурации γ_1 и γ_0 , что γ_v , $v \in \{0, 1\}$, является одновременно S-v -валентной и T-v -валентной конфигурацией.

Рассмотрим тот процесс p , переход в котором преобразует конфигурацию γ_1 в конфигурацию γ_0 :

$$\gamma_1 \xrightarrow{p} \gamma_0 \ .$$

Невозможность консенсуса

Доказательство утверждения 7.

В последовательности конфигураций выполнения, состоящей из всех конфигураций, следующих за γ и предшествующих δ_0 , можно выделить две последовательно идущие конфигурации γ_1 и γ_0 , что γ_v , $v \in \{0, 1\}$, является одновременно S-v -валентной и T-v -валентной конфигурацией.

Рассмотрим тот процесс p , переход в котором преобразует конфигурацию γ_1 в конфигурацию γ_0 :

$$\gamma_1 \xrightarrow{p} \gamma_0 \ .$$

Тогда этот процесс p не может принадлежать группе S , поскольку конфигурация γ_1 является S-1 -валентной, а конфигурация γ_0 является S-0 -валентной.

Подобным же образом можно убедиться в том, что $p \notin T$. В результате мы приходим к противоречию. $\square$

Невозможность консенсуса

Доказательство теоремы 3.

Невозможность консенсуса

Доказательство теоремы 3.

Предположим противное: такой протокол *Prot* существует.

Невозможность консенсуса

Доказательство теоремы 3.

Предположим противное: такой протокол *Prot* существует.

Тогда согласно утверждению 3

- Для всякого алгоритма консенсуса *A* , робастного по отношению к выходу из стороя одного процесса, существует бивалентная начальная конфигурация.

А согласно Утверждению 7

- В протоколе *Prot* нельзя достичь бивалентной конфигурации.

Невозможность консенсуса

Доказательство теоремы 3.

Предположим противное: такой протокол *Prot* существует.

Тогда согласно утверждению 3

- Для всякого алгоритма консенсуса *A* , робастного по отношению к выходу из стороя одного процесса, существует бивалентная начальная конфигурация.

А согласно Утверждению 7

- В протоколе *Prot* нельзя достичь бивалентной конфигурации.

Значит, такого протокола консенсуса *Prot* не существует в случае выхода из строя большей части процессов.

Задача консенсуса для синхронных систем

На практике надежные системы всегда оказываются синхронными, поскольку в них используются таймеры и верхние оценки времени доставки сообщений. В таких системах достижим более высокий уровень робастности, алгоритмы оказываются проще.

Если система является синхронной, то неисправные процессы, не передающие информацию, не могут вводить в заблуждение правильно работающие процессы.
Как только какой-либо процесс не получит ожидаемой информации, он будет использовать вместо нее заранее обусловленное значение и будет считать неисправным тот процесс, который должен был отправить ему эту информацию.
Таким образом, вышедшие из строя процессы будут немедленно обнаружены.
Основное внимание уделяется неисправностям византийского типа.

Задача консенсуса для синхронных систем

Теорема 5.

В том случае, если существует криптографически стойкая схема цифровой подписи, существует также алгоритм широковещательного распространения информации, который устойчив относительно любого числа t неисправностей византийского типа, где $t < N$.

КОНЕЦ ЛЕКЦИИ 12.