Научный семинар по теме кандидатской диссертации:

Исследование и разработка вероятностных методов верификации систем распределенного реестра

выступающий: Федотов Иван Андреевич

научный руководитель: кандидат физико-математических наук, доцент Хританков Антон Сергеевич

Московский физико-технический институт (национальный исследовательский университет)

Специальность 2.3.5

Математическое и программное обеспечение вычислительных систем,

комплексов и компьютерных сетей

Москва

2022

МФТИ

Актуальность работы

Системы распределенного реестра получили активное распространение в областях, связанных с финансами. Ошибки могут привести к существенным финансовым потерям.

Многие блокчейн сети, такие как Ethereum и Bitcoin не позволяют изменять код или параметры системы после развертывания. Работа предлагает алгоритмы для устранения ошибок до введения в эксплуатацию.

Технология новая: средства тестирования и автоматизации все еще в разработке.

Верификация систем распределенного реестра исследуется в ведущих исследовательских центрах: университет Иннополис в России, институт Макса Планка программного обеспечения в Германии, ETH в Швейцарии.

Цель исследования - решение проблемы выявления и устранения уязвимостей в системах распределенного реестра с вероятностными характеристиками.

Научная новизна

- Применен метод статистической проверки моделей для верификации уязвимостей систем распределенного реестра;
- Предложены алгоритмы построения модели и спецификации различных видов консенсуса в виде цепи Маркова с дискретным временем. Алгоритмы реализованы в виде программного комплекса и интегрированы в программный каркас Hyperledger Fabric;
- Применен метод статистической проверки моделей для верификации различных видов протоколов консенсуса;
- Предложен способ отправки сообщения на подтверждение для увеличения вероятности принятия транзакции в сети.

Основные положения диссертационной работы

- Исследованы и систематизированы методы и программные средства для верификации и тестирования систем распределенного реестра.
- Проведена проверка моделей атак на системы распределенного реестра, которые затрагивают их наи более важные составляющие. Также проведен анализ как можно снизить вероятность успешной атаки злоумышленника;
- Разработан алгоритм построения модели протокола консенсуса в виде Марковской цепи с дискретным временем и спецификации в виде логики вероятностной линейной временной логики. Разработан алгоритм оптимизации модели многостороннего соглашения в случае, если модель не удовлетворяет спецификации. На основе построенных алгоритмов разработан программный комплекс по построению модели, верификации и оптимизации многосторонних соглашений в блокчейн сетях. Проведена интеграция программного комплекса в программный каркас Hyperledger Fabric.

Публикации в журналах:

- Систематический обзор исследований в области автоматической верификации кода смарт-контрактов. Федотов И. Хританков А. Программная инженерия, 2020, стр. 3-13. Входит в перечень журналов ВАК.
- Statistical Model Checking of Common Attack Scenarios on Blockchain. Fedotov I., Khritankov A., EPTCS 342, 2021, стр. 65-77. Индексация Scopus.
- Towards verification of probabilistic multi-party consensus protocols: Constructing algorithms for verification of multi-party protocols with probabilistic properties. Fedotov I., Khritankov A., Barger A., ACM, ICSIM, 2022, стр. 100–105. Индексация Scopus.
- Автоматическая верификация многосторонних соглашений и планирование отправки сообщений в системах распределенного реестра. Федотов И., Хританков А., Обидаре М., 2022, стр. 200-213. Входит в перечень журналов ВАК.
- Optimizing multi-party agreement protocols / Fedotov I., Khritankov A., Barger A. // 2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2022, стр. 55–58.

Выступление на конференциях:

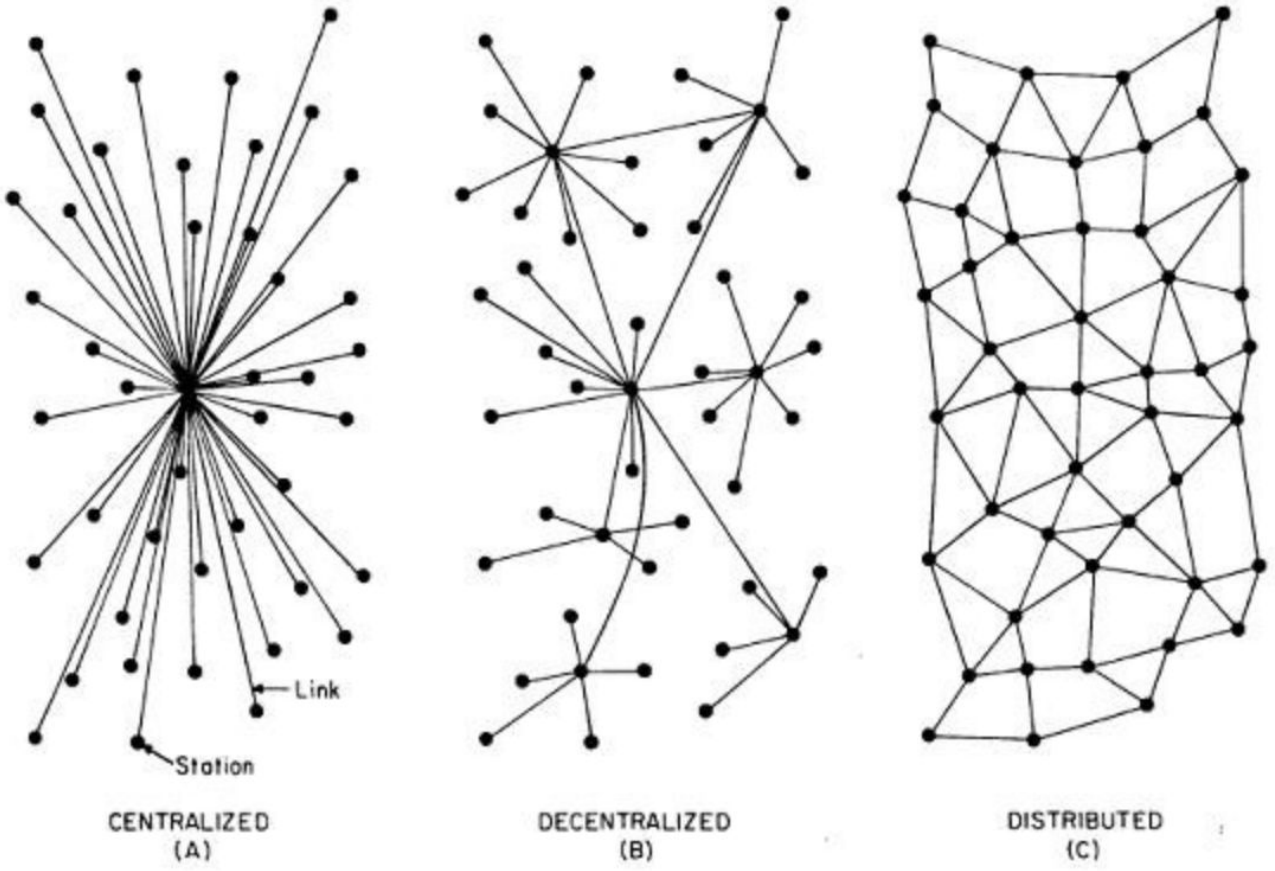
- Увеличение пропускной способности гибридного алгоритма консенсуса. 61-я Всероссийская научная конференция МФТИ. Москва, 2018.
- Верификация смарт-контрактов на основе статистической проверки моделей. 63-я Всероссийская научная конференция МФТИ. Москва, 2020.
- Статистическая проверка моделей для протоколов консенсуса под твержденя транзакций // 64-я Всероссийская научная конференция МФТИ. - Москва, 2021.
- Statistical Model Checking of Common Attack Scenarios on Blockchain. SCSS, Линц, Австрия, 2021.
- Towards verification of probabilistic multi-party consensus protocols. Fedotov I., Khritankov A., Barger A., ICBDSC, Йокогама, Япония, 2022.
- 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Париж, Франция, 2022.

Технология распределенного реестра

Распределенный реестр – база данных, распределенная между несколькими сетевыми узлами.

Технология распределенного реестра

Распределенный реестр – база данных, распределенная между несколькими сетевыми узлами



Server-client

Bitcoin

Social networks graph

Технология блокчейн

- Блокчейн - это распределенное хранилище, состоящие из множества узлов. Каждый узел хранит историю транзакций
- Каждый узел может предложить транзакцию для подтверждения сети
- Подтверждение транзакции происходит в соответствии с протоколом консенсуса
- Сеть блокчейн защищает историю транзакций: каждый блок транзакций хранит хеш значение предыдущего блока
- Это обеспечивает прозрачность и сохранность транзакций.

Технология блокчейн

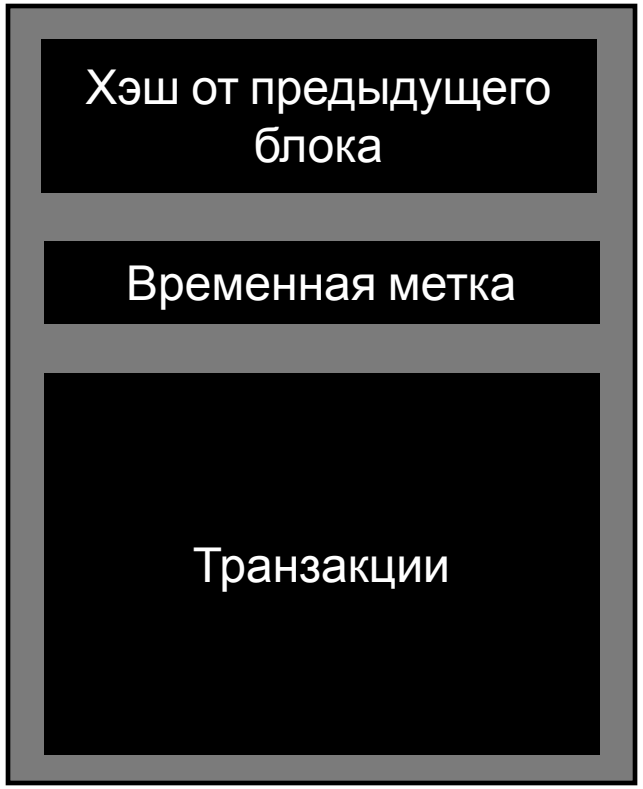


Схема блока

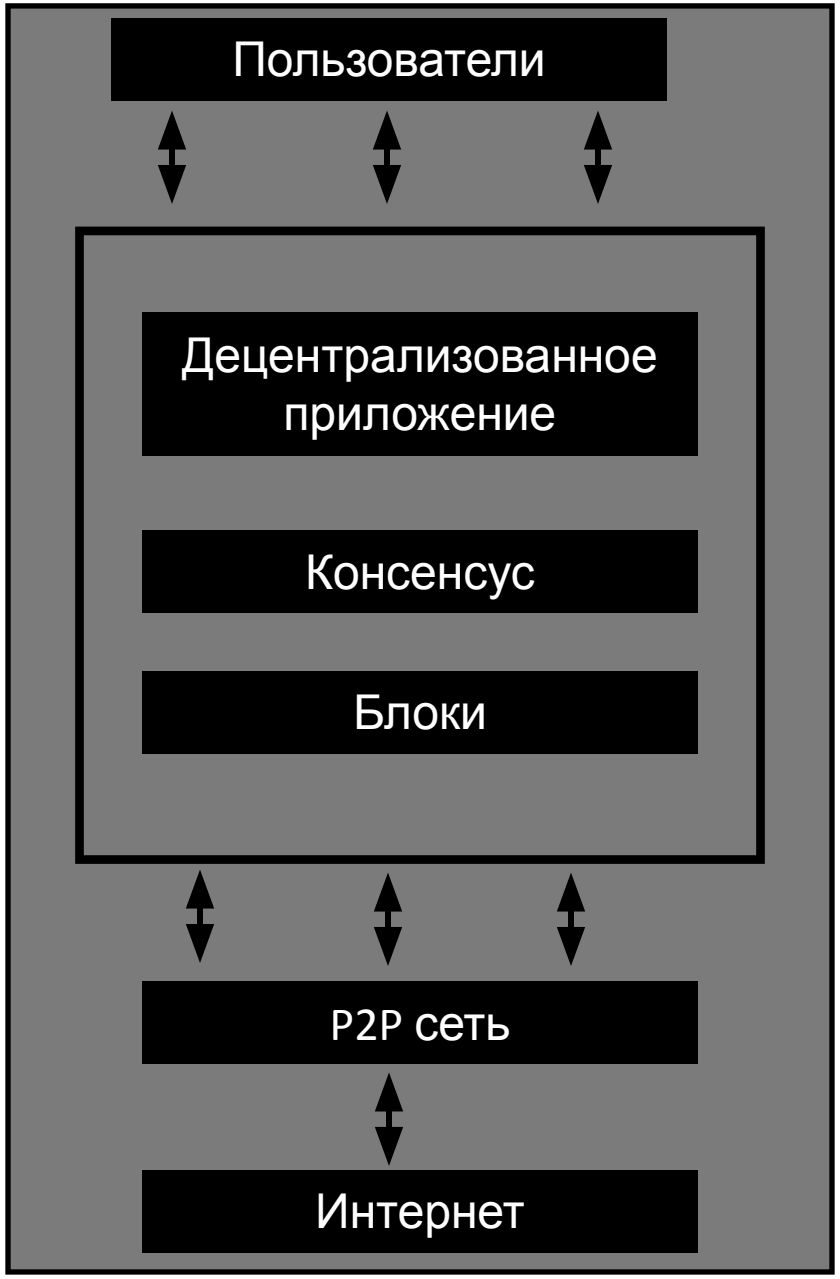


Схема сети

Методы и инструментальные средства для верификации систем распределенного реестра[1]

Цель: ознакомиться и систематизировать методы верификации систем распределенного реестра. Выявить перспективные направления исследования.

Метод	Описание	Инструментальные средства
Дедуктивный анализ		
Сопоставление предусловия с постусловием в контексте работы функций, процедур, методов для определения корректности их исполнения.		SolidiKeY [39]
Аудит и анализ		
Анализ результатов работы смарт-контракта на соответствие спецификациям.		S-gram [57]

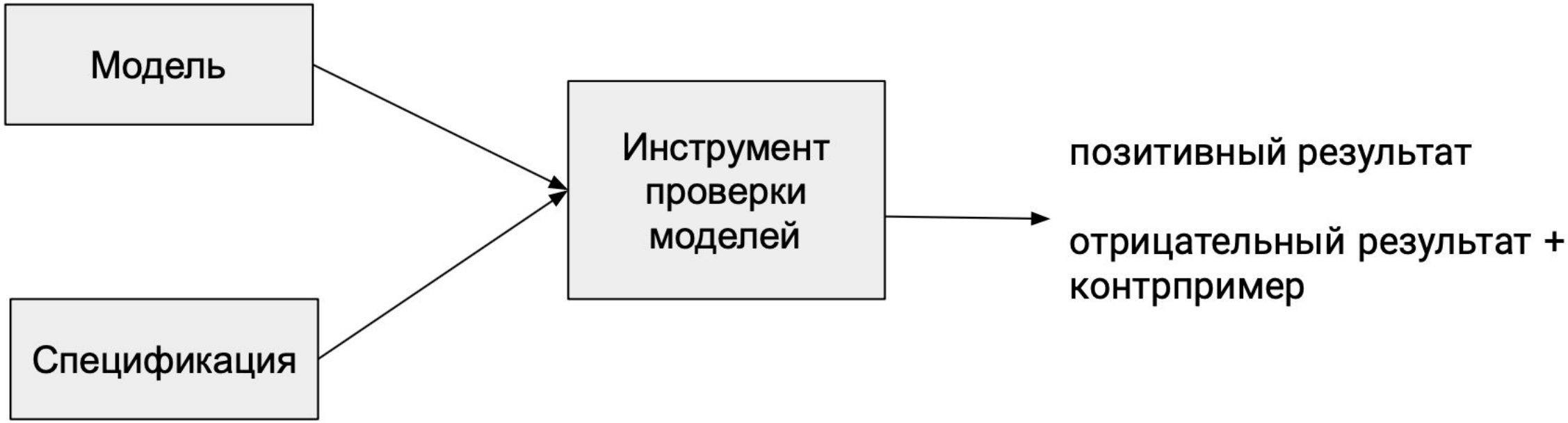
[1] Федотов И.А. Хританков А.С. Систематический обзор исследований в области автоматической верификации кода смарт-контрактов. https://www.elibrary.ru/item.asp?id=42347754

Метод	Описание	Инструментальные средства
Формальная верификация		
Используются формальные методы для доказательства соответствия модели системы установленным спецификациям требований.		VeriSol [26], F* [29], SOLAR [30], программный каркас на основе теории игр [33], решающие устройства SMT [34,35], SPIN [36], NuSMV [38]
Динамическая верификация		
Проверка поведения программы в момент ее выполнения. Полезна при отсутствии доступа к коду программы.		ContractFuzzer [51], ContractLarva [53], извлечение спецификаций из журналов [54], тензорный анализ [55]
Статический анализ кода		
В отличии от динамического анализа происходит без запуска программ. Статический анализ обладает высокой степенью надежности и применяется в жизненно-важных областях [63].		SmartCheck [23, 24, 25], Intellij Idea Solidity plugin [64], онлайн среда разработки Remix [65]
Проверка доказательств теорем		
Использование инструментариев доказательств теорем для подтверждения соответствия спецификациям.		Isabelle/HOL [40, 41], FEther [43], VeriSolid [45], FSPVM-E [49]

Перспективное направление: использование формальных методов верификации для систем распределенного реестра со стохастическими свойствами.

Метод проверки моделей

- Фаза моделирования: моделирование системы; формализация свойств для проверки
- Фаза запуска: запуск инструмента проверки моделей для верификации спецификаций
- Фаза анализа: если система удовлетворяет спецификации, то можно проверять следующую спецификацию; в противном случае необходимо генерировать и анализировать контрпример.



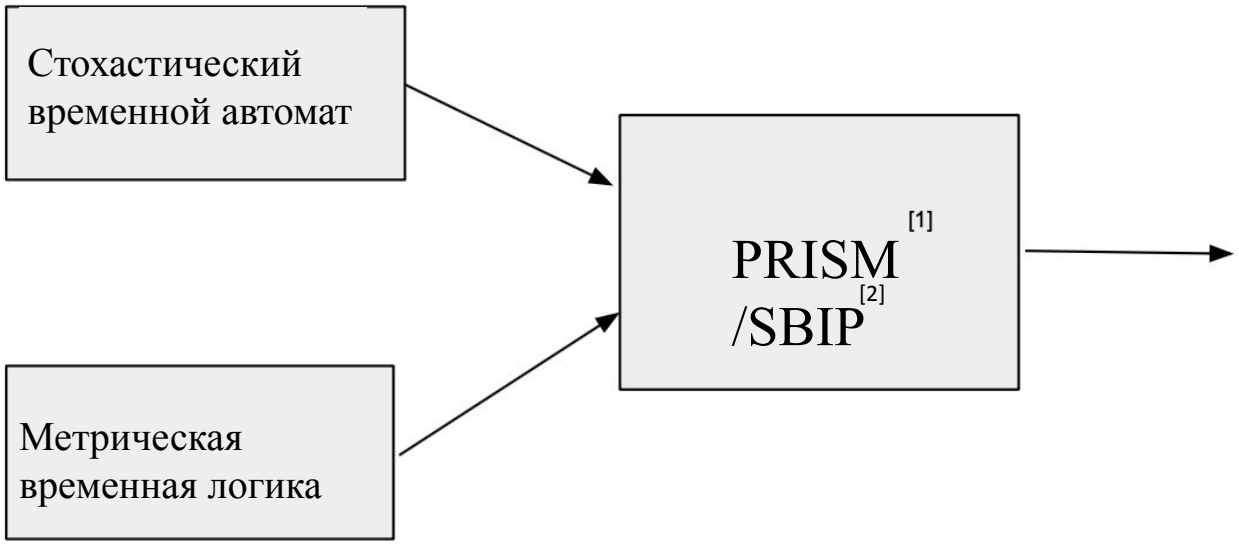
Виды моделей:

- Автоматы с конечным числом состояний
- Структуры Крипке
- Стохастические автоматы
- Марковские цепи

Виды логик для задания спецификаций:

- Логические формулы (нулевого, первого и второго порядков)
- Линейная временная логика
- Метрическая временная логика
- Вероятностная линейная временная логика
- Вычислительное дерево логики

Статистическая проверка моделей

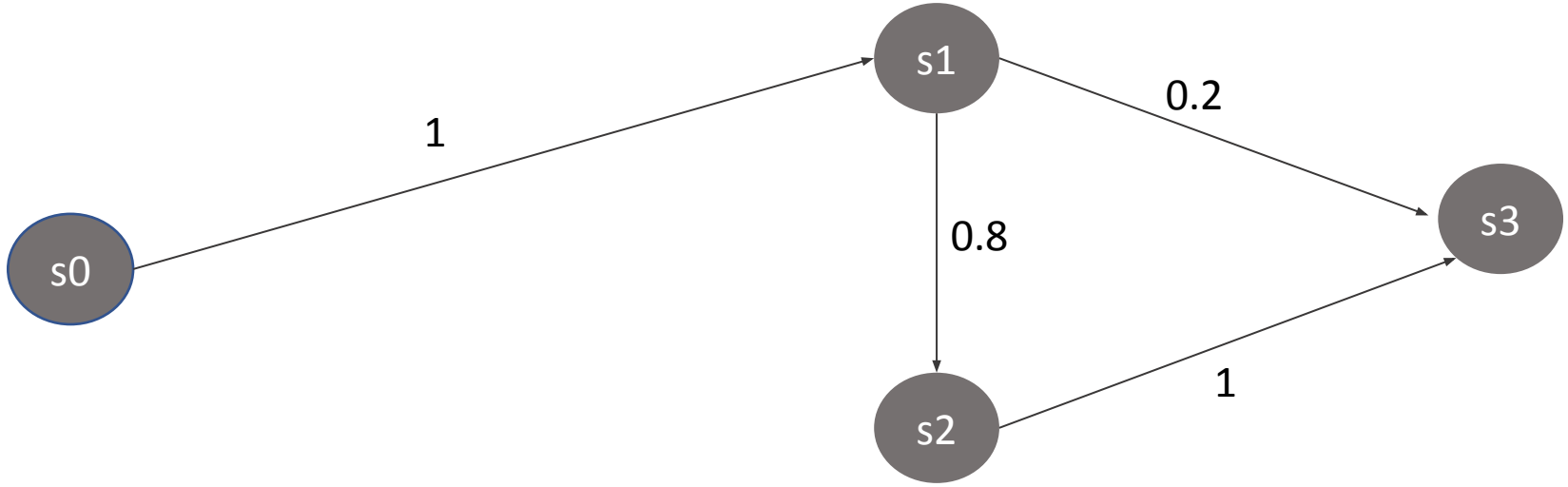


- Качественный вопрос: вероятность удовлетворить свойству больше порогового значения?
- Количественный: какова вероятность, что система удовлетворяет свойству? $\rho(|p'-p| < \delta) \geq \alpha$, δ - точность, α - параметр риска.

[1] PRISM 4.0: Verification of probabilistic real-time systems. Kwiatkowska, Marta, Gethin Norman, and David Parker. 2011.
[2] SBIP 2.0: Statistical Model Checking Stochastic Real-Time Systems Braham Lotfi Mediouni et al. 2018.

Статистическая проверка моделей

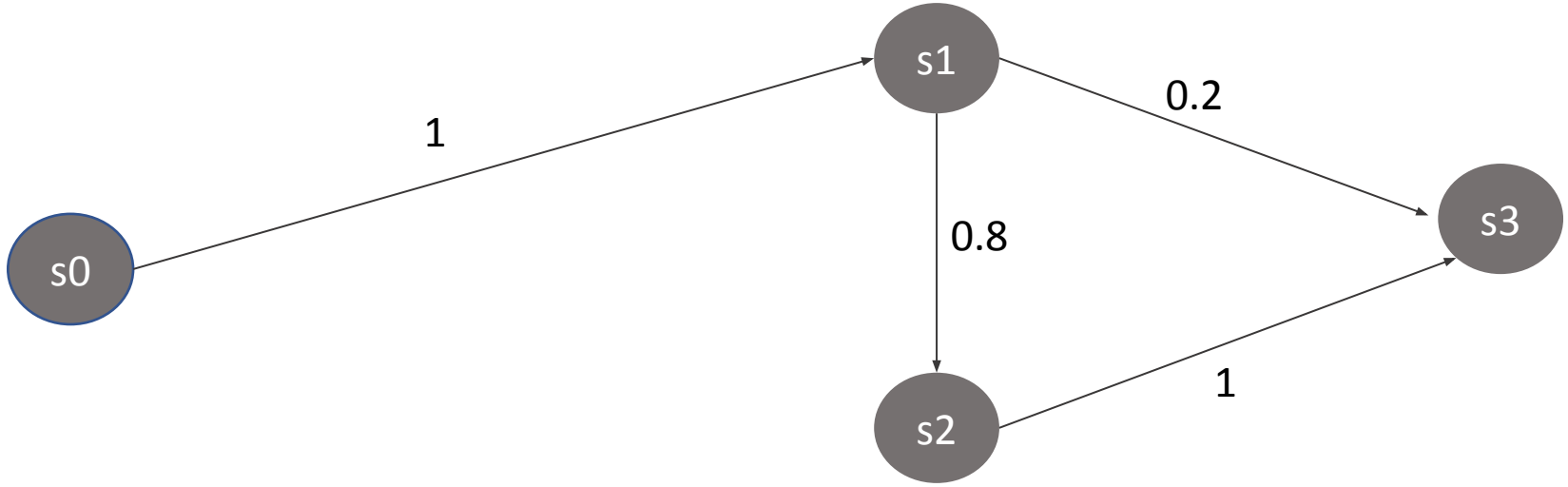
Марковская цепь с дискретным временем



Вероятностная линейная временная логика

Статистическая проверка моделей

Марковская цепь с дискретным временем



Вероятностная линейная временная логика

Расширение временной логики LTL. Временные операторы дополнены операторами вероятностными.
P > 0.5 Eventually (s3)

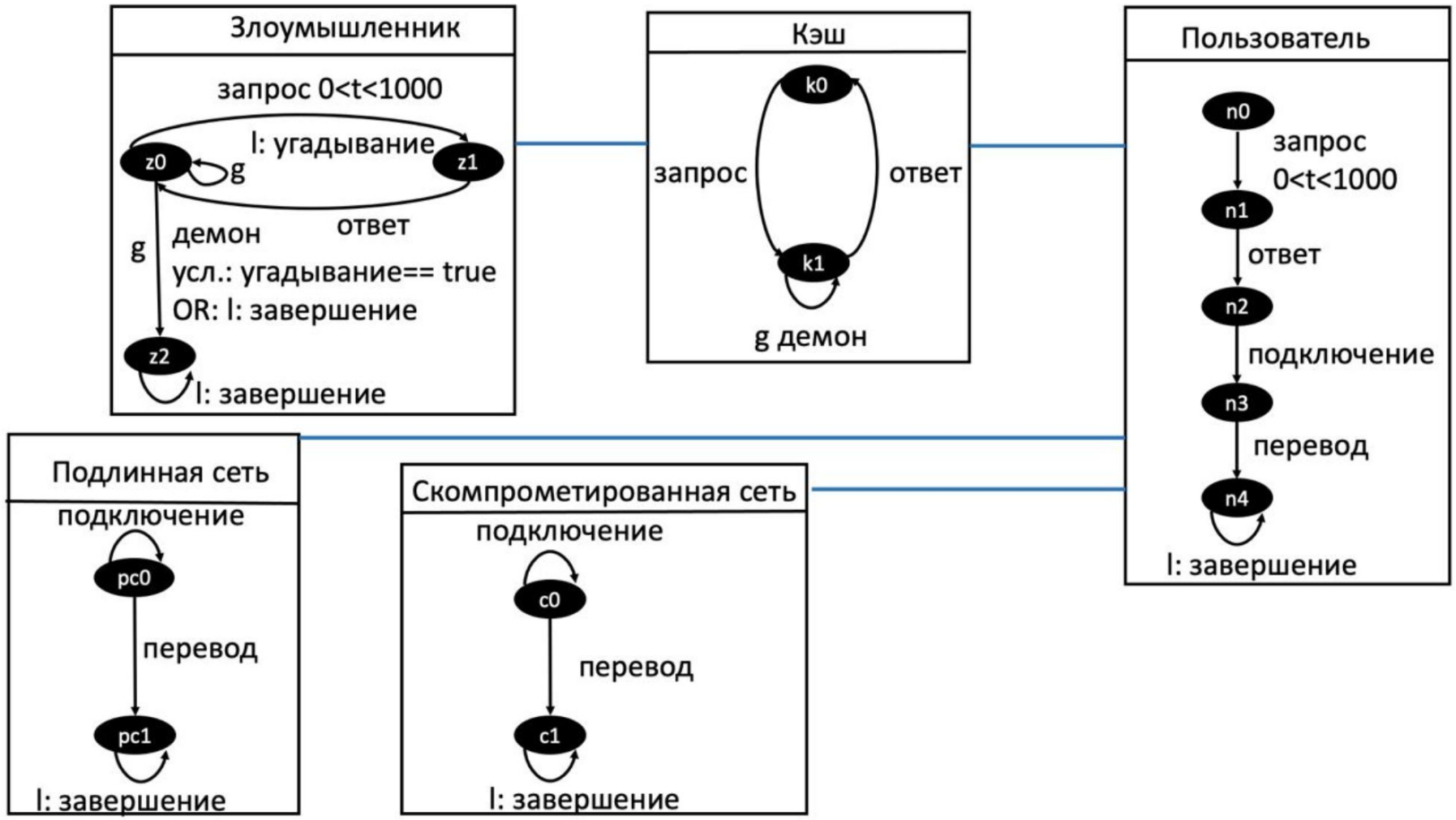
Моделирование атак на блокчейн системы с помощью фреймворка BIP (Behavior, Interaction, Priority)[1].

Исследовательский вклад, поставленные цели и используемые методы.

- Применить метод статистической проверки моделей для моделирования блокчейн атак, которые оказывают воздействия на наиболее значимые части блокчейн систем
- Из экспериментов получить вероятностную картину успешной атаки
- На основе экспериментальных результатов предложить решения как избежать атаки
- Оценить модель на реальных данных сети блокчейн, тем самым доказав применимость моделей в промышленной эксплуатации
- Использовать для верификации метод статистической проверки моделей и программный комплекс SBIP.

[1] Statistical Model Checking of Common Attack Scenarios on Blockchain. Fedotov I., Khritankov A. https://www.elibrary.ru/item.asp?id=47046151

DNS атака


Злоумышленник
запрос 0<t<1000
l: угадывание
z0
g
z1
g
демон
ответ
усл.: угадывание== true
OR: l: завершение
z2
l: завершение
Кэш
k0
запрос
ответ
k1
g демон
Пользователь
n0
запрос
0<t<1000
n1
ответ
n2
подключение
n3
перевод
n4
l: завершение
Подлинная сеть
подключение
pc0
перевод
pc1
l: завершение
Скомпрометированная сеть
подключение
c0
перевод
c1
l: завершение

DNS атака

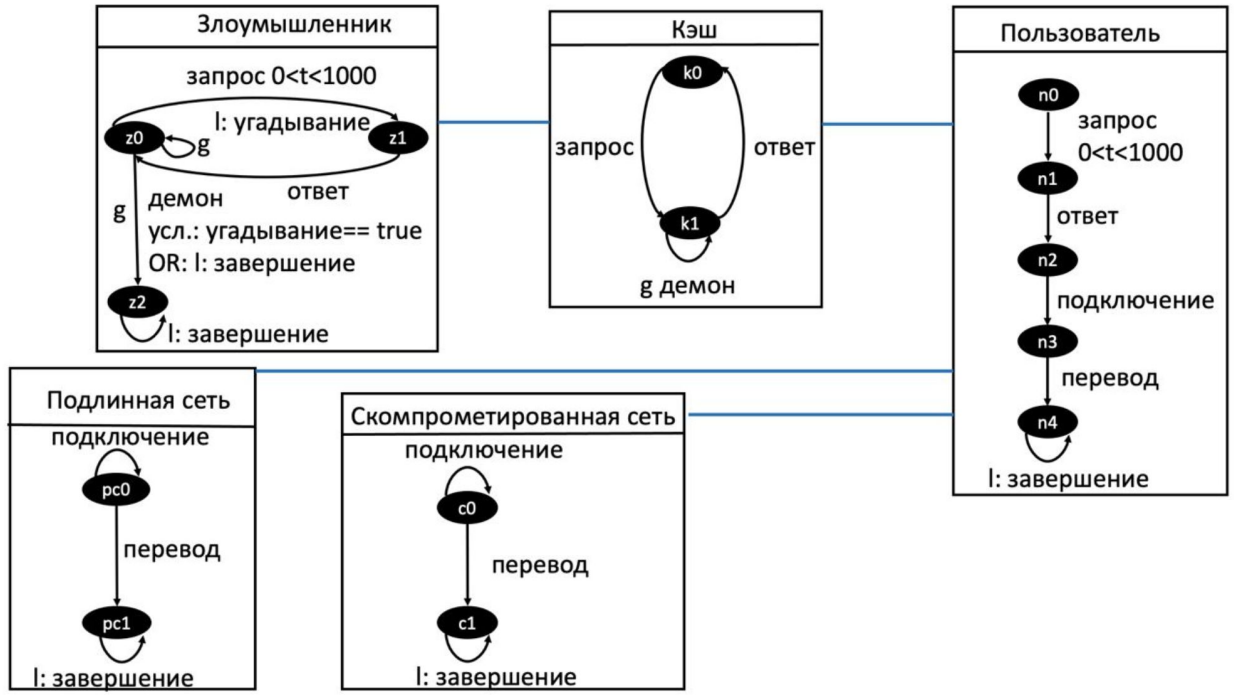
Пример математического представления модели.

Атомарный компонент *Пользователь*.

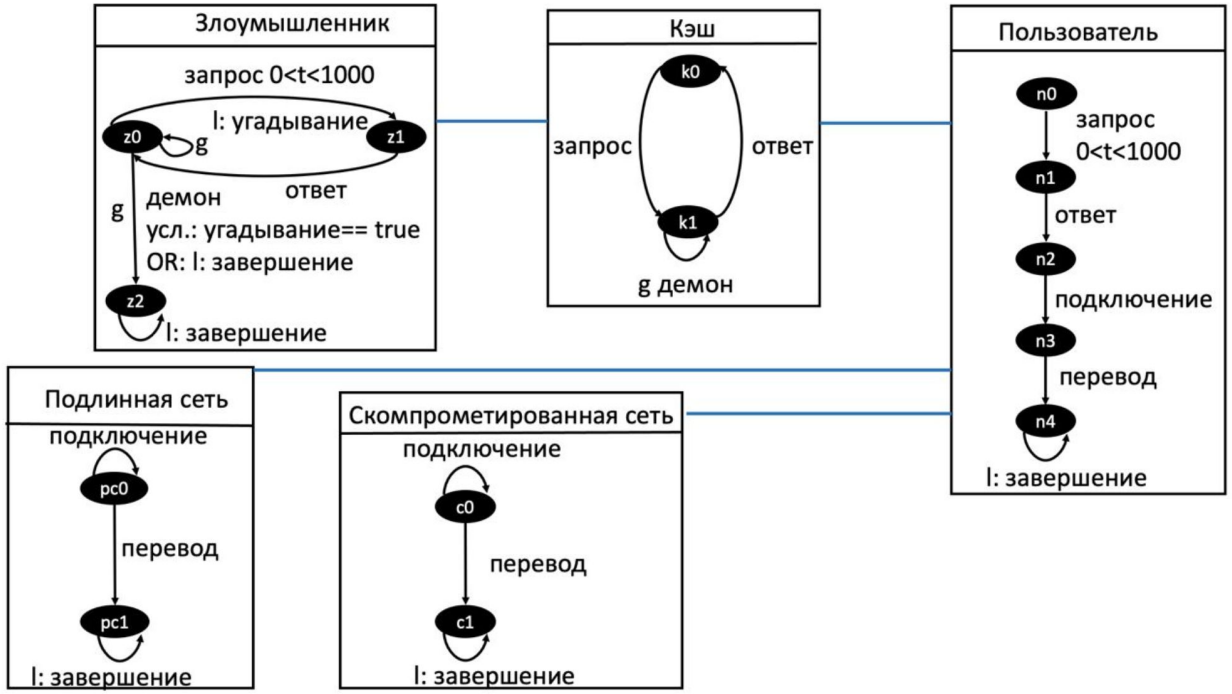
- $L = \{n_0, n_1, n_2, n_3, n_4\}$
- P = {запрос, ответ, подключение, перевод, завершение}
- T = $\{n_0 \times$ запрос $\times\ n_1,\ n_1 \times$ ответ $\times\ n_2,\ n_2 \times$ подключение $\times\ n_3,$ $n_3 \times$ перевод $\times\ n_4,\ n_4 \times$ завершение $\times\ n_4\ \}$
- $X = \{t, balance, transfer\_sum\}$
- $\{t_{n+1} = t_n + 1, 0 \leq n < 1000, t_0 = 0\},\ balance_{new} = balance - transfer\_sum$

Атомарный компонент *Скомпрометированная сеть*.

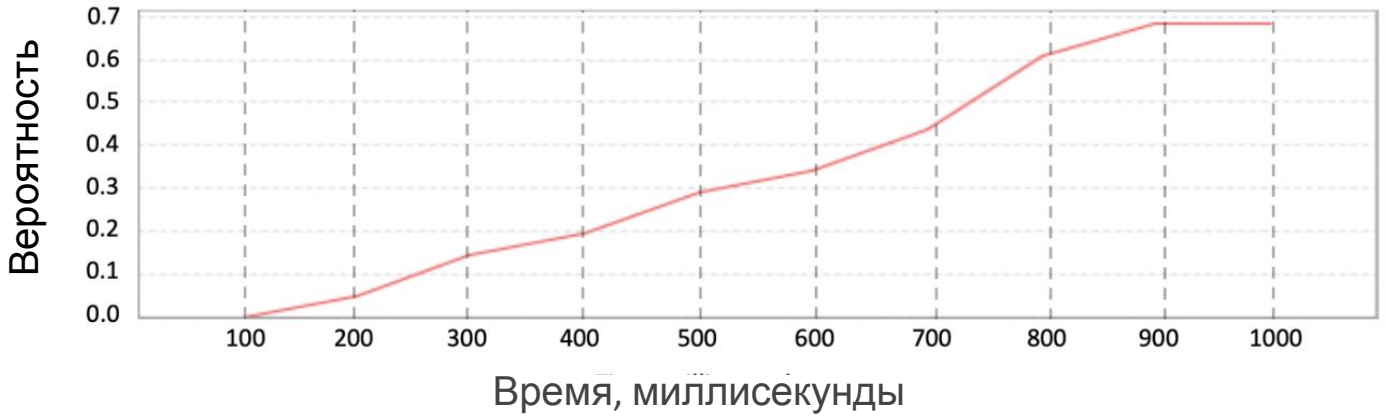
- $L = \{c_0, c_1\}$
- P = {подключение, перевод, завершение}
- T = $\{c_0 \times$ подключение $\times\ c_0,\ c_0 \times$ перевод $\times\ c_1,\ c_1 \times$ завершение $\times$ $c_1\ \}$
- $X = \{balance, transfer\_sum\}$
- $balance = \begin{cases} balance + transfer\_sum, & \text{демон} = \text{true} \\ balance, & \text{демон} \neq \text{true} \end{cases}$



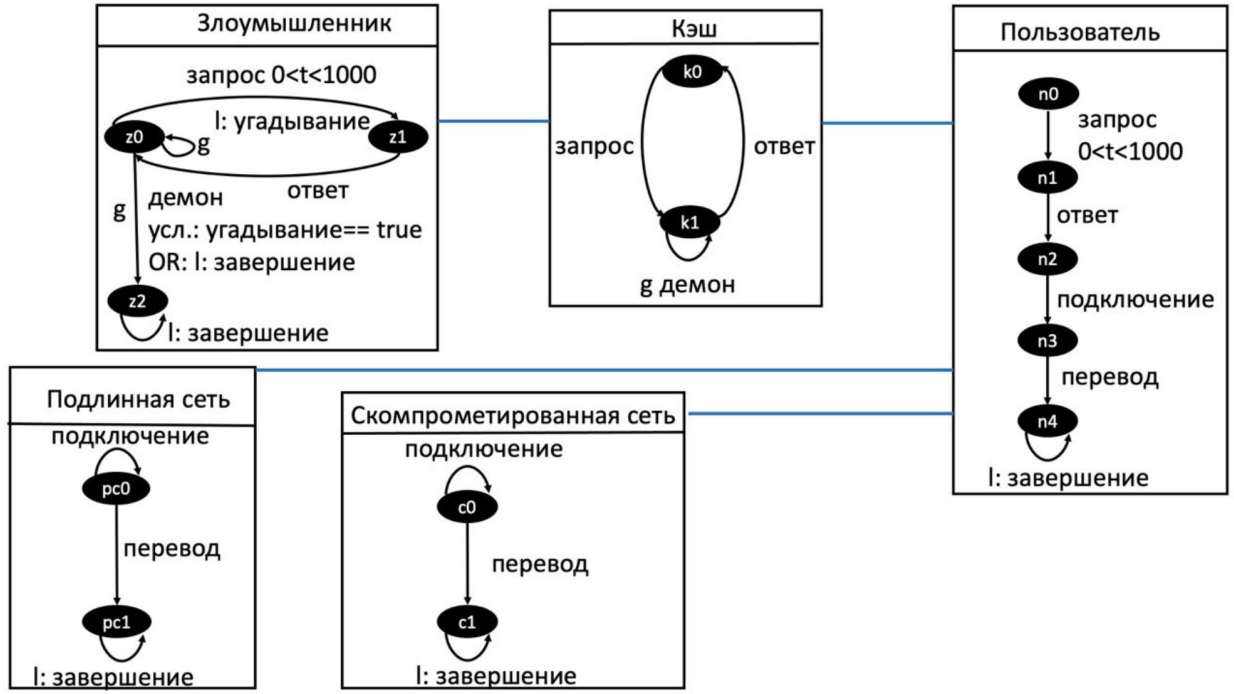
DNS атака



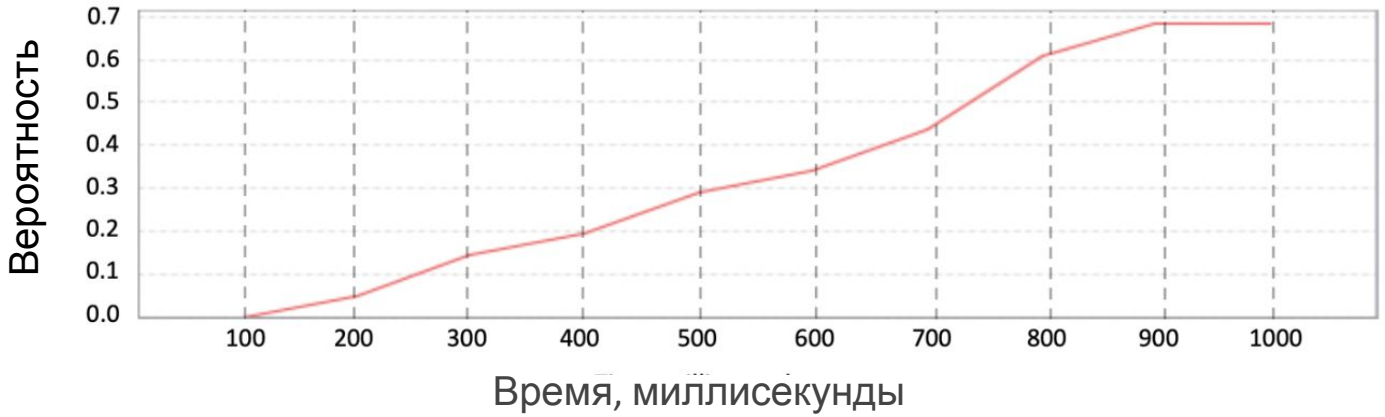
Зависимость времени с начала эксперимента от вероятности успеха злоумышленника



DNS атака

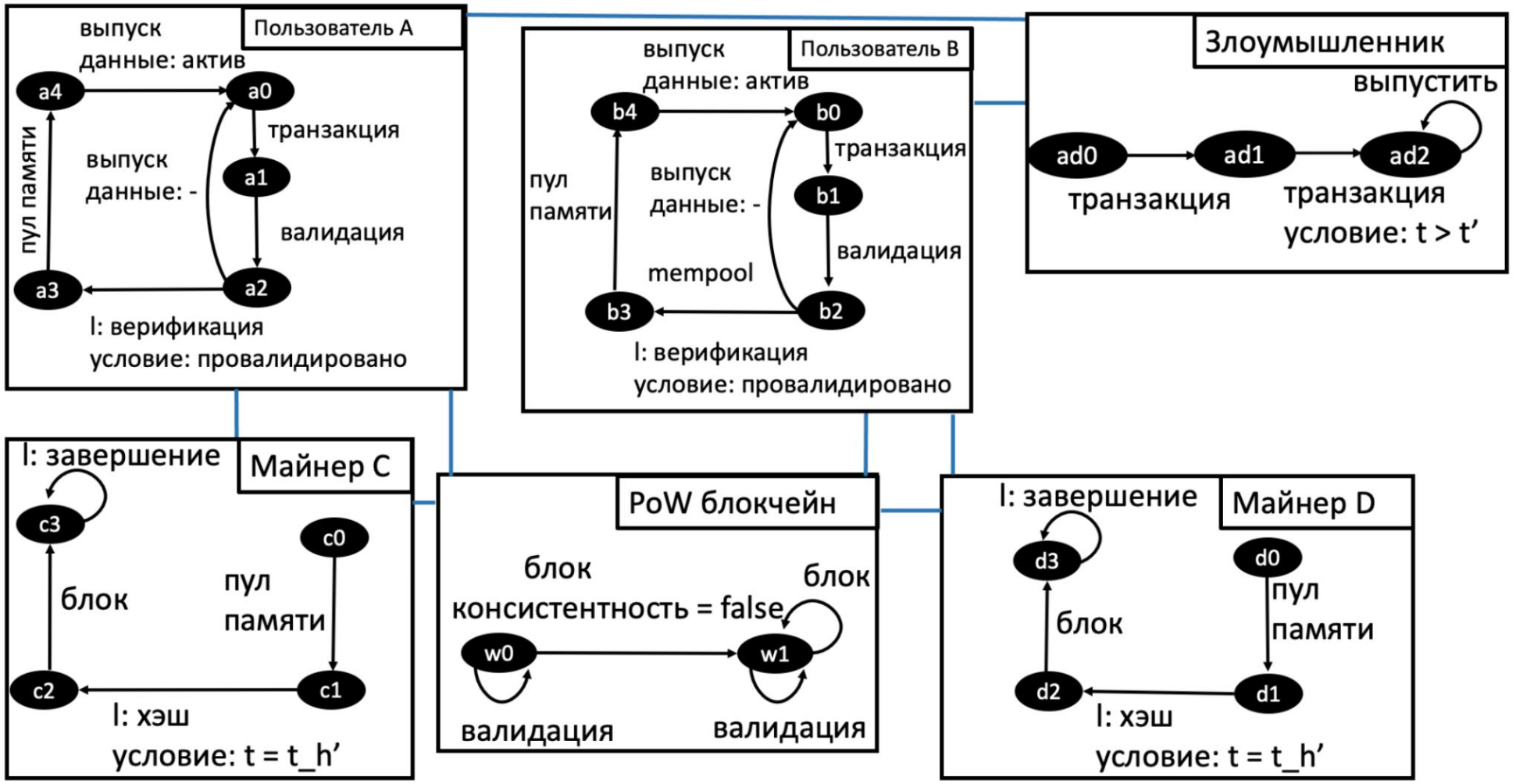


Зависимость времени с начала эксперимента от вероятности успеха злоумышленника



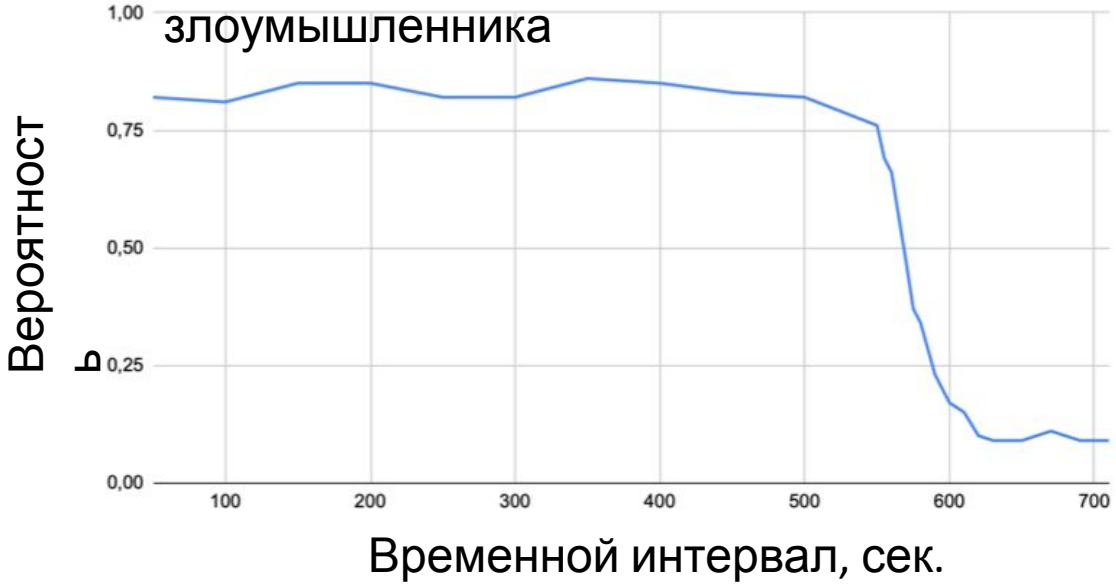
Решение: использовать криптографическое шифрование сервера DNS методом DNSSEC

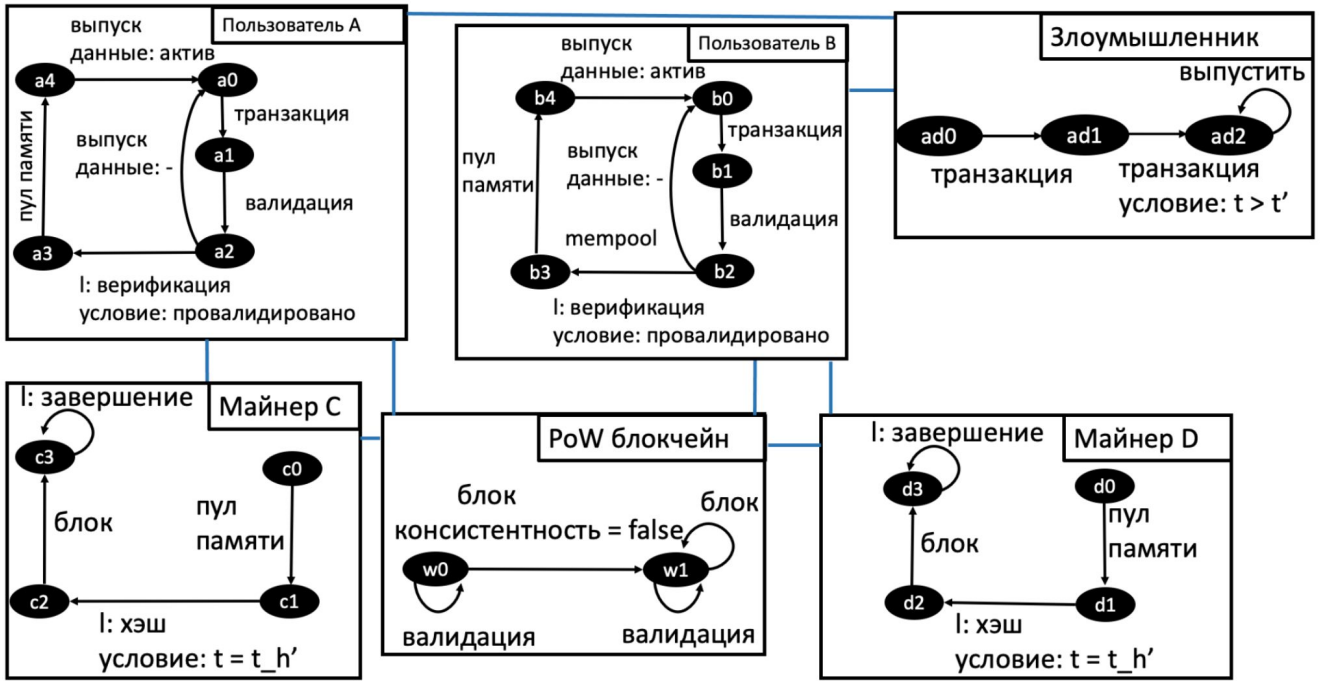
Двойная трата с заполнением пула памяти



Двойная трата с заполнением пула памяти

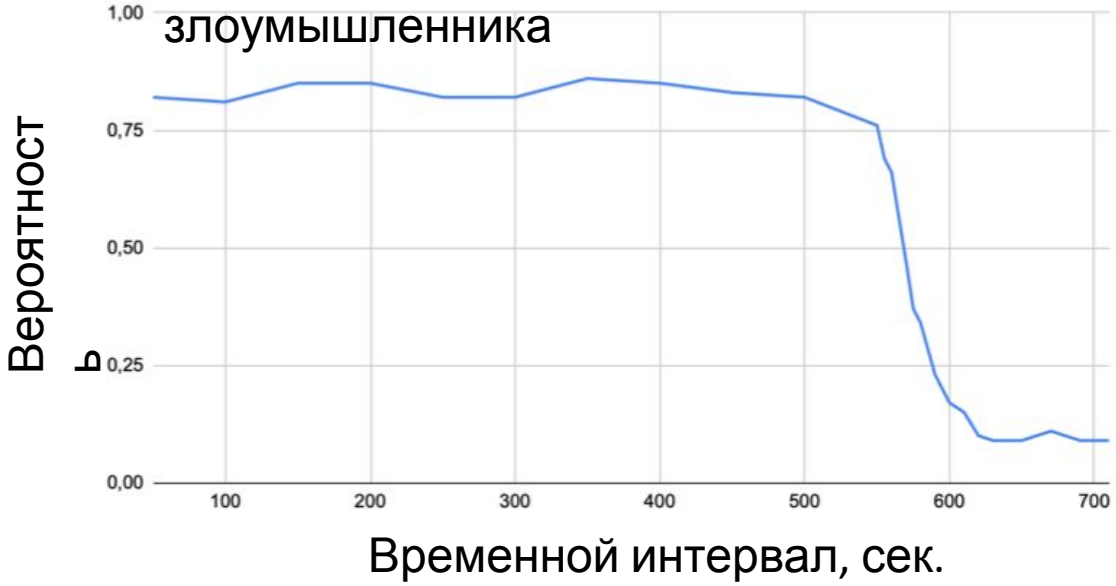
Зависимость времени между транзакциями от вероятности успеха злоумышленника

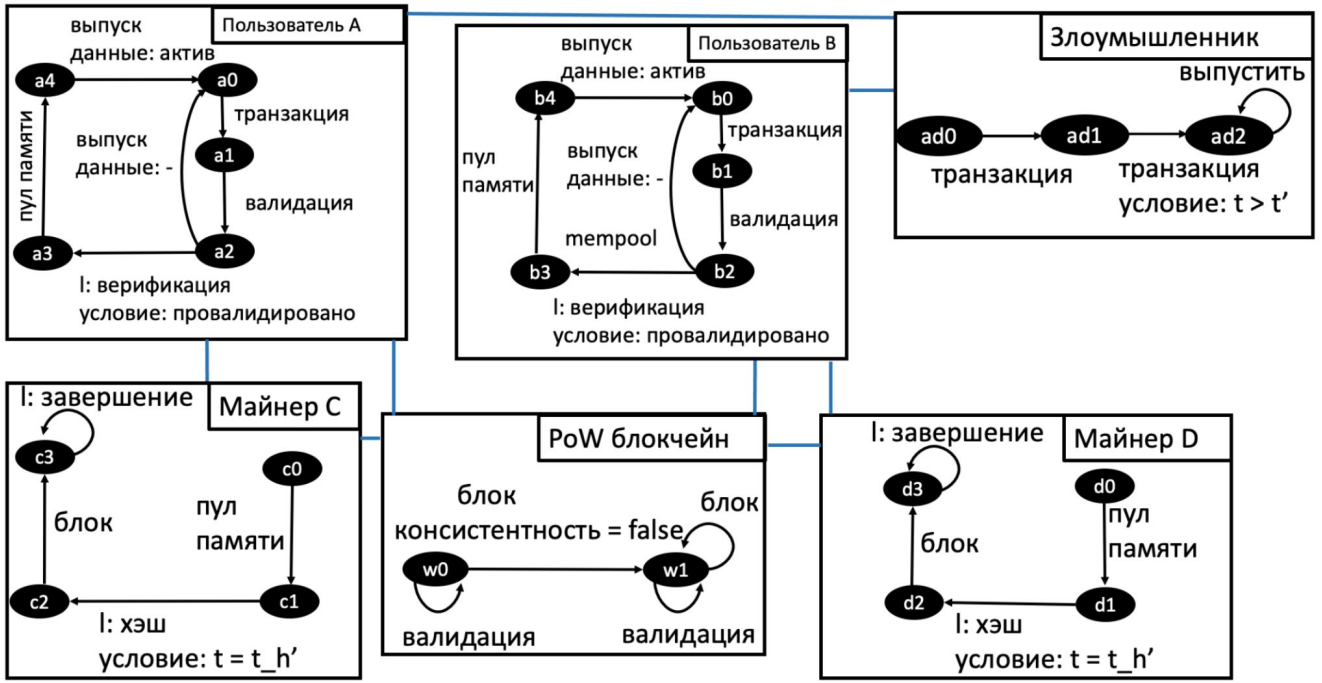




Двойная трата с заполнением пула памяти

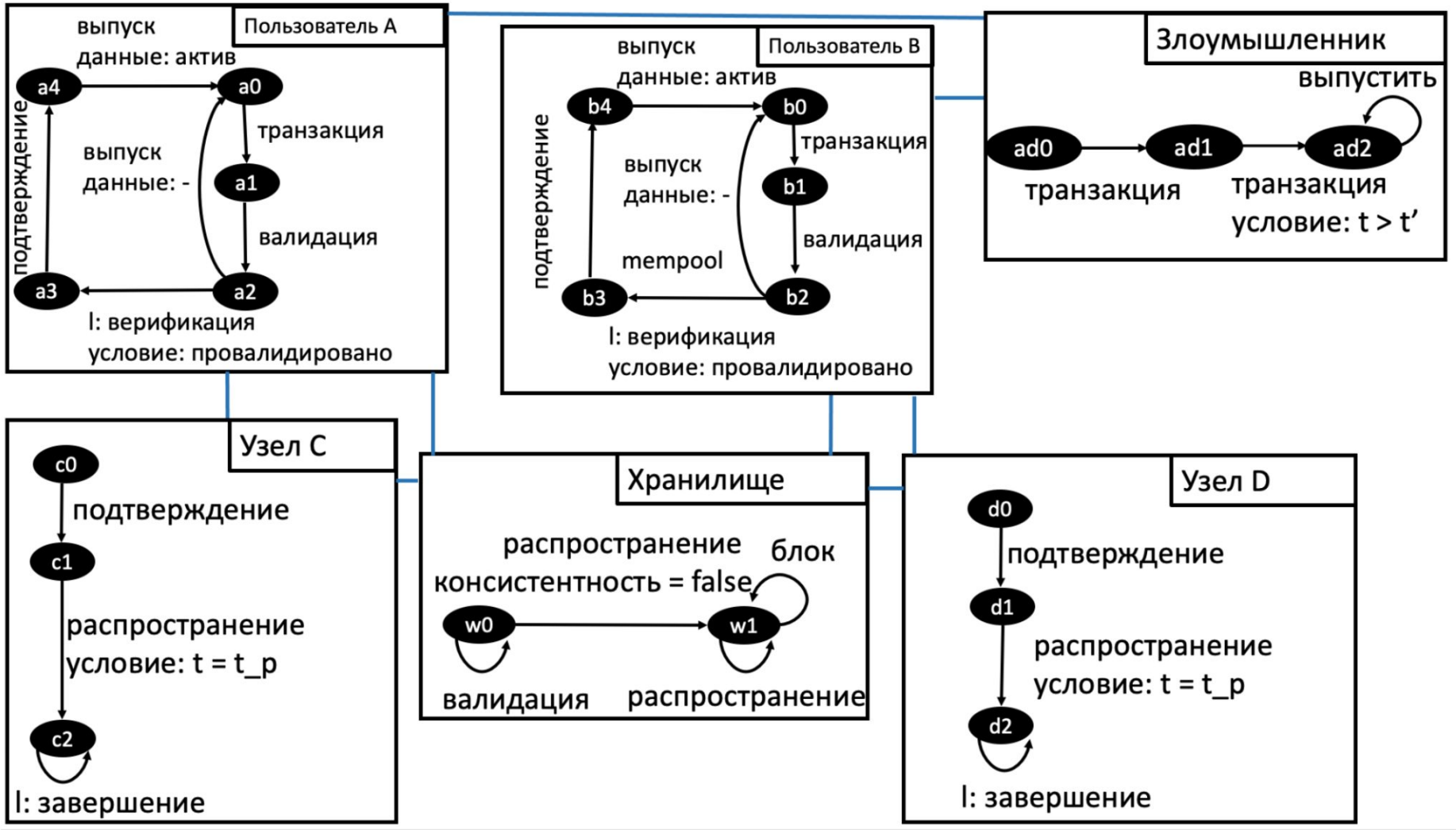
Зависимость времени между транзакциями от вероятности успеха злоумышленника





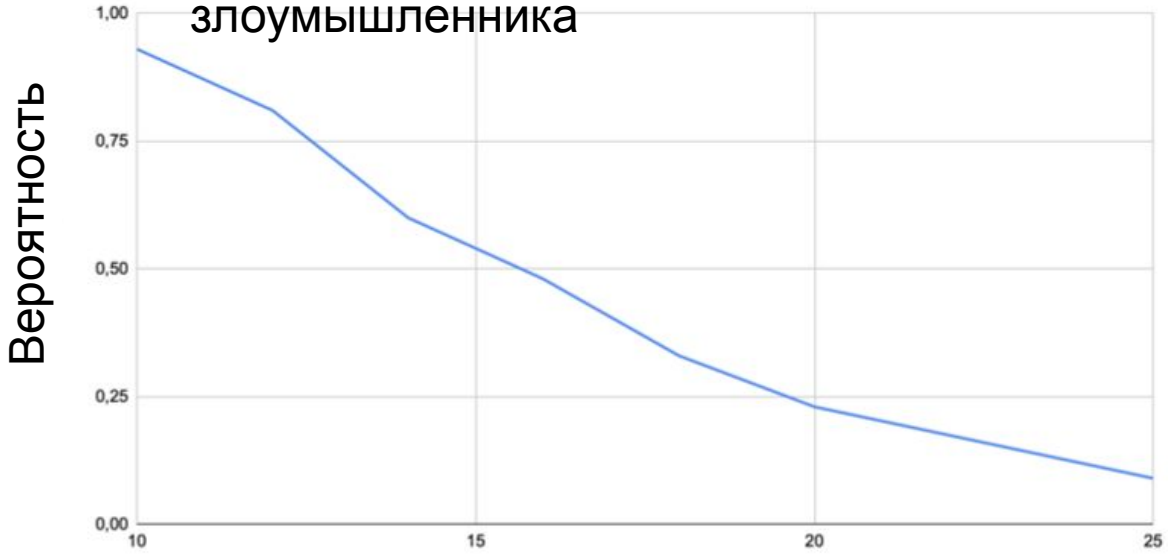
Решение: ограничить время между двумя транзакциями от одного пользователя

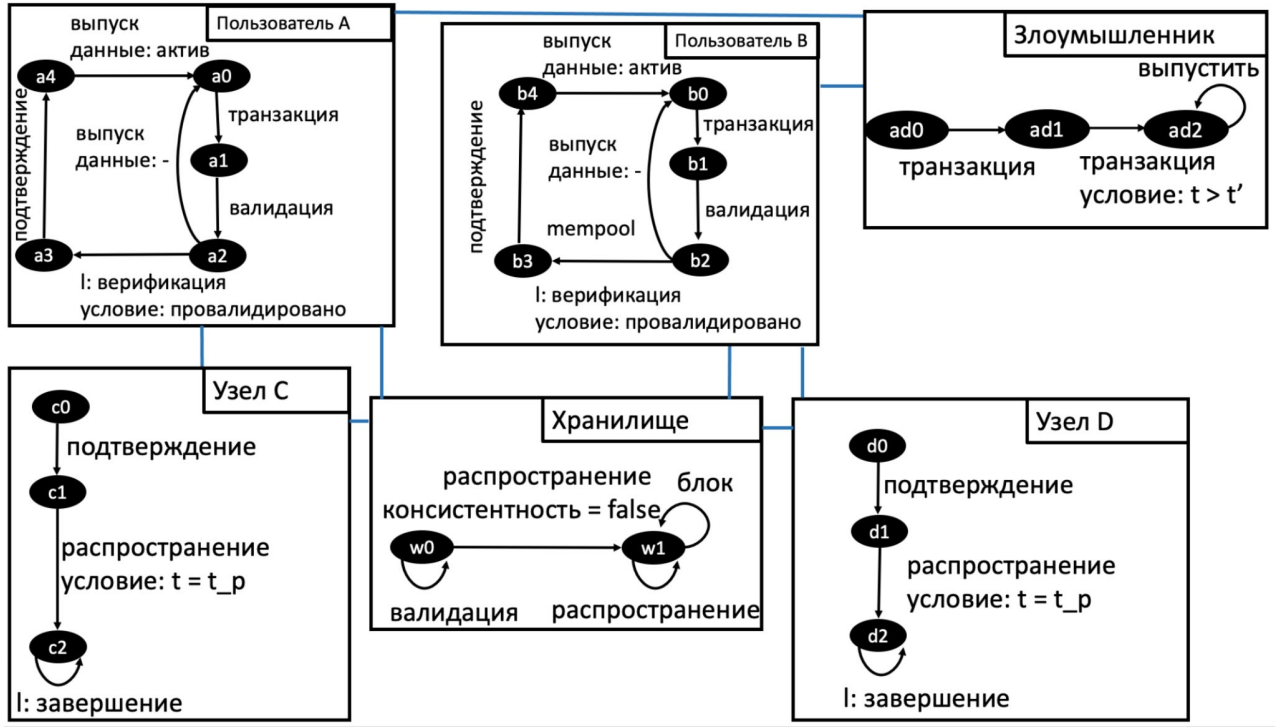
Двойная трата и задержка консенсуса



Двойная трата и задержка консенсуса

Зависимость времени между транзакциями от вероятности успеха злоумышленника





Решение: ограничить время между двумя транзакциями от одного пользователя

Верификация блокчейн систем с помощью статистической проверки моделей. Результаты.

- Смоделированы атаки, затрагивающие основные составляющие блокчейн систем
- Использованы реальные данные для оценки вероятности. Тем самым доказана возможность использования моделей в системах промышленной эксплуатации.
- Произведена оценка вероятности успешных атак. Предложены и проанализированы решения для предотвращения атак.
- Модели можно использовать как в публичных, так и в приватных сетях блокчейн.

Верификация протоколов консенсуса. Цели исследования.

- Разработать алгоритмы для моделирования протоколов консенсуса
- Разработать алгоритмы для представления спецификации протоколов консенсуса в формальном виде. Применить метод статистической проверки моделей для верификации консенсуса
- Разработать алгоритмы для оптимизации протоколов консенсуса
- Экспериментально подтвердить корректность алгоритмов.

Входные параметры для построения модели взвешенного консенсуса[1]

Параметры системы

- Множество организаций
- Вес для каждой организации
- Вероятность, что организация подтвердит/отклонит транзакцию

Спецификация системы

- Пороговое значение веса. Если организации с суммарным весом больше порогового значения примут транзакцию, то транзакция считается принятой всей системой.
- Пороговое значение вероятности. Если вся система достигает консенсуса по транзакции с вероятностью больше порогового значения, то транзакция считается валидной для системы

[1] Towards verification of probabilistic multi-party consensus protocols. Fedotov I., Khritankov A., Barger A., 2022

Алгоритм создания модели из параметров системы

Алгоритм 1: создание модели DTMC-creation

Входные данные: множество пар <организация, вероятность, вес> *orgs,* корень *root,* множество узлов *nodes*

Результат: Множество с узлами модели *nodes,* хранящих информацию о модели

если лист *orgs* не пустой, то:

извлечь из листа организацию *nextOrg* с $P_{nextOrg}$ и W

// построить поддерево с подтверждающим ответом:

создать узел $N_{nextOrg}$, родителем которого является *root*, с параметрами $P_{nextOrg}$ и W и подтверждающим ответом;

добавить узел $N_{nextOrg}$ с подтверждающим ответом родителя в множество *nodes*;

//запустить алгоритм рекурсивно для построения поддерева:

DTMC-creation(copy(orgs), $N_{nextOrg}$*, nodes)*;

//построить поддерево с ответом отказа:

создать узел $N_{nextOrg'}$, родителем которого является *root*, с параметрами $P_{nextOrg}$ и W и с ответом отказа;

добавить узел $N_{nextOrg'}$ с ответом отказа родителя в множество *nodes*;

// запустить алгоритм рекурсивно для построения поддерева:

DTMC-creation(orgs, $N_{nextOrg'}$*, nodes)*;

если лист *orgs* **пустой, то:**

создать два листовых узла из корня *root* с двумя переходами, соответствующими ответам подтверждения и отказа;

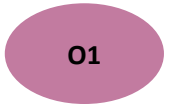
добавить листья в множество *nodes*;

завершение алгоритма;

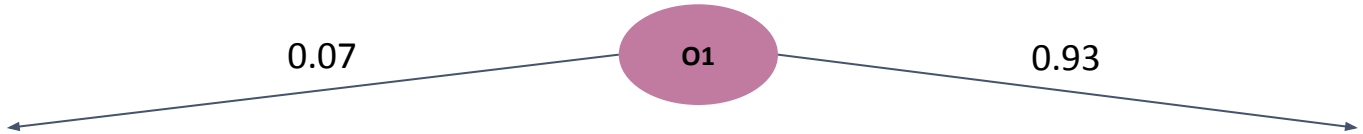
Пример построения модели и спецификации

- Организация 1. Вес: 1, вероятность отклонения транзакции: 0.07
- Организация 2. Вес: 3, вероятность отклонения транзакции: 0.01
- Организация 3. Вес: 2, вероятность отклонения транзакции: 0.02

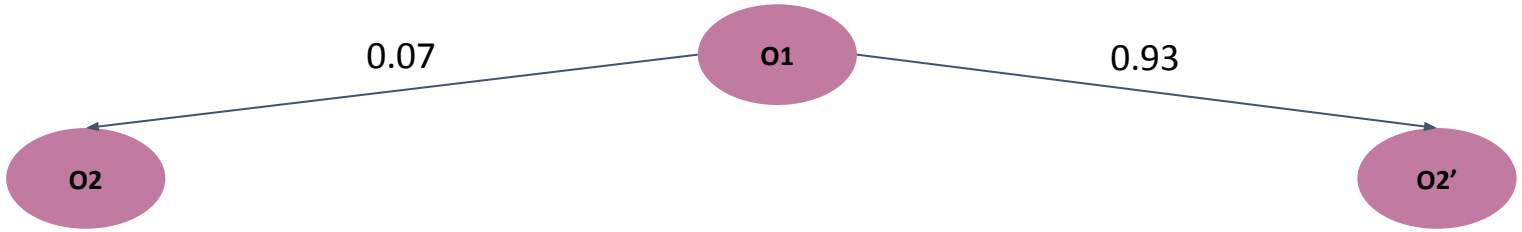
Многостороннее соглашение -> цепь Маркова



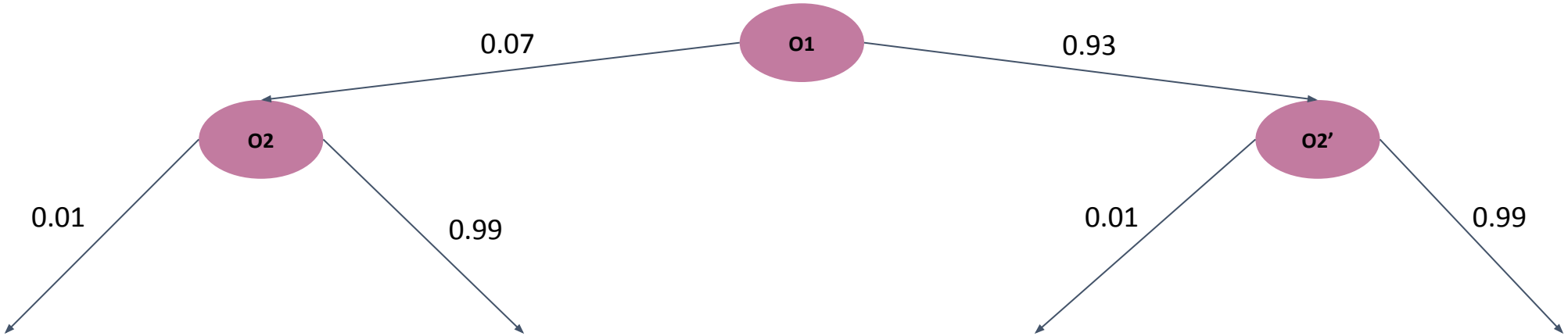
Многостороннее соглашение -> цепь Маркова



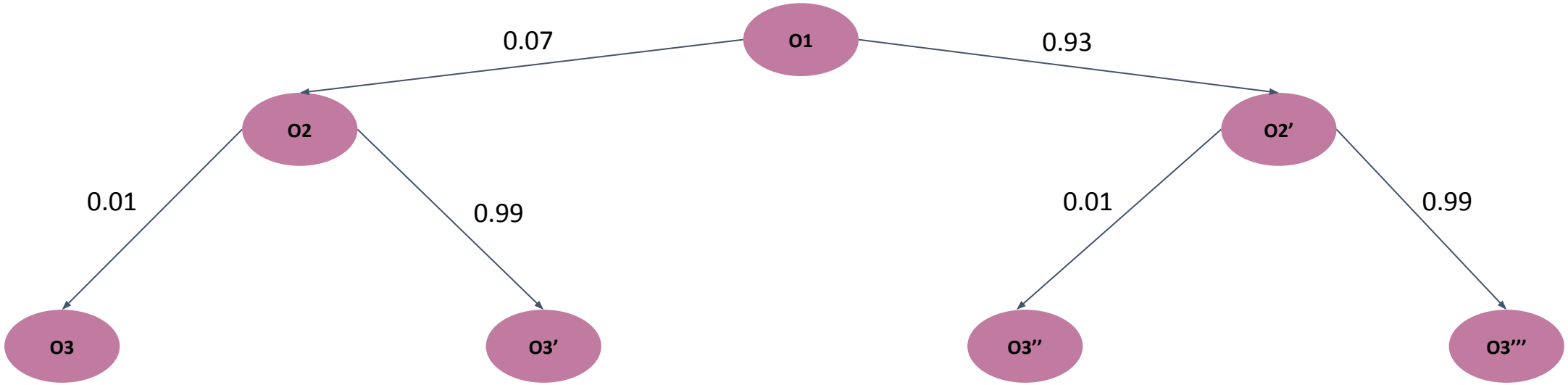
Многостороннее соглашение -> цепь Маркова



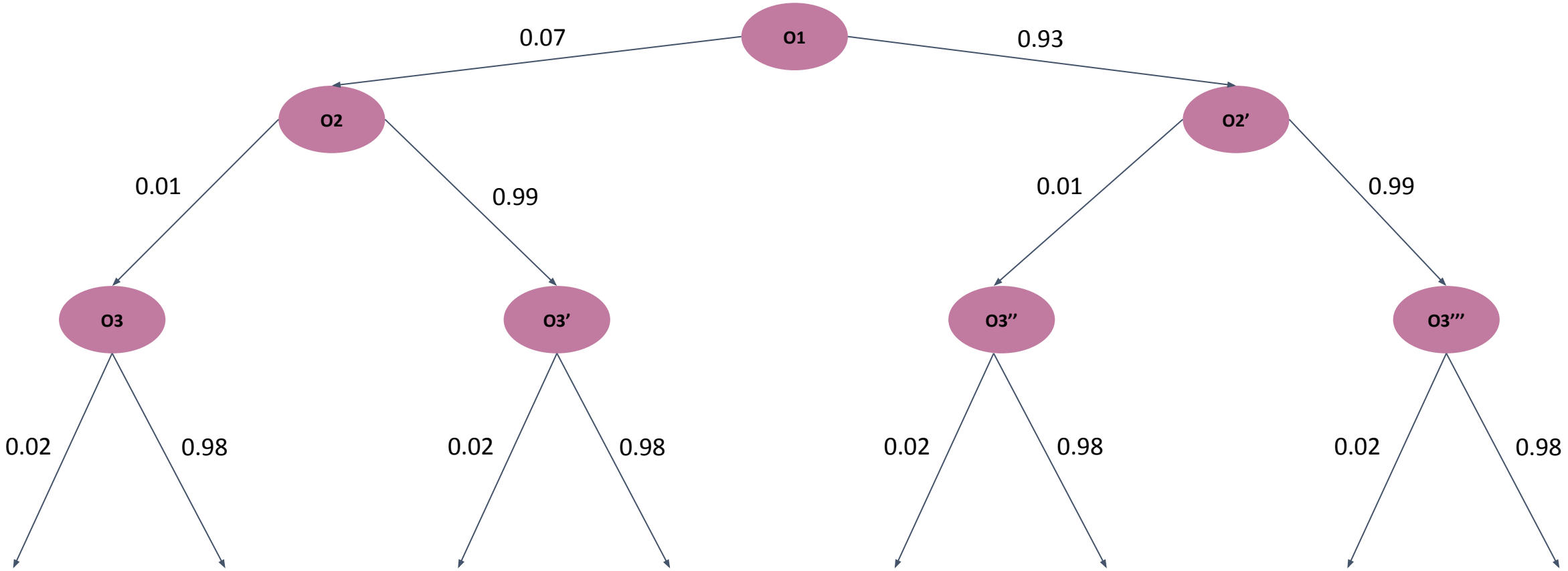
Многостороннее соглашение -> цепь Маркова



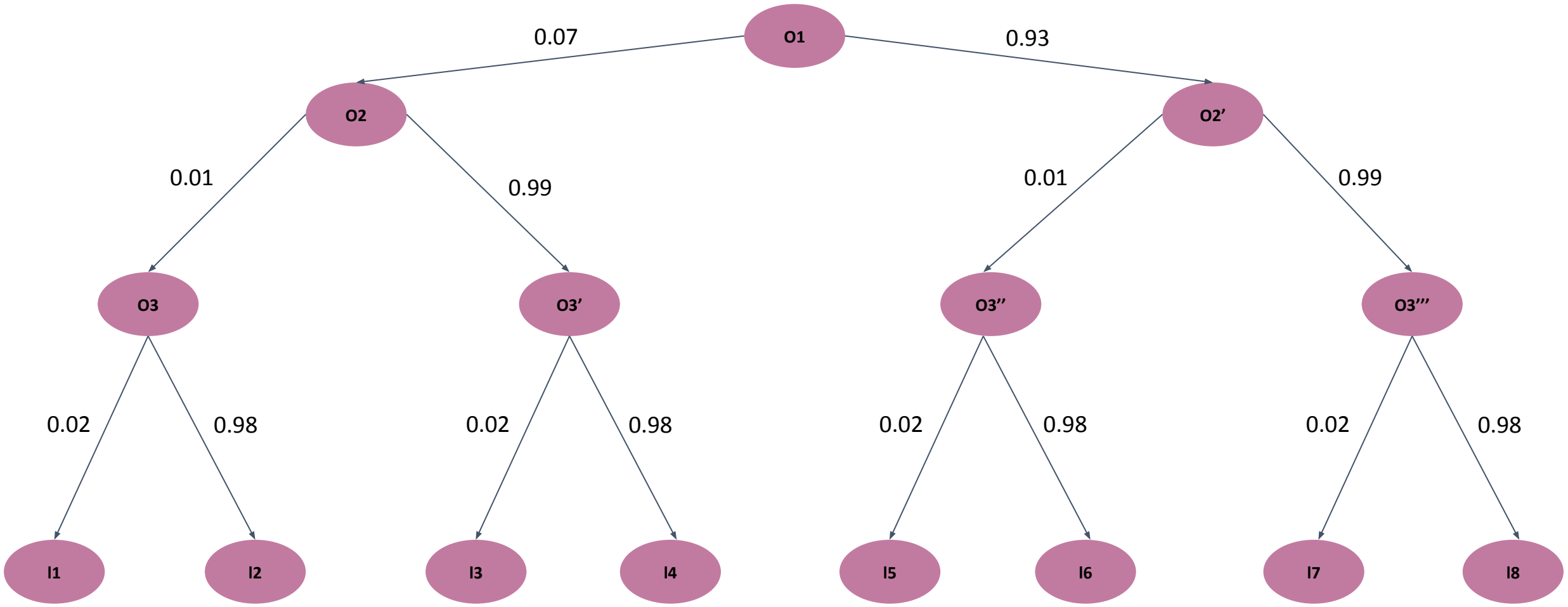
Многостороннее соглашение -> цепь Маркова



Многостороннее соглашение -> цепь Маркова



Многостороннее соглашение -> цепь Маркова



Алгоритм создания спецификации

Алгоритм 2: маркировка узлов labeling-sum

Входные данные: Множество узлов модели *nodes*, корневой узел *root*

Результат: множество узлов, маркированный общей суммой

Для каждого узла преемника N_s корня *root*:

    извлечь из *nodes* узел N_s

    присвоить N_s общий вес корня;

    если *root* дал утверждающий ответ, то

        добавить вес корня к общему весу N_s

    labeling-sum(N_s, nodes)

возвратить nodes;

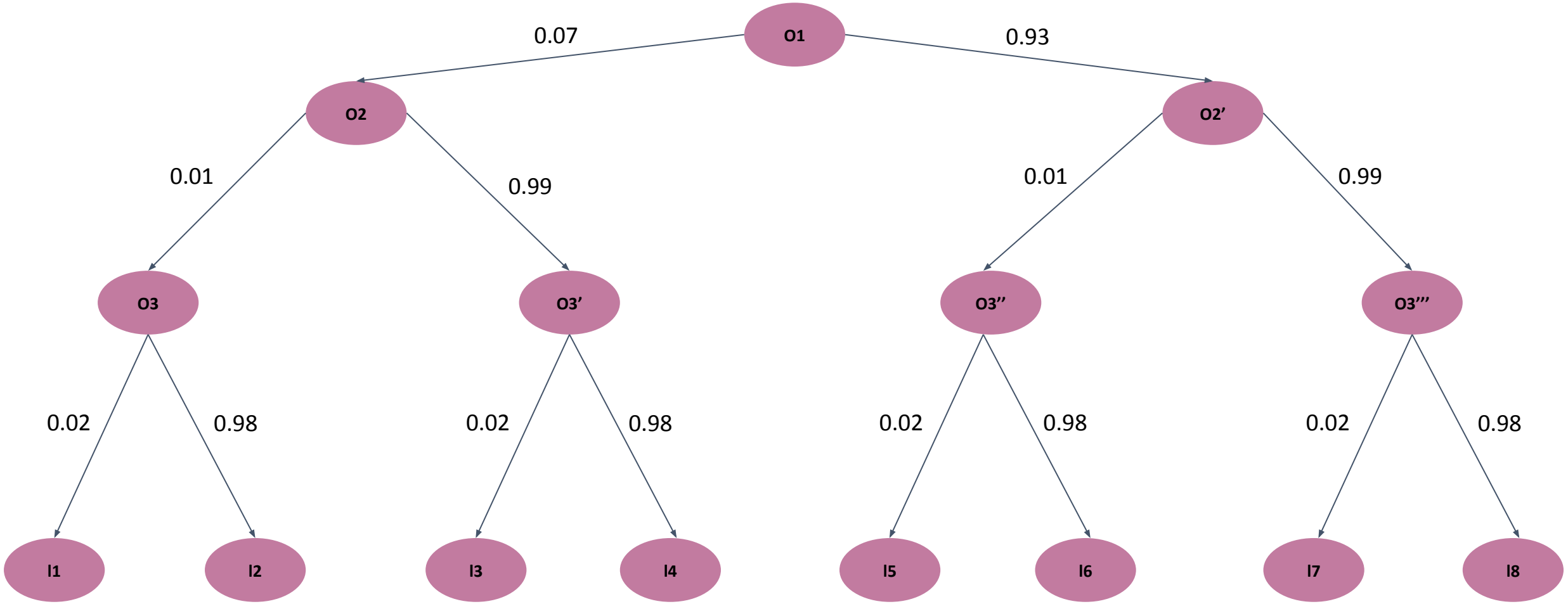
Многостороннее соглашение -> спецификация

- Организация 1. Вес: 1, вероятность отклонения транзакции: 0.07
- Организация 2. Вес: 3, вероятность отклонения транзакции: 0.01
- Организация 3. Вес: 2, вероятность отклонения транзакции: 0.02

- Порог вероятности: 0.95
- Порог веса: 5

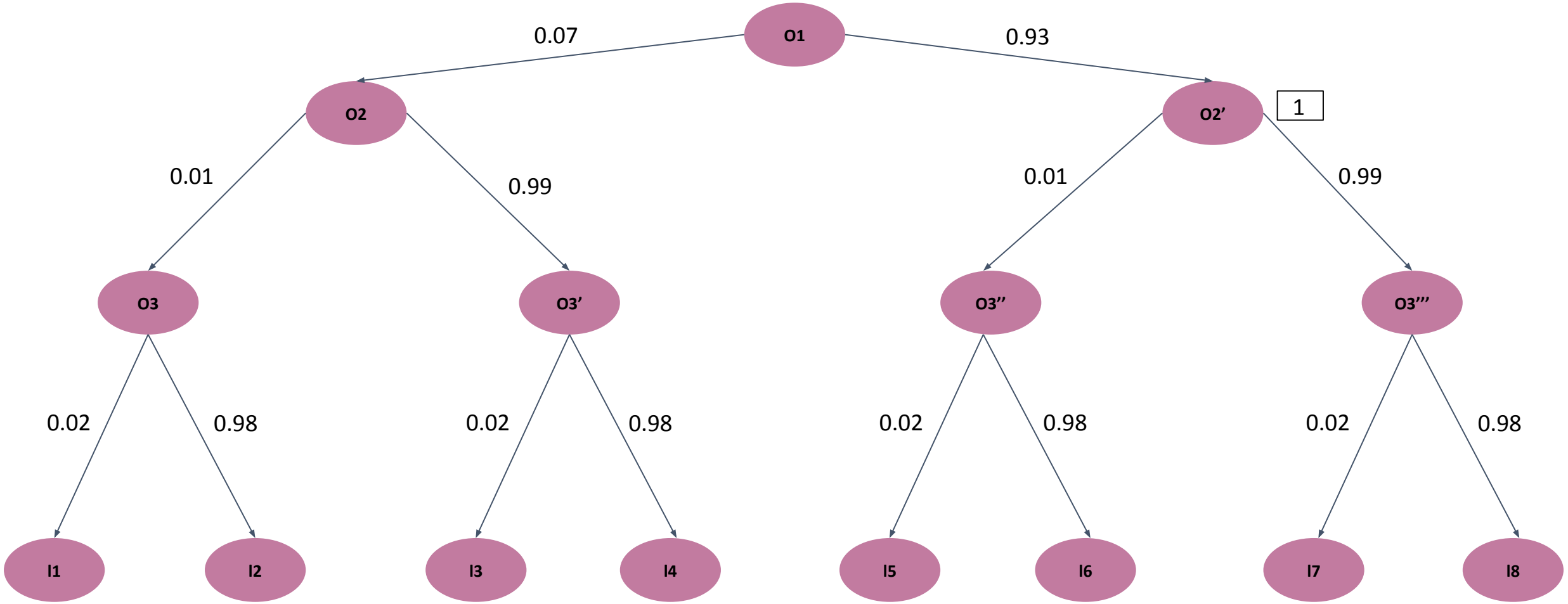
Многостороннее соглашение -> спецификация

- Порог вероятности: 0.95
- Порог веса: 5



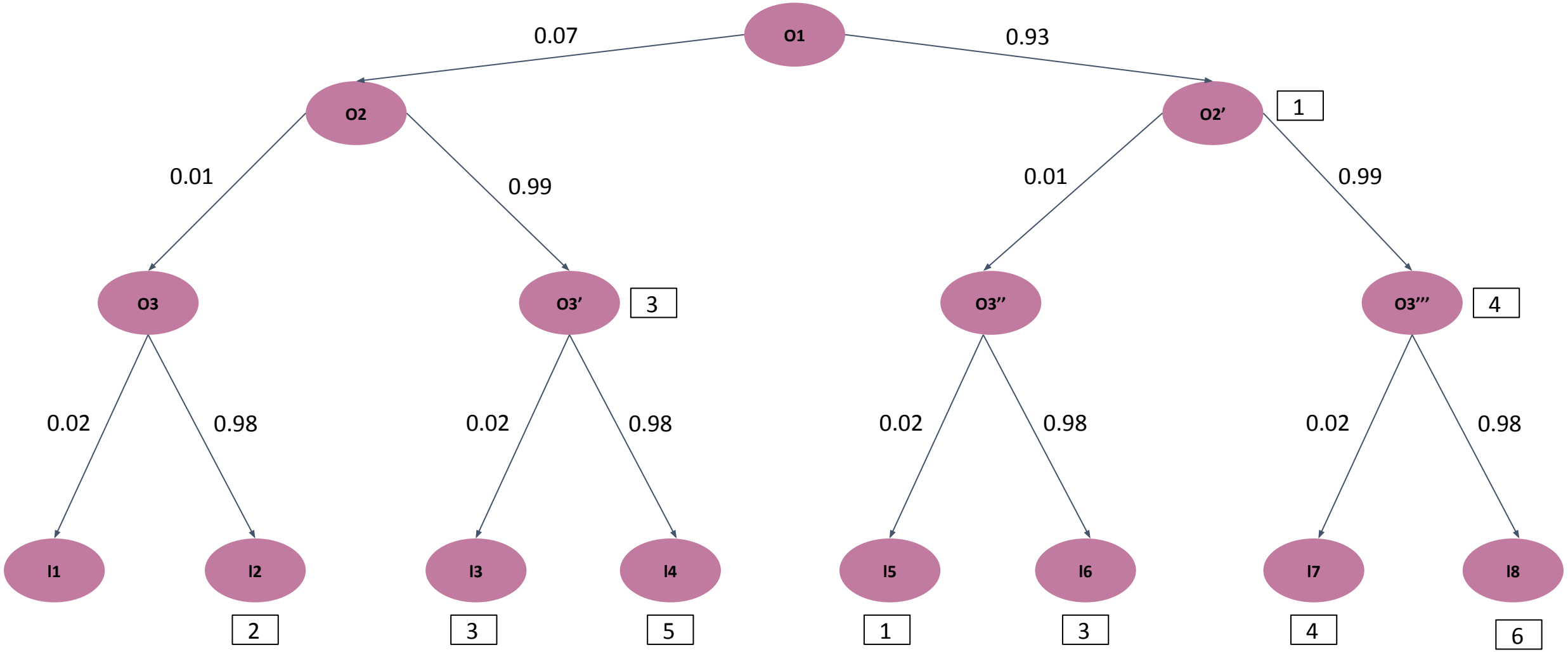
Многостороннее соглашение -> спецификация

- Порог вероятности: 0.95
- Порог веса: 5



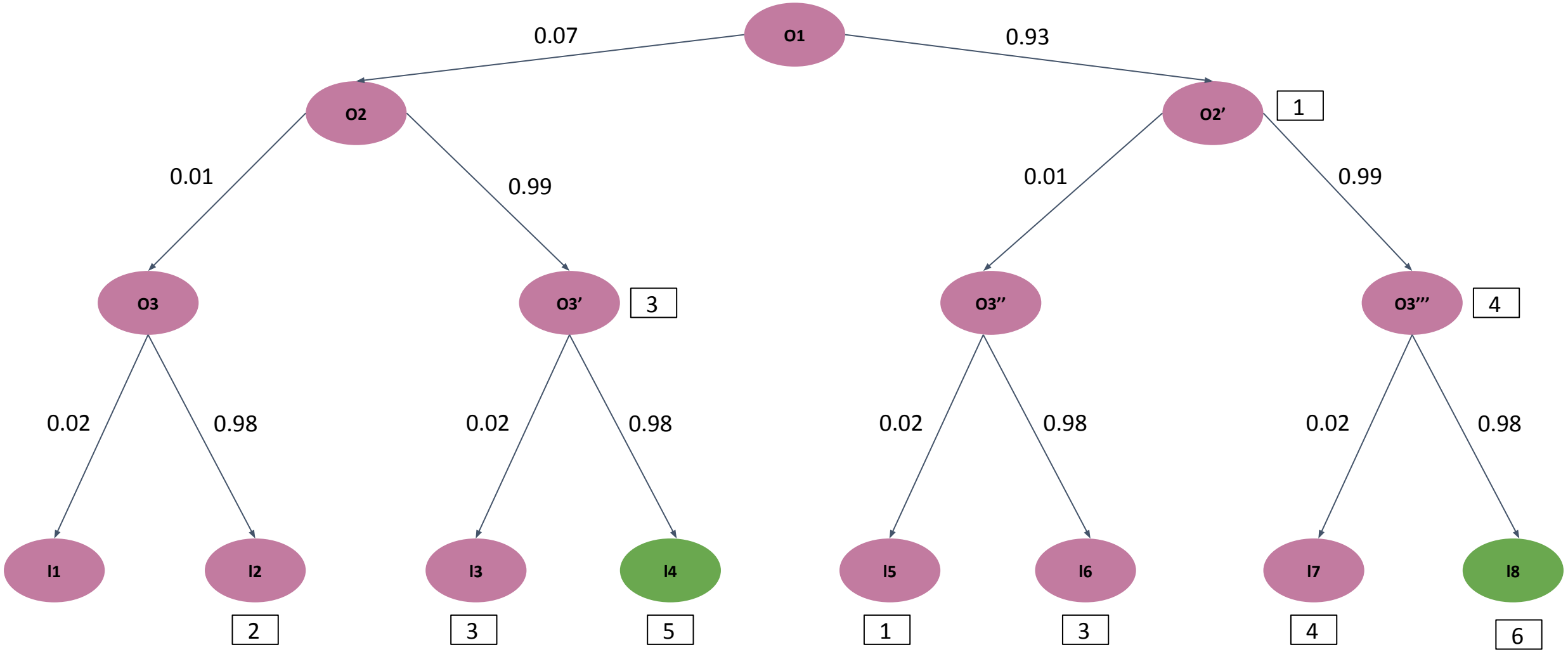
Многостороннее соглашение -> спецификация

- Порог вероятности: 0.95
- Порог веса: 5



Многостороннее соглашение -> спецификация

- Порог вероятности: 0.95
- Порог веса: 5



Экспериментальные результаты

Зависимость вероятности принятия транзакции одной организацией от вероятности подтверждения всей системой

Зависимость порога вероятности от порога значения веса

Weight threshold	Probability threshold
1	0.9990 +/-0.0004
2	0.9997 +/-0.0005
3	0.9989 +/- 0.001
4	0.9894 +/- 0.0003
5	0.9717 +/- 0.0005
6	0.9035 +/- 0.0009
7	0

Верификация взвешенного консенсуса, заданного в виде логической формулы[1]

Параметры системы

- Множество организаций
- Вероятность, что организация подтвердит/отклонит транзакцию

Спецификация системы

Логическая формула. Каждой переменной логической формулы присваивается значение ответа соответствующей организации.

[1] Автоматическая верификация многосторонних соглашений и планирование отправки сообщений в системах распределенного реестра. Федотов И., Хританков А., Обидаре М., 2022

Алгоритм создания модели из параметров системы для консенсуса в виде логической формулы

Алгоритм 1: создание модели DTMC-creation

Входные данные: множество пар <организация, вероятность> *orgs,* корень *root,* множество узлов *nodes*
Результат: Множество с узлами модели *nodes,* хранящих информацию о модели
если лист *orgs* не пустой, то:
извлечь из листа организацию *nextOrg* с вероятностью $P_{nextOrg}$
// построить поддерево с подтверждающим ответом:
создать узел $N_{nextOrg}$, родителем которого является *root*, с параметром $P_{nextOrg}$ и подтверждающим ответом;
добавить узел $N_{nextOrg}$ с подтверждающим ответом родителя в множество *nodes*;
//запустить алгоритм рекурсивно для построения поддерева:
DTMC-creation(copy(orgs),$N_{nextOrg}$*, nodes)*;
//построить поддерево с ответом отказа:
создать узел $N_{nextOrg}$, родителем которого является *root*, с параметром $N_{nextOrg}$ и с ответом отказа;
добавить узел $N_{nextOrg}$ с ответом отказа родителя в множество *nodes*;
// запустить алгоритм рекурсивно для построения поддерева:
DTMC-creation(orgs, $N_{nextOrg}$*, nodes)*;
если лист *orgs* **пустой, то:**
создать два листовых узла из корня *root* с двумя переходами, соответствующими ответам подтверждения и отказа;
добавить листья в множество *nodes*;
завершение алгоритма.

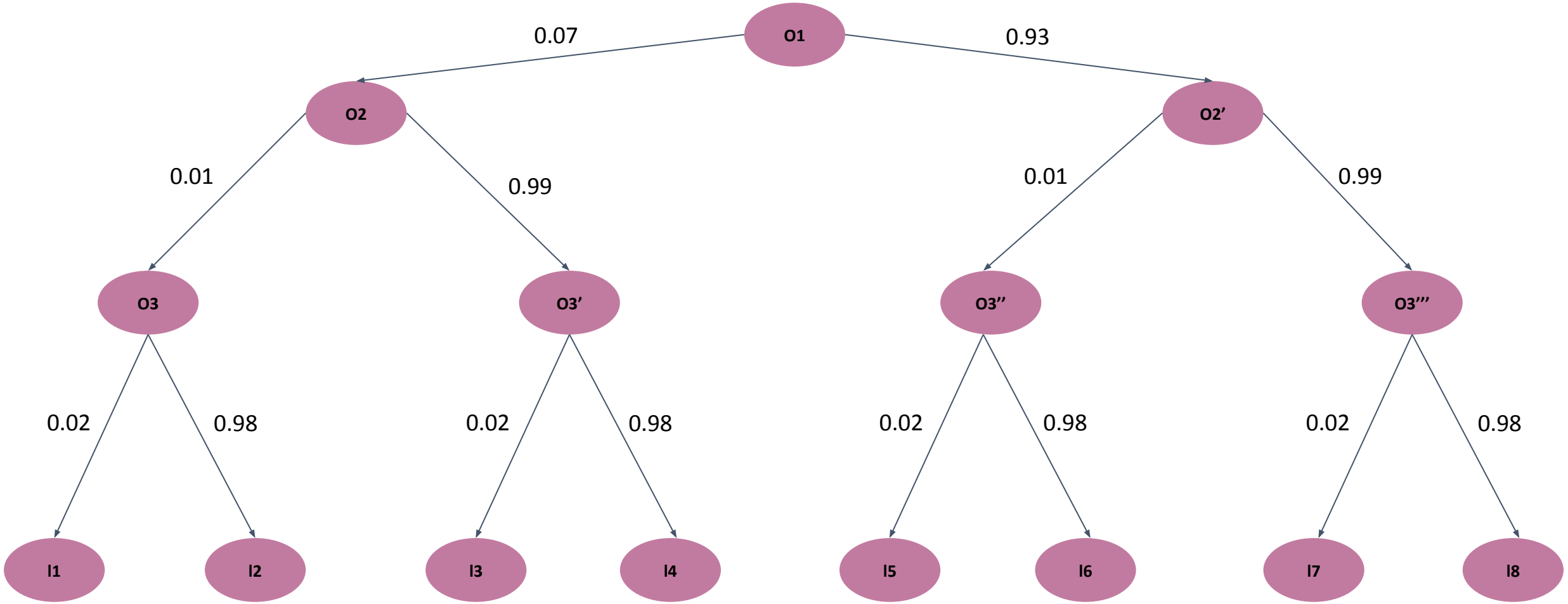
Алгоритм создания спецификации на модели для для консенсуса в виде логической формулы

Алгоритм 2: создание спецификации для модели DTMC

Входные данные: Множество узлов модели *nodes*, спецификация в виде ДНФ *s*
Результат: спецификация в виде pLTL *s_new*
инициализировать множество листовых узлов *leafs*;
Для каждого литерала *lit* из спецификации *s* обойти дерево вглубь:
если организация, соответствующая узлу, встречается в *lit*, то совершить переход, соответствующий подтверждению транзакции;
если не встречается, то продолжить переход по двум поддеревьям: одно соответствует подтверждению, второе — отклонению транзакции;
каждый лист, который был достигнут при обходе вглубь, добавить в множество *leafs*;
s_new = дизъюнкция узлов из множества *leafs*;
добавить в начало *s_new* временной оператор *F*;
добавить в начало *s_new* вероятностный оператор *P*;
возвратить формулу *s_new*.

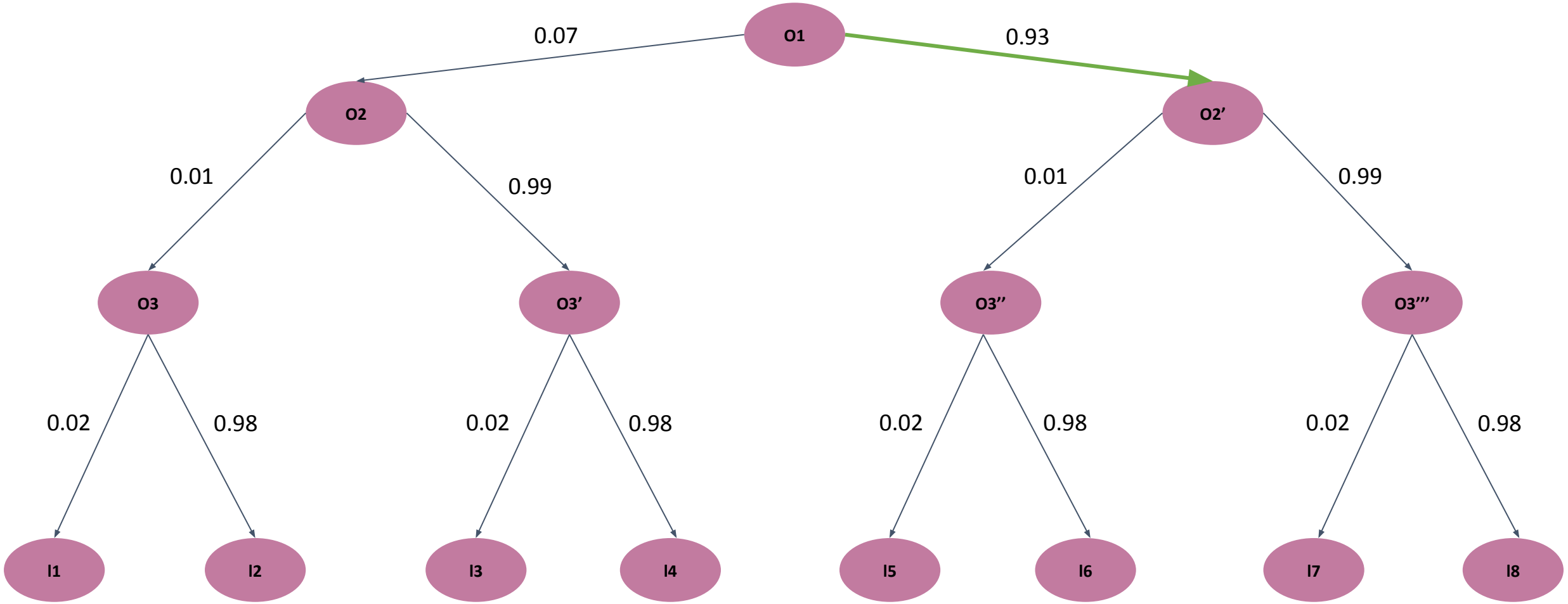
Многостороннее соглашение -> спецификация

(Организация 1 И Организация 2) ИЛИ (Организация 2 И Организация 3)



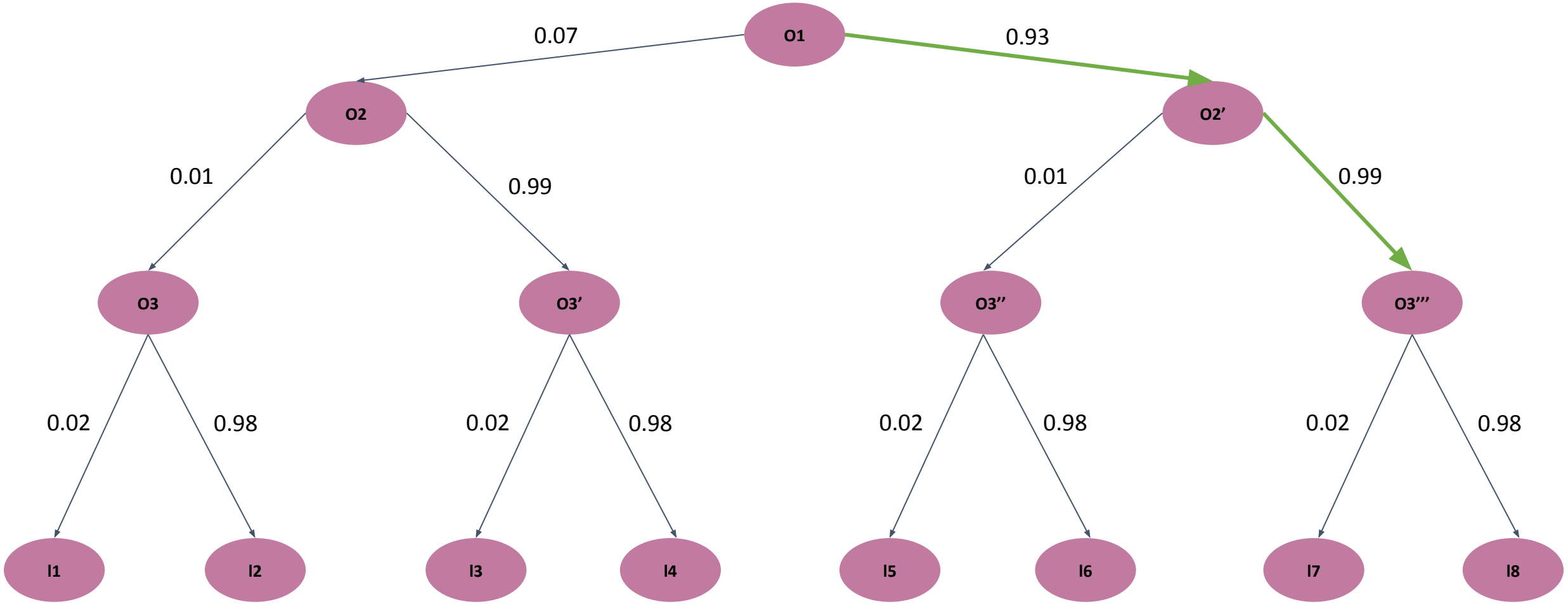
Многостороннее соглашение -> спецификация

(Организация 1 И Организация 2) ИЛИ (Организация 2 И Организация 3)



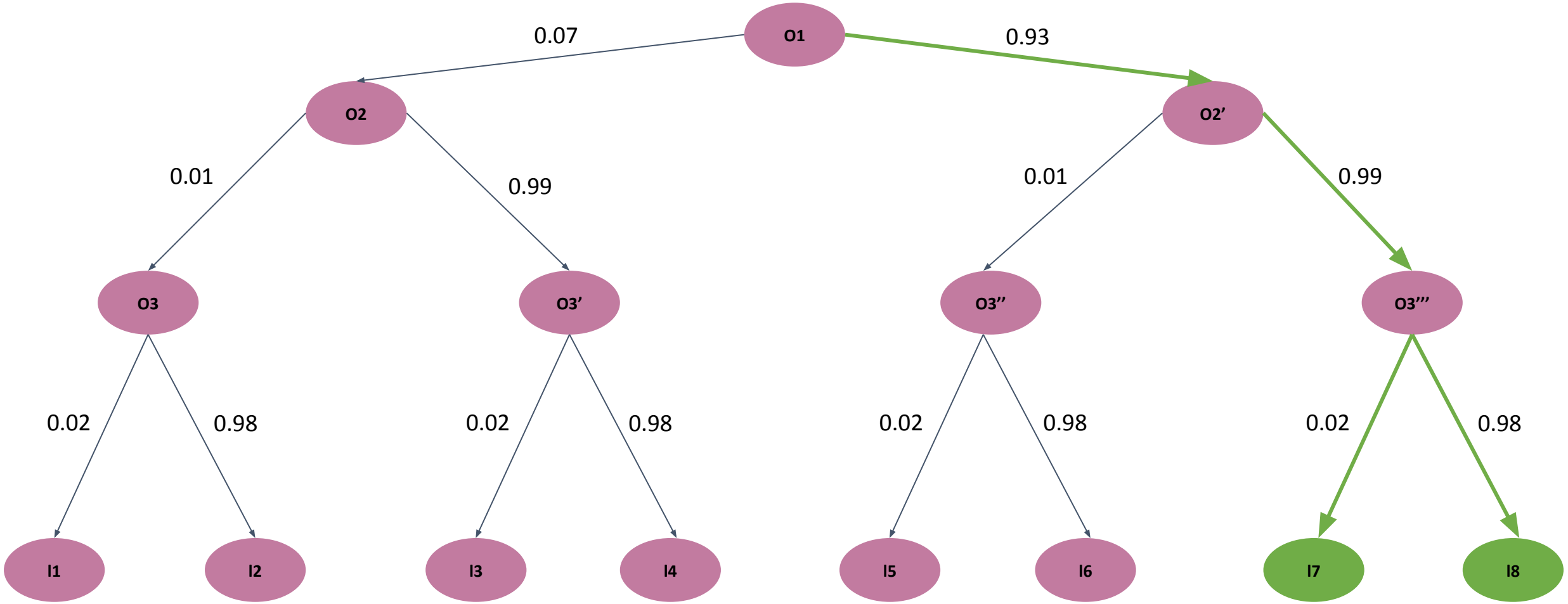
Многостороннее соглашение -> спецификация

(Организация 1 И Организация 2) ИЛИ (Организация 2 И Организация 3)



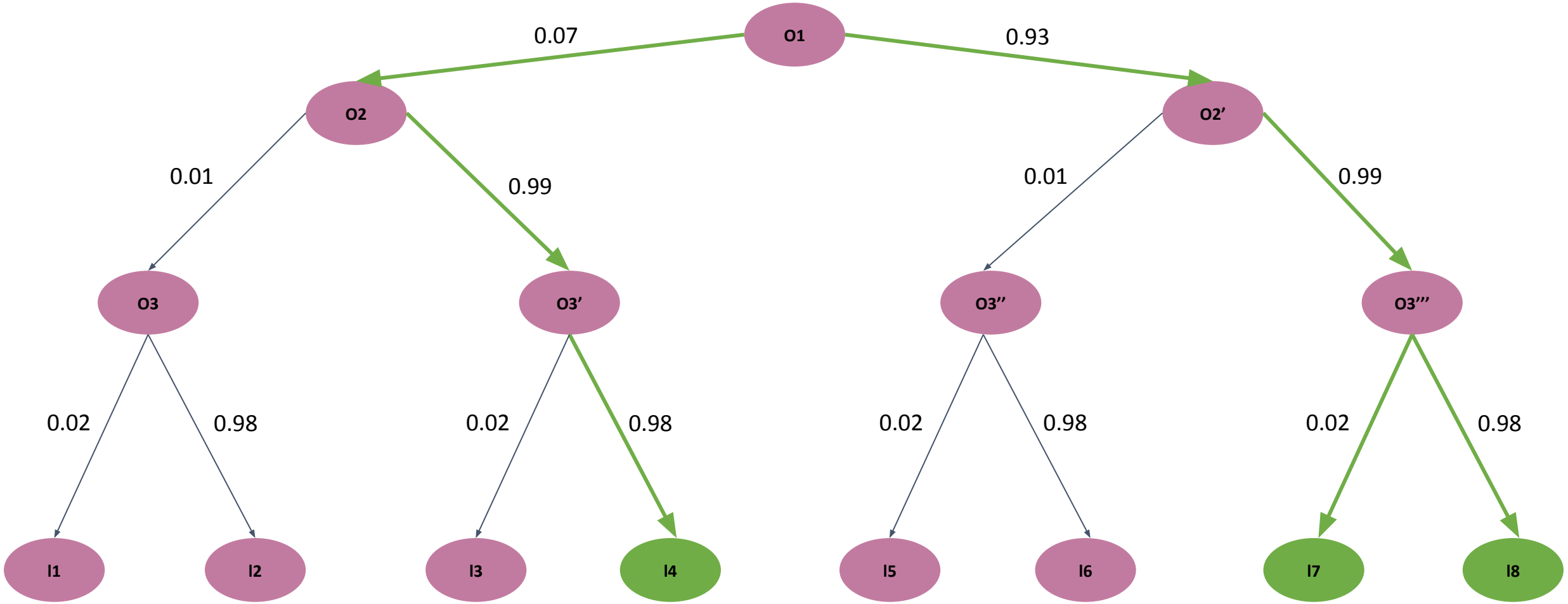
Многостороннее соглашение -> спецификация

(Организация 1 И Организация 2) ИЛИ (Организация 2 И Организация 3)



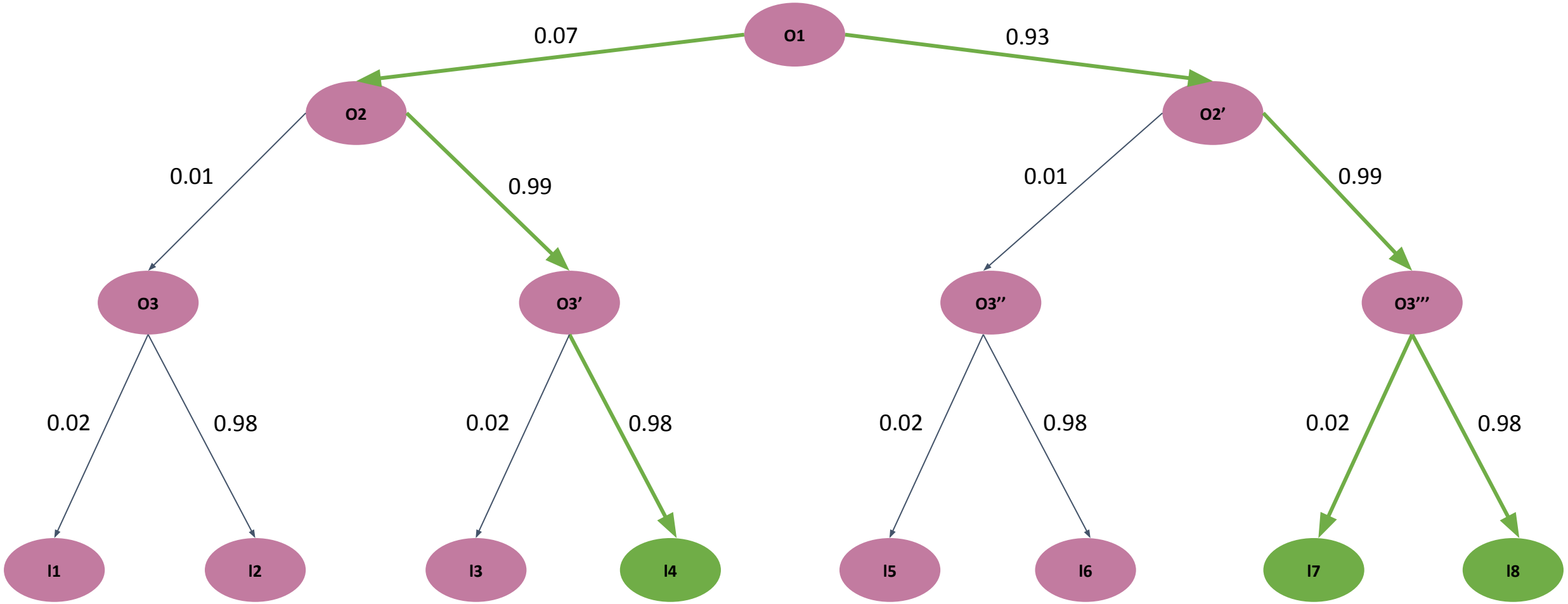
Многостороннее соглашение -> спецификация

(Организация 1 И Организация 2) ИЛИ (Организация 2 И Организация 3)

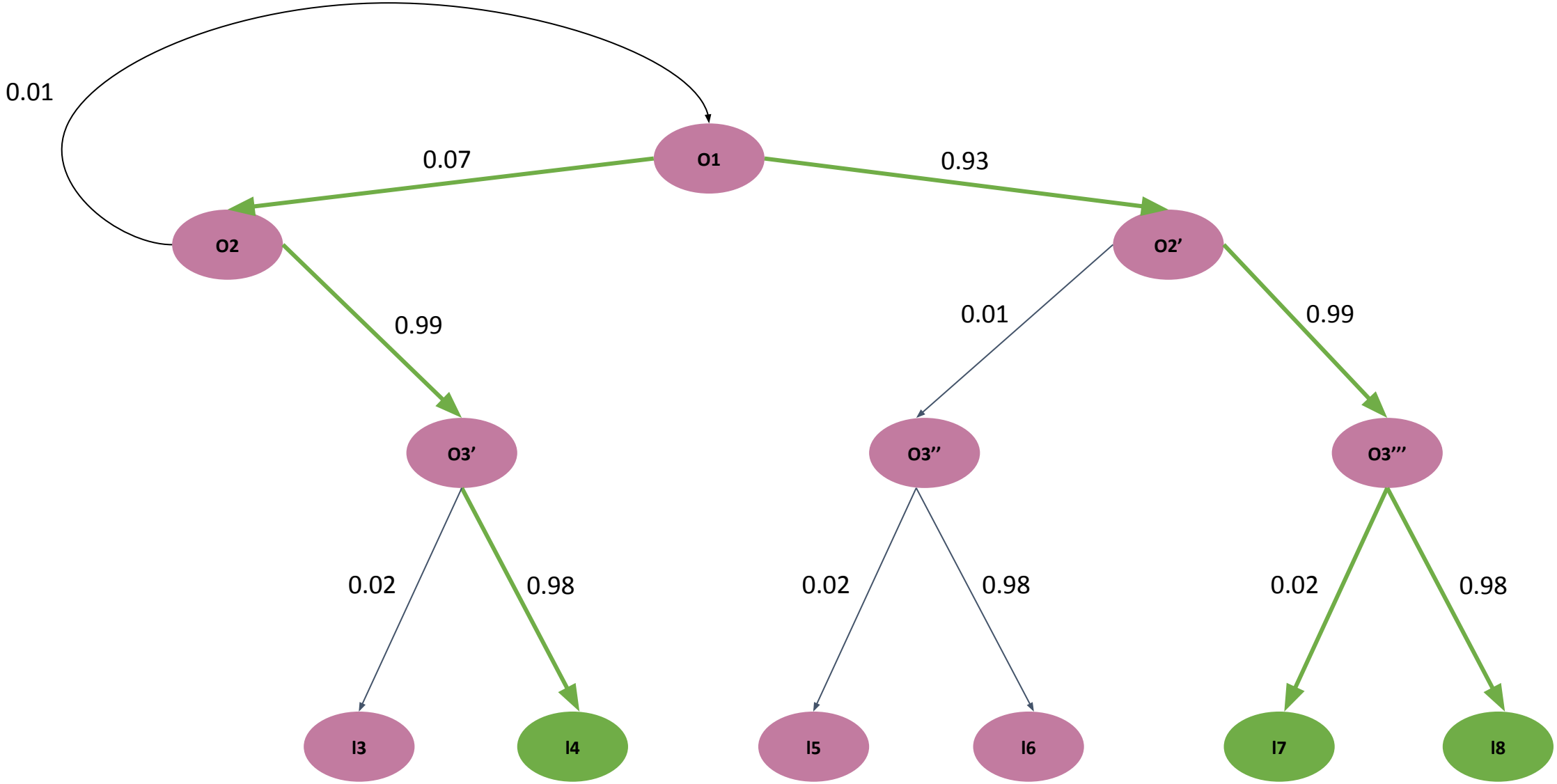


Многостороннее соглашение -> спецификация

P = ? F [**I4** OR **I7** OR **I8**]



Модель с обратными переходами.[1]



[1] Optimizing multi-party agreement protocols. I. Fedotov, A. Khritankov, A. Barger, 2022

Программный комплекс для анализа консенсуса и планирования отправки сообщений

Цели:

- Разработать программный комплекс анализа и планировки консенсуса на основе алгоритма для консенсуса, представленного в виде логической формулы
- Программный комплекс должен вычислять вероятность достижения консенсуса, а также предлагать планировку отправки сообщений на подтверждение с целью увеличить вероятность
- Интегрировать с промышленным фреймворком для построения блокчейн сетей

Структура программного комплекса:

- Модуль анализатора. Принимает на вход параметры консенсуса. Строит модель и вычисляет вероятность достижения консенсуса. Опционально строит все модели с обратными переходами. Выдает представление моделей с вероятностью и средним числом сообщений для каждой модели. Реализован на языке Java.
- Модуль планировщика. Принимает результат анализатора. Отправляет сообщение на подтверждение в соответствии с выбранной моделью. Может быть интегрирован с клиентом Hyperledger Fabric Java Gateway.

Постановка серии экспериментов

1. **Эксперимент 1.** Запуск тестовой сети из 3 организаций на базовой и модифицированной реализациях Java клиента HLF. Эксперимент преследует цель проверить, что без обратных переходов поведение программ ного комплекса в модифицированной реализации клиента не имеет статистически значимых отличий от поведения в случае использования базовой реализации клиента.
2. **Эксперимент 2.** Запуск тестовой сети из 3 организаций. Эксперимент нацелен проверить, что предсказания симуляции совпадают с результатами, полученными в тестовой сети. Подтверждение транзакции опре деляется правилом большинства.
3. **Эксперимент 3**, частичный многофакторный эксперимент. Эксперимент проводится с целью выявить, что популяция достижения консенсуса с помощью программного модуля не имеет статистически значимых отличий при разных конфигурациях сети. При разных парамет рах сети, как и в предыдущем эксперименте, сравниваются популяции, полученные в симуляции модуля анализатора и популяции из эксперимента на тестовой сети.

Фазы проведения эксперимента

1. **Фаза 1.** Подготовить участников консенсуса, которым будет отправляться транзакция на подтверждение. Локально запустить узлы и симулировать потери пакетов в сети. Это обеспечит разные вероятности принятия транзакций для разных участников соглашения. Исходя из этого каждому участнику задать вероятность подтверждения транзакции. Подключить к сети модифицированный клиент Fabric Gateway Java, который позволяет отправлять транзакции в соответствии с выбранной моделью.
2. **Фаза 2.** Задать файл конфигурации консенсуса. Запустить модуль consensus-analyzer. Передать результат в модифицированный клиент. Запустить процесс подтверждения транзакции.
3. **Фаза 3.** Собрать ответы клиента Fabric Gateway Java. Оценить изменение числа подтверждений при использовании модели с обратными переходами.

Постановка эксперимента в виде проверки гипотез

1. **H0:** Результат достижения консенсуса при использовании модифицированного клиента на модели без обратных переходов не имеет статистически значимых отличий от результата достижения консенсуса в базовой реализации клиента при одинаковых параметрах сети.
2. **H0':** Результат достижения консенсуса при использовании модифицированного клиента на модели без обратных переходов и результат достижения консенсуса в базовой реализации клиента имеют статистически значимые отличия.
3. **H1:** Результат достижения консенсуса при использовании модели с обратными переходами не имеет статистически значимых отличий от результата достижения консенсуса, полученного в модуле анализатора в ходе симуляции при использовании той же модели.
4. **H1':** Отличия в достижениях консенсуса при использовании модели с обратными переходами и в результате, полученном в ходе симуляции, являются статистически значимыми.
5. **H2:** Результат достижения консенсуса при использовании модели с обратными переходами с разными факторами не имеет статистически значимых отличий от результата достижения консенсуса, полученного в модуле анализатора в ходе симуляции при использовании той же моде ли.
6. **H2':** Результат достижения консенсуса при использовании модели с обратными переходами имеет статистически значимые отличия от результата достижения консенсуса, полученного в модуле анализатора в ходе симуляции при использовании той же модели

Результаты экспериментов

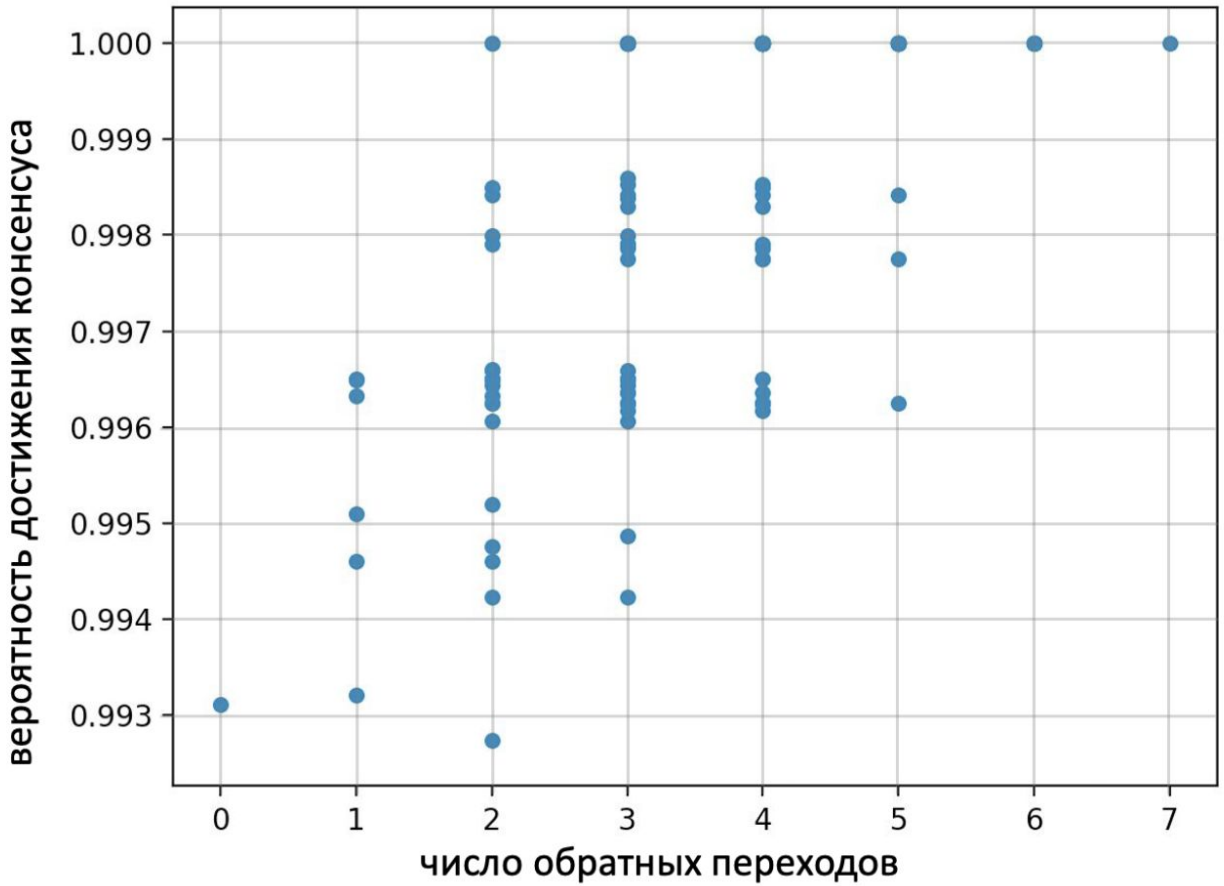


График зависимости вероятности достижения консенсуса от числа обратных переходов.

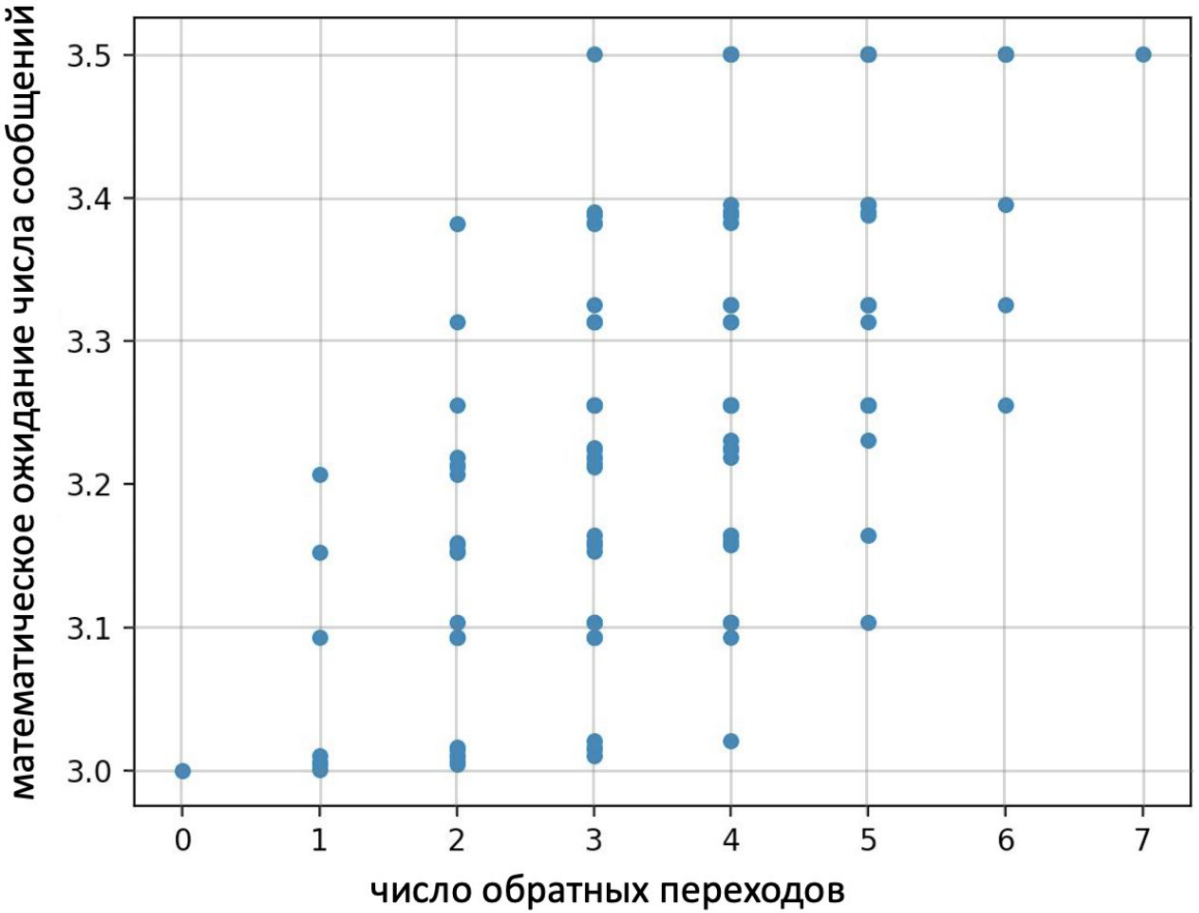


График зависимости математического ожидания числа сообщений от числа обратных переходов.

Результаты экспериментов

1. Статистически значимых отличий в распределении достижения консенсуса в модели без обратных переходов и распределении достижения консенсуса в базовой реализации клиента обнаружено не было.
2. Не выявлено статистически значимых отличий между данными достижения консенсуса на тестовой сети и данными, полученными в ходе симуляции.
3. Также не было найдено статистически значимых различий в многофакторном эксперименте. При заданном уровне значимости можно утверждать, что поведение модели соответствует поведению тестовой сети.

Статистически значимые различия искались с помощью теста Манна-Уитни при критерии значимости 0.05.

Внедрение результатов исследования

1. Внедрение в учебный курс «Автоматизация программирования». Был составлен учебный план, который включает в себя курс лекций по формальным методам верификации систем распределенного реестра. Была подготовлена и проведена лабораторная работа с применением программного комплекса, разработанного в ходе исследования по диссертационной работе.

2. Проведено внедрение разработанных алгоритмов и программного комплекса совместно с исследователями центра Idea. Проведены исследования и анализ потребностей верификации многостороннего соглашения в распределенных системах со стохастическими характеристиками. Проведена верификация многосторонних соглашений с последующей оптимизацией логики отправки на подтверждение.

Результаты диссертационной работы

- Сделан обзор методов и программных средств выявления и устранения уязвимостей в системах распределенного реестра
- Применен метод статистической проверки моделей для верификации атак на блокчейн системы и предложены пути устранения атак
- Предложены алгоритмы для построения модели и спецификации протоколов консенсуса различных видов
- Предложен алгоритм для увеличения вероятности достижения консенсуса
- Разработан программный комплекс для анализа консенсуса и планирования отправки сообщений. С помощью серии статистических экспериментов показана корректность работы программного комплекса.