Введение в нулевое разглашение, 18.10.2005

Ю. Лифшиц*

October 25, 2005

План лекции

1. Интерактивные доказательства, языки
2. Примеры интерактивных доказательств
3. Нулевое разглашение (Определение, доказательство)
4. Задачи для самостоятельного решения

1 Интерактивные доказательства, языки

1.1 Введение

Нулевое разглашение является одним из главных достижений современной теоретической информатики в целом (не только криптографии) за последние 20 лет и имеет очень много приложений – как теоретических, так и конкретных практических. Эта лекция – вводная, на ней мы познакомимся с нулевым разглашением, как со свойством интерактивного доказательства.

Неформально, интерактивное доказательство – диалог двух лиц, одно из которых проверяет доказательство (в дальнейшем – **V** или **V**erifier), а другое пытается убедить первого в том, что действительно знает доказательство (в дальнейшем – **P** или **P**rover).

1.2 Обычные доказательства

Обычное доказательство мы будем рассматривать в понимании Дэвида Гильберта, (книга "Основания Геометрии").

Вначале задаётся базовый список *аксиом*, на которые мы опираемся (или определений). Есть *правила вывода* (например индукция, доказательство от противного, и т. д.), и мы строим доказательство, как список утверждений – от условия задачи (и аксиом) к утверждению задачи.

*Законспектировал П. Никитин.

1.3 Понятие языка

Понятие языка является основным в теории сложности. Язык – это набор L строк конечной длины n из нулей и единиц, т. е. $L \subset \bigcup_{x_i \in \{0,\ 1\}^n} x_i$, где $1 \leqslant i \leqslant 2^n$; вот некоторые примеры:

- Все строки только из нулей.
- Все строки, в кототрых нулей и единиц поровну.
- Все строки вообще

Языков много, но мы остановимся на только одном из них подробно и умомянем ещё более общий по сравнению с ним.

1.4 Языки NP-класса

Язык L принадлежит NP, если существует полиномиальный алгоритм P, такой что $x \in L \Leftrightarrow \exists y : P(x, y) = 1$, т. е. алгоритм проверки принадлежности x к языку L по подсказке y. Тогда для $x \in L$ значение такого y, что $P(x, y) = 1$, является "доказательством" принадлежности языку. Заметим, что y должен быть полиномиального размера от x, иначе P даже не успел бы прочитать подсказку.

Примеры:

- Составные числа. Подсказка – два нетривиальных простых множителя в его разложении. Алгоритм P – перемножение этих чисел.
- Язык, содержащий пары изоморфных графов. Тогда подсказка для проверки принадлежности этой пары к языку – просто явно заданный изоморфизм.

Кроме того, существует язык $PSPACE$, который определяется так же, как NP, но с ограничением на память, используемую алгоритмом P. Т.е. $PSPACE$ – класс языков, такой что Язык L принадлежит $PSPACE$, если существует алгоритм P, использующий полиномиальный объем памяти, такой что $x \in L \Leftrightarrow P(x) = 1$.

1.5 Интерактивные доказательства

Давайте опишем связь между языками и задачами. *Массовая задача* – набор вопрос-ответ. Тогда язык, соответствующий задаче – набор вопросов, для которых ответ "да". Значит, например, для задачи коммивояжёра языка не существует. Пример языка, не принадлежащего NP: множество программ, которые заканчивают работу. Задача проверки принадлежности программы к этому языку алгоритмически неразрешима, значит, не существует подсказок: ведь тогда существовал бы и алгоритм проверки, основанный на переборе подсказок.

Итак, NP – задача, которую можно решить перебором всех y (ведь y должен быть полиномиального размера от x, и для каждого y мы можем запустить полиномиальный P. Только если хоть раз $P(x,y) = 1$, наш $x \in L$).

В этой лекции мы будем доказывать только теоремы вида "x принадлежит языку L". Если $L \in NP$, то доказательство очень простое – предъявить y, и запустить алгоритм P. Значит, обычные теоремы с константной или даже полиномиальной (от условия) длинной доказательства образуют язык класса NP, где доказательство и есть подсказка. И наоборот: самый общий вид теоремы: "строка $\{0,\ 1\}^n$ принадлежит к языку L".

Итак, в интерактивном доказательстве участвуют **P**rover и **V**erifier. **P** хочет убедить **V**, что рассматривамый ими x лежит в заданном языке L. Они обмениваются сообщениями, и через конечное число раундов **V**должен либо признать доказательство верным, либо высказать отрицание этого. Требования к доказательству:

- Полнота: $\forall x \in L, \exists \mathbf{P} : [\mathbf{P}(x), \mathbf{V}(x)] = 1$.
- Корректность: $\forall x \notin L, \forall \mathbf{P}' : Pr([\mathbf{P}'(x), \mathbf{V}(x)] = 1) = \nu(|x|)$.

($\nu(|x|)$ – фукция, убывающя быстрее обратной к полиному). То есть: сначала **P** шлет **V** первое приближение доказательства. Тот задаёт какой-то вопрос. **P** уточняет доказательство, и так далее. Полнота означает, что про любой x из L можно так доказать теорему, а корректность – что вероятность того, что **V** поверит в доказательство для $x \notin L$ очень мала.

2 Примеры интерактивных доказательств

Мы считаем, что **V**erifier пользуется полиномиальным вероятностным алгоритмом, а **P**rover вычислительно не ограничен. Это в частности означает, что **V** возможно проверяет доказательство не целиком, а выборочно. То есть условно он предлагает **P** некий тест из нескольких раундов, который тот пытается пройти. Заметим, что очень просто доказать за один раунд принадлежность x к языку NP (подумайте сами, ответ уже звучал в пункте **1.5**.) Тогда IP – класс всех языков, принадлежность к которым можно интерактивно доказать. Учёные задумались над видом этого класса. Заметим, что $IP \subset PSPACE$, ведь **V** может просто перебрать все подсказки, каждый раз при переходе к новой стирая память, каждый раз вызывая P, ведь полиномиальный алгоритм не может занять более чем полиномиальный объём памяти за время своей работы. Верно и более сильное: Теорема [Шамир, 1990]: $IP = PSPACE$.

Первый пример доказательства (изоморфизм графов): **P** собирается доказать $G_0 \cong G_1$. Он знает изоморфизм ϕ.

1. **P** выбирает случайную перестановку π и посылает $\pi\ (G_1) = H$
2. **V** посылает случайное b ($b = \{0,\ 1\}$), т. е. просит либо изоморфизм $H \cong G_0$, либо $H \cong G_1$.

3. В зависимости от b, **P** посылает π или $\pi \circ \phi$

4. Шаги 1-3 повторяются 1000 раз

Полнота следует из транзитивности изоморфизма. Каковы шансы пройти тест при $G_0 \ncong G_1$? Вне зависимости от количества вершин шансы равны $1/2^{1000}$ (вероятность). Значит, имеет место и корректность.

Второй пример доказательства (не-изоморфизм графов): **P** собирается доказать $G_0 \ncong G_1$.

1. **V** выбирает случайное b и случайную перестановку π и посылает $\pi\ (G_b) = H$

2. **P** пытается угадать b по H

3. Шаги 1-2 повторяются 1000 раз

Полнота верна, так как посылаемый граф H изоморфен ровно одному из $\{G_0,\ G_1\}$ Если же графы изоморфны, то H Может быть получено как из G_0, так и из G_1. Значит, вероятность угадать для **P** в таком случае равна опять же $1/2^{1000}$, то есть доказательство корректно.

Третий пример доказательства (3-раскрашиваемость, т. е. возможность раскрасить данный граф в 3 цвета так, чтобы инцидентные вершины были разных цветов, это *NP-полная* задача). (То есть если есть $\forall G_0, G_1\ \exists H$: $G_0 \cong G_1 \Leftrightarrow H$ – 3-раскрашиваем).

Покажем доказательство 3-раскрашиваемости:

1. **P** случайным образом переставляет три цвета между собой

2. **P** коммитит (т.е. использует привязку к биту) цвета всех вершин

3. **V** выбирает случайную пару вершин, соединённых ребром

4. **P** открывает цвета этих вершин

5. Шаги 1-4 повторяются $1000n^2$ раз

Если существовала раскраска – то всё верно, значит корректность есть. Если же нет, то вероятность угадать за n^2 раз равна $(1 - \frac{1}{n^2})^{n^2} = \frac{1}{e}$, а вероятность угадать за $1000n^2$ раундов равна $(1 - \frac{1}{1000n^2})^{1000n^2} = \frac{1}{e^{1000}}$. Давайте заметим, что **V** ничего не узнал про раскраску графа (здесь важно, что он может спрашивать только про соединённые ребром вершины, и на том, что он не может расшифровать зашифрованные цвета). Так мы подходим к понятию нулевого разглашения, как к ещё одному (необязательному) свойству интерактивного доказательства, однако оно нуждается в более формальном определении.

3 Нулевое разглашение (Определение, доказательство)

3.1 Подготовка

Напомним, у **P** и **V** есть теорема о принадлежности строчки x к языку L. Мы хотим определить, что значит, что доказательство не раскрывает никакой информации о ... о чём же? Об обычном доказательстве? Или о каких-то "тайных знаниях" **P**rover'а? А что это такое? Может быть, нулевое разглашение – это такое свойство, которое после интерактивного доказательства не гарантирует **V**erifier'у возможность самостоятельно "не-интерактивно" доказать теорему? Но ведь он *всегда* может это сделать, так как мы уже доказали, что $IP \subset PSPACE$! И тогда **V**erifier может разобраться, верна ли теорема, перебирая подсказки. Можно сказать так – мы должны "мало" узнать про x. Но мы хотим в общем случае определить нулевое разглашение, не определяя каждый раз заново секрет (для изоморфных графов это был сам изоморфизм, для 3-раскрашиваемости – сама раскраска.) Но мы уже можем определить "не-секрет": им является то, что можно вывести из условия полиномиальным алгоритмом. Итак, набросок определения: интерактивное доказательство обладает *нулевым разглашением*, если все что **V** узнал об x, он мог вычислить самостоятельно.

3.2 Формальное определение

Первая попытка: пара алгоритмов (**P**, **V**), образующих интерактивное доказательство, обладает нулевым разглашением, если:

$$\exists \mathsf{S}_{PPT} \forall x \in L : VIEW_{\mathbf{P},\mathbf{V}}[x] = S[x],$$

где $VIEW_{\mathbf{P},\mathbf{V}}$ – последовательность сообщений, полученных **V**.

То есть **V** может самостоятельно "симулировать" диалог с **P**. $\exists \mathsf{S}_{PPT}$ означает, что существует полиномиальный вероятностный алгоритм S, который может не зная ничего, кроме условия теоремы симулировать **P**rover, то есть сгенерировать последовательность сообщений между ним и **V** (на самом деле в этом нет ничего удивительного, ибо он может её *угадать*). Но это определение обладает каким-то недостатком. Каким? Верно ли, что мы гарантировали "точное" неразглашение? Как мы здесь комментируем неразглашение? "Мы могли бы запустить симулятор и узнать из него." Значит, подвох в том, что мы можем узнать из **P**rover'а то, чего нельзя узнать из симулятора. Это можно сделать, если **V** жульничает при диалоге. Тогда мы будем иметь дело не с $VIEW_{\mathbf{P},\mathbf{V}}[x]$, а с $VIEW_{\mathbf{P},\mathbf{V'}}[x]$! И нам важно гарантировать, что и в этом случае **V** не получит никакой информации. Например, если **V** в приведённых выше интерактивных доказательствах ведёт себя не случайным образом, а с каким-то подвохом.

Вторая попытка (окончательное определение):

$$\forall \mathbf{V'}\ \exists \mathsf{S}_{PPT}\ \forall x \in L : VIEW_{\mathbf{P},\mathbf{V'}}[x] = \mathsf{S}[x]\ (*)$$

Так же вводится еще более сильное свойство (симулятор с оракульным доступом):

$$\exists \mathsf{S}_{PPT}\ \forall \mathbf{V'}\ \forall x \in L : VIEW_{\mathbf{P},\mathbf{V'}}[x] = \mathsf{S}^{V'}[x]$$

Здесь мы усилили свойство, требуя один единственный алгоритм, не интересующийся внутренностью **V**erifier'a.

3.3 Применения нулевого разглашения

- Многосторонние вычисления — взаимный контроль участников.
- Протоколы авторизации – подслушивание бесполезно! Это настоящая революция: вместо того, чтобы присылать пароль, пользователь проходит набор тестов, не подсказавающих злоумышленнику этого пароля!!!
- Электронные выборы – возможность избирателя доказать, что он не зашифровал голоса за несколько человек, не раскрыв свой голос.
- Электронные деньги: мы подписываем у банка банкноту в 100\$, не раскрывая сам текст, написанный на ней.
- Или византийские генералы: если генерал может послать зашифрованные приказы, и, например, не раскрывая их, доказать, что они одинаковы.

3.4 Доказательство нулевого разглашения для изоморфизма графов

Напомним наш алгоритм доказательства изоморфности:

1. **P** выбирает случайную перестановку π и посылает $\pi\ (G_1) = H$
2. **V** посылает случайное b ($b = \{0,\ 1\}$), т. е. просит либо изоморфизм $H \cong G_0$, либо $H \cong G_1$.
3. В зависимости от b, **P** посылает π или $\pi \circ \phi$
4. Шаги 1-3 повторяются много раз

Давайте сначала обсудим его неразглашаемость неформально. **V**erifier видит некий граф H, изоморфный одному из двух исходных и изоморфизм к нему. Помогает ли это нам установить исходный изоморфизм? Ничуть. Но на самом деле всё не так просто, ведь **V** может "чуть-чуть приглядеться" к H и понять, какому из G_0, G_1 он вероятнее изоморфен (если, к примеру, **P** не очень силино его изменил). Давайте убедимся, что это всё-таки не облегчает жизнь **V**erifier'у. Мы должны доказать, что для каждого выбора теста существует программа S, которая для любого входа x генерирует настоящий диалог **V** с **P**. Вопрос: что за объекты стоят

слева и справа в равенстве $(*)$? Ответ: потоки сообщений. И равенство это значит, что программа S генерирует те и только те потоки сообщений, которые могли получиться в результате общения **P** и **V'**, и каждый конкретный поток получается с точно такой же вероятностью, т. е. распределения вероятности и справа и слева одинаковы. Итак, алгоритм симулятора:

1. Выбираем случайное b, случайную перестановку π
2. Скармливаем граф π (G_b) алгоритму **V'**
3. Если **V'** просит показать изоморфизм для G_b – показываем π, если для $G_{\bar{b}}$ — сбрасываем память **V'** и пробуем еще раз
4. Цикл по шагам 1-3 повторяем до 1000 успешных итераций

Вопрос: какова вероятность того, что **V'** попросит на шаге n тот изоморфизм, который мы знаем? Ответ: на каждом шаге $1/2$. Следовательно, математическое ожидание работы симулятора — полиномиально. Точнее, в данном случае оно равно 2000 раундов. Значит, вероятность *не пройти* весь симулятор за 10^6 шагов – очень мала, и равна $1/2^{1000}$. Вопрос: а вдруг нельзя сбросить память у **V'**? Но ведь **V'** с S заодно, значит, всё-таки можно.

Итак, на данный момент мы симулятором построили последовательность сообщений, которая *могла быть* в действительности. Но мы не проверили, что так можно сгенерировать *каждый* из возможных диалогов, причём равновероятно. Давайте убедимся в этом:

- Фазы независимы между собой
- Мы включаем/не включаем фазы *независимо* от их содержания

Аналогия:

- Студент стоит спиной к доске. Профессор выписал случайную последовательность. Студент говорит, какие символы вычеркнуть. То, что осталось – случайная последовательность (если профессор честный)!

С симулятором всё абсолютно так же: мы выбираем *случайно* первую фазу, так же случайно, как и **P**rover. Затем **V**erifier, не зная, из какого мы построили граф H, (из G_0 или G_1), говорит, какой изоморфизм ему хотелось бы получить. Значит, те попытки, которые мы не выкинем, останутся в нашей окончательной последовательности сообщений с такой же вероятностью, как и в настоящем диалоге интерактивного доказательства. Значит, мы доказали, что для нашего S и справа, и слева в равенстве $(*)$ получится одно и то же. Что и требовалось!

3.5 Ещё более сложная проблема

До сих пор открытой является проблема для случая, когда **P**rover *одновременно* присылает **V**erifier'у графы, а тот сразу для всех них генерит тест (какой он требует изоморфизм - к G_0 или к G_1), и **P** сразу на все вопросы теста отвечает. Для такого протокола пока ещё никто не умеет доказать свойство нулевого разглашения!!! Наше доказательство не проходит - ведь теперь шанс закончить работу равен $1/2^{1000}$.

3.6 Задачи на дом

1. Доказать, что $IP \subseteq PSPACE$.

2. Нулевое разглашение для квадратичных вычетов по составному модулю: рассмотрим только те остатки x в классе вычетов по модулю $N = pq$, где p и q – простые, которые дают либо квадратичный вычет по модулю N (т. е. $\exists y \in \mathbb{Z}_n : \ x \equiv y^2 \mod N$), либо не являются квадратичным вычетом ни по модулю p, ни по модулю q. Наш язык Li состоит только из тех остатков, которые всё же дают квадратичный вычет по модулю N. Задание состоит в том, чтобы придумать такое интерактивное доказательство с нулевым разглашением, которое доказывает принадлежность рассматриваемого остатка к языку Li. Пример: $N = 15$, тогда $Li = \{1,\ 4\}$ (а рассматриваем мы в этом примере все $x : x \in \{1, 4, 7, 11, 13\}$).

3.7 Использованные материалы

1. Презентация, предоставленная докладчиком – Ю. Лифшицом

2. Бумажный конспект лекции

3. Звукозапись слов докладчика

4. Текстовый редактор UltraEdit-32 v.10.10c

5. TεX-поставка EXTEX

6. Программа для просмотра PostScript – GsView v.4.3

7. Adobe Acrobat Distiller Professional Version 6.0