Сложность доказательств

Никита Гаевой

Санкт-Петербургский Государственный Университет

6 ноября 2020 г.

Определения

В начале нас будет волновать в первую очередь язык тавтологий, то есть язык невыполнимых формул в КНФ. Есть два почти эквивалентных определения.

Definition

Системой пропозициональных доказательств называется полиномиальная функция, действующая из строк в тавтологии.

Определения

В начале нас будет волновать в первую очередь язык тавтологий, то есть язык невыполнимых формул в КНФ. Есть два почти эквивалентных определения.

Definition

Системой пропозициональных доказательств называется полиномиальная функция, действующая из строк в тавтологии.

Definition

Системой пропозициональных доказательств называется полиномиальный алгоритм P удовлетворяющий следующим двум условиям:

1. Для любого $x \in \text{TAUT}$ существует такой y, что $P(x, y) = 1$ (completeness)
2. Для любого $x \notin \text{TAUT}$ и любой строчки y $P(x, y) = 0$ (soundness)

Игрушечные примеры

Функция проверки доказательства работает за время, зависящее от длины доказательства, поэтому существует система доказательств, где все доказательства представляют из себя просто достаточно длинные строки из единиц.

Игрушечные примеры

Функция проверки доказательства работает за время, зависящее от длины доказательства, поэтому существует система доказательств, где все доказательства представляют из себя просто достаточно длинные строки из единиц.
А если $P = NP$, то существует и система доказательств, где все доказательства пустые.

Игрушечные примеры

Функция проверки доказательства работает за время, зависящее от длины доказательства, поэтому существует система доказательств, где все доказательства представляют из себя просто достаточно длинные строки из единиц.

А если $P = NP$, то существует и система доказательств, где все доказательства пустые.

Существование системы доказательств полиномиального размера эквивалентно вопросу равенства классов $NP \stackrel{?}{=} coNP$.

Опередение

Нам дана формула $\varphi := C_0 \wedge C_1 \wedge \ldots \wedge C_m$, хотим ее доказать.

- C_i называются *аксиомами*.
- Resolution rule: $\dfrac{A \vee c \qquad B \vee \neg c}{A \vee B}$
- Weakening rule: $\dfrac{A}{A \vee B}$
- Доказательством в системе резолюций называется последовательность клозов $\pi = (D_1, D_2, \ldots, D_s)$, где каждый следующий клоз получен при помощи правил вывода из аксиом или клозов, выведенных ранее, а D_s — пустой клоз.

Опеределение

Нам дана формула $\varphi := C_0 \wedge C_1 \wedge \ldots \wedge C_m$, хотим ее доказать.

- C_i называются *аксиомами*.
- Resolution rule: $\dfrac{A \vee c \qquad B \vee \neg c}{A \vee B}$
- Weakening rule: $\dfrac{A}{A \vee B}$
- Доказательством в системе резолюций называется последовательность клозов $\pi = (D_1, D_2, \ldots, D_s)$, где каждый следующий клоз получен при помощи правил вывода из аксиом или клозов, выведенных ранее, а D_s — пустой клоз.

Если запретить одну и ту же переменную элиминировать дважды на одном пути, то получим Regular Resolution.

Опеределение

Нам дана формула $\varphi := C_0 \wedge C_1 \wedge \ldots \wedge C_m$, хотим ее доказать.

- C_i называются *аксиомами*.
- Resolution rule: $\dfrac{A \vee c \qquad B \vee \neg c}{A \vee B}$
- Weakening rule: $\dfrac{A}{A \vee B}$
- Доказательством в системе резолюций называется последовательность клозов $\pi = (D_1, D_2, \ldots, D_s)$, где каждый следующий клоз получен при помощи правил вывода из аксиом или клозов, выведенных ранее, а D_s — пустой клоз.

Если запретить одну и ту же переменную элиминировать дважды на одном пути, то получим Regular Resolution.
А если попросить, чтобы все клозы (кроме аксиом) при каждом использовании выводились заново, мы получим систему tree-like Resolution.

Пример вывода

$$\varphi := (\neg x_0 \vee x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_0) \wedge (x_0 \vee x_1 \vee x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2)$$

Пример вывода

$$\varphi \coloneqq (\neg x_0 \vee x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_0) \wedge (x_0 \vee x_1 \vee x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2)$$

$$\frac{x_0 \vee x_1 \vee x_2 \qquad \neg x_0 \vee x_1}{x_1 \vee x_2}$$

Пример вывода

$$\varphi := (\neg x_0 \vee x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_0) \wedge (x_0 \vee x_1 \vee x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2)$$

$$\dfrac{\dfrac{x_0 \vee x_1 \vee x_2 \qquad \neg x_0 \vee x_1}{x_1 \vee x_2} \qquad \neg x_1 \vee x_2}{x_2}$$

Пример вывода

$$\varphi := (\neg x_0 \vee x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_0) \wedge (x_0 \vee x_1 \vee x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2)$$

$$\dfrac{\dfrac{\neg x_2 \vee x_0 \qquad \dfrac{\neg x_0 \vee \neg x_1 \vee \neg x_2 \qquad \neg x_0 \vee x_1}{\neg x_0 \vee \neg x_2}}{\neg x_2}}{}$$

Пример вывода

$$\varphi := (\neg x_0 \vee x_1) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_0) \wedge (x_0 \vee x_1 \vee x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2)$$

$$\dfrac{\dfrac{\dfrac{x_0 \vee x_1 \vee x_2 \qquad \neg x_0 \vee x_1}{x_1 \vee x_2} \qquad \neg x_1 \vee x_2}{x_2} \qquad \dfrac{\neg x_2 \vee x_0 \qquad \dfrac{\neg x_0 \vee \neg x_1 \vee \neg x_2 \qquad \neg x_0 \vee x_1}{\neg x_0 \vee \neg x_2}}{\neg x_2}}{\varepsilon}$$

Чем хороши резолюции?

Имеют алгоритмический смысл. Например, если мы запустим алгоритм DPLL на невыполнимой формуле, мы автоматически получим доказательство в системе tree-like Resolution. А чтобы получить доказательство в Regular Resolution, надо запустить одну из версий алгоритма CDCL.

Чем хороши резолюции?

Имеют алгоритмический смысл. Например, если мы запустим алгоритм DPLL на невыполнимой формуле, мы автоматически получим доказательство в системе tree-like Resolution. А чтобы получить доказательство в Regular Resolution, надо запустить одну из версий алгоритма CDCL.
Некоторые формулы имеют минимальные доказательства экспоненциального размера. В числе таких формул Цейтинские формулы и некоторые обобщения принципа Дирихле.

Nullstellensatz

Theorem

Система уравнений

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ f_2(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \ldots, x_n) &= 0 \end{aligned}$$

не имеет решений тогда и только тогда, когда существует такой набор многочленов g_i, *что* $\sum_i f_i g_i = 1$.

Как из этого сделать систему доказательств?

Возьмем отрицание всей формулы, это превратит КНФ в ДНФ. Все клозы теперь конъюнкты, а конъюнкция хорошо представляется в виде произведения.

Как из этого сделать систему доказательств?

Возьмем отрицание всей формулы, это превратит КНФ в ДНФ. Все клозы теперь конъюнкты, а конъюнкция хорошо представляется в виде произведения.
Что делать с тем, что значения переменных не всегда равны 0 или 1? Добавить много уравнений вида $x_i^2 = x_i$.

Как из этого сделать систему доказательств?

Возьмем отрицание всей формулы, это превратит КНФ в ДНФ. Все клозы теперь конъюнкты, а конъюнкция хорошо представляется в виде произведения.
Что делать с тем, что значения переменных не всегда равны 0 или 1? Добавить много уравнений вида $x_i^2 = x_i$.
Мы уже получили систему доказательств!

Как из этого сделать систему доказательств?

Возьмем отрицание всей формулы, это превратит КНФ в ДНФ. Все клозы теперь конъюнкты, а конъюнкция хорошо представляется в виде произведения.
Что делать с тем, что значения переменных не всегда равны 0 или 1? Добавить много уравнений вида $x_i^2 = x_i$.
Мы уже получили систему доказательств!
Как измерять сложность такого доказательства?

- $\max_i \deg g_i$
- Количество мономов в $g_1, \ldots, g_m$
- Размер схем, вычисляющих $g_1, \ldots, g_m$.

Polynomial Calculus

На самом деле, $\sum_i f_i g_i = 1$ можно выводить и при помощи правил вывода.

- Из g и h можно вывести $g + h$
- Из h можно вывести gh, где g — произвольный полином.
- Доказательство — последовательность многочленов $\pi = (f_1, \ldots, f_m, h_1, \ldots, h_s, 1)$, полученная последовательным применением правил вывода.

Polynomial Calculus

На самом деле, $\sum_i f_i g_i = 1$ можно выводить и при помощи правил вывода.

- Из g и h можно вывести $g + h$
- Из h можно вывести gh, где g — произвольный полином.
- Доказательство — последовательность многочленов $\pi = (f_1, \ldots, f_m, h_1, \ldots, h_s, 1)$, полученная последовательным применением правил вывода.

Definition

Система доказательств P p-симулирует систему Q (пишут $P \leq_p Q$), если существует полиномиальная функция F такая, что $P(F(x)) = Q(x)$.

Polynomial Calculus p-симулирует Nullstellensatz и даже резолюции.

Определение

Нам понадобится

- Набор пропозициональных переменных.
- Произвольный функционально полный набор булевых операторов K.
- Произвольный набор правил вида

$$\frac{B_1 \quad B_2 \quad \dots \quad B_n}{B},$$

где все B_i состоят из переменных, связанных при помощи операторов из K.

Определение

Нам понадобится

- Набор пропозициональных переменных.
- Произвольный функционально полный набор булевых операторов K.
- Произвольный набор правил вида

$$\frac{B_1 \quad B_2 \quad \dots \quad B_n}{B},$$

где все B_i состоят из переменных, связанных при помощи операторов из K.

Под доказательством опять понимаем последовательность формул, каждая из которых получается при помощи правил вывода из предыдущих или аксиом.

Определение

Нам понадобится

- Набор пропозициональных переменных.
- Произвольный функционально полный набор булевых операторов K.
- Произвольный набор правил вида

$$\frac{B_1 \quad B_2 \quad \dots \quad B_n}{B},$$

где все B_i состоят из переменных, связанных при помощи операторов из K.

Под доказательством опять понимаем последовательность формул, каждая из которых получается при помощи правил вывода из предыдущих или аксиом.
И если то, что у нас получилось, оказалось системой доказательств (то есть, если набор правил sound и complete), то это называется *системой Фреге*.

Что про них известно?

- Все системы Фреге p-эквивалентны.

Что про них известно?

- Все системы Фреге p-эквивалентны.
- С нижними оценками на них все плохо. Лучшие известные оценки на размер доказательства квадратичны, а на количество операций — линейны. Тут стоит отметить, что резолюции не являются системой Фреге, поскольку работают только на дизъюнктах.

Что про них известно?

- Все системы Фреге p-эквивалентны.
- С нижними оценками на них все плохо. Лучшие известные оценки на размер доказательства квадратичны, а на количество операций — линейны. Тут стоит отметить, что резолюции не являются системой Фреге, поскольку работают только на дизъюнктах.
- Бывают еще расширенные системы Фреге. В них добавляется возможность завести переменную, обозначающую какую-нибудь выведенную формулу, что позволяет писать dag-like доказательства вместо древовидных.

А что бывает еще?

Мы уже немного обсудили понятие p-симуляции, которое позволяет говорить, что одна система доказательств «не хуже» другой. Довольно естественным образом возникает вопрос о существовании p-оптимальной системы доказательств.

А что бывает еще?

Мы уже немного обсудили понятие p-симуляции, которое позволяет говорить, что одна система доказательств «не хуже» другой. Довольно естественным образом возникает вопрос о существовании p-оптимальной системы доказательств. К сожалению, этот вопрос так и остается открытым.
Иногда рассматривают более слабое понятие симуляции (или слабой p-симуляции).

Definition

Система доказательств P симулирует систему Q, если для любого x существует такой y, что $P(y) = Q(x)$ и $|y| \leq \text{poly}(|x|)$.

Про существование оптимальной системы доказательств относительно такой симуляции тоже ничего не известно.

Другие языки и классы

Могло показаться, что системы доказательств — это обязательно про тавтологии и классы P и coNP. На самом деле это совсем не так и можно рассматривать системы доказательств для других языков.
Например, для языка SAT есть система доказательств «выполняющий набор».

Другие языки и классы

Могло показаться, что системы доказательств — это обязательно про тавтологии и классы P и coNP. На самом деле это совсем не так и можно рассматривать системы доказательств для других языков.
Например, для языка SAT есть система доказательств «выполняющий набор».
Более того, даже проверяющий алгоритм не обязательно лежит в классе P.
Например, бывают системы доказательств, проверяемые схемами из класса NC^0 (то есть каждый бит теоремы зависит от $O(1)$ бит доказательства), и вопрос существования такой системы доказательств для TAUT остается открытым.

Bonus fact

Definition

Системой доказательств с $t(n)$ битами подсказки (advice) называется полиномиальный алгоритм $\Pi(x, w, \alpha)$ и последовательность $(\alpha_n)_{n\in\mathbb{N}}$, где $\alpha_n \in \{0,1\}^{t(n)}$, такой что для всех x

$$x \in L \Leftrightarrow \exists w\ \Pi(x, w, \alpha_{|x|+|w|}) = 1$$

Theorem

Для любого языка L существует система доказательств с одним битом подсказки, симулирующая все системы доказательств с $k(n) = O(\log n)$ битами подсказки. Причем симуляция может быть вычислена за полиномиальное время с $k(n)$ битами подсказки.

Идея конструкции

- Занумеруем все системы доказательств.
- Мы построим оптимальную систему доказательств Π_*, которая будет буквально симулировать все остальные системы доказательств.

Идея конструкции

- Занумеруем все системы доказательств.
- Мы построим оптимальную систему доказательств Π_*, которая будет буквально симулировать все остальные системы доказательств.
- Доказательством в системе Π_* будет тройка из номера системы доказательств Π, доказательства в ней, и используемой в Π подсказки.

Идея конструкции

- Занумеруем все системы доказательств.
- Мы построим оптимальную систему доказательств Π_*, которая будет буквально симулировать все остальные системы доказательств.
- Доказательством в системе Π_* будет тройка из номера системы доказательств Π, доказательства в ней, и используемой в Π подсказки.
- Для проверки доказательства Π_* будет просто симулировать работу того, что ей дали в качестве доказательства. Completeness и симуляции тривиальны.

Идея конструкции

- Занумеруем все системы доказательств.
- Мы построим оптимальную систему доказательств Π_*, которая будет буквально симулировать все остальные системы доказательств.
- Доказательством в системе Π_* будет тройка из номера системы доказательств Π, доказательства в ней, и используемой в Π подсказки.
- Для проверки доказательства Π_* будет просто симулировать работу того, что ей дали в качестве доказательства. Completeness и симуляции тривиальны.
- Осталось гарантировать soundness Π_*. Для этого нам нужно научиться проверять, что данный в доказательстве алгоритм на самом деле составляет какую-то систему доказательств. Для этого воспользуемся нашим одним битом равномерной подсказки.