Математические методы верификации схем и программ

mk.cs.msu.ru → Лекционные курсы
→ Математические методы верификации схем и программ

Блок 31

Проблема выполнимости булевых формул (SAT)

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2022/2023, осенний семестр

Вступление

Задача MC-LTL трудна (PSPACE-полна), а значит, не имеет достаточно эффективного решения

BMC (проверку соотношения $M \models_k \varphi$) можно расценивать как попытку «упростить» задачу MC-LTL, сохранив общность и открыв возможность изменять сложность за счёт выбора параметра k

Разумно предположить, что для нетривиальных значений k эта задача всё равно останется трудной как минимум в теоретическом смысле

Однако в современном мире задачи, трудные в теоретическом смысле, иногда считаются достаточно простыми и эффективно решаемыми на практике

Проблема выполнимости булевых формул (SAT)

КНФ (конъюнктивная нормальная форма) над n переменными — это булева формула вида

$$D_1 \,\&\, \ldots \,\&\, D_k,$$

где $k \geq 0$ (для $k = 0$ полагаем КНФ равной 1) и для каждого i, $1 \leq i \leq k$, множитель D_i представляет собой дизъюнкт

$$L_1 \vee \cdots \vee L_m,$$

где $m \geq 0$ (для $m = 0$ полагаем дизъюнкт равным 0) и для каждого j, $1 \leq j \leq m$, L_i представляет собой литеру: одну из булевых переменных $x_1, \ldots, x_n$ или её отрицание

Булева формула выполнима, если реализуемая ей функция истинна (принимает значение 1) хотя бы на одном наборе значений переменных

Проблема выполнимости булевых формул (SAT) формулируется так: для заданного натурального числа n и заданной КНФ K над n переменными проверить выполнимость K

Проблема выполнимости булевых формул (SAT)

Иногда рассматривается более широкая постановка задачи SAT, в которой вместо КНФ можно использовать произвольную формулу над задарнее заданной системой булевых функций (например, над $\{\&, \vee, \neg\}$)
На практике такое расширение оказывается несущественным:

- Подформула вида $f(\ldots, \varphi, \ldots)$, где φ не переменная, может быть заменена на $f(\ldots, x, \ldots)\,\&(x \leftrightarrow \varphi)$, где x — переменная, не встречавшаяся ранее в КНФ
- Подформулы вида $f(x_{i_1}, \ldots, x_{i_m})$ и $(x \leftrightarrow f(x_{i_1}, \ldots, x_{i_m}))$, где f — предзаданная функция и $x_{i_1}, \ldots, x_{i_m}$ — переменные, могут быть заменены на (настолько же предзаданные, как и функция) равносильные КНФ
- Последовательно применяя эти преобразования, можно получить из произвольной формулы настолько же (не)выполнимую КНФ, добавив линейное (относительно размера исходной формулы) число переменных и литер

Имея в виду эту взаимосвязь, будем в примерах и алгоритмах выбирать наиболее удобную из постановок задачи SAT

Проблема выполнимости булевых формул (SAT)

Известно, что задача SAT вычислительно трудна (NP-полна)

То есть с теоретической точки зрения эта задача скорее всего не имеет эффективного (*полиномиального*) решения

А если обнаружится, что эффективное решение всё-таки есть, то и у теоретиков, и у практиков возникнет очень много проблем, проистекающих из предположения, что такого решения нет — например:

- Ряд методов защиты информации, основанных на трудных (NP-полных или близких к ним) задачах, перестанет работать в теории
- Методы эффективного приближенного или частного решения многих трудных задач окажутся неактуальными

Проблема выполнимости булевых формул (SAT)

Но на практике в современном мире принято считать, что существуют способы эффективного решения задачи SAT (хотя и не настолько эффективного, чтобы можно было столкнуться с упомянутыми ранее проблемами):

- Эффективные решающие алгоритмы
- Эффективные программные средства (SAT-решатели)
  - MiniSAT, zChaff, RSat, Lingeling, WinSAT, ...
- История успеха применения этих средств к КНФ с **миллионами** переменных и множителей, возникающим на практике
- Конференции, соревнования и коммерческие запросы, «подстёгивающие» дальнейшее повышение практической эффективности
- **Но никто не понимает, почему оно так хорошо работает**

Современные SAT-решатели, как правило, основываются на двух видах алгоритмов:

- DPLL/CDCL
- Локального поиска

SAT: локальный поиск

Алгоритмы локального поиска устроены так:

1. Выбирается начальный набор значений переменных
2. Если на текущем наборе КНФ истинна, то завершить работу
3. Иначе выбирается переменная, её значение изменяется на противоположеное, и повторяется (2)
4. По необходимости выбирается новый набор значений переменных либо алгоритм досрочно завершается

Выбор, как правило, выполняется с привлечением случайности

Цель «блужданий» по наборам значений, как правило, — получение как можно большего числа истинный множителей

Если алгоритм завершился досрочно, то констатируется, что выполнимость КНФ неизвестна

SAT: DPLL/CDCL

CDCL (Conflict-driven clause learning) — это способ организации алгоритмов решения задачи SAT, наиболее хорошо зарекомендовавший себя на практике на текущий момент

Этот способ является «умной» надстройкой над способом организации DPLL (по фамилиям создателей: Davis, Putnam, Logemann, Loveland)

Чтобы не перегружать рассказ деталями, остановимся только на кратком описании схемы работы DPLL

SAT: DPLL/CDCL

Общая схема работы DPLL устроена так:

1. Если текущая КНФ имеет вид 1, то работа завершается: «да» (КНФ выполнима)
2. Если текущая КНФ содержит множитель 0, то работа завершается: «нет» (КНФ невыполнима)
3. Иначе
   - 3.1 Выбираются переменная (x) и её значение 0 или 1 (b)
   - 3.2 Подставляется значение x/b, формула оптимизируется, и (рекурсивно) проверяется выполнимость оптимизированной формулы
   - 3.3 Если последняя формула выполнима, то работа завершается: «да»
   - 3.4 Иначе подставляется значение $x/\neg b$, формула оптимизируется, и возвращается результат (рекурсивной) проверки её выполнимости

Основные оптимизации:

- Удаление множителей со слагаемым 1
- Свёртка констант: если есть множитель с одной литерой x или $\neg x$, то подставляется значение $x/1$ (для литеры x) или $x/0$ (для $\neg x$)
- Деполяризация: если переменная x входит в КНФ только без отрицания (только с отрицанием), то подставляется $x/1$ ($x/0$)

SAT: DPLL/CDCL

Пример

$$(x_1 \vee x_2 \vee x_3) \& (x_1 \vee \neg x_2 \vee x_3) \& (\neg x_1 \vee \neg x_2) \& (\neg x_2 \vee \neg x_3)$$

SAT: DPLL/CDCL

Пример

$$(x_1 \vee x_2 \vee x_3)\,\&\,(x_1 \vee \neg x_2 \vee x_3)\,\&\,(\neg x_1 \vee \neg x_2)\,\&\,(\neg x_2 \vee \neg x_3)$$

$x_2/1$, удаление множителей

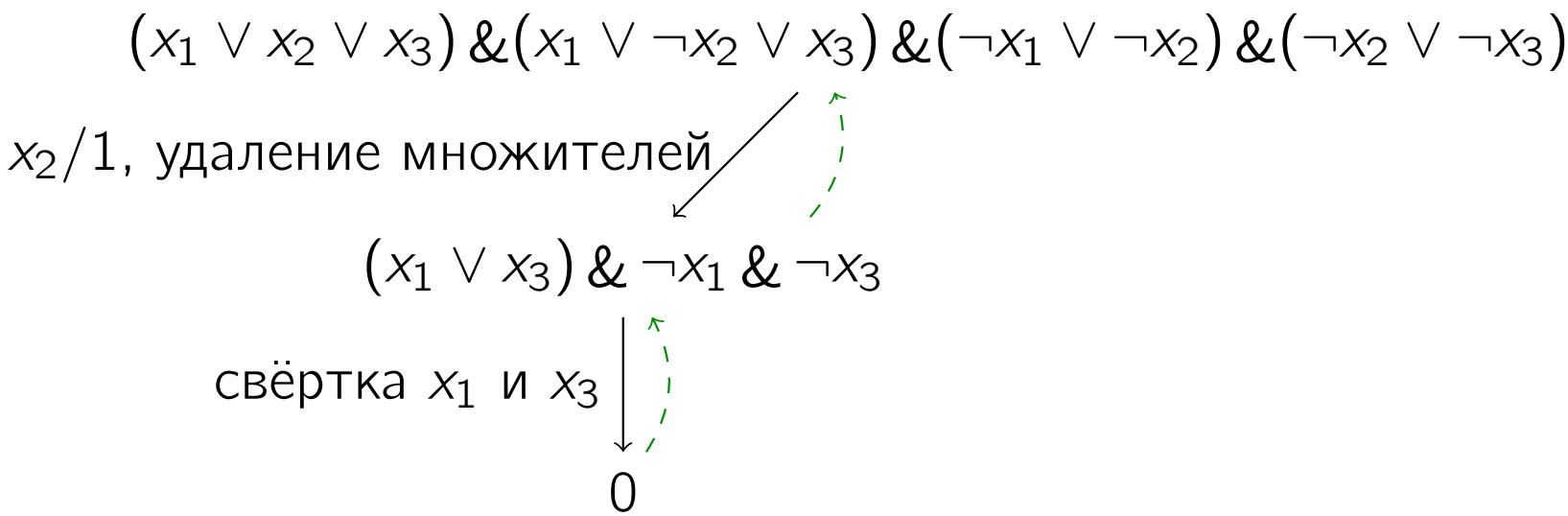
$$(x_1 \vee x_3)\,\&\,\neg x_1\,\&\,\neg x_3$$

свёртка x_1 и x_3

$$0$$

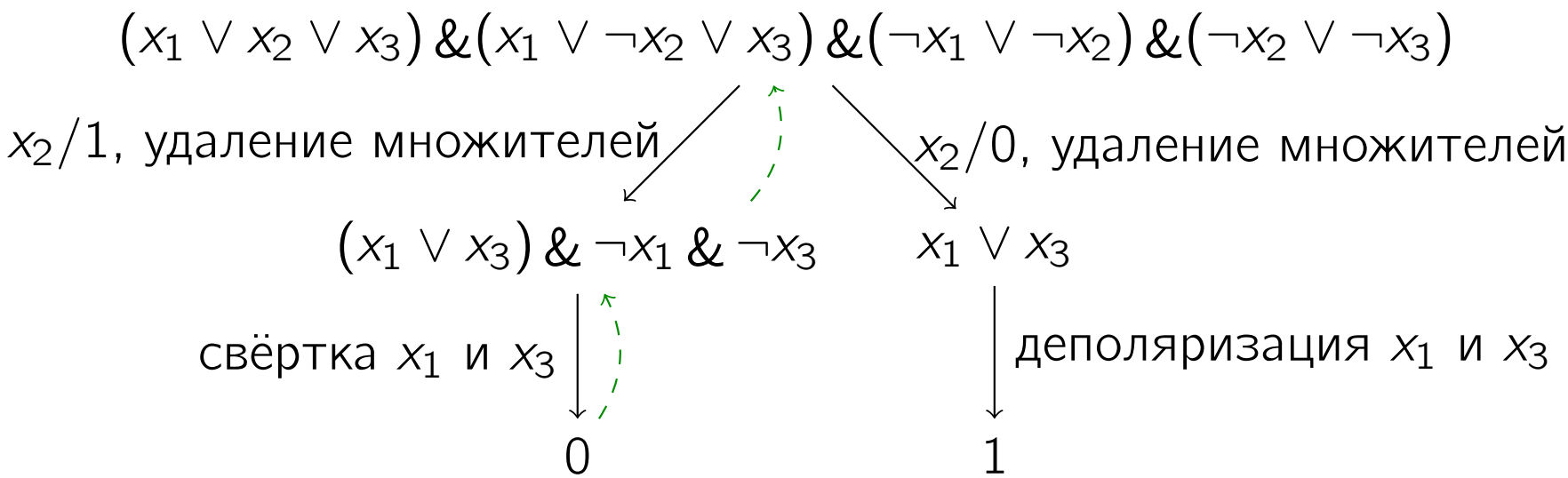
SAT: DPLL/CDCL

Пример



SAT: DPLL/CDCL

Пример



Получили КНФ 1: конец выполнения, исходная КНФ выполнима