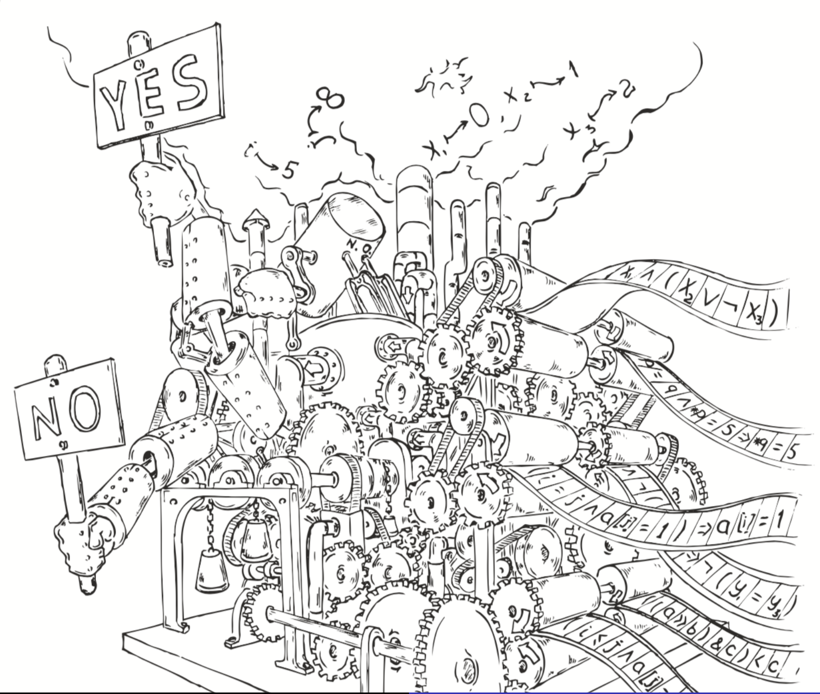

YES
NO
N.O.

Задачи разрешимости логических формул и приложения
Лекция 6. Оптимизации CDCL

Роман Холин

Московский государственный университет

Москва, 2021

Conflict-driven clause learning

```
function CDCL
    while true do
        while BCP() = "conflict" do
            backtrack-level := Analyze-Conflict()
            if backtrack-level < 0 then
                return "Unsatisfiable"
            end if
            BackTrack(backtrack-level)
            if ¬ Decide() then
                return "Satisfiable"
            end if
        end while
    end while
end function
```

Граф следствий

- Будем писать $x_i@dl$, если на уровне принятия решений dl мы присвоили переменной x_i значение истина и $\neg x_i@dl$ - если присволи ложь
- Вершины графа - переменные, определенные частичной оценкой
- Из v_i идет ребро v_j, если v_j оценена в результате BCP() и v_i входит в дизъюнкт-предпосылку c. Эти ребра помечаются меткой c
- Если есть "конфликт то ему соответствует вершина. Пусть c - конфликтный дизъюнкт. Тогда к вершине "конфликт"идут ребра от переменных, входящих в c и они помечаются меткой c

Пример

$$c_1 = (\neg x_1 \vee x_2)$$
$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$
$$c_3 = (\neg x_2 \vee x_4)$$
$$c_4 = (\neg x_3 \vee \neg x_4)$$
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$
$$c_6 = (x_2 \vee x_3)$$
$$c_7 = (x_2 \vee \neg x_3)$$
$$c_8 = (x_6 \vee \neg x_5)$$

Пример

$c_1 = (\neg x_1 \vee x_2)$
$c_2 = (\neg x_1 \vee x_3 \vee x_5)$
$c_3 = (\neg x_2 \vee x_4)$
$c_4 = (\neg x_3 \vee \neg x_4)$
$c_5 = (x_1 \vee x_5 \vee \neg x_2)$
$c_6 = (x_2 \vee x_3)$
$c_7 = (x_2 \vee \neg x_3)$
$c_8 = (x_6 \vee \neg x_5)$
Пусть x_1@6 и $\neg x_5$@3

Пример

$c_1 = (\neg x_1 \vee x_2)$
$c_2 = (\neg x_1 \vee x_3 \vee x_5)$
$c_3 = (\neg x_2 \vee x_4)$
$c_4 = (\neg x_3 \vee \neg x_4)$
$c_5 = (x_1 \vee x_5 \vee \neg x_2)$
$c_6 = (x_2 \vee x_3)$
$c_7 = (x_2 \vee \neg x_3)$
$c_8 = (x_6 \vee \neg x_5)$
Пусть x_1@6 и $\neg x_5$@3
$c_9 = (x_5 \vee \neg x_1)$

Пример

$c_1 = (\neg x_1 \lor x_2)$
$c_2 = (\neg x_1 \lor x_3 \lor x_5)$
$c_3 = (\neg x_2 \lor x_4)$
$c_4 = (\neg x_3 \lor \neg x_4)$
$c_5 = (x_1 \lor x_5 \lor \neg x_2)$
$c_6 = (x_2 \lor x_3)$
$c_7 = (x_2 \lor \neg x_3)$
$c_8 = (x_6 \lor \neg x_5)$
Пусть $x_1@6$ и $\neg x_5@3$
$c_9 = (x_5 \lor \neg x_1)$
Откатимся до 3 уровня принятия решений

- Почему откатились на 3 уровень, а не на 5?

- Почему откатились на 3 уровень, а не на 5?
- Эмпирические исследования показывают, что так быстрее

- Почему откатились на 3 уровень, а не на 5?
- Эмпирические исследования показывают, что так быстрее
- Почему останавливаемся?

- Почему откатились на 3 уровень, а не на 5?
- Эмпирические исследования показывают, что так быстрее
- Почему останавливаемся?
- Докажем от противного

Бинарная резолюция

$$\frac{(a_1 \vee \ldots a_n \vee c)(b_1 \ldots b_m \vee \neg c)}{(a_1 \vee \ldots a_n \vee b_1 \ldots b_m)}$$

Бинарная резолюция

$$\frac{(a_1 \vee \ldots a_n \vee c)(b_1 \ldots b_m \vee \neg c)}{(a_1 \vee \ldots a_n \vee b_1 \ldots b_m)}$$

Известен результат, что КНФ не выполнима тогда и только тогда, когда существует конечное число бинарных резолюций, приводящих к пустому дизъюнкту

Пример

$$
\begin{aligned}
c_1 &= (\neg x_4 \vee x_2 \vee x_5) \\
c_2 &= (\neg x_4 \vee x_{10} \vee x_6) \\
c_3 &= (\neg x_5 \vee \neg x_6 \vee \neg x_7) \\
c_4 &= (\neg x_6 \vee x_7)
\end{aligned}
$$

Пример

$$c_1 = (\neg x_4 \vee x_2 \vee x_5)$$
$$c_2 = (\neg x_4 \vee x_{10} \vee x_6)$$
$$c_3 = (\neg x_5 \vee \neg x_6 \vee \neg x_7)$$
$$c_4 = (\neg x_6 \vee x_7)$$

- Выберем последний оцененый литерал
- Выберем дизъюнкт предпосылку данного литерала
- Применим бинарную резолюцию к конфликтному дизъюнкту и дизъюнкту предпосылке через переменную, соответствующей литералу

Пример

$$c_1 = (\neg x_4 \vee x_2 \vee x_5)$$
$$c_2 = (\neg x_4 \vee x_{10} \vee x_6)$$
$$c_3 = (\neg x_5 \vee \neg x_6 \vee \neg x_7)$$
$$c_4 = (\neg x_6 \vee x_7)$$

- Выберем последний оцененый литерал
- Выберем дизъюнкт предпосылку данного литерала
- Применим бинарную резолюцию к конфликтному дизъюнкту и дизъюнкту предпосылке через переменную, соответствующей литералу

Когда остановиться?

Уникальная точка импликации

- Уникальная точка импликации - любая вершина импликационного графа, которая не является вершиной «конфликт» и которая находится на каждом пути, ведущему от конкретной корневой вершины к вершине конкретной вершине «конфликт» (в теории графов так же называется доминатором вершины)

Уникальная точка импликации

- Уникальная точка импликации - любая вершина импликационного графа, которая не является вершиной «конфликт» и которая находится на каждом пути, ведущему от конкретной корневой вершины к вершине конкретной вершине «конфликт» (в теории графов так же называется доминатором вершины)
- Первая УТИ - ближайшая к конфликтной вершине уникальная точка импликации

Уникальная точка импликации

- Уникальная точка импликации - любая вершина импликационного графа, которая не является вершиной «конфликт» и которая находится на каждом пути, ведущему от конкретной корневой вершины к вершине конкретной вершине «конфликт» (в теории графов так же называется доминатором вершины)
- Первая УТИ - ближайшая к конфликтной вершине уникальная точка импликации
- Эмпирические исследования показывают, что нужно остановится, когда будет добавлена отрицание первой УТИ текущего уровня принятия решений

```
if current-decision-level = 0 then
    return -1
end if
cl := current-conf licting-clause
while ¬ Stop-criterion-met(cl) do
    lit := Last-assigned-literal(cl)
    var := Variable-of-literal(lit)
    ante := Antecedent(lit)
    cl := Resolve(cl, ante, var)
end while
add-clause-to-database(cl)
return clause-asserting-level(cl)
```

Conflict-driven clause learning

```
function CDCL
    while true do
        while BCP() = "conflict" do
            backtrack-level := Analyze-Conflict()
            if backtrack-level < 0 then
                return "Unsatisfiable"
            end if
            BackTrack(backtrack-level)
            if ¬ Decide() then
                return "Satisfiable"
            end if
        end while
    end while
end function
```

- Для каждого литерала посчитаем $J(l) = \Sigma_{w \in B, l \in w} 2^{|w|}$

Dynamic Largest Individual Sum (DLIS)

- Для каждого литерала посчитаем в скольких неразрешенных дизъюнктах он находится

Dynamic Largest Individual Sum (DLIS)

- Для каждого литерала посчитаем в скольких неразрешенных дизъюнктах он находится
- Очень дорого

Variable State Independent Decaying Sum (VSIDS)

- Для каждого литерала посчитаем в скольких неразрешенных дизъюнктах он находится
- Иногда всё делим на 2
- При каждом конфликте, увеличиваем на 1 балл литерала

Variable State Independent Decaying Sum (VSIDS) в MiniSAT

- При каждом конфликте, увеличиваем на балл литерала на Inc
- Inc сначала 1, затем увеличивается на 1.5 после каждого конфликта
- Всё делится на 10^{-100}, если есть балл, выше 10^{100}

- Для каждого литерала и переменной посчитаем в скольких неразрешенных дизъюнктах он находится
- Иногда оценки переменной делим на 2
- При каждом конфликте, увеличиваем на 1 балл литерала
- Каждый конфликтный дизъюнкт помещаем в стек
- Когда нужно выбрать, какую перпеменную оценить, находим в стеке самый верхний дизъюнкт, который неразрешен, а в нём выбираем переменную с самой большой оценкой, а затем литерал с самой большой оценкой

Эвристики, основанные на дизъюнктах (Clause-Move-To-Front)

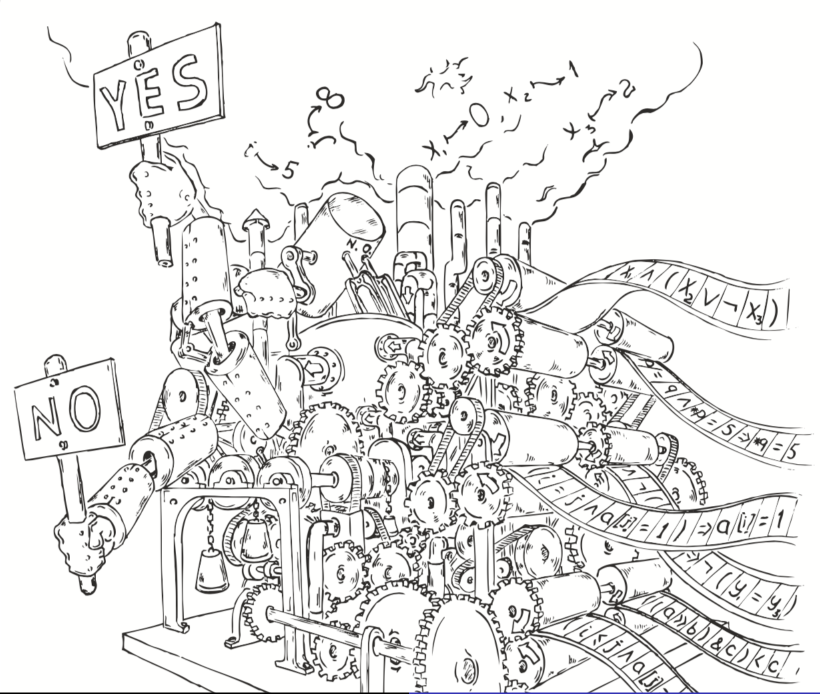
- То же самое, что и Berkmin
- Перед новым дизъюнктом кладем k дизъюнктов, которые участвовали в процессе получения нового дизъюнкта в процессе бинарных резолюций

Противоречивые ядра

- Противоречивое ядро противоречивой формулы f - подформула формулы f, которое является противоречивой.
- Минимальное противоречивое ядро противоречивой формулы f - противоречивое ядро, такое, что удаление любый дизъюнктов из этой формулы делает её выполнимой подформулу выполнимой.

Граф резолюций

- Вершины - дизъюнкты
- Корни - первоначальные дизъюнкты
- Есть ребро - если вершина, к которой было проведено ребро, получена в процессе бинарной резолюции через вершину, от которой было проведено ребро


YES
NO