Специальные формы КНФ

Содержание

- 1 КНФ в форме Крома
- 2 КНФ в форме Хорна
- 3 См.также
- 4 Примечания
- 5 Источники информации

Рассмотрим две формы, с помощью которых можно представить формулы, заданные в конъюнктивной нормальной форме, то есть имеющей вид конъюнкции выражений в скобках, каждое из которых представляет собой дизъюнкцию одного или нескольких литералов. Для двух этих форм существует алгоритм, который может за полиномиальное время проверить, существует ли набор аргументов, на которых данная функция будет принимать значение 1, в то время, как для обычной функции, не представленной данной формой, эта задача является NP-полной. Этот факт интересен потому, что, имея большое количество функций, которые можно свести к форме Хорна или Крома, мы сможем гарантированно вычислять необходимое нам условие за полиномиальное время. Поэтому с помощью применения данных форм мы сможем решать очень быстро целый класс задач, например, задачи на графах, которые, как известно, имеют большое практическое применение.

КНФ в форме Крома

Определение:

Конъюнктивная нормальная форма (англ. *conjunctive normal form, CNF*) **в форме Крома, 2-КНФ**[1] (англ. *2-CNF*) — конъюнкция выражений в скобках, каждое из которых представляет собой дизъюнкцию ровно двух литералов.

Пример :

$$(x_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_3) \wedge (\overline{x}_3 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_2) \wedge \ldots$$

Утверждение:

Существует алгоритм, который за полиномиальное время проверяет, что формулу, заданную в форме Крома, можно удовлетворить.

Утверждение:

Функцию F можно задать в форме Крома $\iff$ выполнено следующее следствие : $F(x_1, \ldots, x_n) = F(y_1, \ldots, y_n) = F(z_1, \ldots, z_n) = 1 \Rightarrow F(\langle x_1, y_1, z_1 \rangle, \langle x_2, y_2, z_2 \rangle, \ldots, \langle x_n, y_n, z_n \rangle)$

КНФ в форме Хорна

Определение:

Конъюнктивная нормальная форма (англ. *conjunctive normal form, CNF*) **в форме Хорна[2]** (англ. *Horn clause*) — это конъюнкция выражений в скобках, каждое из которых представляет собой дизъюнкцию литералов, в которой присутствует не более одного литерала без отрицания.

Пример:

$$(\overline{x}_1 \vee \overline{x}_2 \vee \ldots \vee \overline{x}_n) \wedge (x_1 \vee \overline{x}_2 \vee \ldots \vee \overline{x}_n) \wedge \ldots$$

Каждая скобка представляет собой Дизъюнкт Хорна[3].

Утверждение:

Существует алгоритм, который за полиномиальное время проверяет, что функцию, заданную в форме Хорна, можно удовлетворить.

$\triangleright$

Далее будет приведено доказательство, предлагающее алгоритм решения.

- **Шаг 1. Одиночное вхождение переменных.** Найдем в данной формуле одиночно стоящие переменные. Например, для формулы $x \wedge (x \vee \neg y \vee \neg z)$ такой переменной является x.
  1. Присутствуют одиночно стоящие переменные.

     Присвоим всем таким переменным значение 1, если переменная входит без отрицания и 0 иначе, так как в конъюнкции они должны дать 1. Заметим, что если какая-либо скобка после этого обратилась в 0, то решения не существует.

  2. Отсутствуют одиночно стоящие переменные.

     Всем переменным надо присвоить значение 0 и булева формула разрешится. Это следует из того, что в каждом дизъюнкте есть хотя бы одна переменная с отрицанием, подставив в нее значение 0 мы получим 1 в результате дизъюнкции. В итоге мы получим выражение вида: $1 \wedge 1 \wedge \ldots \wedge 1$, что в результате даст нам 1. В таком случае дальнейшие шаги выполнять не нужно.

- **Шаг 2.**

  Опустим одиночно стоящие переменные и скобки, в которых значение стало равным 1. Перейдём к 1 шагу алгоритма. По определению формы Хорна, в каждой из скобок, где мы опустили переменную, не больше 1 переменной без отрицания. Либо какая-то из переменных внутри скобки будет иметь отрицание, т.е. при подстановке 0 станет равна 1, либо мы рассмотрим переменную без отрицания как отдельно стоящую переменную. Значит 1 шаг алгоритма выполнится верно. Будем проделывать алгоритм, начиная сначала, пока 1 шаг не найдёт ответ.

Обозначим за N число вхождений переменных в формулу.

Итерация состоит из шагов, каждый из которых выполняется за $O(N)$. Всего итераций будет не больше N, так как если первый шаг не завершил алгоритм, то уменьшил размер формулы на одно вхождение. Итого, асимптотика алгоритма составляет $O(N^2)$.

◁

Утверждение:

Функцию F можно задать в форме Хорна $\iff$ выполнено следующее следствие:
$F(x_1, \ldots, x_n) = F(y_1, \ldots, y_n) = 1 \Rightarrow F(x_1 \wedge y_1, x_2 \wedge y_2, \ldots, x_n \wedge y_n)$

См.также

- СКНФ
- 2SAT
- ДНФ

Примечания

1. Wikipedia — 2-satisfiability (https://en.wikipedia.org/wiki/2-satisfiability)
2. Wikipedia — Horn clause (https://en.wikipedia.org/wiki/Horn_clause)
3. Википедия — Дизъюнкт Хорна (https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B7%D1%8A%D1%8E%D0%BD%D0%BA%D1%82_%D0%A5%D0%BE%D1%80%D0%BD%D0%B0)

Источники информации

- Wikipedia — CNF (https://en.wikipedia.org/wiki/Conjunctive_normal_form)