DOI:10.14096/j.cnki.cn34-1069/n.2014.04.009

第 31 卷第 4 期 阜阳师范学院学报(自然科学版) Vol. 31, No. 4
2014 年 12 月 Journal of Fuyang Teachers College (Natural Science) Dec. 2014

DPLL 算法及其改进的新方法

李培培

(阜阳师范学院 数学与统计学院,安徽 阜阳 236037)

摘 要: 对命题公式可满足性问题的判别方法进行了深刻的剖析,基于启发式算法,定义命题公式的核心文字,通过改进 DPLL 算法给出求解 SAT 问题的新方法。

关键词: 命题公式; 可满足性问题; 核心文字; 判别方法

中图分类号: TP301.5 **文献标识码:** A **文章编号:** 1004-4329(2014)04-037-03

DPLL algorithm and its new improvement

LI Pei-pei

(*School of Mathematics and Statistics Science, Fuyang Teachers College, Fuyang, Anhui, 236037, China*)

Abstract: This paper dissects the comparison method on the satisfiability of propositional formula. Defining the core literals of propositional formula, the paper designs a new improving method of DPLL algorithm for the satisfiability problem based on heuristic algorithm.

Key words: propositional formula; satisfiability problem; core literals; comparison method

命题公式的可满足性问题是计算机科学的中心问题之一,很多学者在这一领域做了大量的研究,命题逻辑内容虽然简单,但其在计算机科学、人工智能、数学等学科中发挥着重要的作用。文中对命题公式可满足性问题的判别方法进行了深刻的剖析,在启发式算法[1]的基础上,通过定义命题公式的核心文字,给出用于求解 SAT 问题的 DPLL 算法[2]的一种改进方法。

1 基本概念

命题变元及其否定统称为文字,命题变元称为正文字,而变元的否定称为负文字。若干个文字的析取称为子句,形式为 $C = L_1 \vee \cdots \vee L_n$,其长度是所含文字的个数。有时将一个子句 C 表示成文字集合 $\{L_1, \cdots, L_k\}$,k 称为子句 C 的长度,长度为 k 的子句称为 k-子句。只有一个文字的子句称为单子句,没有文字的子句称为空子句。子句的合取 ($F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$) 称为一个 CNF 公式(简称公式)。记 $Var(F)$ 为出现在公式 F 中的变元集合,$Lit(F)$ 为出现在公式 F 中的文字集合。有时将一个公式表示成一个子句集合 $\{C_1, \cdots, C_m\}$ 或一张表 $[C_1, \cdots, C_m]$[3]。若 $m = 0$,则称子句集为空,记为 Φ。含有一对互补文字的子句称为重言子句。如果公式 F 中的所有子句的长度不超过 k,称 F 为一个 k - CNF 公式[4]。命题公式 φ 是可满足的,如果存在某个真假赋值 τ,有 $\tau(\varphi) = 1$。一般来说,子句中的文字越多,该子句就越容易被满足,因此文字越少的子句越难满足。一个极端的情形是,空子句不可满足,但空公式 Φ 可满足[5]。

收稿日期: 2014-09-16

基金项目: 国家特色专业(TS11496); 安徽省专业综合改革试点项目(2013zy167); 阜阳师范学院专业综合改革试点项目(2013ZYSD05); 校级精品课程(2012KFKC10); 阜阳师范学院科研项目(2014FSKJ13)资助。

作者简介: 李培培(1982 -),女,硕士,讲师。研究方向: 计算复杂性。

2 DPLL算法及其改进的新方法

2.1 DPLL算法

解决可满足性问题的一个重要算法是DPLL算法。该算法是由Davis和Putnam提出的,Logemann和Loveland对它作了进一步的描述。有关DPLL算法的详细描述,请参阅文献[2]。命题公式可满足性问题的判定可使用真值表法。真值表与主合取范式一一对应,二者是描述公式等价的不同标准形式,因此使用主合取范式也可判断命题公式的可满足性。但当命题变项的个数较多时,真值表法与主合取范式法的计算量较大。下面简单叙述一下DPLL算法,设F是一个公式,在DPLL算法中,应用如下的4条规则对问题进行简化或分解[4]:

(1)重言式规则,(2)单子句规则,(3)纯文字规则,(4)分裂规则。

假设公式F应用上述4条规则中的任意规则后所得到的公式为F^*,则公式F和F^*有相同的可满足性[5]。目前,命题公式可满足性问题的判定算法大多是基于DPLL算法通过引进新的规则来实现的[7]。影响最大、应用最广的推导规则是"消解"规则及其改进[8],它是DPLL算法中单子句规则的推广。

2.2 DPLL算法改进

每一个CNF公式均可在多项式时间内转换成一个3-CNF公式,并且保留相同的可满足性,所以判定CNF公式的可满足性时,只考虑3-CNF公式[5]。下面给出核心文字的定义。

设F是一个CNF公式,$A\subseteq F$是F的一个子公式,L是$Var(F)$上的一个文字。

任何满足A的指派τ,均使得L取真值,则称L是A可满足的一个核心文字。对应的变元称为核心变元。反之,若L取假,则对任意的指派τ,都有$\tau(A)=0$。

通常可以由真值表、消解法或启发式算法[1]来寻找3-CNF公式的核心文字,但当变元较多时真值表法计算量较大。在启发式算法中,定义了一个启发式函数$g:Lit(F)\to M$。$g(w)$的计算结果越大,表明与w的关联强度大,则$\max\{g(w)\}$中的w为F可满足的核心文字。$g(w)$的计算依赖子句集$c(w)$,$c(w)$表示从包含w的三元子句中删去w后得到的二元子句集合或单子句集。

计算$g(w)$的算法如下:

```
int g(w)
{ i = i + 1;
  计算 c(w) 的值;
   if  i < M
```

$$\text{return}\sum_{(u\vee v)\in c(w)}(g(\neg u)g(\neg v))$$

```
  else
```

$$\text{retrrn}\sum_{(u\vee v)\in c(w)}((2c_1(\neg u)+c_2(\neg u))(2c_1(\neg v)+c_2(\neg v)))$$

```
}
```

注:$c_1(u)$,$c_2(u)$分别表示一元子句和二元子句的个数。

对于CNF公式F,确定三元子句中出现的文字集$\{t_1,t_2,\cdots,t_k\}$,调用计算$g(w)$的算法得到$g(t_1)$,$g(t_2)\cdots g(t_k)$,从而确定$\max\{g(w)\}$,并得到对应的核心文字w。根据核心文字优先原则对公式进行化简,化简公式时使用以下两种规则。

核心文字规则:设L是公式F的核心文字,从公式集S中删去包含L的所有子句,得到集合S_1。如果S_1是空集,则S可满足;否则对S_1中的每个子句,如果包含文字$\neg L$,则从该子句中删掉$\neg L$,得到集合S_2。S可满足当且仅当S_2可满足。

纯文字规则:如果一个文字L出现在当前公式F的某些子句中,并且文字$\neg L$不出现在F的其它任何子句中,那么L被称为纯文字。从S中删去包含文字L的所有子句,得到子句集S_1,那么S可满足当且仅当S_1可满足。

改进的DPLL算法描述如下:

假定所给定的3-CNF公式F的子句集为S,不断地按照核心文字规则和纯文字规则对S作变换,直到不能再进行为止。使用上述规则化简3-CNF公式,可以快速找到公式的可满足指派。如果化简后子句集合为空,则F可满足;否则,就不能判定公式的可满足性。

算法步骤:

步骤1:输入3-CNF公式F;

步骤2:调用启发式算法计算$\max\{g(w)\}$,找F的核心文字;

步骤3:对公式F的子句集S按顺序使用核心文字规则、纯文字规则进行化简,得到公式集S_1。若S_1为空,则F可满足;

步骤4:若得到的公式为2-CNF公式,则调用2-SAT算法[9];

步骤5:否则,重复步骤2)、3)、4)、5),直到不

能化简为止;

步骤6: 输出3-CNF 公式 F 是否可满足。

按照上述步骤和规则,不断的对子句集合进行化简。显然,算法是可以终止的。由于可以在线性时间内判定2-CNF 公式的可满足性[9],所以改进的 DPLL 算法可以在多项式时间内完成。

3　数值例子

用下面的例子验证改进的 DPLL 算法的可行性。

例1 设3-CNF 公式 $A = \{a \vee c \vee d, a \vee \neg c \vee \neg d, \neg a \vee \neg c \vee \neg d, \neg a \vee b \vee \neg f,$

$a \vee b \vee f, a \vee b \vee g, \neg a \vee \neg b \vee \neg g, \neg a \vee \neg c \vee d, \neg b \vee c \vee \neg e, b \vee d \vee \neg e, b \vee d \vee e, b \vee c \vee \neg d, \neg b \vee c \vee e\}$。

解: (1) 计算 $\max\{g(w)\}$ 对应的文字为 $\neg c$,对公式 A 使用核心文字规则化简得到: $\{\neg a \vee b \vee \neg f, a \vee b \vee f, \neg a \vee \neg b \vee \neg g, a \vee b \vee g,$

$b \vee d \vee \neg e, b \vee d \vee e, b \vee \neg d, \neg b \vee e, \neg b \vee \neg e\}$。

(2) 针对(1)中公式的三元子句,计算 $\max\{g(w)\}$ 且得到对应的文字为 a,使用核心文字规则得到: $\{b \vee \neg f, \neg b \vee \neg g, b \vee d \vee \neg e, b \vee d \vee e, b \vee \neg d\}$。

(3) e、$\neg e$ 对应的函数值一样,取其一为核心文字,使用核心文字规则化简得到:

$\{b \vee \neg f, \neg b \vee \neg g, b \vee d, b \vee \neg d\}$ 为2-CNF 公式,调用2-SAT 算法,判定知该公式是可满足的。

4　结束语

本文对命题公式的可满足性判定问题进行了深刻的剖析,通过定义命题公式的核心文字和引进核心文字规则,对 DPLL 算法进行改进,从而给出求解 SAT 问题的新算法。此算法可以在多项式时间内判定一些 CNF 公式的可满足性问题。

参考文献:

[1] Dubois O, Seymour P. A backbone-search heuristic for efficient solving of hard 3-SAT formula [C] //The 17th International Joint Conference on Artificial Intelligence, Seattle, San Francisco: Morgan Kauffman. Publishers, 2001: 248-253.

[2] 张　健. 逻辑公式的可满足性判定 [M]. 北京: 科学出版社,2000: 8-20.

[3] 许道云,王晓峰. 可满足性问题的归约技术 [J]. 逻辑学研究,2012,5(1): 35-49.

[4] 聂国霞,秦永彬,许道云. 基于因子图求解(3,4 =)-CNF 公式类下可满足问题 [J]. 计算机与数字工程,2013,41(5): 686-689.

[5] 张秋菊. 关键文字和极小不可满足公式 [D]. 贵州: 贵州大学理学院. 2008.

[6] 许道云,刘长云. 带文字改名策略的 DPLL 算法 [J]. 计算机科学与探索,2007,1(1): 116-125.

[7] 王晓峰,许道云,唐瑞雪. 基于 DPLL 的混合遗传算法求解 SAT 问题 [J]. 计算机工程与科学,2010,32(5): 54-56.

[8] Aharoni R, Linial N. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas [J]. Journal of Combinatorial Theory, Series A, 1986, 43(2): 196-204.

[9] Aspvall B, Plass M F, Tarjan R E. A linear-time algorithm for testing the truth of certain quantified Boolean formulas [J]. IPL, 1979, 8(3): 121-123.