基于 MOVR 启发式的求差知识编译算法

牛当当[1],吕 帅[2,3],王金艳[4]
(1. 西北农林科技大学信息工程学院,陕西杨凌 712100;2. 吉林大学计算机科学与技术学院,吉林长春 130012;
3. 符号计算与知识工程教育部重点实验室(吉林大学),吉林长春 130012;
4. 广西师范大学计算机科学与信息工程学院,广西桂林 541004)

摘 要: DKCHER 算法是基于超扩展规则的求差知识编译算法,也是目前为止表现最好的 EPCCL 理论编译算法. 本文通过研究 DKCHER 算法的执行流程,设计了一种新的启发式策略 MOVR(maximum occurrence number of variables in middle result),用于动态地从输入子句集中选择所包含变量在中间结果中出现次数最多的子句. 将 MOVR 启发式策略与 DKCHER 算法相结合,设计了 MOVR_DKCHER 算法. 实验结果表明,MOVR 启发式策略能够显著提高 DKCHER 算法的编译效率和编译质量,编译效率平均可提升 70 倍左右,最高可以提高 237 倍.

关键词: 知识编译; 扩展规则; 超扩展规则; EPCCL 理论; 启发式策略
中图分类号: TP301, TP181 **文献标识码:** A **文章编号:** 0372-2112 (2019)11-2299-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.11.009

Knowledge Compilation Algorithm of Computing Difference Based on MOVR Heuristics

NIU Dang-dang[1], LÜ Shuai[2,3], WANG Jin-yan[4]
(1. *College of Information Engineering, Northwest A&F University, Yangling, Shaanxi* 712100, *China*;
2. *College of Computer Science and Technology, Jilin University, Changchun, Jilin* 130012, *China*;
3. *Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun, Jilin* 130012, *China*;
4. *School of Computer Science and Information Engineering, Guangxi Normal University, Guilin, Guangxi* 541004, *China*)

Abstract: DKCHER is a knowledge compilation algorithm of computing difference based on hyper extension rule, which is the best EPCCL compiler so far. We research the executing process of DKCHER algorithm in this paper, and design MOVR (maximum occurrence number of variables in middle result) heuristics, which is used to dynamically select the clause with maximum occurrence number of variables in middle result. By combining MOVR heuristics with DKCHER, MOVR_DKCHER algorithm is designed. Experimentally, MOVR heuristics can significantly improve the compilation efficiency and compilation quality of DKCHER, in which the compilation efficiency of DKCHER can be averagely improved about 70 times, and the improvement can be up to 237 times at most.

Key words: knowledge compilation; extension rule; hyper extension rule; EPCCL theory; heuristic strategy

1 引言

知识编译是命题逻辑推理方法的重要组成部分,具有广泛的应用前景. 知识编译主要通过结合离线编译和在线推理求解重复性推理问题. 针对可分解否定范式(DNNF),Darwiche 提出了高效的完备知识编译方法[1],并给出了知识编译图谱[2]. Broeck 等人研究了约简的 SDDs(sentential decision diagrams)语言上多项式时间内的推理方法[3]. Lai 等人设计了带有蕴含文字的有序二元决策图,是一种新的知识编译语言[4],同时还基于合取的可分解性设计了几种新的知识编译目标语言[5]. Amarilli 等人研究了知识编译中的连接宽度和结构[6]. 上述研究提高了相关知识编译目标语言的应用范围、表示能力以及编译算法的性能.

收稿日期:2018-11-02;修回日期:2019-03-22;责任编辑:马兰英
基金项目:国家自然科学基金(No. 61502197, No. 61503044, No. 61763003, No. 61502111);吉林省科技发展计划资助项目(No. 20180101053JC);广西省自然科学基金(No. 2016GXNSFAA380192);西北农林科技大学博士科研启动基金(No. Z109021813)

2003 年,Lin 等人提出了扩展规则推理方法(extension rule,ER)[7],被著名人工智能专家 M. Davis 称为与归结方法互补的方法. 基于扩展规则,Lin 等人提出了知识编译算法 KCER,可以将子句集编译为 EPCCL 理论[8]. 针对 KCER 算法,谷文祥等人设计了 MCN 和 MO 两种启发式策略,提高了其编译效率和编译质量[9]. 刘大有等人提出了新的 EPCCL 理论编译器 C2E,该编译器具有较高的编译效率和编译质量[10]. 刘磊等人基于超扩展规则提出了求差知识编译算法 DKCHER,具有最优的 EPCCL 理论编译性能[11]. EPCCL 理论是一种高效的知识编译目标语言,能够在线性时间内支持知识编译图谱中的全部查询操作,因此提高 EPCCL 编译器的编译效率和编译质量具有重要意义.

启发式策略在多个领域中被成功用来提升算法的性能[12~14]. 显然,有效的启发式策略也能够提高 EPCCL 理论编译算法的编译效率和编译质量,如谷文祥等人针对 KCER 算法设计的高效启发式策略 MCN 和 MO[9]. 然而不同算法的执行流程差距较大,其中的选择过程和计算过程也有较大差别,因此将 MCN 和 MO 等现有启发式策略直接应用到 DKCHER 算法是不可行的.

本文通过深入研究 DKCHER 算法的执行流程,并从中提取出关键选择过程. 结合所提取选择过程,设计启发式策略 MOVR,从输入子句集中选择合适的与 EPCCL 理论完成求差操作的子句. 基于 MOVR 启发式,设计并实现 MOVR_DKCHER 算法. 最后,通过对比测试 MCN_DKCHER、MOVR_DKCHER 和 DKCHER 三种算法的编译效率和编译质量,验证 MOVR 启发式策略的有效性.

2 超扩展规则

为了便于表示,本文用 $V(C)$ 和 $V(F)$ 分别表示子句 C 和子句集 F 中出现的变量集,用 $J_N(C)$ 和 $J_N(F)$ 分别表示子句 C 和子句 F 在变量集 N 上所能扩展出的极大项集. 本文所研究的子句集中不包含重言式.

定义 1(超扩展规则,Hyper extension rule)[11] 给定两个子句 C 和 A,$D=\{C\vee A, C\vee\neg A\}$,其中 $V(C)\cap V(A)=\varnothing$,将 C 到 D 的推导过程称为超扩展规则,D 中的元素为 C 应用超扩展规则的结果.

超扩展规则具有以下特殊性质:基于超扩展规则能够利用 EPCCL 理论保存两个子句所能扩展出极大项集的差集[11].

定义 2(EPCCL 理论)[8] 子句集 F 是一个 EPCCL 理论,则 F 中任意两个子句间均含有互补文字对.

给定任意子句集 $F=\{C_1,\cdots,C_m\}$,令 N 表示 F 中出现的变量集. DKCHER 算法利用超扩展规则的性质计算 $J_N(\square)-(J_N(\square)-J_N(C_1)-\cdots-J_N(C_m))$,进而得到与 F 等价的 EPCCL 理论[11]. 显然,DKCHER 算法分两步执行:第一步计算 $J_N(\square)-J_N(C_1)-\cdots-J_N(C_m)$,并将计算结果保存为 EPCCL 理论. 假设第一步所得结果为 $\{E_1,\cdots,E_h\}$,则第二步计算 $J_N(\square)-J_N(E_1)-\cdots-J_N(E_h)$,得到与 F 等价的 EPCCL 理论. 在计算中间结果或输入子句集所能扩展出极大项的差集时,DKCHER 算法按照子句编号顺序求解. 事实上,DKCHER 算法求解过程中子句的选择顺序会极大地影响其编译效率和编译质量.

3 MOVR 启发式

根据 DKCHER 算法的计算过程,在每步计算中,需要顺序计算已得 EPCCL 理论与输入子句集中所有子句所能扩展出极大项集的差集. 好的子句选择顺序能够帮助 DKCHER 算法更快、更高质地完成编译工作,因此本节首先将针对 DKCHER 算法编译过程中的子句选择设计一种高效的启发式策略.

DKCHER 算法分两步执行,其中每步计算都是计算空子句□与输入子句集中所有子句所能扩展出极大项集的差集. 由于 DKCHER 算法的两步执行过程相同,因此在考虑子句选择策略时,可以仅考虑单步计算过程. 在 DKCHER 算法的单步计算过程中,假设输入的子句集(DKCHER 算法第二步计算的输入是一个 EPCCL 理论,同样也是子句集)的子句数为 m,则其需要完成 m 次 EPCCL 理论与某个子句的求差操作. 传统的 DKCHER 算法使用子句输入顺序作为编译过程中的求差操作顺序.

事实上,由 DKCHER 算法编译所得到的 EPCCL 理论可以通过规约得到编译质量更优的 EPCCL 理论. 刘大有等人实现了多项式时间的 REDUCE 算法,能够将 EPCCL 理论中任意满足规约规则的子句对规约为一条子句[10]. 然而使用 REDUCE 算法会较大地增加知识编译算法的整体时间开销,应避免在 DKCHER 算法中直接融合该算法. 若能在编译过程中通过选择子句使所得结果 EPCCL 理论中尽可能少地包含满足规约规则的互补文字对,将会使 DKCHER 算法的编译效率和编译质量得到双重提升. 综上,需要设计好的启发式选择策略帮助 DKCHER 算法在求差操作过程中选择合适的子句.

MCN 启发式策略作用于 KCER 算法编译过程中的子句选择,可以选择将 MCN 启发式作为子句选择策略与 DKCHER 算法相结合,用于其单次求差操作过程中的子句选择. 然而 MCN 启发式是结合 KCER 算法的特性所设计,因此该启发式并不一定适用于 DKCHER 算法. 而且 DKCHER 算法分两步执行,且第二步输入的子句集是一个 EPCCL 理论,在 EPCCL 理论中所有子句之间均包含互补关系,而 MCN 启发式通过选择与其它子

句互补最少的子句来完成选择过程,因此在 DKCHER 算法第二步的编译过程中 MCN 启发式的启发作用会失效. 因此有必要结合 DKCHER 算法的特性设计新的启发式策略.

本文通过选择变量在中间结果中出现次数最多的子句来确定 DKCHER 算法求差操作中的子句选择. 之所以选择变量出现次数最多的子句,是由于这样的子句较为活跃,会与较多的子句产生互补、蕴含、相交等关系. 若将较为活跃的子句放在算法执行末段处理,会与已得部分结果中的子句产生较多联系,而使得已得部分结果的规模随着算法的执行而剧烈增加,会极大地降低算法的性能. 因此本文优先处理较为活跃的子句.

给定一个子句集 F 和一个变量 v,本文用 $O_F(v)$ 表示 v 在 F 中出现的次数. 假设 DKCHER 算法单步求差计算的输入子句集为 $F=\{C_1,\cdots,C_m\}$,已得中间结果 EPCCL 理论 $E=\{D_1,\cdots,D_h\}$,对于 F 中任意子句 $C_x=l_1\vee\cdots\vee l_k$,文字 $l_1,\cdots,l_k$ 对应的变量分别为 $v_1,\cdots,v_k$,则 C_x 中文字在 E 中的出现次数为 $\sum_{1\leqslant j\leqslant k}O_E(v_j)$. DKCHER 算法分两步计算,每一步开始时中间结果 $E=\{\square\}$,这时输入子句集中所有子句包含的变量在 E 中出现的总次数都为 0,无法依据中间结果抉择出较为活跃的子句. 根据选择最活跃子句的原则,当中间结果 $E=\{\square\}$ 时,本文通过计算输入子句集 F 中每个子句包含的变量在 F 中出现的总次数来抉择出最优子句,即通过选择变量在输入子句集 F 中出现次数最多的子句确定第一步求差操作的子句选择;当 $E\neq\{\square\}$,本文则从输入子句集中选择变量在中间结果 E 中出现次数最多的子句. 将上述选择过程称为 MOVR 启发式策略,该策略的执行流程如算法 1 所示.

算法 1　MOVR

输入:子句集 $F=\{C_1,\cdots,C_m\}$,EPCCL 理论 $E=\{D_1,\cdots,D_h\}$,当前为第 d 次求差操作
输出:F 中的某个子句 C

```
1  初始化:b=0,i=1
2  If d=1
3      While i≤m
4          k=Σ_{1≤j≤|C_i|} O_F(v_j)
5          If k>b
6              C=C_i
7              b=k
8          i++
9  Else
10     While i≤h
11         k=Σ_{1≤j≤|C_i|} O_E(v_j)
12         If k>b
13             C=C_i
14             b=k
15         i++
16 Return C
```

算法 1 第 4 行和第 11 行中,v_j 为 C_i 中第 j 个文字对应的变量. 本文将 MOVR 启发式策略与 DKCHER 算法结合,设计了 MOVR_DKCHER 算法,该算法的执行流程如算法 2 所示.

算法 2　MOVR_DKCHER

输入:令子句集 $F_1=\{C_1,\cdots,C_n\}$,M 包含了 F 中出现的所有变量
输出:与 F_1 等价的 EPCCL 理论

```
1  初始化:F2={□},h=i=j=1
2  While i≤|F1|
3      C=MOVR(F1,F2,i)
4      F1=F1-{C}
5      While j≤|F2|
6          If C 与 Cj互补 Then skip
7          Else If C|=Cj Then F2=F2-{Cj}
8          Else Cj={Cj∨¬(C-Cj)}
9          j++
10     i++
11     j=1
12 If h=1 Then
13     F1=F2
14     h--
15     Goto 2
16 Else Return F2
```

本文设计 MOVR 启发式策略的目的在于:通过控制两阶段编译过程中的子句选择顺序,减少 DKCHER 算法所得中间结果 EPCCL 理论和最终结果 EPCCL 理论中满足规约规则的子句对数量,从而提高中间结果 EPCCL 理论和最终结果 EPCCL 理论的质量. MOVR 启发式策略将较为活跃的子句优先提供给 DKCHER 算法用于单次求差操作,能够在一定程度上避免 EPCCL 理论规模和满足规约规则子句对数的剧烈增长.

4　实验结果与分析

为了验证 MCN 启发式在 DKCHER 算法中的启发效果,本文将 MCN 启发式与 DKCHER 算法相结合,设计并实现了 MCN_DKCHER 算法. 本文在随机问题和通用测试用例上对比测试 MOVR_DKCHER 算法、MCN_DKCHER 算法和 DKCHER 算法[11]的编译效率和编译质量(主要使用编译结果中子句数量进行衡量). 本文实验平台如下:CPU:Intel(R) Core(TM) i5-6600 CPU @ 3.30 GHZ 3.31 GHZ,内存:8GB,操作系统:Windows 10.

4.1　固定子句长度的 3-SAT 子句集上的测试

本文用随机产生器生成了子句长度固定的 3-SAT 测试样例,即每个子句均包含三个文字,随机产生器的结果为包含两个参数 $<n,m>$ 的子句集,其中:n 为变量

个数,m 为子句个数. n 个变量对应 $2*n$ 个文字,每个文字出现的概率是相同的,同时控制生成过程,避免生成重言式.

表 1 给出了 $<20,m>$、$<25,m>$ 和 $<30,m>$ 三种随机测试样例的实验结果,所得结果为 50 次实验的平均值. 其中,size 表示子句个数,time 表示运行时间(单位为 ms);表中数据加粗表示当前行最优编译效率和质量. 下表类似,恕不赘述.

表 1 固定子句长度 3-SAT 实例上的实验结果

instances	MOVR_DKCHER		MCN_DKCHER		DKCHER	
	size	time(ms)	size	time(ms)	size	time(ms)
<20,65>	**395**	**9**	627	47	525	21
<20,75>	**140**	**4**	190	37	176	18
<20,85>	**39**	**2**	70	34	56	17
<20,95>	**12**	**1**	13	34	14	18
<25,77>	**905**	**34**	3449	419	1358	203
<25,87>	**518**	**14**	1666	372	790	165
<25,97>	**209**	**7**	289	283	326	182
<25,107>	**48**	**2**	170	285	62	145
<30,88>	**6311**	**1491**	12286	3865	10317	2156
<30,98>	**2324**	**217**	6279	2976	3708	1038
<30,108>	**763**	**30**	2375	2646	1421	1180
<30,118>	**223**	**10**	707	2313	396	970

首先,对比分析 MCN_DKCHER 算法和 DKCHER 算法的编译效率和编译质量. 显然,MCN 启发式策略并未提升 DKCHER 算法的编译效率和编译质量,反而使之有所下降. 原因正如第三节的分析,MCN 启发式策略并不适用于 DKCHER 算法的编译过程. 并且在 DKCHER 算法的第二阶段,MCN 启发式计算所有子句的信息值均相同,其启发作用完全失效. 接下来对比分析 MOVR_DKCHER 算法和 DKCHER 算法的编译效率和编译质量.

从编译质量上看:(1)MOVR 启发式策略始终能够提高 DKCHER 算法的编译质量,平均可提升 1.556 倍;(2)在子句集的规模较小时,MOVR 启发式对 DKCHER 算法编译质量的提升较为明显;(3)子句集规模较大时,虽然 MOVR_DKCHER 算法能始终保持较优的编译质量,然而 MOVR 启发式对 DKCHER 算法编译质量的提升效果并不明显,甚至并未提升. 造成上述现象的原因在于:当子句数和变量数的比值较小时,DKCHER 算法并不擅长编译此类问题,而 MOVR 启发式策略将与中间结果联系最紧密的子句放在最开始处理,越晚处理的子句与中间结果的关联越不明显,能够减少扩展中间结果的次数,因此 MOVR 启发式策略能够提升 DKCHER 算法的编译质量;而在子句数和变量数的比值较大时,DKCHER 算法本身就极擅长处理此类实例,其所得编译结果已经足够精简,所留提升空间并不大.

从编译效率上看:MOVR 启发式策略在大部分实例中能够提高 DKCHER 算法的编译效率. 跟编译质量方面观察的现象相反,当子句数和变量数的比值较小时,MOVR 启发式策略对于编译效率的提升并不明显,甚至有所降低;反而当子句数和变量数的比值较大时,MOVR 启发式策略能够显著提升 DKCHER 算法的编译效率. 造成上述现象的原因在于:MOVR 每次都选择与中间结果关联最紧密的子句,这样做虽然能够降低最终编译结果的规模,然而会使一开始中间结果的规模增长较快,当子句数与变量数的比值较小时,MOVR 启发式未能充分发挥其优势;而对于子句数与变量数的比值较大的实例,随着编译算法较多次求差操作的执行,MOVR 启发式策略能够快速降低中间结果的规模,因此能够显著提高 DKCHER 算法的编译效率.

4.2 相变点附近的 3-SAT 子句集上的测试

为了能够更全面地展示本文提出的启发式策略的特性,参照文献[11],本文同样选择了 SATLIB 中一些通用测试用例,并随机生成了相变点附近(子句数/变量数≈4.26)的 3-SAT 实例. 由于 MCN 启发式并不适用于 DKCHER 算法,因此本小节的对比测试仅在 DKCHER 算法和 MOVR_DKCHER 算法之间展开,所得结果同样为 50 次实验的平均值. 测试结果如表 2 所示.

表 2 相变点附近的 3-SAT 实例上的实验结果

instances	MOVR_DKCHER		DKCHER	
	size	time(ms)	size	time(ms)
uf20-01	**58**	**4**	78	16
uf20-02	**50**	**2**	100	14
uf20-03	**20**	**4**	**20**	10
blockworld-anomaly	**48**	**15**	**48**	109
pigeon-hole-6	**1**	**859**	**1**	3624
par8-1-c	**64**	**21**	**64**	31
<31,133>	**239**	**12**	354	1648
<32,137>	**121**	**11**	158	2442
<33,141>	**192**	**12**	246	2521
<34,146>	**101**	**16**	138	3720
<35,150>	**105**	**18**	189	4276

表 2 进一步验证了表 1 中子句数和变量数比值较大时,MOVR 启发式所具备的特性,即:MOVR 启发式策略能够提升 DKCHER 算法的编译质量,但提升效果并不明显;而在编译效率方面,MOVR 启发式效率能够发挥巨大启发作用,最高可提升 237 倍. 同时,通过表 2 也进一步说明了,MOVR 启发式在通用测试用例上同样能够起到很好的启发效果.

5 结论与展望

本文重点研究了 DKCHER 算法中的关键选择过程,设计了启发式策略 MOVR,能够动态地从输入子句集中选择变量在中间结果中出现次数最多的子句. 结

合 MOVR 启发式策略和 DKCHER 算法,实现了 MOVR_DKCHER 算法. 实验结果表明,MOVR_DKCHER 算法具有最优的编译质量和编译效率,在最好情况下,其编译效率是 DKCHER 算法的 237 倍.

未来将研究如何进一步提高 MOVR 启发式策略的启发效果,使其在子句数和变量数比值较小时依然能够较显著提升 DKCHER 算法的编译效率,以及如何使 DKCHER 算法在处理这类实例时具备更好的编译质量.

参考文献

[1] Darwiche A. Decomposable negation normal form[J]. Journal of the ACM,2001,48(4):608 – 647.

[2] Darwiche A, Marquis P. A knowledge compilation map[J]. Journal of Artificial Intelligence Research,2002,17:229 – 264.

[3] Broeck G V, Darwiche A. On the role of canonicity in knowledge compilation[A]. Proceedings of 29th AAAI Conference on Artificial Intelligence[C]. Austin, Texas, 2015. 1641 – 1648.

[4] Lai Y, Liu D Y, Wang S S. Reduced ordered binary decision diagram with implied literals: A new knowledge compilation approach[J]. Knowledge and Information Systems,2013,35(3):665 – 712.

[5] Lai Y, Liu D Y, Yin M H. New canonical representations by augmenting OBDDs with conjunctive decomposition[J]. Journal of Artificial Intelligence Research,2017,58:453 – 521.

[6] Amarilli A, Monet M, Senellart P. Connecting width and structure in knowledge compilation[A]. Proceedings of 21st International Conference on Database Theory[C]. Vienna, Austria, 2018. 6:1 – 6:17.

[7] Lin H, Sun J G, Zhang Y M. Theorem proving based on the extension rule[J]. Journal of Automated Reasoning,2003,31(1):11 – 21.

[8] Lin H, Sun J G. Knowledge compilation using the extension rule[J]. Journal of Automated Reasoning,2004,32(2):93 – 102.

[9] 谷文祥,王金艳,殷明浩. 基于 MCN 和 MO 启发式策略的扩展规则知识编译方法[J]. 计算机研究与发展,2011,48(11):2064 – 2073.
GU Wen-Xiang, WANG Jin-Yan, YIN Ming-Hao. Knowledge compilation using extension rule based on MCN and MO heuristic strategies[J]. Journal of Computer Research and Development, 2011, 48(11): 2064 – 2073. (in Chinese)

[10] 刘大有,赖永,林海. C2E:一个高性能的 EPCCL 理论编译器[J]. 计算机学报,2013,36(6):1254 – 1260.
LIU Da-You, LAI Yong, LIN Hai. C2E: An EPCCL compiler with good performance[J]. Chinese Journal of Computer,2013,36(6):1254 – 1260. (in Chinese)

[11] 刘磊,牛当当,吕帅. 基于超扩展规则的知识编译方法[J]. 计算机学报,2016,39(8):1681 – 1696.
LIU Lei, NIU Dang-Dang, LÜ Shuai. Knowledge compilation methods based on the hyper extension rule[J]. Chinese Journal of Computers,2016,39(8):1681 – 1696. (in Chinese)

[12] Niu D D, Liu L, Lü S. Knowledge compilation methods based on the clausal relevance and extension rule[J]. Chinese Journal of Electronics,2018,27(5):1037 – 1042.

[13] 柳强,何明,刘锦涛,牛彦杰,黄倩. 无人机“蜂群”的蜂拥涌现行为识别与抑制机理[J]. 电子学报,2019,47(2):374 – 381.
LIU Qiang, HE Ming, LIU Jin-Tao, NIU Yan-Jie, HUANG Qian. A mechanism of for identifying and suppressing the emergent flocking behaviors of UAV swarms[J]. Acta Electronica Sinica,2019,47(2):374 – 381. (in Chinese)

[14] Li H B, Liang Y C, Zhang N, Guo J S, Xu D, Li Z S. Improving degree-based variable ordering heuristics for solving constraint satisfaction problems[J]. Journal of Heuristics,2016,22(2):25 – 145.

作者简介

牛当当(通信作者)　男,1990 年 2 月出生,陕西周至人. 现为西北农林科技大学信息工程学院讲师,主要研究方向为自动推理和抽象论辩.
E-mail:niudd@nwafu.edu.cn

吕　帅　男,1981 年 7 月出生,吉林公主岭人. 2010 年获得吉林大学博士学位,现为吉林大学计算机科学与技术学院副教授,主要研究方向为人工智能、智能规划与自动推理.
E-mail:lus@jlu.edu.cn