第20卷 第2期
2010年2月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.20 No.2
Feb. 2010

PSL 可满足问题的计算复杂度

虞 蕾[1,2]

(1.国防科学技术大学 计算机学院,湖南 长沙 410073;
2.第二炮兵工程学院 计算机系,陕西 西安 710025)

摘 要:PSL 是一种用于描述并行系统的属性规约语言,包括线性时序逻辑 FL 和分支时序逻辑 OBE 两部分。由于 OBE 就是 CTL,因此论文重点研究 FL 逻辑。理论上已证明许多难解的问题都可多项式变换为"可满足性"问题,"可满足性"问题是研究时序逻辑的核心问题之一,并已成为程序验证的一种有力工具;而计算复杂度是"可满足性"问题需要解决的最深刻的方向之一,其研究意义在于它可作为解决一类问题的难度的标准。文中在利用"铺砖模型"基础上,推导并得出 FL 的"可满足性"问题的计算复杂度为 EXPSPACE - hard,这对正确评价解决该问题的各种算法的效率,进而确定对已有算法的改进余地具有重要的指导意义。

关键词:PSL;可满足性问题;计算复杂度

中图分类号:TP301.5 **文献标识码**:A **文章编号**:1673-629X(2010)02-0016-05

Computational Complexity of Satisfiability Problems for PSL

YU Lei[1,2]

(1.Computer School, National University of Defense Technology,Changsha 410073,China;
2.Department of Computer Science, Second Artillery Engineering College,Xi'an 710025,China)

Abstract: PSL is a property specification language which describes parallel systems and is divided into two parts, i.e. FL and OBE. Because OBE is essentially the temporal logic CTL(computation tree logic), and PSL formulas with clock statements can be easily rewritten to unclocked formulas, this work studies with emphasis on the unclocked FL logic. It has long been verified that many hard problems can be translated into the correspongding satisfiability(SAT) problems in polynomial time, and SAT problems are central for temporal logic in order to be a useful tool in program verification. The computational complexity is one of the most impotant theories to be resloved for SAT problems in order to bring on the hardness criterion for one series of problems. The complexity of FL has been studied with the help of the computation model of"tiling" and the result shows the satisfiability of FL is EXPSPACE - hard, which has important values for evaluating the efficiency and further determining the improvement of FL SAT problems' algorithms.

Key words: property specification language;satisfiability problem;computational complexity

0 引 言

PSL[1]是由 EDA 界的标准化组织 Accellera 提出,并于2005年确定为 IEEE 工业标准(IEEE - 1850)的用于描述并发系统的属性规约语言。IBM 的 Sugar 语言是形成 PSL 的基础[2,3]。PSL 主要有三大用途:(1)形式化验证;(2)动态验证仿真;(3)功能规约描述(作为功能规约文档)。由于 PSL 易于读写,语法精简,语义清晰,表达能力强,因此作为一种功能描述语言得到了广泛的应用。

收稿日期:2009-05-31;**修回日期**:2009-08-17

基金项目:中国博士后科学基金(20080431401)

作者简介:虞 蕾(1978-),女,浙江浦江人,博士后,讲师,研究方向为模型检验和航迹规划。

PSL 是一种分层语言,包括4层:布尔层、时序层、验证层和建模层。时序层是语言的核心层,用于实现语言的主要功能,即描述设计模块的时序属性,可分为 FL(Foundation Language,基础语言)和 OBE(Optional Branching Extension,可选分支扩展)两部分。FL 是线性时序逻辑,而 OBE 是一种分支时序逻辑。时序层的 FL 包含命题操作子、LTL 操作子、定义时间粒度的时钟操作子、连续扩展正规表达式(SEREs)以及 abort 操作子。FL 公式的时钟声明可视为语法 Sugar,具有时钟声明的公式很容易改写成非时钟公式(改写规则见官方语言标准[4])。由于 OBE 就是 CTL(computation tree logic)[5],因此文中重点研究 PSL 的线性非时钟逻辑 FL 的"可满足性"问题的复杂度。

"可满足性"问题(satisfiability,SAT)最初是由

Cook 于 1971 年首先提出的：已知 F 是关于变量 $x_1, x_2, \cdots, x_n$ 的合式公式，如果真值赋值 $t: \{x_1, x_2, \cdots, x_n\} \to \{0,1\}$ 使得 $t(F)=1$，则称 t 为 F 的一个成真赋值；如果合式公式 F 有成真赋值，则称 F 是可满足的。因此“可满足性”问题表述为：任给一个合取范式，问 F 是否是可满足的？此定义可作为“可满足性问题”的描述原型。在此基础上进一步推广，定义时序逻辑公式可满足性问题：任给某一时序逻辑形式系统 TLFS 和时序逻辑公式 u，问 u 是否是可满足的？理论证明许多难解的问题都可多项式变换为“可满足性”问题。立足于计算机算法理论，计算复杂度是“可满足性”问题需要研究的最深刻的问题之一；计算复杂度也是计算机理论中一个极其重要的研究领域，其理论发源于 20 世纪 60 年代，以多项式时间上界的图灵机为基本计算模型而奠定了理论基础。研究可满足性问题的计算复杂度的意义在于它可作为研究一类问题的难度的标准。若已知一个问题的计算时间下界，就可以较正确地评价解决该问题的各种算法的效率，进而确定对已有算法还有的改进余地。在许多情况下，确定一个问题内在的计算复杂度是很困难的，但是已创造出的各种分析问题计算复杂度的方法和工具，可以较准确地确定许多问题的复杂度。文中研究的 PSL 的“可满足性”问题的计算复杂度就借助了著名的“铺砖问题”[6] 的这一计算模型。

1　相关工作

文献[7～9]详细给出了命题线性时序逻辑的可满足性问题的计算复杂度，这个问题的较为广泛的研究直接以它们在从规约(specification)到得出合成的并行系统的应用为驱动的。Sistla 和 Clarke[7]指出 LTL 表示的可满足性问题的计算复杂度是 NP－完全或 PSPACE－完全的；但是如果有命题操作子的限制下，复杂度将降低。其中的一种限制方法就是允许语言中存在某一“命题操作子的度”(degree of propositionality)，也即限制允许使用的命题操作子的集合[10]。Lewis[11]通过例举命题逻辑案例最早分析出“命题词限制”对问题可满足性问题复杂度的影响。文献[12]推导出模态命题逻辑复杂度的“三分法”：可满足性问题为 PSAPCE－完全、coNP－完全或 P；这个根据对命题操作子的限制不同而得出的完整分类利用了布尔函数闭集合的“Post 格(Post Lattice)”结构。文献[10]利用分析时序和命题逻辑结合的可满足性问题，得出 LTL 的广义计算复杂度是 PSPACE－完全、NP－完全或 P 的。

文献[13]指出 CTL 可满足性问题复杂度是 EXPTIME－完全的。文献[14]进一步指出 Emerson 和 Halpera 提出的 CTL＊的复杂度为 2EXPTIME－完全。研究的方法是：利用混合树自动机可将 CTL＊与 Vardi 和 Wolper 提出的 process logic(YAPL)的满足性问题简化(reduction)为判空问题，从而使其复杂度成为非确定型双指数时间可识别的语言类；文献进一步使用 Kripke 结构对图灵机运算进行编码，Kripke 结构中的每一个状态对应于图灵机工作带上的一个方格，这使得 CTL＊和 YAPL 的复杂度具有确定型双指数时间下界。文献[15]给出在时序操作子限制下的 CTL 和 CTL＊片断(fragment)的可满足性问题的复杂度存在三种可能：CTL 片断可为 EXPTIME、PSPACE 和 NP；CTL＊片断为 2EXPTIME、PSPACE 和 NP。PCTL (Probabilistic CTL，概率 CTL)是对 CTL 的进行概率扩展的逻辑，即用量化满足某一路径公式后所有运行(run)的可能性的概率操作子替代全称或存在路径量词而得到的逻辑；迄今为止，PCTL 的可满足性问题复杂度还未得到验证结果，但是文献[16]给出了 PCTL 的性质片断(quanlitative fragment)的复杂度。

μ－演算是命题模态逻辑上添加最大和最小不动点操作子组成的逻辑，而完全增强型 μ－演算 μfull enriched μ－calculas)则是在 μ－演算基础上再添加逆程序、分级模态词以及名词性词(nominal)，文献[17]利用在无穷计算树上运行且具有奇偶性接收条件的完全增强型自动机(full enriched automata，FEAs)，得到有关各种完全增强型 μ－演算片断的可满足性问题的计算复杂度。

Reset－LTL 是由 LTL 和 ForSpec 语言的 reset 操作子共同组成的逻辑，文献[18]利用“铺砖问题”的计算模型证明其复杂度为 PSPACE－完全；Aborts－LTL 是由 LTL 和 Sugar2.0 的 abort 操作子组成的逻辑，其复杂度为 SPACE(exp(k, n))－完全的(其中 k 为 abort 的个数，n 为公式长度)。文献[19]研究了不包含 abort 操作子情况下的 PSL 的各种片断的可满足性问题的复杂度。

另外，一些文献还研究了其它一些逻辑的可满足性问题，如文献[20]给出了各种 PDL(Propositional Dynamic Logics，命题动态逻辑)的扩展片断可满足性问题的计算复杂度。

2　预备知识

2.1　PSL 语法与语义

2.1.1　PSL 语法

定义 1[1](连续扩展正规表达式 Sequential Extended Regular Expression，SERE)。

每个布尔表达式 b 是一个 SERE；如果 r, r_1, r_2 是 SEREs，则以下各项都是 SERE：

· $\{r\}$ · $r_1;r_2$ · $r_1:r_2$ · $r_1 \mid r_2$

· $r_1\&\&r_2$ · $[*0]$ · $r[*]$

定义 2[1]（基础语言公式，Foundation Language Formulas，FL formulas）。

如果 b 是布尔表达式，则 b 和 $b!$ 都是 FL formulas；

如果 φ 和 Ψ 是 FL formulas，r 是 SERE，b 是布尔表达式，则以下各项都是 FL formulas：

· (φ) · $\neg\varphi$ · $\varphi \wedge \Psi$ · $r!$ · r · $X!\varphi$

· $[\varphi \cup \Psi]$ · φ abort b · $r \mapsto \varphi$

定义 3[1]（可选分支扩展公式，Optional Branching Extension Formulas，OBE formulas）。

每个布尔表达式是一个 OBE 公式；如果 f, f_1, f_2 是 OBE 公式，则以下各项都是 OBE 公式：

· (f) · EX f · $\neg f$ · E$[f_1 \cup f_2]$ · $f_1 \wedge f_2$ · EG f

定义 4[1]（Accellera PSL 公式）。

每个 FL 公式都是一个 Accellera PSL 公式；每个 OBE 公式都是一个 Accellera PSL 公式。

2.1.2 PSL 的语义

2.1.2.1 非时钟 SEREs 语义

$w \vdash r$ 表示 w 紧满足（tightly satisfies）r，其中，w 是定义在 $\sum = 2^p \cup \{\perp, \top\}$ 上的有穷字，原子命题 $p \in AP$，b 为布尔表达式，r, r_1, r_2 是非时钟 SEREs，则：

(1) $w \vdash \{r\} \Leftrightarrow w \vdash r$

(2) $w \vdash b \Leftrightarrow |w| = 1$ 且 $w^0 \vdash b$

(3) $w \vdash r_1;r_2 \Leftrightarrow \exists w_1, w_2$, s.t. $w = w_1w_2$, $w_1 \vdash r_1$ 且 $w_2 \vdash r_2$

(4) $w \vdash r_1:r_2 \Leftrightarrow \exists w_1, w_2, l$, s.t. $w = w_1lw_2$, $w_1l \vdash r_1$ 且 $lw_2 \vdash r_2$

(5) $w \vdash r_1 \mid r_2 \Leftrightarrow w \vdash r_1$ 或 $w \vdash r_2$

(6) $w \vdash r_1 \&\& r_2 \Leftrightarrow w \vdash r_1$ 且 $w \vdash r_2$

(7) $w \vdash [*0] \Leftrightarrow w = \varepsilon$

(8) $w \vdash r[*] \Leftrightarrow w \vdash [*0]$ 或 $\exists w_1, w_2$, s.t. $w_1 \neq \varepsilon$, $w = w_1w_2$, $w_1 \vdash r$ 且 $w_2 \vdash r[*]$

2.1.2.2 非时钟 FL 公式语义

式 $w \models \varphi$ 表示 w 满足 φ，其中，w 是一个有穷或无穷字，b 是布尔表达式，r 是非时钟 SERE，φ 和 Ψ 是非时钟 FL 公式，则：

(1) $w \models (\varphi) \Leftrightarrow w \models \varphi$

(2) $w \models \neg\varphi \Leftrightarrow \neg\, w \models \varphi$

(3) $w \models \varphi \wedge \Psi \Leftrightarrow w \models \varphi$ 且 $w \models \Psi$

(4) $w \models b! \Leftrightarrow |w| > 0$ 且 $w^0 \vdash b$

(5) $w \models b \Leftrightarrow |w| = 0$ 或 $w^0 \vdash b$

(6) $w \models r! \Leftrightarrow \exists j < |w|$, s.t. $w^{0\cdots j} \vdash r$

(7) $w \models r \Leftrightarrow \forall j < |w|$, s.t. $w^{0\cdots j}T^w \models r!$

(8) $w \models X!\varphi \Leftrightarrow |w| > 1$ 且 $w^{1\cdots} \models \varphi$

(9) $w \models [\varphi \cup \Psi] \Leftrightarrow \exists k < |w|$, s.t. $w^{k\cdots} \models \Psi$，且 $\forall j < k, w^{j\cdots} \models \varphi$

(10) $w \models \varphi$ abort $b \Leftrightarrow w \models \varphi$ 或 $\exists j < |w|$ s.t. $w^j \vdash b$ 且 $w^{0\cdots j-1}T^w \models \varphi$

(11) $w \models r \mapsto \varphi \Leftrightarrow \forall j < |w|$, s.t. $w^{0\cdots j} \vdash r$, $w^{j\cdots} \models \varphi$

根据 Cook 提出的“可满足性”问题的描述，定义 PSL（PSL 的非时钟 FL 逻辑部分）的“可满足性问题”：对 SERE 公式 r，如果存在真值赋值 $t_1:(x_1, x_2, \cdots, x_{n1}\} \to \{0,1\}$，$w_1 = x_1x_2\cdots x_{n1}$，其中 $x_i \in \sum = 2^p \cup \{\perp, \top\}$，$i = 1, \cdots, n_1$ 且存在正整数 k 有 $n_1 \leq k$，使得 $t_1(r) = 1 \Leftrightarrow w \vdash r$；或对 FL 公式 φ，如果存在真值赋值 $t_2:(x_1, x_2, \cdots, x_{n2}\} \to \{0,1\}$，$w_2 = x_1x_2\cdots x_{n2}$，其中 $x_i \in \sum = 2^p \cup \{\perp, \top\}$，$i = 1, \cdots, n_2$ 且 $n_2 = +\infty$ 或存在正整数 k 有 $n_2 \leq k$，使得 $t_2(\varphi) = 1 \Leftrightarrow w \models \varphi$，则称 PSL 公式 r 或 φ 是可满足的。

2.2 计算复杂度理论

借助图灵机这一基本计算模型，可定义各种复杂性类，进而计算“可满足性问题”的复杂度。

定义 5[21,22]（图灵机的空间复杂度）已知确定型图灵机 DTM M，$\sum$ 为字母表，对 $\forall w \in \sum^*$，M 总停机。把 M 关于 w 的计算中在每一条动作带上使用的方格数的最大值记为 $s_M(w)$；或已知非确定图灵机 NTM M，M 关于 w 的每一个计算都停机，把 M 关于 w 的所有计算中在每一条动作带上使用的方格数的最大值记为 $s_M(w)$，则称 $s_M(n) = \max\{s_M(w) \mid w \in \sum^*$，且 $|w| \leq n, n \in N\}$ 为 M 的空间复杂度。

设全函数（非递减函数）$s: N \to N$，若对几乎所有 n 有 $s_M(n) \leq s(n)$（即 $s_M(n) = O(s(n))$），则称 $s(n)$ 是 M 的空间复杂度上界，称 M 是 $s(n)$ − 空间 DTM 或 NTM。

定义 6[21,22] L 表示某一语言，DSPACE$(s) = \{L \mid$ 存在 $s(n)$ − 空间 DTM 接受语言 $L\}$；NDSPACE$(s) = \{L \mid$ 存在 $s(n)$ − 空间 NTM 接受语言 $L\}$。

若 F 是个函数族，则多项式空间可识别的语言类：DSPACE$(F) = \bigcup_{s \in F}$DSPACE(s)；

非确定型多项式空间可识别的语言类：NDSPACE$(F) = \bigcup_{s \in F}$NDSPACE(s)；

指数空间可识别的语言类：EXPSPACE $= \bigcup_{F>0}$DSPACE$(2^{Fn}) = \bigcup_{k<0}$DSPACE$(2^{O(n^k)})$。

定义 7[21,22]　(多对一归约)设 A,B 为两个语言,A 多对一归约到 B 指存在可计算的函数 f: $\sum^* \to \Gamma^*$,对 $\forall x \in \sum^*$,有 $x \in A \Leftrightarrow f(x) \in B$。记为 $A \leq_m B$,f 简称为归约函数。

定义 8[21,22]　设 C 是某个复杂度类:

(1)A 为 C - 困难的(C - hard)$\Leftrightarrow \forall_{B \in C}(B \leq_m A)$;

(2)A 为 C - 完全的(C - complete) $\Leftrightarrow A \in C \wedge A$ 为 C - 困难的。

由定义 8 可知如果一问题是对另一属于 C - 完全问题的适当简化形式,则此问题的复杂度可减为 C - hard。

3　PSL 可满足性问题的计算复杂度推理

首先介绍一种能表示图灵机运算的计算模型——"铺砖问题"[6]:将大小相同的正方形瓷砖按一定要求覆盖一给定的区域。瓷砖的每条边上都有固定的颜色,并且边有上下左右之分。每一种瓷砖的数量不限,铺砖时瓷砖相邻的边必须颜色相同,同时瓷砖不允许旋转和翻转。而超指数型铺砖问题[18]是"铺砖问题"的一个特例,其复杂度被证明是 SPACE($\exp(k,n)$) - complete,其中 $\exp(1,n) = 2^n$,$\exp(k,n) = 2^{\exp(k-1,n)}$。超指数型铺砖问题描述为:给定有限瓷砖集合 T,瓷砖水平相邻关系 $V \subseteq T \times T$ 表示 T 中所有可以按水平方向铺放的瓷砖对的集合,即第一块瓷砖的右边与第二块瓷砖的左边的颜色相同,瓷砖垂直相邻关系 $H \subseteq T \times T$ 表示 T 中所有可以按垂直方向铺放的瓷砖对的集合,初始瓷砖和末瓷砖分别为 t_0 及 t_a,限度 $n > 0$。判断是否存在一种有 $\exp(k,n) \times m$ 格的瓷砖铺法且 $m > 0$,有:(1)t_0 位于左下角,t_a 位于左上角;(2)瓷砖水平相邻关系属于 H,瓷砖垂直相邻关系属于 V。存在函数 f:$(\exp(k,n) \times m) \to T$,有(1)$f(0,0) = t_0$,$f(0,m-1) = t_a$;(2)对于任意 $0 \leq i < \exp(k,n)$,以及 $0 \leq j < m$,有$(f(i,j), f(i+1,j)) \in H$;(3)对于任意 $0 \leq i < \exp(k,n)$,以及 $0 \leq j < m-1$,有$(f(j,i), f(j,i+1)) \in V$。

其次,给出具有 $\exp(k,n)$ 个字母的字的定义方法:对于一个长度为 $\exp(k,n)$ 的字,字中每个字母用一个$(k-1)$ - block 表示,每个$(k-1)$ - block 长度为 $\exp(k-1,n)$,且用 $\#_{k-1}$ 标示开始;$(k-1)$ - block 表示了每个字母和字母在字中的位置信息。(1)$(k-1)$ - block 可作为一个 $\exp(k-1,n)$ 的计数器,也即第一个$(k-1)$ - block 计数值为 0,以后的$(k-1)$ - blocks 计数值为与 $\exp(k,n)$ 取模后的结果;(2)若存在两个$(k-1)$ - blocks 记为 b_1 和 b_2 分别位于第 1 和第 2 个字的同一位置的字母,则这两个字母相同。为表达(1)和(2),须分析$(k-1)$ - block 各个字节位的组成信息。$(k-1)$ - block 由 $\exp(k-1,n)$ 个$(k-2)$ - blocks 组成;同样,$(k-2)$ - blocks 可作为 $\exp(k-2,n)$ 的计数器。用标记$(k-2)$ - blocks 起始位置的命题 C^{k-1} 值表示此时$(k-1)$ - block 的字节数。

进一步,将此铺砖问题简化(reduction)为 PSL 的可满足性问题。已知一铺砖问题 $\tau = < T, H, V, t_0, t_a, n >$,其中 T, H, V, t_0, t_a 的含义同上,n 表示铺砖布局中每行的瓷砖块数。构建 PSL 公式 φ_τ,τ 允许上述瓷砖铺法当且仅当 φ_τ 是可满足的。其思想就是将一个铺砖布局编码成定义在 T 上的一个字,这个字由一系列长度为 $l = \exp(k,n)$ 的块组成(其中 k 为 PSL 中 abort 的个数,n 为公式长度),每个块定义瓷砖布局的一行。每一个$(k-1)$ - block 表示一个字母及其在字中的位置信息。若一个字从初始瓷砖 t_0 开始到表示 t_a 的块结束,则这个字代表一个正确的瓷砖铺法。

在已知一超指数型铺砖问题 $< T, H, V, t_0, t_a, l = \exp(k,n) >$ 的前提下,构建一 k 层公式 φ_k。当且仅当存在一 $l \cdot m$ 的 $< T, H, V, t_0, t_a, l >$ 瓷砖布局时,φ_k 是满足的。每个字母用一$(k-1)$ - block 表示,同时每行用相应的 k - block 表示。用命题 \$ 标记 $l \cdot m$ 长度字后的第一个单元(瓷砖),由于 m 是无界的,因此要求 \$ 首先出现在表示 t_a 瓷砖的 k - block 之后;同时用命题 @ 标记表示此时 k - block 的第一个单元。此铺砖问题模型必须满足 4 个条件,针对 PSL,分别将这 4 个条件转化为利用 PSL 操作算子构造的公式。φ_k 即为这 4 个公式的合取形式。下面给出具体的实现过程:

① 首先满足第一个 @ 出现在表示最末瓷砖 t_a 的 k - block 行首。

令 $\alpha(n,k) = T; (\neg \chi_0)[*]; \chi_0; (\neg \chi_0)[*] \&\& (c_0; T[*]; c_0) \mid (\neg c_0; T[*]; \neg c_0): (c_1; T[*]; c_1) \mid (\neg c_1; T[*]; \neg c_1) \cdots: (c_{\exp(k-1,n)}; T[*]; c_{\exp(k-1,n)}) \mid (\neg c_{\exp(k-1,n)}; T[*]; \neg c_{\exp(k-1,n)})$,$\alpha(n,k)$ 是应用 PSL 的所有 SERE 操作算子的组成的表达式,其长度为 $\exp(k,n)$,其中,$\chi_0 := \wedge_{i=0}^{n-1} \neg c_i$ 表示铺砖布局中所有行的开端(即每行的第一单元);$c_0, \cdots, c_{\exp(k-1,n)}$ 表示任意行中每个单元与 $\exp(k,n)$ 取模后的位置信息。$(c_i; T[*]; c_i) \mid (\neg c_i; T[*]; \neg c_i)$ 表示某行内某个单元与 $\exp(k,n)$ 取模后的位置信息为 c_i 或 $\neg c_i$,而后每前进一个单元加 1 计数,当计数个数满 $\exp(k-1,n)$ 即一轮后,其位置信息值又返回相应的 c_i 或 $\neg c_i$。

则满足要求 ① 的 PSL 公式表示为:

$$\varphi_{a1} = G((\neg @)U(@ \wedge (\#_{k-1} \wedge (\#_{k-2} \rightarrow C_0^{k-1})U(\alpha(n,k)) \mid\!\rightarrow \#_{k-1}))) \qquad (1)$$

其中,C_0^{k-1} 表示标记在$(k-2)$ - blocks 开端的由这些$(k-2)$ - blocks 组成的$(k-1)$ - block 的字节数。

② 其次,要求这样的 @ 最终会出现,并且第一个 \$ 紧跟在包含第一个 @ 的 k - block 行后。

$$\varphi_{a2} = G(F@ \wedge (\neg @ \wedge \neg(\$)U(@ \wedge X((\neg \#_k \wedge \neg \$)U(\#_k \wedge \$))))) \qquad (2)$$

③ 若当前字母是 t_i,则下一个字母 t_j 需要满足 $(t_i, t_j) \in H$,则有

$$\varphi_{h1} = G(\wedge_{t_i \in T}((\#_{k-1} \wedge t_i) \rightarrow X((\neg \#_{k-1})U(\#_{k-1} \wedge \vee_{t_j \mid (t_i,t_j) \in H} t_j)))) \qquad (3)$$

除每行的最后一个字母外,其它字母都必须满足上述条件,则有

$$\varphi_{h2} = G(((\neg \#_{k-1} \wedge ((\#_{k-2} \rightarrow C_1^{k-1})U\alpha(n,k) \mid\!\rightarrow \#_{k-1}))) \rightarrow \varphi_{h1})U\$) \qquad (4)$$

④ 对每个 $0 \le i < m, 0 \le j < l$,有$(f(j,i), f(j, i+1)) \in V$,利用文献[18]已定义的 $\theta_=^{k-1}$ 公式要求当前$(k-1)$ - block 与以 S_k 标记开始的下一行对应位置的$(k-1)$ - block 等价。

对于每块瓷砖 t_i,定义公式:

$$\varphi_{nxti} = G(\theta_=^{k-1} \wedge ((\neg \#_k)U(\#_k \wedge ((\#_{k-1} \wedge \theta_=^{k-1}) \rightarrow t_i)U\alpha(n,k+1) \mid\!\rightarrow \#_k))) \qquad (5)$$

再定义:$\varphi_{nextti} = \varphi_{nxti} \text{abort } \$_{k-1}$ (6)

这样,使得下一行具有同一$(k-1)$ - block 位置的瓷砖用 t_i 标记。

$$\varphi_{v_1} = G(\wedge_{t_i \in T}(t_i \rightarrow \vee_{t_j \mid (t_i,t_j) \in V} \varphi_{nexttj})) \qquad (7)$$

式(7)定义了 V 中垂直相邻瓷砖之间的关系。这个条件成立直至标记最末瓷砖行的命题 @ 出现,也即:

$$\varphi_{v_2} = G(\varphi_{v_1} U @) \qquad (8)$$

其中,式(2)、(3)、(6)、(7) 和(8) 分别参考文献[18]得出。至此,可以得到 $\varphi_k = \varphi_{a_1} \wedge \varphi_{a_2} \wedge \varphi_{h_1} \wedge \varphi_{v_2}$。文献[18] 还给出了 $\theta_=^k$ 的定义并指明其复杂度为 $O(4^k)$,上述超指数型铺砖问题实现的过程是 SPACE($\exp(k,n)$) - complete,由于 PSL 公式 φ_k 的"可满足性" 问题是超指数型铺砖问题计算模型的简化形式,由此得出 PSL 的可满足性问题的复杂度为 EXPSPACE - hard。

4 结束语

"可满足性"问题是研究时序逻辑的核心问题之一,并已成为程序验证的一种有力工具[15];而且理论证明许多难解的问题都可多项式变换为"可满足性"问题。研究可满足性问题的计算复杂度的意义在于它可作为研究一类问题的难度的标准。

文中在借助计算模型"铺砖问题"的基础上研究了 PSL 的线性时序逻辑部分 FL 的"可满足性"问题的复杂度,比文献[19]研究的 PSL 的线性时间逻辑增加了操作算子"abort",且与文献[19]比较,文中不局限于研究 FL 逻辑各个片断的复杂度,而是给出同时考虑 FL 所有操作子情况下表达"铺砖问题"的性质的逻辑公式,从而得出 FL 的可满足性问题的计算复杂度是 EXPSPACE - hard。

参考文献:

[1] Accellera Organization, Inc. Formal syntax and semantics of Accellera property specification language[EB/OL]. 2004. In Appendix B of http://www.eda.org/vfv/docs/PSL-v1.1.pdf.

[2] Accellea. Property specification language reference manual(version1.0.1)[EB/OL]. 2003. http://www.haifa.il.ibm.com/projects/verification/sugar.

[3] Beer I, Ben-David S, Eisner C, et al. The temporal logic Sugar[C]//In: Gerard B, Hubert C, Alain F. Proc. of 13th Int'l Conf. on Computer Aided Verification (CAV). LNCS 2102. Heidelberg: Springer-Verlag, 2001: 363-367.

[4] Accellea. Property specification language reference manual(version1.1)[EB/OL]. 2004. http://www.eda.org/vfv/docs/PSL-v1.1.pdf.

[5] Emerson E A, Clarke E M. Using branching-time temporal logic to synthesize synchronization skeletons[J]. Science of Computer Programming, 1982, 2(3): 241-266.

[6] van Emde Boas P. The convenience of tilings[M]//In Complexity, Logic and Recursion Theory, volume 187 of Lecture Notes in Pure and Applied Mathetaics. [s. l.]: [s. n.], 1997, 331-363.

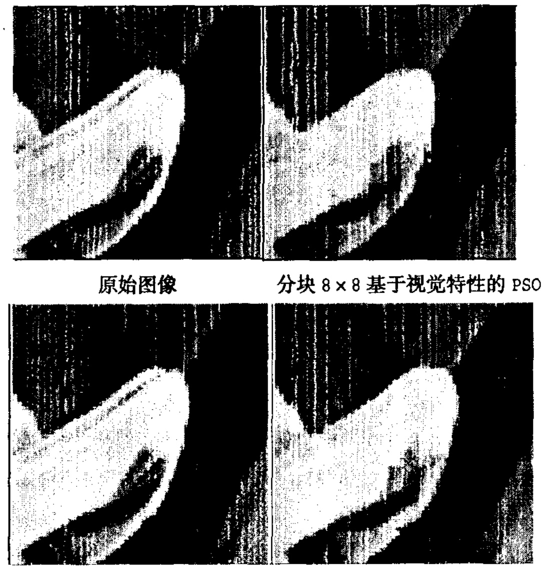
[7] Sistla A P, Clarke E M. The complexity of propositional linear temporal logics[J]. Journal of ACM, 1985, 32(3): 733-749.

[8] Chen C C, Lin I P. The computational complexity of satisfiability of temporal horn formulas in propositional linear-time temporal logics[J]. Information Processing Letters, 1993, 45(3): 131-136.

[9] Pemri S, Schnoebelen P H. The complexity of propositional linear temporal logics in simple cases[C]//In: Proc. of the STACS'98, LNCS 1373. [s. l.]: Springer-Verlag, 1998: 61-72.

[10] Bauland M, Schneider T. The complexity of generalized satisfiability for linear temporal logic[C]//In: Proc. of 10th International Conference, FOSSACS' 2007, LNCS 4423. [s. l.]: Springer-Verlag, 2007: 48-62.

(下转第 24 页)



原始图像 分块 8×8 基于视觉特性的 PSO

原始图像 分块 4×4 基于视觉特性的 PSO

图 3 边缘放大后图像质量对比(轻微的块效应)

3 结束语

通过对算法的模拟实验,所提出的基于视觉特性的粒子群分形块编码算法明显优于传统的分形块编码算法,其中认为基于视觉特性的 PSO(固定块)分形编码算法具有较高的性价比。其编码时间比全搜索算法加快超过 100 倍,比文献[5]中基于视觉特性的搜索方法加快 10 倍以上,且图像质量只有小幅度降低(0.6~0.9dB),且通过人眼的综合判断无明显失真。基于视觉特性的 PSO(固定块)分形编码算法是可行和高效的算法。

参考文献:

[1] 陈守吉,张立明．分形与图像压缩[M].上海:上海科技教育出版社,1998.

[2] 陈衍仪．图像压缩的分形理论和方法[M].北京:国防工业出版社,1997.

[3] Mitra S K, Murthy C A, Kundu M K. Technique for fractal image compression using genetic algorithm[J]. IEEE Transactions on Image Processing,1998(4):586-593.

[4] Fisher Y. Fractal Image Compression: Theory and Application [M].[s.l.]:Springer,1995.

[5] Jacquin A E. Image Coding Based on Fractal Theory of Iterated Contractive Image Transformations[J]. IEEE Trans. Image Proc,1992,1(1):10-30.

[6] Ramamurthi B, Gersho A. Classified Vector Quantization of Images[J]. IEEE Transactions on Communications,1986,34(11):1105-1115.

[7] Maurice C, Kennedy J. The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space [J]. IEEE Transaction on Evolutionary Computation,2002,6(1):58-73.

[8] 薛 猛,刘 兵.图像编码标准化的发展与现状[J].计算机技术与发展,2007,17(6):90-93.

(上接第 20 页)

[11] Lewis H. Satisfiability problems for propositional calculi[J]. Mathematical Systems Theory,1979,13:45-53.

[12] Bauland M, Hemaspaandra E, Schnoor H, et al. Generalized modal satisfiability [C]//In: Proc. of the STACS' 2006, LNCS3884. [s.l.]:Springer-Verlag,2006:500-511.

[13] Fischer M J, Ladner R E. Propositional modal logic of program [J]. Journal of Computer and System Science,1979,18:194-211.

[14] Vardi M Y, Stockmeyer L J. Improved upper and lower bounds for modal logics of programs[C]//In: Proc. of the 17th annual ACM symposium on Theory of computing. Rhode Island, USA: ACM,1985:240-251.

[15] Meier A, Mundhenk M. The complexity of satisfability for fragment of CTL and CTL * [C]//The 2nd Workshop on Reachability Problems in Computational Models. Amsterdam: Elsevier Science Pubilishers, Electronic Notes in Theoretical Computer Science,2008:201-213.

[16] Brazdil T, Forejt V, Kretinsky J, et al. The satisfiability problem for Probabilistic CTL[C]//In: Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science. Washington DC, USA: IEEE Computer Society Press,2008:391-402.

[17] Bonatti P A, Lutz C. The complexity of enriched mu-calculi [C]//In: Proc. of ICALP' 2006. LNCS 4052. [s.l.]: Springer-Verlag,2006:540-551.

[18] Armoni R, Bustan D. Resets vs. Aborts in Linear Temporal Logic[EB/OL]. 2003. http://www.cs.rice.edu/vardi/.

[19] Lange M. Linear time logics around PSL: Complexity, Expressiveness and a little bit of f Succinctness[C]//In Proc. of CONCUR 2007, LNCS 4703. [s.l.]: Springer-Verlag, 2007:90-104.

[20] Goller S. Computational complexity of propositional dynamic logics[EB/OL]. 2005. http://www.fmi.uni-leipzig.de/promotion/abstract.goeller.pdf.

[21] 堵丁柱,葛可一,王 杰.计算复杂性导论[M].北京:高等教育出版社,2008.

[22] 张立昂.可计算性与计算复杂性导引[M].第 2 版.北京:北京大学出版社,2004.