

基于状态转换图同构求解的等价性验证算法

卢英, 李炜

(安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

摘要:文中提出了一种基于状态转换图同构求解的时序电路等价性验证算法。算法将两时序电路的等价性问题转化为验证相应状态转换图的同构性。首先将初始状态对匹配为待验证对,然后采用递归的方法验证以初始状态对的下一状态对为初始状态的子状态转换图是否同构,从而得到时序电路是否等价的信息。若两状态转换图同构,则两图中的状态均可一一配对为待验证状态对,即所有的代验证状态对均为等价状态对。该方法可以有效地克服算法级描述到底层实现之间跨度太大的问题。

关键词:状态转换图;同构;时序电路;等价状态对

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2009)03-0074-03

Verifying Algorithm of Equivalence Based on State Transfer Graph's Isomorphism Solving

LU Ying, LI Wei

(Institute of Computer Science and Technology, Anhui University, Hefei 230039, China)

Abstract: Present a sequential circuit's equivalence verifying algorithm, which is based on the graph's isomorphism solving. Adopt recursive method to verify whether all the sub-STG(state transfer graph) is isomorphic, thus can obtain some information about sequential circuits' equivalence. If the state transfer graphs are isomorphism, all the states in different graphs can be matched as equal state pairs. The method based on STG can effectively overcomes the big span from algorithm-level description to low-level implementation.

Key words: state transfer graph; isomorphism; sequential circuit; equal state pair

0 引言

电路按逻辑功能可以划分为组合电路和时序电路,组合电路由门电路组成,不包含记忆单元,而时序电路一定包含存储记忆的单元电路。由于组合电路相对比较简单、验证方法和工具都已经很成熟,例如模拟验证、形式验证等^[1]。而时序电路相对复杂,而且研究方法比较少。目前对时序电路的验证是把时序电路转换为组合电路来验证的。但是此方法有很强的局限性。传统的遍历每个状态的方法^[2]虽然不受状态编码和结构相似性的限制,但是都存在一定的缺陷。SET方法部分地缓解了“内存爆炸”^[3]问题,但是它不事先对电路进行模型抽象,所以它所能验证的内容有限,并且不能对活性进行验证。基于BDD^[2,4]的验证方法

仅限于25k门以下的设计规模,而且速度较慢。所以提出一种基于状态转换图的时序电路等价性验证算法。该算法也是采用遍历每个状态的方法,不同的是通过验证两状态转换图同构的方法来验证两时序电路的等价性^[5]。算法将状态转换图存储为待验证状态对的形式,并为待验证状态对加上唯一的特征值。该算法既不依赖于组合电路的相似性,也不依赖于时序电路部分寄存器的匹配。结果证明该算法具有速度快、验证规模大等特点。

1 时序电路的等价性证明方法

时序电路^[3]就是一种有限状态机。有限状态机是一个抽象的模型,可以用它来描述系统的输入输出等操作特性。有限状态机有Mealy和Moore两种形式,采用Mealy型状态机来描述时序电路。Mealy型状态机描述如下:

$$M = (S, \Sigma', Y, \delta, S_0)$$

S : 状态 S_0, S_1, \dots, S_n 的有限非空集。

Σ' : 输入符号的有限非空集。

收稿日期:2008-06-04

基金项目:安徽省高等学校省级自然科学基金项目资助(2006KJ013A)

作者简介:卢英(1984-),女,安徽宿州人,硕士研究生,研究方向为计算机与VLSI设计自动化;李炜,副教授,硕士生导师,研究方向为嵌入式系统和CIMS技术。

Y : 输出字母表。

δ : 状态转换函数, 表示在当前输入和当前状态到下一状态和输出的映射, 是个多输出的函数, 即 $S \times \Sigma' \rightarrow (S, Y)$ 。

S_0 : 初始状态。

算法通过验证两时序电路的状态转换图是否同构, 得到两电路的等价信息。首先从初始顶点开始先将两个状态转换图的初始状态匹配为待验证状态对, 然后递归地验证每一个初始状态所形成的可达空间, 即以当前初始状态的下一状态为初始状态的所有子状态转换图。如果以此递归的算法验证了两状态转换图的所有状态, 则两时序电路等价; 如果在验证过程中发现有待验证状态对的下一状态对不能匹配成待验证状态对, 则两时序电路不等价。

2 基于状态转换图的验证算法

该算法是在两状态转换图的初始顶点已知, 转换关系函数完全确定的前提下进行的, 目的是让算法更高效些。

2.1 基本思想

此算法只适应于已给出初始状态的时序电路。已知电路的初始状态, 时序电路可用初始状态以及在该初始状态的可达状态空间组成的子状态转换图来表示。验证两时序电路对应的状态转换图是否同构等同于验证两时序电路是否等价^[6]。如果在一个状态转换图中任取一个顶点都可以在另一个状态转换图中找到等价的顶点, 那么称两状态转换图同构, 也就是说这两个状态转换图代表的时序电路等价。

算法首先将两状态转换图的初始状态集合中的初始状态匹配为待验证状态对, 然后递归地验证子状态转换图的初始状态是否也是待验证状态对。如果在验证的过程中出现多个状态和同一个状态匹配, 则先验证这多个状态是否是同一个状态, 若是, 只保留其中的一个状态而将其余的从图中删除, 然后再将其与另一个图中的状态匹配为待验证状态对。其中的 S_1, S_2 为待验证状态转换图的初始状态集合。

(1) 首先匹配初始状态 S_1, S_2 , 形成待验证状态对并加上唯一的标志。每验证完一个状态对之后都把相应的内存释放掉, 只需要继续保留其唯一的标志, 以防止后面出现相同的待验证状态对。

(2) 设 $S_1 = S_1 \rightarrow \text{NEXT}, S_2 = S_2 \rightarrow \text{NEXT}$ (NEXT 表示当前的下一状态), 验证以 S_1, S_2 为初始

顶点的子状态转换图是否同构。首先判断 S_1, S_2 是否已匹配为待验证状态对, 是转向步骤(3); 否则判断逻辑表达式 $S_1 = S_2$ 是否成立 (其中的 $=$ 表示 S_1, S_2 匹配为待验证状态对), 成立转向步骤(2), 否则转向步骤(4)。

(3) 结束, 两状态转换图表示的时序电路等价。

(4) 结束, 两时序电路不等价。

定义1 待验证状态对是等价状态对的条件是它的前一状态对和后一状态对均为待验证状态对。

根据定义1, 有以下的结论: 两电路等价的充分必要条件是它们所对应状态转换图的所有待验证状态对都是等价状态对。

图1和图2分别是时序电路1和时序电路2的状态转化图, 图中的顶点代表状态, 顶点之间的箭头表示状态之间的转换关系, 而转换关系的条件在箭头被标注出来, 格式为输入/输出, 其中的“-”表示取任意值。为了区别两图的顶点, 它们采用不同的编号。 S_{00} 和 S_{10} 是两状态转换图的初始顶点。

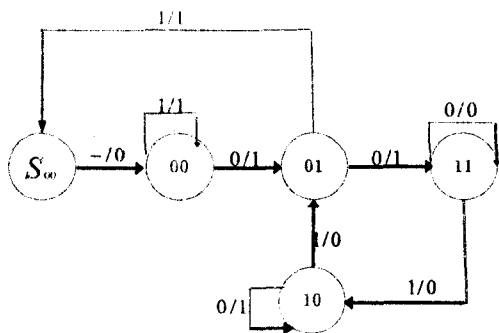


图1 时序电路1的状态转换图

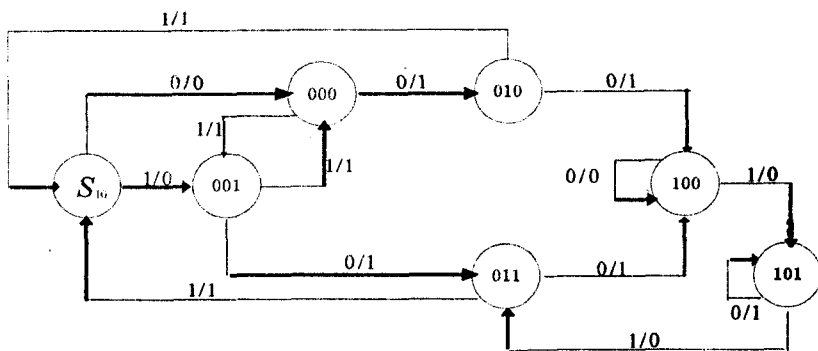


图2 时序电路2的状态转换图

验证的具体过程如下:

首先将两状态转换图的初始顶点 S_{00} 和 S_{10} 匹配为待验证状态对, S_{00} 下一状态为 00, S_{10} 的下一状态为 000、001, 其中 00 和两个状态 000、001 匹配。先验证状态 000 和 001 是否为同一个状态, 它们在相同的输入 1 下输出均为 1, 且互为下一状态, 在输入 0 下输出均为 1, 下一状态分别为 010 和 011, 而它们的上一状态均为

S_{10} , 只需要验证状态 010 和 011 是否是同一个状态。状态 010 和 011 在输入 1 下输出均为 1, 下一状态均为 S_{10} ; 在输入 0 下输出均为 1, 下一状态都为 100。由此可见 010 和 011 是同一状态, 可知状态 000 和 001 是同一状态。化简图 2 所示的状态转换图: 即删除图中的状态 001、011, 删除后的状态转换图如图 3 所示。状态 00 与 000 构成待验证状态对。00 的下一状态 01 与 000 的下一状态 010 匹配, 它们在输入 0 下输出均为 1, 在输入 1 下输出均为 1, 所以 01 与 010 为待验证状态对。状态 01 和 010 在输入 1 下输出均为 1, 且都回到初始状态, 在输入 0 下输出均为 1, 下一状态分别为 11 和 100。11 和 100 匹配, 它们在输入 0 下输出均为 0, 在输入 1 下输出均为 0, 所以 11 和 100 为待验证状态对。它们的下一状态 10 和 101 匹配, 它们在输入 1 下输出输出均为 0, 输入 0 下输出均为 1, 它们的下一状态分别为 01 和 010, 而 01 和 010 已匹配为待验证对。至此, 状态转换图中的所有状态已验证完毕, 每个待验证状态对的上一状态对和下一状态对均为待验证状态对, 根据定义 1 可得所有的待验证状态对都是等价状态对, 所以两状态转换图表水的电路等价。

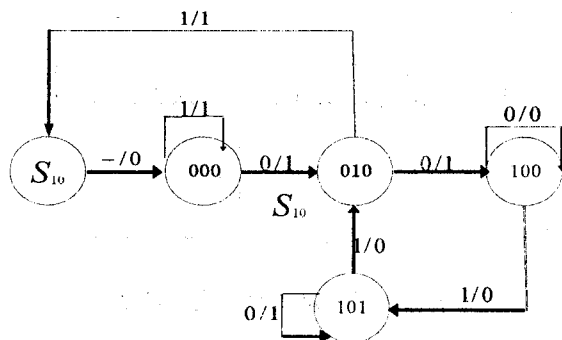


图 3 化简后的状态转换图

2.2 算法实现

文中的算法是在初始顶点已知, 并且初始顶点唯一的情况下进行的。图中顶点的结构如下面的结构体所描述。

```
typedef struct node {
    int currentstate; // 状态标志
    int out[2]; // 分别代表在输入 0、1 下的输出
    bool flag; // 为待验证对的标记
    node * nextstate[2]; // 分别指向在输入 0、1 下的下一状态
};

bool STG-equal(node * a, node * b)
{
    // 验证以 a, b 指向的顶点为初始顶点的状态转换图是否同构
    flag(a, b) = 1; // a, b 指向的顶点匹配为待验证状态对的标记
    a->flag = TRUE;
```

```
    b->flag = TRUE;
    node * p1, * p2, * q1, * q2;
    p1 = a->nextstate[0]; p2 = a->nextstate[1];
    q1 = b->nextstate[0]; q2 = b->nextstate[1];
    bool n1, n2, n3;
    n1 = flag(p1, q1) && ! p2->flag && ! q2->flag; // 一对没有配对, 另一对彼此配对
    n2 = ! p1->flag && ! q1->flag && flag(p2, q2);
    n3 = ! (p1->flag || p2->flag || q1->flag || q2->flag); // 下一状态均没有匹配
    if ((a->out[0] == b->out[0]) && (a->out[1] == b->out[1])) // 如果输出相等
    {
        if (flag(p1, q1) && flag(p2, q2)) // 下一状态均互配为待验证状态对
            printf("两时序电路等价");
        else if (n1 || n2 || n3) // 至少有一状态对没被配对, 而其它的彼此配对
            VerifyEquivalence(p1, q1, p2, q2); // 匹配下一状态
        else printf("两时序电路不等价");
    }
    else printf("两时序电路不等价"); // 初始状态对的输出不相等
}

VerifyEquivalence(node * s1, node * s2, node * t1, node * t2) // 分别匹配 s1, t1 和 s1, t2 指向的顶点
{
    if (flag(s1, t1) && ! s2->flag && ! t2->flag) STG-equal(s1, t1);
    else if (! s1->flag && ! t1->flag && flag(s2, t2)) STG-equal(s2, t2);
    else if (! (s1->flag || s2->flag || t1->flag || t2->flag))
    {
        if (出现一个状态 a 与两个状态 b, c 匹配)
        {
            if (equal-state(b, c)) STG-equal(a, b);
            else printf("两时序电路不等价");
        }
        else { STG-equal(s1, t1); STG-equal(s2, t2); }
    }
}
else;
```

函数 `equal-state(node * s, node * t)` 的功能是验证 `s` 和 `t` 指向的状态是否是同一个状态。

3 结束语

文中提出一种基于 STG(状态转换图)的验证算法, 该算法采用递归的方法验证两状态转换图是否同构。首先将初始顶点匹配为待验证状态, 然后再分别验证以当前初始状态的下一状态为初始顶点的子状态转换图。实践表明该方法可以有效地克服算法级描述

(下转第 83 页)

增加了复杂度等一些不完善的问题,还需日后进一步的深入研究。

参考文献:

- [1] 沈波,张世永,钟亦平.无线传感器网络分簇路由协议[J].软件学报,2006,17(7):1588-1600.
- [2] 金骥,徐昌庆,葛颖君.无线传感器网络基于类的 LEACH 路由算法研究[J].计算机应用与软件,2006,23(11):137-138.
- [3] 李秉智,赵娜.基于无线传感器网络的 LEACH 算法的改进[J].传感器与仪器仪表,2006,23(8):153-155.
- [4] 吴臻,金心宇.无限传感器网络的 LEACH 算法的改进[J].传感技术学报,2006,19(1):34-36.
- [5] Kim J M, Joo H K, Hong S S. An efficient clustering scheme through estimate in centralized hierarchical routing protocol [C]//International Conference on Hybrid Information Technology. Washington, DC, USA: IEEE Computer Society, 2006:145-152.
- [6] Chang R S, Kuo C J. An energy efficient routing mechanism for wireless sensor networks[C]//Proceedings of the 20th International Conference on Advanced Information Networking and Applications. Washington, DC, USA: IEEE Computer Society, 2006:308-312.
- [7] 李利民,李建中,陈渝.无限传感器网络[M].北京:清华大学出版社,2005:95-96.
- [8] Heinzelman W. Application-Specific protocol architectures for wireless networks [D]. Boston: Massachusetts Institute of Technology, 2000.
- [9] Heinzelman W R, Chandrakasan A, Balakrishnan H. An Application-Specific Protocol Architecture for Wireless Microsensor Networks[J]. IEEE Transactions on Wireless Communications, 2002,1(4):660-670.
- [10] Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks[C]//In: the proceedings of the Hawaii International Conference on System Sciences. Maui, Hawaii: [s. n.], 2000.

(上接第 76 页)

到底层实现之间跨度太大的问题。

参考文献:

- [1] Gupta A. Formal Hardware Verification Methods: A Survey [J]. Formal Methods in System Design, 1992(1):151-238.
- [2] Samman G A, Schmaltz J, Toma D, et al. TheoSim: combining symbolic simulation and theorem proving for hardware verification[C]//In: Proc of the Integrated Circuits and Systems Design. [s. l.]: [s. n.], 2004:60-65.
- [3] 边计年,薛宏熙,苏明,等.数字系统设计自动化[M].北京:清华大学出版社,2005.
- [4] Touati H J, Savoj H, Lin B, et al. Implicit state enumeration of finite state machines using BDDs[C]//Proceedings of International Conference on Computer-Aided Design. Santa Clara: [s. n.], 1990:130-133.
- [5] Huang S Y, Cheng K T, Chen K C, et al. Aquila: An Equivalence Checking System for Large Sequential Designs[J]. IEEE Transactions on Computers, 2000,49(5):443-464.
- [6] 方敏,张雅顺,李辉.混合系统的形式验证方法[J].系统仿真学报,2006,18(10):2921-2928.

(上接第 79 页)

的速度提高很多。图 3 给出了网络用于识别字符图像的模块大小分别为 2×2 、 4×4 、 8×8 、 32×32 时,图像噪声为 30% 的情况下网络的收敛性。通过比较可以知道,新网络通过 2 次迭代就可以达到传统网络 10 次迭代的效果。

4 结束语

提出了一种利用距离学习算法来减少 Hopfield 网络连接权值的新方法。

仿真结果证实了数据分析的结论,与传统 Hopfield 网络相比,新方法在复杂度、速度和学习能力上都有改善。距离因子的属性使得它可以滤除一些远处神经元的连接权值,这就使网络无论在软件还是硬件的执行过程中减少了复杂性。

参考文献:

- [1] de Sa Marques J P. 模式识别——原理、方法及应用[M]. 吴逸飞译. 北京:清华大学出版社,2002.
- [2] Kuhlmann P, Garces R, Eissfeller H. A dilution algorithm for neural networks[J]. Journal of Physics, 1992(25):593-598.
- [3] Gazzaniga M S. Organization of the human brain[J]. Science, 1989(245):947-952.
- [4] 韩力群. 人工神经网络理论、设计及应用[M]. 北京:化学工业出版社,2002.
- [5] Hopfield J J. Neural networks and physical systems with emergent collective computational abilities[J]. Proceedings of the National Academy of Science, 1982(79):2554-2558.
- [6] Hebb D O. The organization of behavior: a neuropsychological theory[M]. New York: Wiley, 1949.
- [7] 孙兆林. Matlab 6. x 图像处理[M]. 北京:清华大学出版社,2002.