

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2016 年 1 月 14 日 (14.01.2016)



(10) 国际公布号
WO 2016/004808 A1

- (51) 国际专利分类号:
G06F 11/36 (2006.01)
- (21) 国际申请号: PCT/CN2015/081062
- (22) 国际申请日: 2015 年 6 月 9 日 (09.06.2015)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201410320943.2 2014 年 7 月 7 日 (07.07.2014) CN
- (71) 申请人: 西安交通大学 (XI'AN JIAOTONG UNIVERSITY) [CN/CN]; 中国陕西省西安市碑林区咸宁西路 28 号, Shaanxi 710049 (CN)。
- (72) 发明人: 刘焱 (LIU, Ting); 中国陕西省西安市碑林区咸宁西路 28 号, Shaanxi 710049 (CN)。 张晓东 (ZHANG, Xiaodong); 中国陕西省西安市碑林区咸宁西路 28 号, Shaanxi 710049 (CN)。 俞乐晨 (YU, Lechen); 中国陕西省西安市碑林区咸宁西路 28 号, Shaanxi 710049 (CN)。 刘沛 (LIU, Pei); 中国陕西省

西安市碑林区咸宁西路 28 号, Shaanxi 710049 (CN)。 郑庆华 (ZHENG, Qinghua); 中国陕西省西安市碑林区咸宁西路 28 号, Shaanxi 710049 (CN)。

- (74) 代理人: 西安智大知识产权代理事务所 (XI'AN CYDA INTELLECTUAL PROPERTY AGENCY); 中国陕西省西安市雁塔区科技路 195 号世纪颐园 A707, Shaanxi 710075 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA,

[见续页]

(54) Title: METHOD FOR DATA CONTENTION TESTING AND PROOF-GENERATION, BASED ON MULTITHREADED PROGRAM CONSTRAINT CONSTRUCTION

(54) 发明名称: 基于多线程程序约束构建的数据竞争检测与证据生成方法

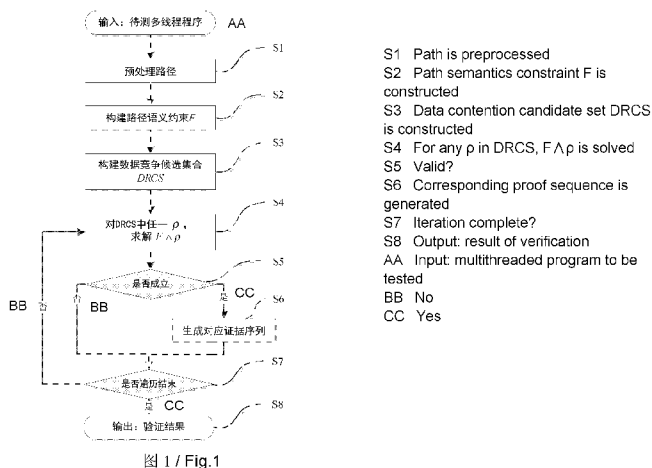


图 1 / Fig.1

(57) Abstract: Provided is a method for data contention testing and proof-generation, based on multithreaded program constraint construction; according to multithreaded program semantics, a constraint expression is constructed; a data contention testing problem is converted to a constraint solving problem; a constraint solver is used to detect data contention that could exist, and generate a program execution path triggering data contention; first, a tested program is stubbed, and the program is executed to obtain an execution path; next, according to the multithreaded program execution semantics, the execution path is converted to a first-order logic expression having no quantifiers, the constraint expression encompassing all possible thread interleavings; then, according to the timing relationship between statements when data contention occurs, a data contention candidate set is constructed and conditions for contention of candidates are generated; lastly, the candidate set is iterated to determine whether data contention exists; if so, then a corresponding proof sequence is generated. The present method identifies all data contention in a primary execution, and presents no false positives; a proof sequence showing the data contention triggering process is generated for each data contention.

(57) 摘要:

[见续页]

RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

根据细则 4.17 的声明:

- 关于发明人身份(细则 4.17(i))

- 关于申请人有权申请并被授予专利(细则 4.17(ii))
- 关于申请人有权要求在先申请的优先权(细则 4.17(iii))
- 发明人资格(细则 4.17(iv))

本国际公布:

- 包括国际检索报告(条约第 21 条(3))。

一种基于多线程程序约束构建的数据竞争检测与证据生成方法, 根据多线程程序语义构建约束表达式, 将数据竞争检测问题转化为约束求解问题, 采用约束求解器检测可能存在的数据竞争, 并生成触发数据竞争的程序执行路径, 首先插桩被检测程序, 执行此程序并得到执行路径; 其次根据多线程程序执行语义将执行路径转化为无量词一阶逻辑表达式, 此约束表达式涵盖所有可行的线程交织; 然后根据发生数据竞争时语句间时序关系构建数据竞争候选集合, 生成候选发生竞争的条件; 最后遍历候选集合判定是否存在数据竞争, 如有则生成对应的证据序列, 本方法可找出一执行中所有的数据竞争且不存在误报情况, 对每一个数据竞争都生成一个展示了数据竞争触发过程的证据序列。

基于多线程程序约束构建的数据竞争检测与证据生成方法

技术领域

本发明涉及可信软件及软件测试领域，特别涉及一种基于多线程程序约束构建的数据竞争检测与证据生成方法。

背景技术

随着处理器多核化的普及，多线程技术已经成为软件编程中提高 CPU 利用率不可或缺的技术。然而，由于线程之间交织的不确定性，多线程程序执行过程中可能会出现一些难以预料的行为导致程序出错，例如对临界区没有做好同步工作而导致的数据竞争问题。数据竞争是两个不同的线程在没有同步保护的情况同时访问一个内存，并且至少有一个写操作。数据竞争不一定导致程序错误，因为有些程序员故意让程序有数据竞争以提高运行的效率，但是有调查表明 5-24% 的数据竞争会对程序产生坏影响。数据竞争很难以被发现，因为它们经常发生在一些低概率出现的交织序列中，在现实中往往需要花很多时间去定位，其引起的错误如同“corner error”，即使在软件发布时也未必能够完全清除它们。因此，数据竞争检测是多线程程序测试领域最受关注的研究点之一。

过去几十年中数据竞争检测已有大量研究，设计出很多杰出的自动化检测工具，主要分为静态与动态分析技术。静态方法通过静态检测程序所有的路径来推断程序中的所有数据竞争，可以检测出大部分数据竞争；但由于使用大量假设，静态分析方法会产生无效的数据竞争，导致误报率较高。动态方法通过监控一次执行中内存与同步信息以确定是否存在数据竞争，能够提供较高精度的检测结果；但是动态分析方法受到交织与路径的影响，往往要通过多次执行来提高覆盖率。本文将静态代码分析与程序执行过程监测相结合，以提高覆盖率且尽可能消除误报。

现有的动态检测技术主要分为三种：基于 lockset、基于 happens-before 与二者结合的方法。1) 基于 lockset 的方法对线程交织不敏感，但是存在误报情况，即无效竞争。2) 基于 happens-before 的方法只检测某特定交织序列上的数据竞争，检测结果虽可靠，但敏感于线程交织。3) 混合方法结合了两者的优点，并且试图减小各自的缺点，但也面临如不能够搜索出隐藏的错误、lockset 高误报引起的无效报警等问题。

发明内容

为了克服上述现有技术的缺点，本发明的目的在于提供一种基于多线程程序约束构建的数据竞争检测与证据生成方法，根据多线程程序语义构建约束表达式，将数据竞争检测问题转化为约束求解问题，采用约束求解器检测可能存在的数据竞争，并生成触发数据竞争的程序执行路径。

为了实现上述目的，本发明采用的技术方案是：

一种基于多线程程序约束构建的数据竞争检测与证据生成方法，包括如下步骤：

S1) 在给定输入下，通过执行已插桩的待测程序以生成路径记录文件，且识别出执行路径中公有变量的访问点以便于约束构建；

S2) 根据程序执行语义将执行路径中状态转移、线程交织关系转化为无量词一阶逻辑表达式，构建蕴含了所有可能的交织序列的多线程程序执行路径约束模型 F ；

S3) 将路径中所有线程上可能发生数据竞争的两点视为数据竞争候选，收集所有候选并构建数据竞争候选集合 $DRCS$ ，同时根据数据竞争的定义构建每个候选的竞争发生条件 ρ ；

S4) 针对每一个候选竞争发生条件 ρ ，利用约束求解器验证 $F \wedge \rho$ 是否有解；

S5) 如果有解，则表示此竞争条件会触发真实的数据竞争；如果无解，则表示此候选不会触发数据竞争；

S6) 当存在数据竞争时，输出该数据竞争的证据序列；

S7) 对于数据竞争候选集合 $DRCS$ ，如果遍历结束，则输出所有结果；否则，

继续遍历下一个竞争候选；

S8) 验证结束后，输出检测到的所有数据竞争以及对应的证据序列。

本发明进一步的改进在于：所述步骤 S1) 中插桩工作并非在源码或者二进制的层面上进行，而是在字节码的层面上完成，具体实施方法为：首先将待测多线程程序源码转化为中间字节码格式，即 LLVM 字节码；然后将具有监控功能的语句植入待测程序；最后将植入监控代码的字节码链接成可执行程序。

本发明进一步的改进在于：所述步骤 S2) 中多线程程序执行路径约束模型 F 蕴含了执行路径所有可能的交织序列，包括五种约束：路径表达式、内存模型约束、读写关系约束、偏序约束以及同步语义约束，定义分别如下：

1) 路径表达式：描述线程内部的定义-使用链，以及控制线程内部状态转换；

2) 内存模型约束：表示程序中语句、变量之间的关系，采用顺序一致性的语义，顺序一致性规定 CPU 按照代码中语句的顺序来执行程序；

3) 读写关系约束：定义线程间的定义-使用链，规定共享变量所读取到的值，必须来自初始值以及最近的写值；

4) 偏序约束：定义线程之间创建线程与终止线程操作语句于被操作线程语句之间的时序关系；

5) 同步语义约束：定义线程之间同步控制操作语句之间的时序关系；

其中，定义-使用链为：将每一个线程序列转化为 SSA 格式，对于每一个 SSA 格式的序列，除去共享访问点都是一个完整的定义-使用链。

本发明进一步的改进在于：所述步骤 S2) 中多线程程序执行路径约束模型 F 的构建方法包括以下操作：

1) 计算路径表达式，以控制线程内部状态转移；

2) 计算内存模型约束，以线程内限制语句之间的关系；

3) 计算读写关系约束，以建立线程间的定义-使用链；

4) 计算同步语义约束，以定义线程间同步关系；

5) 计算偏序约束, 以描述线程创建与终止的语义;

最后, 结合以上五种约束, 构成约束模型 F 。

本发明进一步的改进在于: 定义执行路径事件集合 $\mathbb{E}_\pi = \{T_i | 0 < i \leq k\}$, 其中 k 为线程数量, $T_i = \{e_1, e_2, \dots, e_n\}$ 作为线程 i 的执行序列, e_n 表示 T_i 的第 n 个事件, $O(e_n)$ 表示事件 e_n 的顺序, n 表示 T_i 的事件数量, 则:

所述路径表达式的计算方法:

将每一个线程序列转化为 SSA 格式, 类似于路径条件 (Path Condition) 的收集, 直接将 SSA 格式序列转化为路径表达式;

所述内存模型约束的计算方法:

采用顺序一致性模型, 所有操作完全按程序的顺序执行, 线程内的事件顺序符合约束:

$$\Phi_{mo} = \bigwedge_{T_i \in \tau} \left(\bigwedge_{e_i, e_{i+1} \in T_i} O(e_i) < O(e_{i+1}) \right)$$

其中 e_i 与 e_{i+1} 表示同一线程内连续的两个事件, τ 表示所有线程序列;

所述读写关系约束的计算方法:

使共享变量的读来自于最近的写, 对于同一共享变量 v , 令 R 作为所有对其进行读操作的事件集合, 令 W 作为所有对其进行写操作的事件集合, 给出以下公式:

$$\Phi_{rw} = \bigwedge_{e_r \in R} \bigvee_{e_w \in W} \{ (v_r = v_w \wedge O(e_w) < O(e_r)) \wedge \bigwedge_{e_x \in W} (O(e_x) < O(e_w) \wedge O(e_r) < O(e_x)) \}$$

其中, e_r 为读事件, e_w 与 e_x 为写事件, v_r 和 v_w 为事件 e_r 与 e_w 所操作的变量, 公式所表达的意思是, 如果事件 e_r 中的 v_r 取值来自于事件 e_w 中的 v_w , 首先要满足 e_r 在 e_w 之后, 即 $O(e_w) < O(e_r)$; 然后要满足所有的写要么在 e_w 之前, 要么在 e_r 之后;

所述同步语义约束的计算方法包括 lock/unlock 与 wait/signal 两类操作:

1) lock/unlock 操作的目的是构建锁同步语义约束, 要求在同一互斥锁的 lock/unlock 集合 L 中, 对于任意两个 lock/unlock 事件对: l_i/u_i 与 l_k/u_k , 须满

足公式：

$$\Phi_{syn}^{l/u} = \bigwedge_{l_i/u_i, l_k/u_k \in L} O(u_i) < O(l_k) \vee O(u_k) < O(l_i)$$

其中，锁对 l_i/u_i 要么发生在锁对 l_k/u_k 之前，要么发生在其后；

2) wait/signal 操作的目的是构建条件变量同步语义约束，要满足条件：每一个 wait 操作必须对应一个 signal 操作，而一个 signal 操作至多唤醒一个 wait 操作，对于同一条件变量 cond，令 WT 作为在 cond 上所有 wait 操作的集合，令 SG 作为在 cond 上所有 signal 操作的集合，如要满足之上的条件，须有以下公式：

$$\Phi_{syn}^{w/s} = \bigwedge_{e_{wt} \in WT} \{ (\bigvee_{e_{sg} \in SG_{wt}} O(e_{wt}) > O(e_{sg}) \wedge pair_{wt}^{sg} = 1) \wedge \sum_{e \in SG_{wt}} pair_{wt}^e = 1 \}$$

其中， e_{wt} 为 WT 中的任一元素， SG_{wt} 表示 e_{wt} 可以匹配的 signal 操作的集合， e_{sg} 为 SG_{wt} 中任一 signal 操作事件，利用变量 $pair_{wt}^{sg}$ 是否等于 1 来表示 e_{sg} 是否与 e_{wt} 相匹配。子公式 $\sum_{e \in SG_{wt}} pair_{wt}^e = 1$ 表示，对于每一个 wait 操作 e_{wt} 必须有一个 signal 操作与之匹配；

所述偏序约束的计算方法：

首先规定：如果事件创建一个线程，那么被创建线程的所有事件都要在此事件之后执行；如果事件执行线程终止操作，那么被终止线程的所有事件都要在此事件之前；令 C 为 create/fork 操作的事件集合，令 J 作为 join 操作的事件集合；给定约束：

$$\Phi_{po} = \bigwedge_{e_c \in C} O(e_c) < first(e_c) \wedge \bigwedge_{e_j \in J} O(e_j) > last(e_j)$$

其中， e_c 为线程创建事件， $first(e_c)$ 为 e_c 所创建的线程首个事件的顺序； e_j 为线程终止事件； $last(e_j)$ 为 e_j 所结束的线程末尾事件的顺序；

最终将以上五种约束相与构成约束模型 F 。

本发明进一步的改进在于：所述步骤 S3) 中竞争发生条件 ρ 的构建方法如下：如果有一条路径 $\tau = \langle \tau_1 e_i e_j \tau_2 \rangle$ ，其中 τ_1 是前缀， τ_2 是后缀，事件 e_i 与

e_j 属于不同的线程并且都访问同一内存，至少有一个写，那么二者之间发生了数据竞争，而 τ 看作是 e_i 与 e_j 数据竞争的证据序列；对于访问同一变量的事件 e_i 与 e_k ， e'_i 与 e''_i 分别表示 e_i 的前一个事件与后一个事件；同样， e'_k 与 e''_k 分别表示 e_k 的前一个事件与后一个事件，那么两者发生数据竞争即同时访问同一内存的条件 ρ 为：

$$O(e'_i) < O(e_k) < O(e''_i) \wedge O(e'_k) < O(e_i) < O(e''_k)。$$

本发明进一步的改进在于：所述步骤 S7) 中对于每一个数据竞争都会生成证据序列，以描述其触发过程。

与现有技术相比，本发明的有益效果是：

(1) 提出一种多线程程序约束构建模型，将一次执行中的数据竞争检测问题转化为约束求解问题。此模型按照程序语义进行约束构建，所构建的表达式包含了所有可能的交织序列，进而检测出执行路径中的所有数据竞争。

(2) 对所有数据竞争都产生一个证据序列，以给用户提数据竞争是如何被触发的信息。

(3) 对执行序列进行事后分析，不存在 on-the-fly 技术所产生的巨大运行时开销。

附图说明

图 1 为本发明方法整体流程图。

图 2 为多线程程序路径约束构建方法流程图。

具体实施方式

以下结合附图和实例详细说明本发明的实施方式。

待测程序如下所示，x 与 y 为共享变量，线程 0 创建了线程 1 与线程 2。

共享变量 x, y

线程 0	线程 1	线程 2
1: x=0;	4: lock(m);	8: lock(m);
2: create(1);	5: x = a + b;	9: x ++;
3: create(2);	6: unlock(m);	10: y ++;
	7: y++;	11: unlock(m);

如图 1 所示，本发明数据竞争检测与证据生成方法，包括如下步骤：

步骤 S1)：将监控代码植入被测程序，以对程序的执行过程进行记录。给定输入下，执行示例程序，记录下路径 $\pi=[1,2,3,4,5,6,7,8,9,10,11]$ ；然后，识别出共享变量访问点，包括 $\{1,5,7,9,10\}$ 。

步骤 S2)：根据程序执行语义将执行路径中状态转移、线程交织关系转化为无量词一阶逻辑表达式，构建执行路径 π 的约束模型 F ，包括路径表达式、内存模型约束、读写关系约束、偏序约束、同步语义约束。整个约束模型 F 蕴含了执行路径所有可能的交织序列。具体地，如图 2 所示，按照以下步骤生成对应的逻辑表达式：

S201) 首先，根据识别出的共享访问点，将路径 π 转化为 SSA 格式，如下所示：

initialization : $x_w^0 = 0, y_w^0 = 0$

1: $x_w^1 = 0$;
 2: create(1);
 3: create(2);
 4: lock(m);
 5: $x_w^2 = a + b$;
 6: unlock(m);
 7: $y_w^1 = y_r^1 + 1$;
 8: lock(m);
 9: $x_w^3 = x_r^0 + 1$;
 10: $y_w^2 = y_r^2 + 1$;
 11: unlock(m);

其中，对于全局变量 x 与 y 的下角标表示读（r）或写（w），上角标区分不同的读或写操作，上角标为 0 表示为初始赋值。

然后，根据路径的 SSA 格式，直接计算出路径 π 的路径表达式，如下公式：

$$\begin{aligned} x_w^0 &= 0 \wedge y_w^0 = 0 \wedge \\ x_w^1 &= 0 \wedge x_w^2 = a + b \wedge y_w^1 = y_r^1 + 1 \wedge x_w^3 = x_r^0 + 1 \wedge y_w^2 = y_r^2 + 1 \end{aligned}$$

S202) 构建内存模型约束，采用顺序一致性模型，规定所有操作按程序的顺序执行。按照公式：

$$\Phi_{mo} = \bigwedge_{T_i \in \tau} \left(\bigwedge_{e_i, e_{i+1} \in T_i} O(e_i) < O(e_{i+1}) \right)$$

计算出路径 π 的内存模型约束，如以下公式：

$$\begin{aligned} &O(e_1) < O(e_2) < O(e_3) \wedge \\ &O(e_4) < O(e_5) < O(e_6) < O(e_7) \wedge \\ &O(e_8) < O(e_9) < O(e_{10}) < O(e_{11}) \end{aligned}$$

其中， o_i 表示第 i 行语句交织序列中的排列序号。

S203) 计算读写顺序约束，使共享变量的读来自于最近的写。对于同一共享变量 v ，令 R 作为所有对其进行读操作的事件集合，令 W 作为所有对其进行写操作的事件集合。给出以下公式：

$$\Phi_{rw} = \bigwedge_{e_r \in R} \bigvee_{e_w \in W} \{ (v_r = v_w \wedge O(e_w) < O(e_r)) \wedge \bigwedge_{e_x \in W} (O(e_x) < O(e_w) \wedge O(e_r) < O(e_x)) \}$$

其中， e_r 为读事件， e_w 与 e_x 为写事件。公式所表达的意思是，如果事件 e_r 中的 v_r 取值来自于事件 e_w 中的 v_w ，首先要满足 e_r 在 e_w 之后，即 $O(e_w) < O(e_r)$ ；然后要满足所有的写要么在 e_w 之前，要么在 e_r 之后。

在路径 π 中，对于全局变量 x ， $R = \{e_9\}$ ， $W = \{e_0, e_1, e_5, e_9\}$ ，其读写关系表达式如下公式：

$$\begin{aligned} &\{x_r^0 = x_w^1 \wedge O(e_1) < O(e_9) \wedge (O(e_1) > O(e_5) \vee O(e_5) > O(e_9))\} \vee \\ &\{x_r^0 = x_w^2 \wedge O(e_5) < O(e_9) \wedge (O(e_5) > O(e_1) \vee O(e_1) > O(e_9))\} \end{aligned}$$

其中，对变量 x 的读写可能进行了罗列，当第 9 行 x 的读来自于第 1 行 x 的写时，应该满足：第 1 行在第 9 行之前，且第 5 行对 x 的写不能发生在两者之间。 y 变量的情况类似于 x 。

S204) 计算同步语义约束，包括 lock/unlock 与 wait/signal 两类操作：

1) 构建锁同步语义约束 (lock/unlock 操作) 时，要求在同一互斥锁的 lock/unlock 集合 L 中，对于任意两个 lock/unlock 事件对： l_i/u_i 与 l_k/u_k ，须满足公式：

$$\Phi_{syn}^{l/u} = \bigwedge_{l_i/u_i, l_k/u_k \in L} O(u_i) < O(l_k) \vee O(u_k) < O(l_i)$$

其中，锁对 l_i/u_i 要么发生在锁对 l_k/u_k 之前，要么发生在其后。

2) 构建条件变量同步语义约束 (wait/signal) 时，要满足条件：每一个 wait 操作必须对应一个 signal 操作，而一个 signal 操作至多唤醒一个 wait 操作。对于同一条件变量 $cond$ ，令 WT 作为在 $cond$ 上所有 wait 操作的集合，令 SG 作为在 $cond$ 上所有 signal 操作的集合。如要满足之上的条件，须有以下公式：

$$\Phi_{syn}^{w/s} = \bigwedge_{e_{wt} \in WT} \{ (\bigvee_{e_{sg} \in SG_{wt}} O(e_{wt}) > O(e_{sg}) \wedge pair_{wt}^{sg} = 1) \wedge \sum_{e \in SG_{wt}} pair_{wt}^e = 1 \}$$

其中，令 e_{wt} 为 WT 中的一个元素， SG_{wt} 表示 e_{wt} 可以匹配的 signal 操作的集合， WT_{sg} 表示 e_{sg} 可以匹配的 wait 操作的集合。本文利用变量 $pair_{wt}^{sg}$ 是否等于 1 来表示 e_{sg} 是否与 e_{wt} 相匹配。子公式 $\sum_{e \in SG_{wt}} pair_{wt}^e = 1$ 表示，对于每一个 wait 操作 e_{wt} 必须有一个 signal 操作与之匹配。

在路径 π 中，只有锁 m ，同步语义约束公式如下：

$$o(e_6) < o(e_8) \vee o(e_{11}) < o(e_4)$$

其中，约束表达式表示要么线程 1 先获取锁 $o_6 < o_8$ ，要么线程 2 先获取锁 $o_{11} < o_4$ 。

S205) 计算偏序约束，其规定：如果事件创建一个线程，那么被创建线程的所有事件都要在此事件之后执行。如果事件执行线程终止操作，那么被终止线程的所有事件都要在此事件之前。令 C 为 create/fork 操作的事件集合，

令 J 作为 join 操作的事件集合。给定约束：

$$\Phi_{po} = \bigwedge_{e_c \in C} O(e_c) < first(e_c) \wedge \bigwedge_{e_j \in J} O(e_j) > last(e_j)$$

其中， e_c 为线程创建事件， $first(e_c)$ 为 e_c 所创建的线程首个事件的顺序； e_j 为线程终止事件； $last(e_j)$ 为 e_j 所结束的线程末尾事件的顺序。

在路径 π 中，线程创建语句为 O_2, O_3 ，其偏序关系约束如下公式：

$$o(e_2) < o(e_4) \wedge o(e_3) < o(e_8)$$

其中，约束表示线程创建语句第 2 行在其被创建线程 1 的首个事件前执行。

S206) 将以上五种约束进行相与，得到约束模型 F 。

步骤 S3)：构建数据竞争候选集合，以及生成每一个候选的竞争发生条件。对于访问同一变量的事件 e_i 与 e_k ， e'_i 与 e''_i 分别表示 e_i 的前一个事件与后一个事件；同样， e'_k 与 e''_k 分别表示 e_k 的前一个事件与后一个事件，那么两者发生数据竞争（同时访问同一内存）的条件 ρ 为：

$$O(e'_i) < O(e_k) < O(e''_i) \wedge O(e'_k) < O(e_i) < O(e''_k)。$$

此示例中的候选集合以及竞争发生条件如下：

候选	发生条件
<1,5>	$o(e_5) < o(e_2) \wedge o(e_4) < o(e_1) < o(e_6)$
<1,9>	$o(e_9) < o(e_2) \wedge o(e_8) < o(e_1) < o(e_{10})$
<5,9>	$o(e_4) < o(e_9) < o(e_6) \wedge o(e_8) < o(e_5) < o(e_{10})$
<7,10>	$o(e_6) < o(e_{10}) \wedge o(e_9) < o(e_7) < o(e_{11})$

数据竞争的定义为当两个线程同时访问同一内存，且至少有一个写操作。

以第 4 个候选为例说明， $o(e_6) < o(e_{10}) \wedge o(e_9) < o(e_7) < o(e_{11})$ 表示第 7 行与第 10 行之间发生数据竞争的条件，其中 $o(e_9) < o(e_7) < o(e_{11})$ 表示第 7 行能够发生在第 10 行的前一事件（第 9 行）与后一事件（第 11 行）之间；由于第 7 行为线程末尾，故只用 $o(e_6) < o(e_{10})$ 。此竞争条件成立说明第 7 行与第 10 行可以同时访问同一变量 y ，则出现数据竞争。

步骤 S4—S7)：针对每一个候选，利用求解器求解 $F \wedge \rho$ ，验证数据竞争候选集合 DRCS 中所有的候选是否为有效数据竞争。以下逐一验证每个候选：

验证 $F \wedge o(e_5) < o(e_2) \wedge o(e_4) < o(e_1) < o(e_6)$ ，结果得出第 1 行与第 5 行不发生数据竞争；

验证 $F \wedge o(e_9) < o(e_2) \wedge o(e_8) < o(e_1) < o(e_{10})$ ，结果得出第 1 行与第 9 行不发生数据竞争；

验证 $F \wedge o(e_4) < o(e_9) < o(e_6) \wedge o(e_8) < o(e_5) < o(e_{10})$ ，结果得出第 5 行与第 9 行不发生数据竞争；

验证 $F \wedge o(e_6) < o(e_{10}) \wedge o(e_9) < o(e_7) < o(e_{11})$ ，结果得出第 7 行与第 9 行之间数据竞争，且证据序列为：1,2,3,4,5,6,8,9,10,7,11。

遍历完 DRCS 之后，终止验证工作。

步骤 S8) 收集所有数据竞争以及对应的证据序列。

权利要求书

1、一种基于多线程程序约束构建的数据竞争检测与证据生成方法，其特征在于，包括如下步骤：

S1) 在给定输入下，通过执行已插桩的待测程序以生成路径记录文件，且识别出执行路径中公有变量的访问点以便于约束构建；

S2) 根据程序执行语义将执行路径中状态转移、线程交织关系转化为无量词一阶逻辑表达式，构建蕴含了所有可能的交织序列的多线程程序执行路径约束模型 F ；

S3) 将路径中所有线程上可能发生数据竞争的两点视为数据竞争候选，收集所有候选并构建数据竞争候选集合 $DRCS$ ，同时根据数据竞争的定义构建每个候选的竞争发生条件 ρ ；

S4) 针对每一个候选竞争发生条件 ρ ，利用约束求解器验证 $F \wedge \rho$ 是否有解；

S5) 如果有解，则表示此竞争条件会触发真实的数据竞争；如果无解，则表示此候选不会触发数据竞争；

S6) 当存在数据竞争时，输出该数据竞争的证据序列；

S7) 对于数据竞争候选集合 $DRCS$ ，如果遍历结束，则输出所有结果；否则，继续遍历下一个竞争候选；

S8) 验证结束后，输出检测到的所有数据竞争以及对应的证据序列。

2、根据权利要求1所述基于多线程程序约束构建的数据竞争检测与证据生成方法，其特征在于，所述步骤 S1) 中插桩工作并非在源码或者二进制的层面上进行，而是在字节码的层面上完成，具体实施方法为：首先将待测多线程程序源码转化为中间字节码格式，即 LLVM 字节码；然后将具有监控功能的语句植入待测程序；最后将植入监控代码的字节码链接成可执行程序。

3、根据权利要求1所述基于多线程程序约束构建的数据竞争检测与证据生成方法，其特征在于，所述步骤 S2) 中多线程程序执行路径约束模型 F 蕴含了执行路径所有可能的交织序列，包括五种约束：路径表达式、内存模型约束、

读写关系约束、偏序约束以及同步语义约束，定义分别如下：

1) 路径表达式：描述线程内部的定义-使用链，以及控制线程内部状态转换；

2) 内存模型约束：表示程序中语句、变量之间的关系，采用顺序一致性的语义，顺序一致性规定 CPU 按照代码中语句的顺序来执行程序；

3) 读写关系约束：定义线程间的定义-使用链，规定共享变量所读取到的值，必须来自初始值以及最近的写值；

4) 偏序约束：定义线程之间创建线程与终止线程操作语句同被操作线程语句之间的时序关系；

5) 同步语义约束：定义线程之间同步控制操作语句之间的时序关系；

其中，定义-使用链为：将每一个线程序列转化为 SSA 格式，对于每一个 SSA 格式的线程序列，除去共享访问点都是一个完整的定义-使用链。

4、根据权利要求 3 所述基于多线程程序约束构建的数据竞争检测与证据生成方法，其特征在于，所述步骤 S2) 中多线程程序执行路径约束模型 F 的构建方法包括以下操作：

- 1) 计算路径表达式，以控制线程内部状态转移；
- 2) 计算内存模型约束，以线程内限制语句之间的关系；
- 3) 计算读写关系约束，以建立线程间的定义-使用链；
- 4) 计算同步语义约束，以定义线程间同步关系；
- 5) 计算偏序约束，以描述线程创建与终止的语义；

最后，结合以上五种约束，构成约束模型 F 。

5、根据权利要求 4 所述基于多线程程序约束构建的数据竞争检测与证据生成方法，其特征在于，定义执行路径事件集合 $\mathbb{E}_\pi = \{T_i | 0 < i \leq k\}$ ，其中 k 为线程数量， $T_i = \{e_1, e_2, \dots, e_n\}$ 作为线程 i 的执行序列， e_n 表示 T_i 的第 n 个事件， $O(e_n)$ 表示事件 e_n 的顺序， n 表示 T_i 的事件数量，则：

所述路径表达式的计算方法：

将每一个线程序列转化为 SSA 格式，直接将 SSA 格式序列转化为路径

表达式;

所述内存模型约束的计算方法:

采用顺序一致性模型, 所有操作完全按程序的顺序执行, 线程内的事件顺序符合约束:

$$\Phi_{mo} = \bigwedge_{T_i \in \tau} \left(\bigwedge_{e_i, e_{i+1} \in T_i} O(e_i) < O(e_{i+1}) \right)$$

其中 e_i 与 e_{i+1} 表示同一线程内连续的两个事件, τ 表示所有线程序列;

所述读写关系约束的计算方法:

使共享变量的读来自于最近的写, 对于同一共享变量 v , 令 R 作为所有对其进行读操作的事件集合, 令 W 作为所有对其进行写操作的事件集合, 给出以下公式:

$$\Phi_{rw} = \bigwedge_{e_r \in R} \bigvee_{e_w \in W} \{ (v_r = v_w \wedge O(e_w) < O(e_r)) \wedge \bigwedge_{e_x \in W} (O(e_x) < O(e_w) \wedge O(e_r) < O(e_x)) \}$$

其中, e_r 为读事件, e_w 与 e_x 为写事件, v_r 和 v_w 为事件 e_r 与 e_w 所操作的变量, 公式所表达的意思是, 如果事件 e_r 中的 v_r 取值来自于事件 e_w 中的 v_w , 首先要满足 e_r 在 e_w 之后, 即 $O(e_w) < O(e_r)$; 然后要满足所有的写要么在 e_w 之前, 要么在 e_r 之后;

所述同步语义约束的计算方法包括 lock/unlock 与 wait/signal 两类操作:

1) lock/unlock 操作的目的是构建锁同步语义约束, 要求在同一互斥锁的 lock/unlock 集合 L 中, 对于任意两个 lock/unlock 事件对: l_i/u_i 与 l_k/u_k , 须满足公式:

$$\Phi_{syn}^{l/u} = \bigwedge_{l_i/u_i, l_k/u_k \in L} O(u_i) < O(l_k) \vee O(u_k) < O(l_i)$$

其中, 锁对 l_i/u_i 要么发生在锁对 l_k/u_k 之前, 要么发生在其后;

2) wait/signal 操作的目的是构建条件变量同步语义约束, 要满足条件: 每一个 wait 操作必须对应一个 signal 操作, 而一个 signal 操作至多唤醒一个 wait 操作, 对于同一条件变量 cond, 令 WT 作为在 cond 上所有 wait 操作的集合, 令 SG 作为在 cond 上所有 signal 操作的集合, 如要满足之上的条件, 须

有以下公式：

$$\Phi_{syn}^{w/s} = \bigwedge_{e_{wt} \in WT} \{ (\bigvee_{e_{sg} \in SG_{wt}} O(e_{wt}) > O(e_{sg}) \wedge pair_{wt}^{sg} = 1) \wedge \sum_{e \in SG_{wt}} pair_{wt}^e = 1 \}$$

其中， e_{wt} 为 WT 中的任一元素， SG_{wt} 表示 e_{wt} 可以匹配的 signal 操作的集合， e_{sg} 为 SG_{wt} 中任一 signal 操作事件，利用变量 $pair_{wt}^{sg}$ 是否等于 1 来表示 e_{sg} 是否与 e_{wt} 相匹配，子公式 $\sum_{e \in SG_{wt}} pair_{wt}^e = 1$ 表示，对于每一个 wait 操作 e_{wt} 必须有一个 signal 操作与之匹配；

所述偏序约束的计算方法：

首先规定：如果事件创建一个线程，那么被创建线程的所有事件都要在此事件之后执行；如果事件执行线程终止操作，那么被终止线程的所有事件都要在此事件之前；令 C 为 create/fork 操作的事件集合，令 J 作为 join 操作的事件集合；给定约束：

$$\Phi_{po} = \bigwedge_{e_c \in C} O(e_c) < first(e_c) \wedge \bigwedge_{e_j \in J} O(e_j) > last(e_j)$$

其中， e_c 为线程创建事件， $first(e_c)$ 为 e_c 所创建的线程首个事件的顺序； e_j 为线程终止事件； $last(e_j)$ 为 e_j 所结束的线程末尾事件的顺序；

最终将以上五种约束相与构成约束模型 F 。

6、根据权利要求 1 所述基于多线程程序约束构建的数据竞争检测与证据生成方法，其特征在于，所述步骤 S3) 中竞争发生条件 ρ 的构建方法如下：如果有一条路径 $\tau = \langle \tau_1 e_i e_j \tau_2 \rangle$ ，其中 τ_1 是前缀， τ_2 是后缀，事件 e_i 与 e_j 属于不同的线程并且都访问同一内存，至少有一个写，那么二者之间发生了数据竞争，而 τ 看作是 e_i 与 e_j 数据竞争的证据序列；对于访问同一变量的事件 e_i 与 e_k ， e_i' 与 e_i'' 分别表示 e_i 的前一个事件与后一个事件；同样， e_k' 与 e_k'' 分别表示 e_k 的前一个事件与后一个事件，那么两者发生数据竞争即同时访问同一内存的条件 ρ 为：

$$O(e_i') < O(e_k) < O(e_i'') \wedge O(e_k') < O(e_i) < O(e_k'')。$$

7、根据权利要求 1 所述基于多线程程序约束构建的数据竞争检测与证据

生成方法，其特征在于，所述步骤 S7) 中对于每一个数据竞争都会生成证据序列，以描述其触发过程。

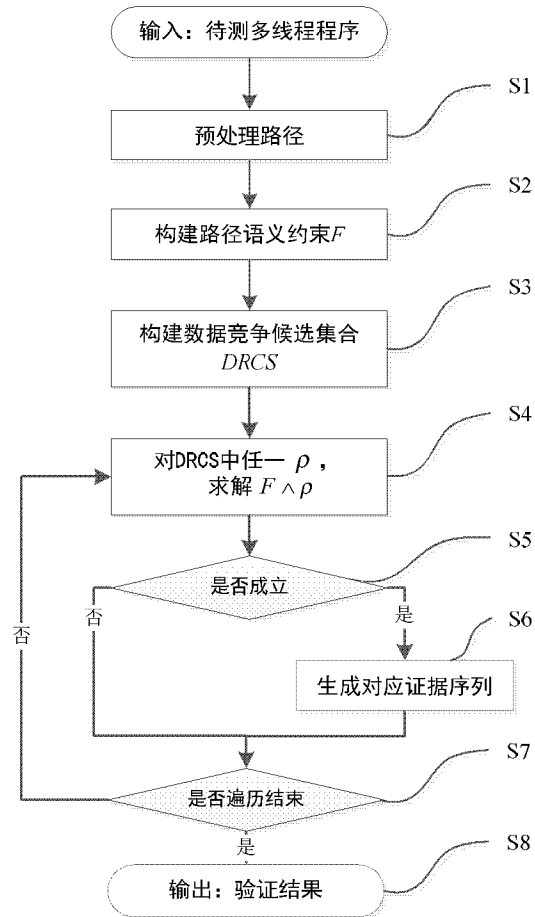


图 1

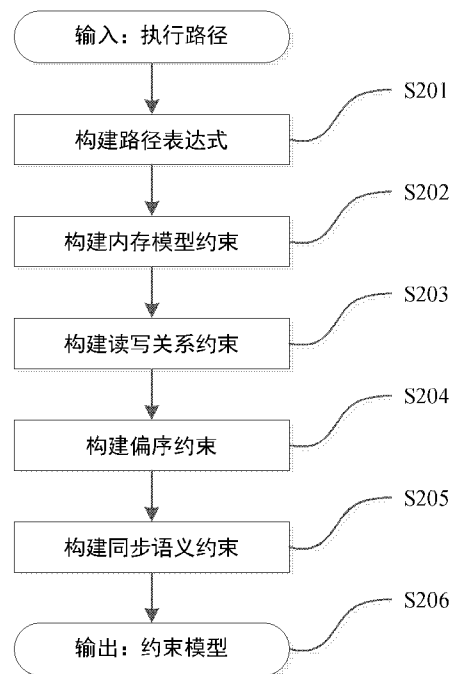


图 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2015/081062

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/36 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI, CNABS, DWPI, SIPOABS: thread, data competition, data race, dynamic

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 102760095 A (UNIV. QINGHUA) 31 October 2012 (31.10.2012) , the whole document	1-7
A	CN 103488563 A (LOONGSON TECHNOLOGY CO., LTD.) 01 January 2014 (01.01.2014) , the whole document	1-7
A	US 2009282288 A1 (NEC LAB AMERICA INC.) 12 November 2009 (12.11.2009) , the whole document	1-7
PX	CN 104077144 A (UNIV. XIAN JIAOTONG) 01 October 2014 (01.10.2014) , the whole document	1-7

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“A” document defining the general state of the art which is not considered to be of particular relevance	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“E” earlier application or patent but published on or after the international filing date	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“&” document member of the same patent family
“O” document referring to an oral disclosure, use, exhibition or other means	
“P” document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 27 July 2015 (27.07.2015)	Date of mailing of the international search report 14.08.2015
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer Zhang, Yiliang Telephone No. (86-10) 62089422

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/CN2015/081062

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 102760095 A	31. 10. 2012	CN 102760095 B	26. 11. 2014
CN 103488563 A	01. 01. 2014	None	
US 2009282288 A1	12. 11. 2009	US 8200474 B2	12. 06. 2012
CN 104077144 A	01. 10. 2014	CN 104077144 B	03. 06. 2015

A. 主题的分类 G06F 11/36 (2006.01) i 按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类		
B. 检索领域 检索的最低限度文献 (标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用)) CNKI, CNABS, DWPI, SIPOABS: 线程, 数据竞争, 动态, thread, data competition, data race, dynamic		
C. 相关文件		
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	CN 102760095 A (清华大学) 2012年 10月 31日 (2012 - 10 - 31) 全文	1-7
A	CN 103488563 A (龙芯中科技术有限公司) 2014年 1月 1日 (2014 - 01 - 01) 全文	1-7
A	US 2009282288 A1 (NEC LAB AMERICA INC) 2009年 11月 12日 (2009 - 11 - 12) 全文	1-7
PX	CN 104077144 A (西安交通大学) 2014年 10月 1日 (2014 - 10 - 01) 全文	1-7
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。		
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件		
国际检索实际完成的日期		国际检索报告邮寄日期
2015年 7月 27日		2015年 8月 14日
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局 (ISA/CN) 北京市海淀区蓟门桥西土城路6号 100088 中国 传真号 (86-10)62019451		授权官员 张一良 电话号码 (86-10)62089422

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2015/081062

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	102760095	A	2012年 10月 31日	CN 102760095 B	2014年 11月 26日
CN	103488563	A	2014年 1月 1日	无	
US	2009282288	A1	2009年 11月 12日	US 8200474 B2	2012年 6月 12日
CN	104077144	A	2014年 10月 1日	CN 104077144 B	2015年 6月 3日