

基于扩展通用图灵机的计算机病毒传染模型

王 剑 唐朝京 张 权 张森强 刘 俭
(国防科学技术大学电子科学与工程学院 长沙 410073)
(xx-wangjian@sina.com)

摘 要 在计算机基础理论模型——图灵机模型的基础上,提出了一种扩展的通用图灵机(EUTM)模型,这种模型突出了计算机病毒的传染特性,极大地简化了计算机病毒传染的形式描述,并根据 EUTM 模型给出了计算机病毒的形式定义,形式化描述了计算机病毒在单机和多机环境下的传染.最后指出了 Fred Cohen 关于病毒检测不可判定性定理证明的不足,并利用 EUTM 模型证明了病毒检测不可判定性定理.

关键词 病毒传染模型; 扩展通用图灵机; 图灵机

中图法分类号 TP309.5

A Computer Viruses' Infection Model Based on an Expanded Universal Turing Machine

WANG Jian, TANG Chao-Jing, ZHANG Quan, ZHANG Sen-Qiang, and Liu Jian
(School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073)

Abstract A model of an expanded universal Turing machine(EUTM) based on the model of Turing machine is proposed, which is the basic theory of computer, and the infection of computer viruses is formally described according to the EUTM model in the circumstance of a single machine and multi-machine environment. The procedure or virus is considered as the code of Turing machine, and the code of Turing machine is regarded as the input of the EUTM model. EUTM reads the inputs, copies the inputs on the storage's tape, and at the same time writes the results on the output's tape. A procedure is a virus iff ① $\| \Phi_v(p) \| = \| v \| \| x \|$; ② $\langle \Phi_v(p) \rangle \supset \langle v \rangle$ or $\langle \Phi_v(p) \rangle \supset \langle V(v) \rangle$; ③ $\Phi_v(p)$ has the properties of procedure v , that is to say, $\Phi_v(p')$ satisfies the condition 1 and 2. The deficiencies of Fred Cohen's proof about virus undetectability theory are pointed out. And the theory of virus undetectability is proved based on the EUTM model. The model of the EUTM simplifies the formal description of computer viruses' infection and emphasizes the basic property of computer viruses.

Key words infection model of computer viruses; expanded universal Turing machine; Turing machine

1 引 言

传染性是计算机病毒最主要的特征,计算机病毒通过传染其他程序来实现传播.图灵机是 A. Turing 于 1936 年提出的一种计算模型,根据图灵机理论,图灵机可以模拟任意一种算法,通用图灵机可以模拟任意算法的图灵机^[1].计算机病毒作为一种具有特定功能的算法,同样可以用图灵机或通用图灵机进行描述,论文从计算机的基础理论模型

——图灵机模型——出发,提出了一种扩展的通用图灵机模型 EUTM,极大地简化了计算机病毒传染机制的形式化描述,对计算机病毒的传染特性和可传播性的形式化描述开辟了新的领域,有助于我们正确地理解计算机病毒.

2 计算机病毒的定义

Fred Cohen 在 1983 年提出了计算机病毒的定义,他认为能够进行自我复制的程序就是计算机病

毒^[2]. 这个定义不是很准确, 按照他的定义, 编译器和编辑器等程序都将被认为是病毒程序. John Inglis 定义病毒为有以下两个特性的代码: ① 至少具备部分自动复制自身的能力; ② 具备一种将自身依附在其他逻辑媒体如程序、磁盘扇区、数据文件的能力^[3]. 但是 John Inglis 的定义不能很好地解释病毒变体的情况

在文献[3]中定义计算机病毒为一段在被激活时能够复制自身或自身的变形到某一个或某些母程序中的机器代码, 当被感染的程序运行时, 病毒代码被执行, 病毒能够进一步传播. 计算机病毒一般包含病毒体(病毒程序)和寄生体, 寄生体为病毒提供一种生存环境, 为一合法程序. 当病毒程序寄生于合法程序后, 病毒就成了程序的一部分, 并在程序中占有合法的地位. 病毒程序一旦寄生于合法程序后, 就随原合法程序的生存而生存, 随它的消失而消失. 计算机病毒一般由引导模块、传染模块和发作模块组成. 病毒的引导模块主要完成病毒体的定位和驻留; 病毒的传染模块主要完成病毒的传染; 病毒的发作模块主要完成病毒的表现和破坏. 本文所有的讨论都是基于这种定义的计算机病毒.

3 图灵机模型

3.1 图灵机及通用图灵机的概念

图灵机是将计算过程抽象化、理想化后所得到的一种计算模型, 它由 3 部分组成: 一条带、一个读写头、一个控制器^[4]. 图灵机的格局由当前状态、当前带内容、读写头的当前位置组成. 图灵机开始运行后, 根据转移函数所描述的规则进行计算, 图灵机就从一个格局到另一个格局进行转换

图灵机本质上是一个程序或算法, 当给定一个输入数据 x 以后, 就可以计算出 $f(x)$. 从这个意义上讲, 一个图灵机相当于一个专用计算机, 它进行一种特定的计算. 所谓通用图灵机是对通用用途计算机的数学模型, 这种机器能够读取程序并执行该程序. 通用图灵机作为一个程序, 在某种意义上相当于一个解释程序, 即给定任意应用程序以及输入数据, 利用解释程序就可以得到输出. 采用通用图灵机模型可以对任意特定的图灵机进行仿真^[4].

3.2 图灵机的形式定义

根据文献[5], 图灵机定义为一个 7 元组 $(Q, \Sigma, \Gamma, \delta, q_0, \beta, F)$. 其中:

Q 为有限状态集, $Q = \{q_1, \dots, q_n\}$;

Γ 为有限带字母表集, $\Gamma = \{a_1, \dots, a_m\}$;

β 为空白符号;

Σ 为输入字母表集, 不包括空白符号 β , 且 $\Sigma \subset \Gamma$;

δ 为转移函数, $\delta: Q \times \Gamma^+ \rightarrow Q \times \Gamma \times \{L, R\}$, 其中 L, R 分别代表左移和右移;

q_0 表示起始状态, $q_0 \in Q$;

F 为结束状态集, $F \subseteq Q$;

d_0, d_1 表示读写头移动的方向.

图灵机 $M = (Q, \Sigma, \Gamma, \delta, q_0, \beta, F)$, M 开始运行后, 根据转移函数 δ 所描述的规则进行计算. 转移函数 $\delta(q_i, a_j) = (q_k, a_l, d_m)$ 表示从一个图灵机格局到另一个图灵机格局的转移, 即当前状态为 q_i , 当前带内容为 a_j , 读写头按照 d_m 给定的方向移动, 这样图灵机的当前状态改变为 q_k , 带内容被改为 a_l . 我们可以通过一系列有效的移动来描述图灵机 M 的结构.

3.3 图灵机模型用于计算机病毒研究的局限

图灵机模型是计算机的一个理论模型, 不能看作真正的机器. 图灵机有一条无穷长的带, 这相当于有无限大的存储, 而实际的计算机的存储量是有限的, 因此所谓图灵机可计算的函数, 也许不可能用任何计算机真正进行计算, 这是图灵机与计算机的根本区别; 图灵机假定机器只有一条带, 一个读写头, 而且读写头又只能一格一格移动, 这相当于只有一个寄存器, 数据都存储在一条带上, 又只有一种存储方式, 这与实际的计算机距离很远, 效率也太低, 即使作为理论研究的工具也是很很不方便的; 图灵机的指令系统与程序同计算机的指令系统与程序也大不相同^[6]. 由于对于不同的输入输出关系必须构建不同的图灵机模型来进行计算, 因此采用图灵机模型不能很好地描述程序与程序之间的关系, 不便于用于分析病毒的传染.

4 扩展的通用图灵机模型 EUTM

4.1 通用图灵机模型

由于通用图灵机模型可以对任意特定的图灵机进行仿真, 现在构建一个通用图灵机 U , U 读取其他图灵机 M 的编码 $\langle M \rangle$ 和 M 的输入字母表 ω , 这样当读取输入 ω 时, U 仿真 M 的计算. U 相当于是 M 的解释器, 操作与其他图灵机一样. 所以所谓通用图灵机 U 是一个图灵机, 它的输入不只是数据 ω , 还有某个图灵机的编码 $\langle M \rangle$, 因此 U 的输入为

数组 $(\langle M \rangle, \omega)$, U 对这个数组的计算等同于图灵机 M 对输入 ω 的计算. 只要对 U 输入任何专用机 M 的程序, U 就可以进行 M 的计算. 如第 3.2 节所述, 可以通过一系列有效的状态转移来描述图灵机, 对于某一个状态转移, $\delta(q_i, a_j) = (q_k, a_l, d_m)$ 我们可以通过二进制编码 $0^i 1 0^j 1 0^k 1 0^l 1 0^m$ 来表示^[5].

定义 1. 定义 $\langle \delta_r \rangle$ 为根据转移函数 δ 和转移次数对一次状态转移的编码, 其中 $r = 1, \dots, p$, p 表示转移次数

根据定义 1, 整个图灵机的二进制编码可以表示为

$$\langle M \rangle = 111\langle \delta_1 \rangle 11\langle \delta_2 \rangle 11 \dots 11\langle \delta_p \rangle 111,$$

图灵机 M 的编码 $\langle M \rangle$ 就作为通用图灵机 U 的一个输入程序, 通用图灵机 U 仿真 M 的计算. 图 1 为通用图灵机模型:

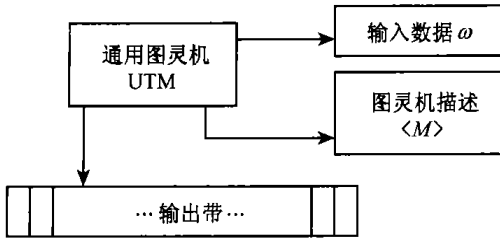


图 1 通用图灵机模型

所谓程序是一系列的指令, 程序的执行即一系列指令的操作, 通过指令的操作来改变计算机的状态, 状态转移函数反映了程序执行的一系列的状态变化. 程序的执行常常改变机器的状态(寄存器的状态、内存), 一个程序的含义是当程序运行时执行了哪些操作. 程序 P 的执行对于基本图灵机来说实际上是一系列转移函数的变化: $\delta(q_1, a_1) \rightarrow \delta(q_2, a_2) \rightarrow \dots \rightarrow \delta(q_p, a_p)$; 对于通用图灵机 UTM, 程序 P 的执行是根据输入 $(\langle M \rangle, \omega)$ 在输出带上输出程序执行的结果, 输出带包括机器的所有状态(如寄存器的状态).

定义 2. 定义机器所有的状态(包括所有的错误条件), 程序文本为机器的表示状态, 对于通用图灵

机来说机器的表示状态包括所有可能的编码形式, 输出带内容、输入数据等. 我们记所有可能的表示状态的集合为 R . 对于任意一个给定的表示状态 $r \in R$, r 包含一个程序 P , 那么程序 P 执行的意义为 $\|\cdot\|: R \rightarrow (R \rightarrow R)$, 即程序运行时机器从一种表示状态转换到另一种表示状态.

4.2 扩展的通用图灵机(EUTM)模型

上述的通用图灵机模型只限于分析一种单一的算法或程序, 如果要分析两个或更多的算法和程序之间的联系, 这种模型显然不方便. 如果要建立程序之间的联系, 需要对上述模型进行修改, 增加一个用于储存程序和程序数据的可读写的储存带. 图 2 为 EUTM 模型.

$(\langle M_1 \rangle, \omega_1) \dots (\langle M_p \rangle, \omega_p)$ 为 p 个图灵机的编码形式, 作为 EUTM 的输入带编码. EUTM 对输入的图灵机编码和数据只是一个只读的操作, EUTM 读取输入带编码 $(\langle M \rangle, \omega)$, 相当于一个程序执行起来, 并且将程序执行的结果输出在输出带上. EUTM 对输出带只是一个可写的操作. EUTM 读取输入带编码 $(\langle M \rangle, \omega)$ 的同时, 将输入带的图灵机编码拷贝到储存带上保存. EUTM 对储存带是一个可读写的操作, 由于输入带编码储存在储存带上, 所以输入带编码可以被改变, 图灵机编码的描述相当于一个程序, 这个程序由 EUTM 执行, 这就相当于程序可以改变自身, 而且通过这种机制, 一个程序可以对另一个程序进行操作. EUTM 自己的内部程序是固定的(相当于是一个黑匣子), 它读入其他图灵机的编码和关于这个图灵机的输入字母表, 来仿真这个图灵机的计算, 把结果输出到输出带上. 我们之所以要加一个储存带是为了将程序和数据复制到这个带上, EUTM 对这个带执行可读写的操作, 这样 EUTM 可以改变程序自身, 而且可以对别的程序进行操作, 这样就可建立两个或更多算法或程序之间的联系, 便于我们进行病毒分析. 这个储存带我们可以看做是实际计算机的磁盘和内存.

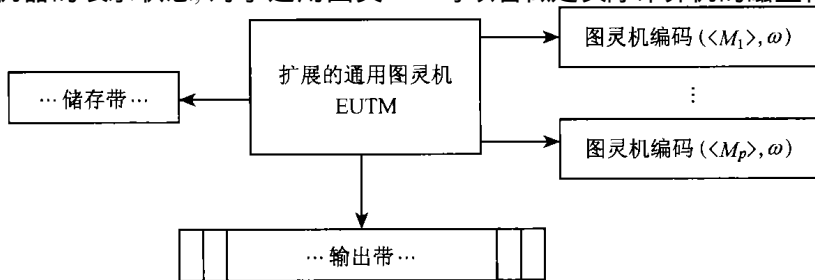


图 2 EUTM 模型

5 计算机病毒传染的 EUTM 描述

5.1 Fred Cohen 的计算机病毒图灵机描述

Cohen 在文献[7]中运用图灵机提出了计算机病毒的典型模型, 建立了病毒集的概念. 病毒集用 (M, W) 来表示. M 是一个图灵机, W 是带字母表上的一组字符串. $\omega \in W$, 每一个字符串 ω 具备如下特性, 当 M 在启动状态时, 开始读 ω , 并且在带上的其他地方写下另一个字符串 ω' , $\omega' \in W$. 这样字符串 ω 就是一个病毒, 并且当 M 读它时, 在带上的某个地方会出现另一个病毒. Cohen 的病毒图灵机模型建立在他定义的病毒基础之上, 上面我们已经讨论了 Cohen 的病毒定义本身就存在缺陷, 那么他的病毒图灵机模型必然存在缺陷. 如编译器在编译一个程序时, 必然将固定的引导模块复制到这个程序中, 根据 Cohen 的病毒图灵机描述, 那么编译器就是病毒. 在文献[8]中详细讨论了 Cohen 病毒模型不完善的地方, 在此就不再详述了.

5.2 计算机病毒的形式定义

定义 3. 程序 p 与程序 p' 相等即 $\forall r \in R$: $\|p\|_r = \|p'\|_r$ (对于任意给定的一个表示状态, 此等式都成立).

上述定义中的等式不同于恒等式, 它不是一个可计算的关系式, 如 2×3 和 6, 合理的解释是相等的但不是完全相同的. 即使对一个小的输入集来检测程序输出的相等性, 也要对整个机器的状态进行彻底的检查. 所以判定程序是否完全相等显然不实际.

定义 4. 为了更贴近实际, 我们定义相似为一个关于 R 的可计算的对数多项式, 用 \approx 来表示, 不相似用符号 $\not\approx$ 表示. 当两个程序对大多数输入产生相似的输出, 则称这两个程序是不可区分的. 即两个程序 p 与 p' 是不可区分的, 当且仅当对于大多数 $r \in R$: $\|p\|_r \approx \|p'\|_r$.

我们把病毒 v 传染程序 p 生成的程序记为 $\Phi_v(p)$, 把 $\Phi_v(p)$ 传染程序 p' 生成的程序记为 $\Phi_v(\Phi_v(p'))$. 如果病毒 v 破坏了原程序的功能, 那么大多数 $r \in R$: $\|p\|_r \not\approx \|\Phi_v(p)\|_r$, 如果病毒 v 没有破坏原程序的功能, 那么对于大多数 $r \in R$: 有 $\|p\|_r \approx \|\Phi_v(p)\|_r$.

定义 5. 一个程序 v 是病毒程序, 当且仅当 $\exists x \in R, \exists \infty p \in R$, 使得:

- ① $\|\Phi_v(p)\| = \|v\| \|x\|$;
- ② $\langle \Phi_v(p) \rangle \supset \langle v \rangle$ 或 $\langle \Phi_v(p) \rangle \supset \langle V(v) \rangle$;

③ $\Phi_v(p)$ 具有程序 v 的性质, 即对于 $\langle \Phi_v(p') \rangle$ 满足以上 ① ② 两个条件.

其中, $V(v)$ 是程序 v 的变体, 被传染后的程序 $\Phi_v(p)$ 的功能为病毒函数和另一个函数的合成, 即 $\|\Phi_v(p)\| = \|v\| \|x\|$, 如果病毒 v 破坏了原程序的功能, 那么 $\|\Phi_v(p)\| = \|v\| \|x\|$, 且对于大多数 $r \in R$: $\|x\|_r \approx \|p\|_r$. 如果病毒 v 没有破坏原程序的功能, 那么 $\|\Phi_v(p)\| = \|v\| \|p\|$, 即被传染后的函数在功能上等于原函数与病毒函数的合成. 那么对于大多数 $r \in R$: 有 $\|p\|_r \approx \|\Phi_v(p)\|_r$ 这里假定病毒在未发作时, 被传染的文件还能够正常工作. $\langle \Phi_v(p) \rangle$ 是指被病毒 v 传染了的程序的图灵机编码形式, $\langle v \rangle$ 指病毒的图灵机编码形式, $\langle V(v) \rangle$ 指病毒变体的图灵机编码形式, 那么 $\langle \Phi_v(p) \rangle \supset \langle v \rangle$ 或 $\langle \Phi_v(p) \rangle \supset \langle V(v) \rangle$ 指被病毒 v 传染了的程序的图灵机编码一定要包括病毒或病毒变体的编码, 这样保证了病毒或病毒的变体被复制到被传染程序中. 第 ③ 条是为了保证被病毒传染的程序具有传染性.

5.3 计算机病毒传染过程的描述

传染的实质是将病毒体或病毒的变体复制到另一个程序中. 传染分静态传染和动态传染. 所谓静态传染是对一个处于静态(非运行状态)的程序的传染. 所谓动态传染指对一个处于动态的程序即正在运行的程序的传染. 我们不详细分析病毒进行的是静态传染还是动态传染, 只考虑传染的结果, 即病毒体或病毒的变种被复制到另一个程序中, 那么就完成了传染. 在我们的模型中, 计算机病毒同样是一段图灵机描述, 即一段编码, 这段编码作为 EUTM 的输入(用 EUTM 来仿真这段编码), 将病毒编码写到 EUTM 的带上(储存带). 具体传染的过程如下:

- (1) EUTM 读取一段图灵机编码 $\langle M' \rangle$, 并将 $\langle M' \rangle$ 复制到储存带上.
- (2) EUTM 读取病毒的编码 $\langle V \rangle$, 并从输出带上找到图灵机编码 $\langle M' \rangle$ (被传染的程序).
- (3) EUTM 将一个特殊的符号插入到 $\langle M' \rangle$ 的起始处.
- (4) EUTM 从输出带上读取这个特殊的字符, 使 $\langle M' \rangle$ 进入一个新的状态的子系统 (即跳转到执行非期望行为).
- (5) EUTM 将病毒的编码 $\langle V \rangle$ 的副本或病毒变体的编码 $\langle V(v) \rangle$ 复制到 $\langle M' \rangle$ 中.
- (6) 然后 EUTM 将控制返回给 $\langle M' \rangle$ 的起始状态, 并且将 $\langle M' \rangle$ 的头部移到原始带内容的第 1 个单

元, 这样 $\langle M' \rangle$ 就被病毒 $\langle V \rangle$ 感染了.

5.4 计算机病毒网络传播模型

以上采用了扩展的通用图灵机模型形式化描述了病毒的传染过程, 但只是描述了在单机情况下病毒的传染, 下面描述通过网络进行的病毒传染. 图 3 为病毒的网络传播模型.

公共带是 EUTM1 和 EUTM2 所共有的带, EUTM1 和 EUTM2 对公共带执行可读写的操作, 具体步骤如下:

(1) 当 EUTM1 要与 EUTM2 进行通信时, 首先 EUTM1 在公共带的标识单元上置一个起始标志位

(2) 然后, EUTM1 把将要发送的程序的编码记为 $\langle M \rangle$, 写到起始标志位后的公共带上.

(3) EUTM1 在公共带上写完程序后, 在带上写一个结束标志位\$.

(4) EUTM2 的读写头读到起始标志位为@ 时, 就从起始标志位后的带上读取内容, 并将带编码存在储存带上

(5) EUTM2 的读写头读到结束标志\$ 时, 表明双方的通信结束, EUTM2 的读写头回到起始标志位的位置.

按照以上步骤, 两个图灵机可以很方便地进行通信. 两个图灵机的公共带就类似现实计算机的网络线, 公共带将两个图灵机紧密地联系起来. 如果 EUTM1 写到公共带上的编码 $\langle M \rangle$ 为一个病毒程序, 病毒就会在两个图灵机之间进行传播.

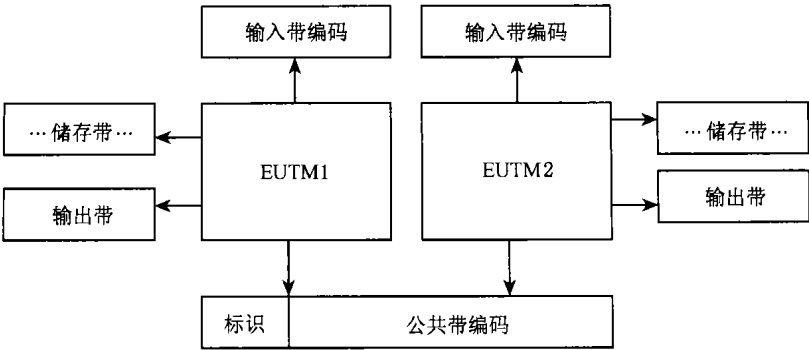


图 3 病毒的网络传播模型

6 病毒的图灵不可检测问题

6.1 Fred Cohen 的病毒检测不可判定性定理的证明

Cohen 在文献[7] 中给出了这个定理的证明, 他的证明运用了著名的证明技巧, 下面我们引入他的证明.

为了判定给定程序 P 是一个病毒, 必须验证 P 是否感染其他程序. 对于任意给定的判定例程 D , P 若被 D 断定为病毒, 则 P 根本无法感染其他程序, 因而不成为病毒, 当且仅当 D 判定 P 不是病毒时, P 才能有机会感染其他程序, 而此时, P 又确实具备病毒的特征, 因此, 这就陷入两难的境地, 说明由其他例程来验证一个程序是否为病毒的做法是不可行的. 在下面的程序 CV (contradictory-virus) 中, 我们利用假设的判定例程 D 来证明病毒检测的不可判定性, 如果是病毒则判定例程 D 返回值为真.

program contradictory-virus :=

```
main-program :=
{if D (contradictory-virus) then
  {infect-executable;
   if trigger-pulled then
     do-damage;
  }
 goto next;
}
```

我们确信, 如果判定程序 D 判定 CV 是一个病毒, CV 不会感染其他程序, 这样 CV 不具备病毒的特性. 如果 D 判定 CV 不是一个病毒, CV 将感染其他程序, 这样 CV 就是一个病毒. 因此假设的判定程序 D 是自相矛盾的, 通过表象无法正确判定程序是否为病毒.

Cohen 的这个证明存在一定的问题, 他基于一个假定, 判定例程 D 自身不是病毒. 假设判定程序 D 自身是病毒, 那么 CV 通过调用 D 会感染病毒, 因此 CV 也是病毒. 当 D 判定 CV 是病毒时, 即使

基于 D 的判定的 CV 不执行自身的病毒代码, D 可能给出一个错误的结果。但是, 在这个限制条件下, 即判定程序 D 必须不是病毒, Cohen 的证明就进行不下去了。

6.2 利用 EUTM 模型证明病毒检测不可判定性定理

病毒的不可判定性定理即不存在一个程序来判定另一个程序是否是病毒。根据 Church-Turing 理论, 如果存在解决一个问题的算法, 那么运用图灵机同样可以解决这个问题。要证明病毒的不可判定性定理即证不可能构建一个图灵机来判断一个程序是否是病毒。在证明中需要用到的 Church-Turing 定理有图灵机停机问题不可判定定理; 通用图灵机可以模拟任意特定的图灵机模型^[4]。下面给出证明

构造一个程序 P , 程序 P 的图灵机编码为 $\langle P \rangle$ 。当图灵机停止在一个可接受的状态时, 程序 P 在输出带上写字符 1; 构造一个病毒程序 V , 病毒体 v 病毒的图灵机编码为 $\langle v \rangle$, 病毒 V 包含程序 P , $\langle V \rangle = \langle P \rangle \langle v \rangle$ 。

随机给定程序 P 一个输入 x , 程序 P 开始运行, 在程序 P 运行之后, 病毒体才开始运行。具体实现可将病毒体附在程序 P 后, 当程序 P 执行完在输出带上写字符 1 的命令后, 将控制权转给病毒体 v , 病毒体在感染时不仅能复制病毒体自身而且能够复制程序 P 和给定的输入 x 。显然, 对于给定的输入 x , 如果程序 P 和图灵机停止了, 那么病毒体就执行起来了, 病毒 V 就能进行传播

假设存在一个图灵机 T , 我们可以用一个通用图灵机 U 来模拟图灵机 T , 通用图灵机 U 读取输入的程序, 如果输入的程序包含病毒, 则通用图灵机 U 在输出带上写字符 1; 如果输入的程序不包含病毒, 则 U 在输出带上写字符 0; 即存在一个通用图灵机能够判断输入的程序是否包含病毒。

根据假设, 通用图灵机 U 的输入为病毒 V (对于通用图灵机实际输入的是病毒 V 的图灵机编码 $\langle V \rangle$), 如果通用图灵机 U 输出为 1, 根据我们构造的病毒 V 可知, 无论接收的输入是什么, 程序 P 或相应的图灵机会停止。如果通用图灵机 U 的输出为 0, 相应的图灵机不会停机。这样通用图灵机 U 能够判定对于任意的输入, 其他图灵机是否会停机 (在这里所谓相应的图灵机是指病毒程序 V 所在的图灵机, 这个相应的图灵机作为通用图灵机 U 的输入)。这与 Church-Turing 的停机不可判定定理相矛

盾, 假设不成立, 所以不存在一个图灵机能够检测病毒, 即不存在一种能够检测病毒的程序(算法)。

7 结束语

本文在文献[3]定义的计算机病毒的基础上, 根据图灵机和通用图灵机模型, 提出了一种扩展的通用图灵机(EUTM)模型, 形式化描述了计算机病毒以及计算机病毒的传染过程, 较好地体现了计算机病毒的传染性和可传播性。在文章的最后, 利用图灵机停机不可判定定理证明了病毒检测不可判定性定理, 至于利用特征代码法、冗余校验法等进行病毒检测的方法本文未做讨论。扩展通用图灵机模型的优点在于它的一般性, 我们还可以利用 EUTM 模型建立操作系统的原始图灵机模型, 并在操作系统图灵机模型的基础上建立网络模型, 进一步描述计算机病毒, 有关这些方面我们还将做更深入的研究。

参 考 文 献

- 1 张立昂. 可计算性与计算复杂性导引. 北京: 北京大学出版社, 1996
(Zhang Li'ang. Introduction to Computability and Complexity (in Chinese). Beijing: Peking University Press, 1996)
- 2 Fred Cohen. Computer virus theory and experiments. Computer & Security, 1987, 6(1): 22~35
- 3 Eugene H Spafford. Computer Viruses as Artificial Life. MA: MIT Press, 1994
- 4 张鸣华. 可计算性理论. 北京: 清华大学出版社, 1984
(Zhang Minghua. Computability Theory (in Chinese). Beijing: Tsinghua University Press, 1984)
- 5 Mäkinen Erkki. Comment on 'A framework for modeling Trojans and computer virus infection'. Computer Journal, 2001, 44(3): 321~323
- 6 H W Thimbleby, S O Anderson, P Cairns. A framework for modeling Trojans and computer virus infection. Computer Journal, 1998, 41(2): 444~458
- 7 Fred Cohen. Computational aspects of computer viruses. Computer & Security, 1989, 8(4): 325~344
- 8 K Kauranen, E Mäkinen. A note on Cohen's formal model for computer viruses. ACM SIGSAC Rev, 1990, 8(1): 40~43



王 剑 男, 1975 年生, 博士研究生, 工程师, 主要研究方向为计算理论、计算机网络攻防对抗技术、信息安全。



唐朝京 男, 1962 年生, 博士, 教授, 主要研究方向为密码学、信息安全



张森强 男, 1978 年生, 博士研究生, 主要研究方向为计算机网络攻防对抗技术、信息安全



张 权 男, 1974 年生, 博士, 主要研究方向为量子信息、信息安全



刘 俭 男, 1978 年生, 硕士, 讲师, 主要研究方向为计算理论、计算机网络攻防对抗

大连民族学院计算机科学与工程系 招聘启事

大连民族学院是隶属国家民委的一所高等院校, 其计算机科学与工程系是学院的重点学科, 设有研究中心, 目前的主要研究方向是计算机图形、图像及非线性技术。24 名教师中具有博士学位的教师 6 人(3 人博士后出站)。另有 6 人在职攻读博士学位。为该系的进一步发展, 现面向国内外招聘计算机专业的教师。我院教师的收入组成是:

基本工资+ 院内岗位津贴+ 课酬+ 科研奖励+ 大连开发区补贴 200 元/月

招聘人员及待遇如下:

(1) 正规院校硕士研究生毕业并具有硕士学位的教师, 享受如下待遇:

- ① 岗位津贴 2020 元/月, 课酬 40 元/课时;
- ② 已婚教师, 学院给予安家费 10 万元, 未婚教师结婚后享受同等待遇;

(2) 具有博士学位的教师, 享受如下待遇:

- ① 岗位津贴 2550 元/月, 课酬 45 元/课时;
- ② 给予安家费 20 万元;
- ③ 学院视其承担科研项目情况, 提供 2~ 10 万元科研启动费;
- ④ 视学院需要, 优先安排配偶工作;

(3) 博士后出站人员, 享受如下待遇:

- ① 岗位津贴 3000 元/月, 课酬 50 元/课时;
- ② 给予安家费 25 万元;
- ③ 学院视其承担科研项目情况, 提供 2~ 15 万元科研启动费;

(4) 优秀学科带头人, 享受如下待遇:

- ① 岗位津贴 3700 元/月;
- ② 给予安家费 30 万元;
- ③ 视承担科研项目情况, 学院提供 2~ 20 万元科研启动费;

(5) 特聘教授, 享受如下待遇:

- ① 给予安家费 35 万元;
- ② 岗位津贴每年 5 万元;
- ③ 视其承担科研项目情况, 学院提供 5~ 30 万元科研启动费;

以上规定中的安家费, 除特聘教授、优秀学科带头人外, 原则上只限于男教师。具有博士学位以上的教师, 学院为其配备住宅电话和电脑。博士后出站人员以上的教师, 学院安排配偶工作。

以上人员除享受学院给予的待遇外, 还可享受以下待遇:

(1) 在国内学术界有一定影响的专家可申请享受大连市政府 3 年每年补贴 5 万元; 博士后可享受大连市政府 3 年每年补贴 2 万元; 博士可享受大连市政府 3 年每年补贴 1 万元。

(2) 博士及博士后, 在开发区落户并购房后, 除享受大连市政府待遇外, 还可享受开发区给予的 3 万元安家费。

联系方式

地址: 大连开发区辽河西路 18 号大连民族学院人才引进办公室, 邮编: 116600

联系人: 袁传军、王喜春; 电话: 0411-7640766

E-mail: rencaiban@dlnu.edu.cn; 网址: www.dlnu.edu.cn