适用于Android 应用程序的系统

Mohamed A. El-Zawawy  
1计算机与信息科学学院，伊斯兰穆罕默德伊本沙特伊斯兰大学（IMSIU），沙特阿拉伯利雅得

2开罗大学理学院数学系，埃及吉萨12613 ，maelzawawy@cu.edu.eg

摘要

今天最常见的计算机形式是手持式，特别是智能手机和平板电脑。智能手机采用的主要平台Android操作系统。因此，分析和验证Android应用程序成为了一个很重要的问题。然而，静态分析的传统技术手段并不适用于Android应用程序的分析。这是因为大多数Android应用程序是使用Java代码开发的，Java代码被翻译成Java字节码并由Dalvik虚拟机执行。因为Android 与用户手势交互等问题，必须开发一种新的用于分析Android应用程序的新技术。

在分析和验证程序的主要工具之一是类型系统[11].文中介绍了一种新的系统AndroidT，它是Android应用程序的特定系统。这类型系统提出会提示诸如“方法未定”之类的错误。这类型系统也适用于检查Android应用程序的健全性。

关键词：安卓应用程序，类型系统，安卓活动；安卓应用程序分析，编程语言，AndroidT，AndroidP

1介绍

今天计算机数量中不断增长的百分比来自手持设备;平板电脑和智能手机。这些计算机中最常见的操作系统是Android。对于手持设备，Android操作系统的一些特性使其成为主要平台[26]. 此文中指出2015年生产的手持电脑数量超过了十亿台设备。这个数量远远超过生产的iOS-MacOS-Windows机器数量[2,4,33]。 由于平板电脑和智能手机的巨大优势，计算的本质受到了极大的影响，发展和变化。 这些设备也广泛传播，安全性，准确性，可靠性对于开发人员和用户来说都是重要的问题。 软件工程是计算机科学的一个分支，它负责通过生成各种用于验证，分析，优化和更正Android应用程序的工具来解决这些问题。

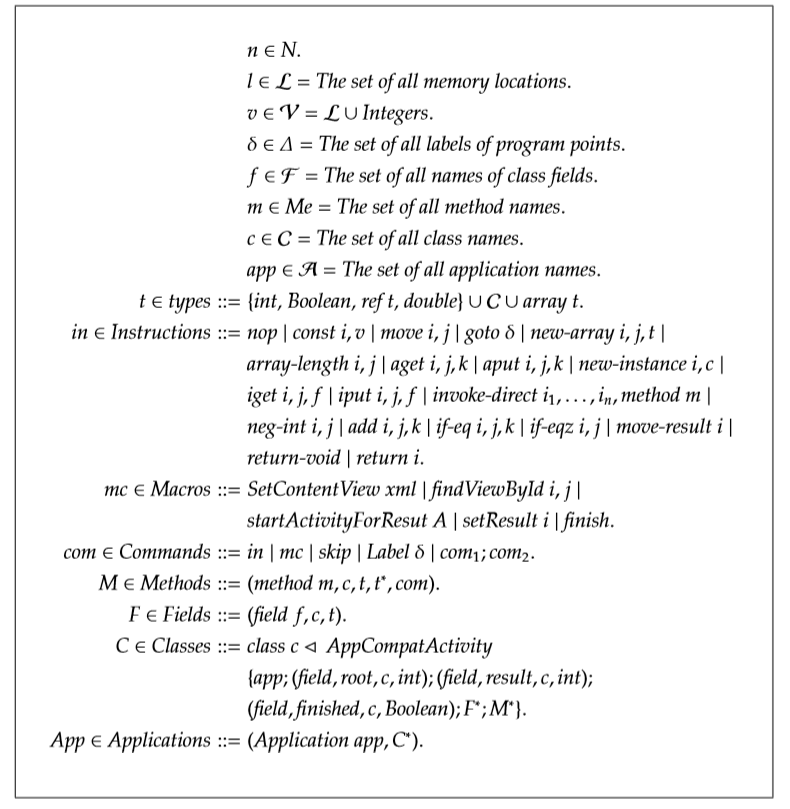


图1 AndroidP：Android应用程序的模型。

更确切地说，静态分析的技术和策略是接近的，是处理这些问题的主要工具。然而，现有的静态和动态分析技术对于研究Android应用程序并不是很好[32,29,20,22]。 这是因为Android应用程序主要是使用Java开发的，它被翻译成Dalvik字节码[33,26]。

关于分析的许多挑战者是Android应用程序。 硬件的功能正在增加（内存容量和处理器速度）。这导致生成复杂和高级Android应用程序的能力更强。这当然会导致研究这些应用程序的更多困难。事件驱动和框架的概念基于 Android平台有助于解决Android应用程序的问题。在Android应用程序中，手势（例如滑动和单击）是用户使用的主要工具应用[15,5]。 Android应用程序负责捕获用户手势并采取相应的行动。这包括识别屏幕像素和事件类型等准确参数。用户和应用程序之间的交互需要执行作为Android框架一部分的宏方法。 以精确的方式处理这些问题和细节使得很难为Android应用程序[33,26]产生高效且准确的静态分析[12]。

在本文中，我们介绍AndroidT，一个用于Android应用程序的简单类型系统[30,7,18]（考虑活动组合）。我们提出的类型系统AndroidT确保在类​​型系统中可键入的Android应用程序不会导致方法是未定义的错误。建议的类型系统处理在通过手势与用户交互期间执行的宏。由于类型系统必须考虑与Android活动的生命周期相关的宏指令，因此涉及开发这种类型系统的问题。此外，在Android应用程序中允许执行不同执行状态的许多活动的组合这一事实使得Android应用程序的类型系统的开发变得复杂。开发该系统的一个关键思想是让打字环境构成两个部分;一个部分报告寄存器和存储器位置的类型，另一部分链接当前活动与存储器位置。

本文开发的类型类型系统有很多应用。除了类型系统保证没有特定错误，类型系统可以用作Android应用程序的验证工具。类型系统可用于通过程序分析技术[25,9,31,24]确保对Android应用程序所做的任何修改的便利性。 尽管提出的类型系统很简单，但它足够强大，因为它是在丰富的Android编程模型AndroidP上设计的，其中包括主要的Dalvik字节代码操作。例如，数组操作包含在我们的编程模型中并在我们的 类型系统。 所提出的类型系统的另一个优点是所使用的类型集是丰富的并且特定于Android应用程序。 例如，类型“视图参考”refv被定义并在类型系统中使用，以捕获托管构成Android活动的视图的地址的寄存器类型。

动机：本文的工作是由对类型系统的需求推动的  
- 特定于Android应用程序和与Android活动生命周期相关的宏指令，  
- 确保没有“方法未定义”等错误，并且 - 适用于检查Android应用程序静态空间的健全性。  
论文大纲：本文的其余部分安排如下。 第2节介绍了用于开发类型系统的语言模型AndroidP。 详细的类型系统，  
AndroidT，在第3节中介绍。第4节介绍了对最相关的最新工作的评论。 该论文在第5节中得出结论。

2.语言模型

图1展示了用于Android应用程序的编程模型AndroidP的语法。模型拥有丰富的类型集，包括整数、布尔值、双位数、类和数组。该模型使用了一组19条Dalvik指令。这些导入是从包含200多条指令的全套指令中精心挑选出来的。androidP的另一个主要组件是一组宏指令[14,1,27]。

在AndroidP中Dalvik指令和宏指令的选择是为了以简单的方式涵盖Dalivk字节码的主要功能。语法中包含的宏几乎用于所有Android应用程序，并控制应用程序的执行。因此，这些指令有望成为Android应用程序大多数分析技术的重点。这将有助于我们的语义通过证明分析技术的数学背景来提供这些技术。

android中的命令集是一系列指令和宏。每个方法都有一个表单(方法m,c,t,t,com)

- m是方法名，

- c是类的拥有方法，

- t是方法的返回类型，

- t是方法参数的类型序列，

- textitcom是方法的主体。

AndroidP中的每个类都有语法(字段f,c,t)，其中c表示有字段的类，f表示字段名，t表示字段类型。AndroidP中的Android类是AppCompatActivity的扩展，因此有以下组件:

- c表示类名，

- app表示具有类的应用程序的名称，

- 一个类型为integer的特殊字段根，用于记录具有类的活动的视图文件的地址，

- 一个类型为integer的特殊字段结果 保持执行类的对象（活动）的结果，

- 一个特殊的字段，用于表示类的对象（活动）是否已完成，

- F \*表示更多字段的序列，

- M \*表示一系列方法结构。

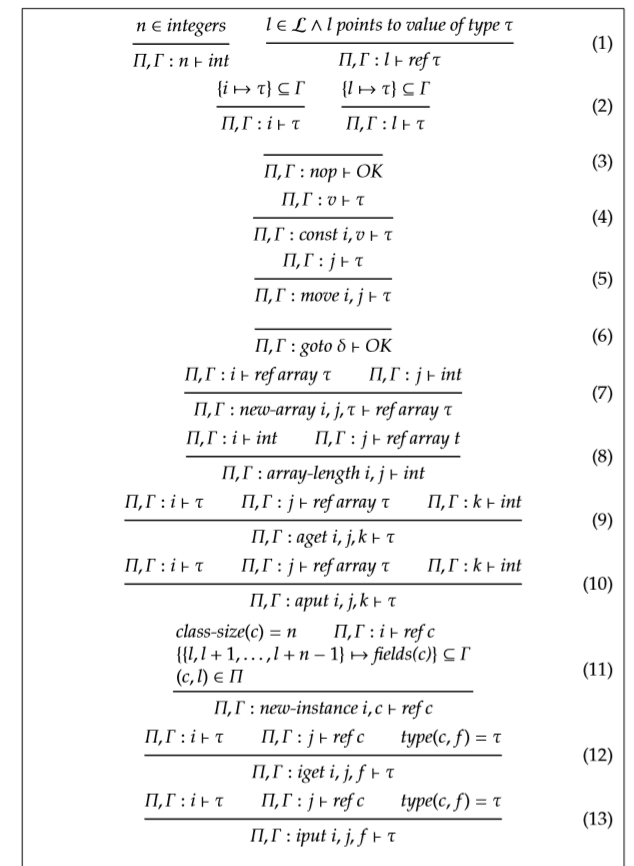
最后，AndroidP中的应用程序具有表单（Applicationapp，C \*），其中包括应用程序名称和类C \*的序列。

图2 类型系统AndroidT的推理规则

我们的模型可以作为[14,1,27]中使用的模型的组合模型来实现，以克服相关模型中的缺点。 [27]遵循了调整宏指令集和类的三个特殊字段的想法。 我们认为[27]中的语言模型非常简单，而[14,1]中的语言模型非常复杂，我们的模型是中等的; 在两个模型之间。我们使用相同的AndroidP模型在[10]中为Android应用程序开发操作语义。

3.类型系统

本节介绍一种用于AndroidP的新型系统AndroidT。类型系统AndroidT旨在保证Android应用程序中类型的健壮性;保证没有动态类型错误，比如方法未找到、变量未找到。对于AndroidP，我们提出的类型系统还支持输入与活动声明的方法如onCreate和onDestroy相关的宏指令。

定义1介绍了AndroidT的类型判断及其组成。

定义1.- 类型判断中使用的判断类型集合T是语言语法AndroidP加上视图引用类型(refv)、文件指向类型 (reff)、void类型和方法类型中定义的类型集合:

τ∈ T = types∪{refv,reff,void,(τ1,...,τn)→τ}.

* 类型环境, Γ,是来自L∪Rto类型的部分映射。
* 类环境，Π，是一对活动类和内存位置的堆栈。
* 类型判断e ∈ {命令, 方法, 字段, 类} 有以下形式之一：

Π,Γ : e⊢τ Π,Γ : e⊢OK.

定义2介绍了在集合T上定义的子类型关系。

定义2. –

1. Boolean⊑int⊑double.

2. ∀τ∈T. τ⊑τ.

3. ∀τ,τ’∈T. τ⊑τ’⇒arraryt⊑arrary t’.

4. ∀τ,τ’∈types. τ⊑τ’⇒ref τ⊑ref τ′.

5. ∀{τ1,...,τn,τ,τ1’...,τn’,τ’} ⊆ types. (∀i. τi ⊑ τi’)∧τ ⊑ τ’⇒ (τ1,...,τn →τ )⊑(τ1’,...,τn’ →τ’).

6. ∀c∈C. c⊑AppCompatActivity.

定义2保证在数组类型、引用类型和方法类型上方便地定义子类型关系。该定义还明确指出，Android应用程序中的所有活动(定义的类)都是继承自AppCompatActivity。然而，子类型关系是部分顺序很重要。

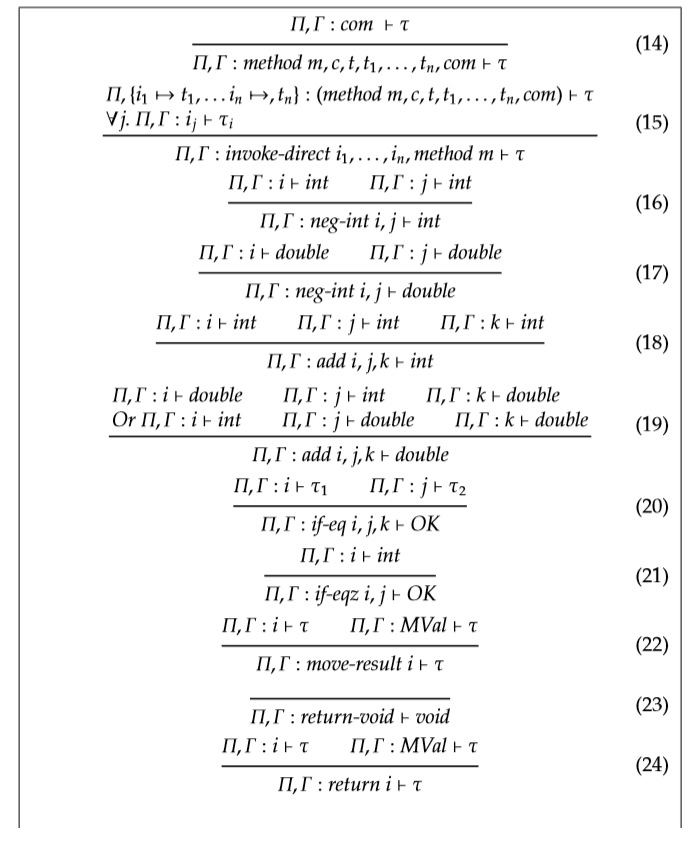
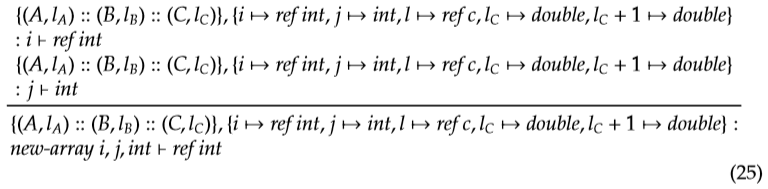


图3 类型系统AndroidT的推理规则（集2）。

引理1.图2展示了针对AndroidP的所提出的类型系统AndroidT的第一组推理规则。 该图显示了Dalvik字节码导入的输入规则。规则7给出了指令new-array i，j，τ的推理规则，它建立了一个类型为τ且长度为j的新数组，并将数组的地址返回到寄存器i。因此，规则要求i为refτ类型，j为int类型。 在这种情况下，规则推断该指令是refτ类型

例1 规则25为应用规则7提供了一个示例。

规则11规定了指令new-instance i，c的输入规则，它建立了类c的新对象，并将对象的地址返回给寄存器i。规则首先调用函数class-size来计算大小，n，内存单元中的类（字节）。 该规则然后假定对象的地址是来自环境（c，l）∈Π的l。 规则还要求根据环境Γ将内存位置与类字段具有相同的类型; {{l，l + 1，...，l + n-1} 7→字段（c）}⊆Γ。

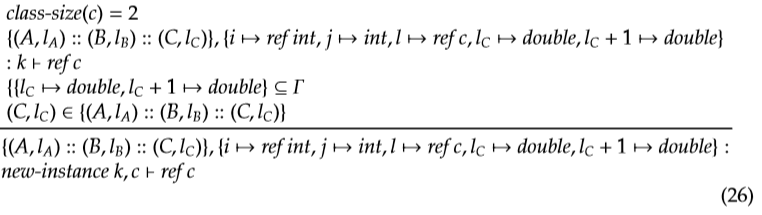
例2 推理规则26给出了一个应用规则11的例子。我们假设类c的大小是2。调用的字段类型为double。

图3展示了提议的用于androidP类型系统AndroidT的第二组推理规则。规则20给出了if-eq i, j,k指令的输入规则，它检查寄存器i和j是否相等。如果是这种情况，则控制转到位置k。规则要求i和j只是可输入的。如果是这样，说明指令正常。

例3 推理规则34给出了一个应用规则20的例子。

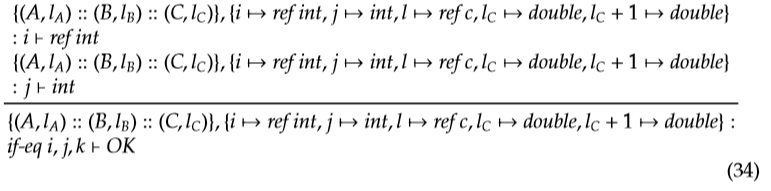


图4展示了提议的用于AndroidP的类型系统AndroidT的第三组推理规则。规则30引入了宏指令setResulti的输入规则，该规则将寄存器i的内容设置为从环境的第一个元素Π得知的当前活动对象的字段结果;Π=（c，l）::Π'。

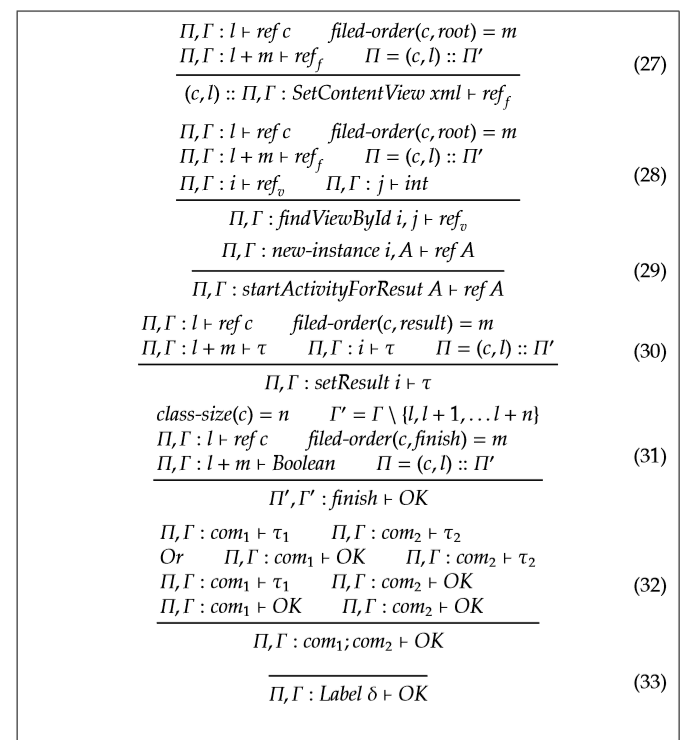


图4类型系统规则AndroidT（第3集）。

该规则调用方法字段顺序来确定字段的顺序，m，类c中的结果;字段顺序（c，结果）= m。 该规则还要求寄存器i和位置l + m具有相等的类型τ。 如果是这种情况，则规则得出结论，指令的类型是τ。

图5展示了提议的用于androidP的类型系统AndroidT的第四组推理规则。规则37说，在环境中为命令分配类型，Π'，Γ'：com⊢τ，保证在更大的环境中为命令分配相同的类型。

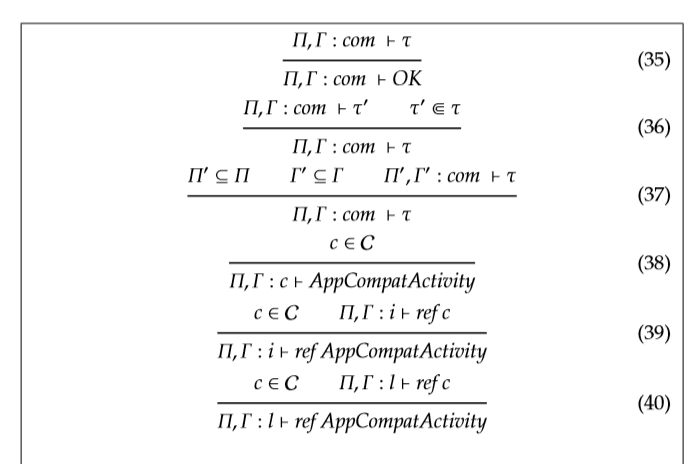


图5 类型系统AndroidT的推理规则（集4）。

例4推理规则41给出了例1和例3的结果应用规则32的一个例子。

(使用结构归纳)来证明以下关于类型系统AndroidT的事实并不难。

定理1 每个结构良好的AndroidP命令都可以在AndroidT中输入。

4 文献综述

安卓移动设备的使用吸引了大量的程序分析者的关注。应用动态监视、信息流分析、修改安卓系统等方法，被应用于揭示、研究、解决安卓应用程序的本质问题和细微结构缺陷[6]。

类型系统是研究一般程序，特别是安卓程序的主要工具之一。对于使用类型系统的安卓应用程序，可以进行各种分析和研究[26,13,23]。安卓应用程序的安全性是一个著名的例子[19,6,28,16]，使用类型系统来分析问题的例子。

在[19]中，对于安卓应用程序，为了分析静态数据流，提出了一种安全类型系统。这种分析被直接应用到了如何揭露隐私泄露里面。这种分析的目的是为了使用类型系统来检查安卓程序的代码是否违反了隐私协议[19]。

在[6]中，类型系统的概念被用来分析安卓应用程序，安卓API被用来进行组件间通讯的原因就是使用了理论计算。本文还提出了一种禁止特权的类型系统，这种思想被视作是防止程序中，类型良好的组件被攻击的保护方案。这项工作是为了生成一个类型检查器，Lintent[6]。

在[26]中提出了一种将安卓应用程序覆盖到Java类中的技术，这种转换包括为应用程序引入一种新的中间形式。本文中还用类型推理的概念来解决强约束问题，一些推理规则被用于转换成完整的DVM操作码集（超过200 个）[3]。本文提供了一种方式来处理无法验证的Dalvik字节码[26]。

在[28]中，有人宣称在安卓应用程序中对于资源的权限（像GPS，照相机，和网络连接）使用以应用为中心而非以资源为中心的方式，能够更好地服务于使用者和开发人员。例如，使用相机的权限必须经过应用程序的验证（例如，纸张扫描），并且，其他权限如网络连接也要被授予，这是为了从特定的服务器下载内容。在[28]中还声称，安卓早已配备了所需的机制来适应以应用为中心的精确而高效的策略结构。

在[13]中，还介绍了针对无干扰概念的的证明声音、DEX字节码类型系统和Dalvik虚拟机的操作语义。本文的目的是为了验证DEX字节码的抗干扰特性。此外，文章还介绍了从Java字节码到DEX字节码的抽象转换（即保持非干扰）。

在[17]中声称，尽管在申请审批的过程中已经付出了高昂的代价，但是一些移动应用商店还是让恶意软件侵入我们的移动设备。显然，应用商店必须采取强有力的机制来保证应用的真实性（而不是恶意的）。在[17]中，提出了一种实现该任务的验证技术。

针对安卓应用程序，在[16]中引入了一种用于污染分析的类型系统。DFlow，是一种被归类于数据流和上下文敏感型的类型系统。这个类型系统与类型推断的分析、机器引入联系在一起。这种类型系统的直接应用和分析是为了揭示隐私泄露，并且在[16]中，还介绍了处理安卓特性的方法，如组件间通讯，宏指令和许多程序的入口点。DFlowhe还使用CFL可达性的方法来分析存储端口的错误。

在[21]中，提出了在应用程序中对于私人信息（如位置和联系人）的访问控制还不够强大。在文中，提出了一种解密策略，通过用户与移动应用程序直接按的交互来控制私有信息的泄露。在应用程序的执行过程中通过依赖适当的事件序列，能够使策略独立于应用程序的实现细节中。

在[8]中，提出一种新的编程语言来帮助表述安卓本地策略和安卓进程间通信逻辑。这可以理解为对策略的高效的形式化的使用的一种尝试。这种建议的语言配备了一个作用域运算符，用于强制将本地策略应用到指定代码。这种技术也被应用于Web服务和面向对象编程[8]。

5，总结

本文中还提出了一个针对安卓应用程序的类型系统，AndroidT。这种类型系统是使用安卓编程的一个丰富的模型，AndroidP来开发的。该模型包括一套完整的Dalvik指令和一套重要的安卓宏指令机制。类型系统AndroidT 能够被用来证明一个安卓应用程序是否是允许类型错误的，例如“方法未定义”。该类型系统还能够用于证明安卓应用程序分析新技术的充分性和正确性。类型系统中使用的一组足够丰富的类型，可以承载其他类型，例如引用View，它是活动的主要组件，也是安卓应用程序的主要组件。

6.参考文献

1. Dalvik bytecode. https://source.android.com/ html.Accessed: 2016-02-1.

2. Dalvik docs mirror. http://www.statista.com/topics/840/smartphones/. Accessed: February 2016.

3. Dalvik docs mirror.http://www.milk.com/kodebase/dalvik-docs-mirror/. Accessed: February 2016.

4. Gartner, inc. worldwide traditional pc, tablet, ultramobile and mobile phone shipments. www.gartner.com/newsroom/id/2692318.

5. Domenico Amalﬁtano, Anna Rita Fasolino, Porﬁrio Tramontana, Salvatore De Carmine, and Atif M Memon. Using gui ripping for automated testing of android applications. In Proceedings of the 27th IEEE/ACM International Conference on Automated SoftwareEngineering, pages 258–261. ACM, 2012.

6. Michele Bugliesi, Stefano Calzavara, and Alvise Span`o. Lintent: towards security type-checking of android applications. In Formal Techniques for Distributed Systems, pages 289–304. Springer, 2013.

7. Luca Cardelli. Type systems. ACM Computing Surveys, 28(1):263–264, 1996.

8. GabrieleCosta. Securingandroidwithlocalpolicies. InProgrammingLanguageswith Applications to Biology and Security, pages 202–218. Springer, 2015.

9. Patrick Cousot. Semantic foundations of program analysis. In PrenticeHall, 1981.

10. Mohamed A El-Zawawy. An operational semantics for android applications. In Computational Science and Its Applications–ICCSA 2016.

11. Mohamed A El-Zawawy. Heap slicing using type systems. In Computational Science and Its Applications–ICCSA 2012, pages 592–606. Springer, 2012.

12. Mohamed A El-Zawawy. Recognition of logically related regions based heap abstraction. Journal of the Egyptian Mathematical Society, 20(2):64–71, 2012.

13. HendraGunadi. Formalcertiﬁcationofnon-interferentandroidbytecode(dexbytecode). In Engineering of Complex Computer Systems (ICECCS), 2015 20th International Conferenceon, pages 202–205. IEEE, 2015.

14. H.S.Karlsen and E.R.Wognsen. Static analysis of dalvik bytecode and reﬂection in android. Master’s thesis, Aalborg University, June 2012.

15. CuixiongHuandIulianNeamtiu. Automatingguitestingforandroidapplications. In Proceedings of the 6th International Workshop on Automation of Software Test, pages 77–83. ACM, 2011.

16. Wei Huang, Yao Dong, Ana Milanova, and Julian Dolby. Scalable and precise taint analysis for android. In Proceedings of the 2015 International Symposium on Software Testingand Analysis, pages 106–117. ACM, 2015.

17. Ren´e Just, Michael D Ernst, and Suzanne Millstein. Collaborative veriﬁcation of informationﬂowforahigh-assuranceappstore.InSoftwareEngineering&Management, page 77, 2015.

18. Aravind Machiry, Rohan Tahiliani, and Mayur Naik. Dynodroid: An input generationsystemforandroidapps.InProceedingsofthe20139thJointMeetingonFoundations of SoftwareEngineering, pages 224–234. ACM, 2013.

19. ChristopherMannandArtemStarostin. Aframeworkforstaticdetectionofprivacy leaks in android applications. In Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 1457–1462. ACM, 2012.

20. Zigurd Mednieks, Laird Dornin, G Blake Meike, and Masumi Nakamura. Programming Android. ” O’Reilly Media, Inc.”, 2012.

21. Kristopher Micinski, Jonathan Fetter-Degges, Jinseong Jeon, Jeﬀrey S Foster, and Michael R Clarkson. Checking interaction-based declassiﬁcation policies for androidusingsymbolicexecution. InComputerSecurity–ESORICS2015,pages520–538. Springer, 2015.

22. GregMiletteandAdamStroud. ProfessionalAndroidsensorprogramming. JohnWiley & Sons, 2012.

23. Martin Mohr, J¨urgen Graf, and Martin Hecker. Jodroid: Adding android support to a static information ﬂow control tool. In Software Engineering (Workshops), pages 140–145, 2015.

24. Kathryn E Newcomer, Harry P Hatry, and Joseph S Wholey. Handbook of practical programevaluation. John Wiley & Sons, 2015.

25. FlemmingNielson,HanneRNielson,andChrisHankin.Principlesofprogramanalysis. Springer, 2015.

26. Damien Octeau, Somesh Jha, and Patrick McDaniel. Retargeting android applications to java bytecode. In Proceedings of the ACM SIGSOFT 20th international symposium on the foundations of softwareengineering, page 6. ACM, 2012.

27. ´EtiennePayetandFaustoSpoto. Anoperationalsemanticsforandroidactivities. In Wei-Ngan Chin and Jurriaan Hage, editors, Proceedings of the ACM SIGPLAN 2014 workshop on Partial evaluation and program manipulation, PEPM 2014, January 20-21, 2014, San Diego, California, USA, pages 121–132. ACM, 2014.