پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

بهبود وارسی مدل با استفاده از نظریه تعبیر مجرد

نگارنده

پویا پرتو

استاد راهنمای اول: دکتر مجید علی‌زاده
استاد راهنمای دوم: دکتر مجتبی مجتهدی

پایاننامه برای دریافت درجه کارشناسی ارشد
در رشته علوم کامپیوتر

تاریخ دفاع

چکیده

تقدیم به

تقدیم به

سپاسگزاری

سپاسگزاری

پیشگفتار

فهرست مطالب

فصل ۱

مقدمه

با توجه به پیشرفت روز افزون علوم کامپیوتر و ورود کاربردهای آن به زندگی روزمره، پیشرفت در روش‌های ساخت و نگه‌داری برنامه‌ها نیازی آشکار به نظر می‌رسد. یکی از مسائل مهم در این زمینه بررسی صحت کارکرد برنامه‌های نوشته‌شده است. عدم صحت کارکرد برنامه‌های نوشته‌شده بسته به حساسیت کاری که یک برنامه انجام می‌دهد می‌تواند تبعات مختلفی داشته‌باشد. پرتاب ناموفق آریان ۵[۱۳] ، از مدار خارج شدن مدارگرد مریخ [۲] و تصادف هلیکوپتر چینوک [۱] چند نمونه از تبعات بزرگ این قضیه در گذشته بوده اند. برای کشف صحت کارکرد برنامه‌های کامپیوتری روش های متفاوتی ابداع شده‌اند که در ادامه به طور مختصر از آن‌ها یاد می‌کنیم اما پیش از آن به یک خاصیت مشترک همه ی این روش‌ها می‌پردازیم که ناکامل بودن است. منظور از ناکامل بودن این است که با استفاده از هیچ یک از روش‌هایی که داریم نمی‌توانیم هر خاصیتی را برای هر برنامه‌ای بررسی کنیم. به عبارت دیگر استفاده از هر روشی، محدودیت‌هایی دارد. و البته قضیه رایس [۱۶] به ما این تضمین را داده که روش کاملی اصلا وجود ندارد. قضیه رایس به طور غیر رسمی بیان می‌کند که مسآله ی بررسی هر خاصیت غیر بدیهی، برای همه ی برنامه‌ها تصمیم ناپذیر است. این دلیلی بر این شده که روش‌های مختلفی برای این کار درست شوند که هر کدام می توانند حالت‌های خاصی از مسأله را حل بکنند.

۱.۱ برخی از روش‌های درستی یابی برنامه‌ها

یک دسته بندی برای این روش‌ها به این شکل است که آن‌ها را به دو دسته ی پویا و ایستا تقسیم‌بندی می کند. روش‌های پویا روش‌هایی هستند که در آن‌ها تست برنامه با اجرای برنامه همراه است و روش‌های ایستا احتیاجی به اجرای خود برنامه ندارند. روش‌های پویا معمولاً با اجرای حالات محدودی از برنامه، تصمیم می‌گیرند که برنامه‌ای که نوشته‌ای ایم، انتظاراتمان را برآورده می‌کند یا خیر. اگر این روش بتواند تشخیص دهد برنامه‌ای درست کار نمی‌کند می‌توانیم با اطمینان نتیجه

بگیریم که برنامه غلط نوشته شده اما اگر برنامه‌ای، از تست های ساخته شده با این روش‌ها با موفقیت رد شود، نمی‌توان اطمینان حاصل کرد که برنامه درست کار بکند زیرا ممکن است حالتی از اجرای برنامه وجود داشته باشد که در تست ها نیامده باشد. در کتاب [۱۴] به توضیح این روش‌ها پرداخته شده. این دسته از روش‌ها از موضوع اصلی کار ما دور هستند. روش‌های ایستا معمولاً روش‌هایی هستند که از نظریه‌های مختلف در منطق ریاضی به عنوان ابزار بهره می‌برند تا بدون اجرای خود برنامه‌ها در مورد صحت اجرای آن‌ها نتیجه‌گیری کنند. به همین دلیل به این روش ها، روش‌های صوری هم گفته می‌شود که اصطلاح متداول تری است. از معروف ترین این روش‌ها وارسی گر مدل، روش‌های استنتاجی و استفاده از نظریه تعبیر مجرد است. در روش وارسی مدل، یک مدل صوری متناهی از برنامه ی مورد بررسی می‌سازیم که همه ی حالات اجرای برنامه با آن قابل توصیف است، سپس با استفاده از یک زبان صوری که بتواند در مورد مدل هایمان صحبت کند، ویژگی‌های مورد بررسیمان را بیان می‌کنیم و در نهایت صحت ویژگی‌های بیان شده را بررسی می کنیم. مقاله [۴] شروع این روش‌ها بوده که این کار را با استفاده از نوعی مدل کریپکی [۱۲] و نوعی منطق زمانی به نام منطق زمانی خطی [۴] انجام داده که روشی است با دقت و البته هزینه ی محاسباتی بسیار بالا. [۱۰] یک منبع بسیار مقدماتی در این زمینه و کتاب[۵] یک مرجع سنتی در این زمینه است. کتاب[۸] نیز می‌تواند یک مرجع برای مطالعه ی یک نظریه ی مرتبط به این موضوع باشد. در روش‌های استنتاجی که شاید بتوان ابتدایی ترین آن‌ها را استفاده از منطق هور[۹] دانست، درستی کارکرد برنامه هایمان را با ارائه ی یک درخت اثبات در یک دستگاه استنتاجی که متناسب با زبان برنامه هایمان ساخته شده، نشان می دهیم. در این روش هم اگر درستی یک برنامه را اثبات کنیم دیگر به طور تئوری خیالی آسوده از درستی برنامه خواهیم داشت اما ساختن درخت اثبات در یک نظریه برهان می‌تواند چالش برانگیز باشد چون این یک مسئله‌ی NP-Hard است. در[۱۰] به منطق هور به طور مقدماتی پرداخته شده. همین‌طور کتاب[۱۵] نیز به پیاده‌سازی منطق هور در زبان coq پرداخته. coq نیز یک اثبات‌یار است که بر اساس یک نظریه نوع وابسته کار می‌کند. برای اطلاعات بیشتر در مورد چگونگی طرز کار این اثبات‌یار و تئوری بنیادین آن می شود که کتاب[۳] مراجعه کرد. تئوری مورد شرح در[۸] نیز می‌تواند در این مسیر به کار گرفته شود. نظریه تعبیر مجرد[۷] نیز یک نظریه ریاضیاتی است که سعی می‌کند از روی معناشناسی یک برنامه ی کامپیوتری[۱۷] ، یک تقریب بسازد.منظور از تقریب، یک دستگاه کوچک‌تر از معناشناسی اصلی است که رفتارش زیرمجموعه ی رفتار های دستگاه اصلی است. سعی بر این است که دستگاه جدیدی که می‌سازیم به لحاظ محاسباتی هم ساده‌تر از معناشناسی اصلی کار کند تا بتوانیم خواص آن را راحت‌تر بررسی کنیم. در این صورت هر نتیجه‌ای که در مورد خواص جدید بگیریم را می‌توانیم در مورد خود برنامه هم بیان کنیم اما می‌دانیم که در این صورت هم به همه ی حقایق دست پیدا نکرده ایم. در مورد این نظریه نیز به تازگی کتاب[۶] منتشر شده که حاصل نزدیک به ۵ دهه کار مبدع این نظریه، پاتریک کوزو، است. همین‌طور[۱۱] نیز در مورد پیاده‌سازی این نظریه بحث کرده.

فصل ۲

عنوان فصل

فصل ۳

عنوان فصل

فصل ۴

عنوان فصل

فصل ۵

نتیجه گیری

واژه‌نامه فارسی به انگلیسی

واژه‌نامه انگلیسی به فارسی

Bibliography

[1] Committee to review chinook zd 576 crash. report from the select committee on chinook zd 576., Feb 2002.

[2] A. S. E. Al. Mars climate orbiter mishap investigation boord phase i report., November 1999.

[3] A. Chlipala. *Certified Programming with Dependent Types: A Pragmatic Introduction to Coq Proof Assistant*. MIT Press, 2022.

[4] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In D. Kozen, editor, *Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[5] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, London, Cambridge, 1999.

[6] P. Cousot. *Principals of Abstract Interpretation.* MIT Press, 2021.

[7] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL '77: Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252. ACM Press, 1977.

[8] D. Harel, D. Kozen, and J. Tiuryn. Dynamic logic. In *Handbook of philosophical logic*, pages 99–217. Springer, 2001.

[9] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[10] M. Huth and M. Ryan. *Logic in computer science : modelling and reasoning about systems*. Cambridge University Press, Cambridge [U.K.]; New York, 2004.

[11] X. R. K. Yi. *Introduction to Static Analysis: An Abstract Interpretation Perspective.* MIT Press, 2020.

[12] S. A. Kripke. A completeness theorem in modal logic1. *The journal of symbolic logic*, 24(1):1–14, 1959.

[13] J. Lions. Ariane 5 Flight 501 Failure: Report of the Inquiry Board, July 1996.

[14] G. J. Myers, C. Sandler, and T. Badgett. *The art of software testing*. John Wiley & Sons, Hoboken and N.J, 3rd ed edition, 2012.

[15] B. C. Pierce, A. Azevedo de Amorimand Chris Casinghino, M. Gaboardi, M. Greenberg, C. Hriţcu, V. Sjöberg, A. Tolmach, and B. Yorgey. *Programming Language Foundations*. Software Foundations series, volume 2. Electronic textbook, May 2018.

[16] H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):358–366, 1953.

[17] G. Winskel. *The formal semantics of programming languages - an introduction.* Foundation of computing series. MIT Press, 1993.

Abstract

Abstract goes here...

College of Science
School of Mathematics, Statistics, and Computer Science

Thesis Title

Author name

Supervisor: name
Co-Supervisor: name
Advisor: name

A thesis submitted to Graduate Studies Office
in partial fulfillment of the requirements for the degree of
B.Sc./Master of Science/Doctor of Philosophy in
Pure Mathematics/ Applied Mathematics/ Statistics/ Computer Science

yyyy