پیش‌گوی آزمون و کاربرد آن در اشکال‌زدایی نرم‌افزار

[۱]علیرضا خلیلیان، دانشجوی دکتری نرم‌افزار

[۲]مریم هاشم‌زاده، دانشجوی کارشناسی ارشد نرم‌افزار

[۳]احمد براآنی دستجردی، دانشیار مهندسی کامپیوتر

[۴]بهمن زمانی، استادیار مهندسی کامپیوتر

[۱، ۲، ۳، ۴]گروه مهندسی نرم‌افزار، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان

{[1]khalilian, [2]m.hashemzade, [3]ahmadb, [4]zamani}@eng.ui.ac.ir

چکیده: شرکت‌های نرم‌افزاری اکنون به‌درستی فهمیده‌اند که برای تضمین کیفیت نرم‌افزار از طریق آزمون و اشکال‌زدایی باید از فنون علمی بهره ببرند و از ابزارهای خودکار استفاده نمایند. یکی از ملزومات این‌کار این است که بدانیم اگر نرم‌افزار بی خطا باشد، موقع آزمون با دادن هر ورودی چه خروجی باید تولید کند. این نیازمندی همان مسئله پیش‌گوی آزمون است که خودکارسازی آن مسئله‌ایست حیاتی و دشوار و مورد نیاز صنعت و پژوهش‌های دانشگاهی. مقاله حاضر به اختصار در مورد این مسئله بحث می‌کند تا ابعاد آن بیشتر روشن گردد.

۱-۱- مقدمه

در بازار روبه رشد نرم‌افزار، تیم‌های توسعه نرم‌افزار با چالش‌های بسیاری مواجهند. متأسفانه، افزایش تقاضا برای نرم‌افزار و رقابت در صنعت نرم‌افزار منجر به کاهش زمان عرضه به بازار و افزایش احتمال عرضه نرم‌افزار معیوب می‌گردد. در نتیجه فعالیت‌های تضمین کیفیت نرم‌افزار اهمیت فوق‌العاده پیدا می‌کنند که در این میان آزمون نرم‌افزار در بهبود کیفیت و قابلیت اطمینان نرم‌افزار نقش حیاتی را ایفا می‌کند.

آزمون نرم‌افزار کاملاً در مقیاس عملی و صنعتی قابل استفاده است و هدفش شناسایی نواقص موجود در محصول نرم‌افزاری است. مشکل اینجاست که آزمون نرم‌افزار فرآیندی زمان‌گیر و پرهزینه است که بیش از ۵۰ درصد از هزینه‌های چرخه حیات توسعه نرم‌افزار را مصرف می‌کند. آزمون نرم‌افزار به دو صورت دستی و خودکار انجام می‌شود.

در آزمون دستی آزمون‌گران آزمون‌هایی را برای پیدا کردن خطاها و اشکال‌های نرم‌افزار طراحی می‌کنند. سپس آزمون‌گران انسانی که برای نرم‌افزار نقش کاربر نهایی را ایفا می‌کنند، تمام ویژگی‌های برنامه را بکار می‌گیرند و برنامه را اجرا می‌کنند که از رفتار صحیح نرم‌افزار اطمینان حاصل نماید. آزمون دستی مشکلاتی دارد که از عمده‌ترین آن‌ها می‌توان به هزینه و زمان بالا، دشواری و دقت پایین اشاره کرد. خودکارسازی فنون آزمون و اشکال‌زدایی در دو دهه اخیر کمک شایانی به حل مشکلات مذکور کرده است از این جهت که در مدت زمان کمتر و با هزینه کمتر می‌توان نرم‌افزار را از جنبه‌های بیشتری آزمود. حاصل چنین آزمون گسترده‌تر، یافتن نقص‌های بیشتر و افزایش بیشتر کیفیت نرم‌افزار است.

در حال حاضر فنون بسیار متعددی برای آزمون خودکار وجود دارد. مسئله این است که با وجود ابزارهای متعدد برای آزمون خودکار، هنوز آزمون را نمی‌توان کاملاً خودکار کرد زیرا بخشی از

ملزومات آن را نمی‌توان بطور خودکار به‌دست آورد. آزمون خودکار بـه اجـرای خودکـار و بررسـی نتایج به‌صورت خودکار نیاز دارد؛ یعنی فن خودکار باید بداند که خروجی منتظره برنامـه در قبـال اجرای هر ورودی چیست تا بتواند خروجی واقعی را با خروجی مورد انتظار مقایسه کند و تشخیص دهد برنامه اجرای درستی دارد یا خیر. خروجی منتظره بخشی از داده آزمون است.

داده آزمون شامل ورودی‌های مناسب و خروجی‌های مورد انتظـار اسـت کـه بـرای خودکارسـازی آزمون باید داده آزمون را نیز به‌طور خودکار تولید کنیم. ورودی‌های مناسب داده‌هایی هستند کـه معیارهای خاصی را برآورده سازند. این معیارها چنان هستند که به آزمون‌گران درصـدی اطمینـان دهند که برنامه در شرایط مختلف اجرای مطلوبی دارد. بنابراین تولید خودکار ورودی مناسب یکی از زمینه‌های تحقیقاتی فعال و مؤثر در خودکارسازی آزمون است. برای ایـن منظـور فنـون بسـیار سودمند و کارامدی طراحی شده است. اما مشکل فعلی تولید خودکار خروجـی‌هـای مـورد انتظـار است.

۱-۲- طرح مسئله

مسئله این است که بعد از اجـرای برنامـه بـا داده‌هـای ورودی، چگونـه درسـتی اجـرای برنامـه را تشخیص دهیم؟ در آزمون دستی، آزمون‌گران افرادی هستند که معمولاً در جریان تحلیل معنـایی نرم‌افزار قرار گرفته‌اند و با این دانش، خروجی برنامه را مشاهده می‌کنند و تشخیص می‌دهنـد کـه درست است یا خیر. اما اگر آزمون خودکار شود، خروجی‌های مورد انتظار هم بایـد خودکـار تولیـد شود و خودکار مقایسه شوند و حل این مسئله کار ساده‌ای نیست.

در ادبیات آزمون و اشکال‌زدایی نرم‌افزار به خروجی مورد انتظار، پیش‌گو یـا اوراکـل[۱] آزمـون مـی-گویند. اما در متون و مقالات، پیش‌گوی آزمون به‌طور گسترده‌تری هم استفاده می‌شود: خروجـی-های منتظره و الگوریتم و مکانیزم تولید آن‌ها و روش مقایسه خروجی واقعی و خروجـی منتظـره. فرآیند یافتن خروجی‌های منتظره صحیح و قابل اعتماد را مسئله‌ی پیش‌گو می‌نامند.

در مقایسه با بسیاری از جنبه‌های خودکارسازی آزمون، مسئله خودکارسـازی پـیش‌گـوی آزمـون کمتر مورد توجه قرار گرفته و یـک علـت آن البتـه بـه سـختی ذاتـی خـود مسـئله برمـی‌گـردد. خودکارسازی تولید پیش‌گوی آزمون باعث جهش مهمّی در سایر فنون خودکـار تضـمین کیفیـت

[۱] برای اینکه بیشتر فارسی بنویسیم و با نرم‌افزار مدیریت پایگاه داده‌ها اشتباه نگیریم، در این متن اوراکل آزمون را پـیش‌گـوی آزمـون خـواهیم گفت.

همچون آزمون‌های مختلف، مکان‌یابی اشکال‌ها و تعمیر خطاها می‌گردد.

یکی از عوامل تأثیرگذار بر دشواری مسئله پیش‌گوی آزمون، تنوع و پیچیدگی سیستم‌ها و بن-سازه‌های[1] نرم‌افزاری است. در حقیقت نرم‌افزارهای کاربردی امروز با زبان‌های برنامه‌سازی متعدد تولید می‌شوند و تولید پیش‌گوی آزمون برای هر زبان مسایل مخصوصی دارد. همچنین بسیاری از نرم‌افزارها با سرهم کردن و استفاده مجدد از کدهای آماده ساخته می‌شوند. اغلب اوقات این کدها در قالب واحدهای ترجمه شده در اختیار ما هستند و کد منبع آن‌ها در دسترس نیست. با توجه به اینکه نشان داده شده فن واقع‌بینانه برای تولید پیش‌گو نیاز به تحلیل کد منبع دارد، برای چنین کدهایی تولید پیش‌گو دشوار می‌شود و فنون مخصوصی را می‌طلبد. مشکل دیگر موقعی بروز می-کند که خروجی منتظره مقدار مشخصی نیست و دامنه‌ای از مقادیر همگی مجاز هستند. خیلی از اوقات خروجی‌های نرم‌افزار رشته‌های کاراکتری هستند با حالت‌های بسیار یا اصلا متنی نیستند و نمایش گرافیکی دارند. برای نرم‌افزارهای شبیه‌ساز اساساً خروجی قطعی وجود ندارد و برای ورودی مشخص چندین خروجی معتبرند. تولید خودکار پیش‌گوی آزمون برای این دسته از کاربردها فوق-العاده مشکل می‌شود.

۱-۳- راه‌حل‌ها و ارزیابی

برای تولید خودکار پیش‌گوی آزمون طبیعی است که نیاز به تحلیل کد منبع داشته باشیم. همچنین فن خودکار باید سعی کند به‌طریقی رفتار نرم‌افزار را یاد بگیرد و بداند نرم‌افزار چه می-کند تا بتواند پیش‌گوی آزمون را تولید کند. برای تحقیق این نیازمندی، چند سال اخیر اغلب فنون پیشنهادی از روش‌های یادگیری ماشینی استفاده کرده‌اند که شبکه عصبی تاکنون از مؤثرترین آن‌ها بوده است. شبکه عصبی یا هر روش یادگیری ماشینی تعدادی ورودی آزمون و خروجی منتظره را دریافت می‌کنند تا با آن‌ها رفتار برنامه را یاد بگیرد و مدلی ساخته شود. سپس با این مدل سعی می‌کنند برای سایر ورودی‌ها پیش‌گوی آزمون را پیش‌بینی و تولید نمایند. به‌همین دلیل اوزان سنجشی همچون دقت[2]، صحت[3] و بازیابی[4] و نرخ مثبت‌های کاذب و واقعی[5] مطرح می-شوند و پیشرفت در این حوزه با افزایش دادن دقت و صحتِ پیش‌بینی محقق می‌گردد. چون تولید

---

[1] Platform

[2] Precision

[3] Accuracy

[4] Recall

[5] False and true positives

پیش‌گوی آزمون برای هر نگارش برنامه فقط یک‌بار انجام می‌شود و حالت پیش پردازش دارد، بنابراین می‌توان از زمان و پیچیدگی محاسباتی فنون و الگوریتم‌های طراحی شده صرف‌نظر کرد و معیار مقایسه و برتری فنون را اوزان سنجش مذکور قرار داد.

با توجه به مشکلات عمده‌ای که در حل مسئله پیش‌گوی آزمون هست، امروزه هر فن مناسب در ابتدا فرض‌های محدود کننده بسیاری باید در نظر بگیرد تا مسئله آن‌قدر کوچک شود که بتوان آن‌را مهار و حل کرد. لذا پیدا کردن فن عمومی برای حل مسئله پیش‌گو حداقل تا چند سال آینده دور از دسترس است و فنون طراحی شده قطعاً خاص منظوره‌اند.

۱-۴- کاربرد

یکی از حوزه‌هایی که تولید خودکار پیش‌گوی آزمون برای آن‌ها سودمند است، نرم‌افزارهای موروثی هستند که عمدتاً به زبان C++ نوشته شده‌اند. برای نمونه شرکت گوگل حجم بسیاری از این نرم‌افزارها دارد که با هزینه بسیار تولید شده‌اند و هنوز کار می‌کنند. متأسفانه چنین برنامه-هایی هنوز خطادارند و کاربران مایلند که هنوز از این نرم‌افزارها استفاده کنند ولی تمایل دارند خطاهای آن‌ها هم رفع شود. اشکال‌زدایی چنین نرم‌افزارهایی هزینه بسیار دارد و ساده هم نیست چون اغلب ممکن است مستندات و توصیفاتی وجود نداشته باشد، در دسترس نباشد، مبهم و ناکامل باشد یا نویسنده نرم‌افزارها در دسترس نباشند. اینجاست که فنون خودکار تولید پیش‌گوی آزمون ارزشمند و کاربردی می‌شوند.

تمام بخش‌های فرایند آزمون و اشکال‌زدایی نرم‌افزار به پیش‌گوی آزمون نیاز دارند؛ پس آزمون و اشکال‌زدایی کاملاً خودکار زمانی واقعاً محقق می‌شود که این نیازمندی تأمین شود. تعمیر خطاهای نرم‌افزار آخرین و مهم‌ترین بخش اشکال‌زدایی است که خودکارسازی آن اساساً وابسته به تولید خودکار پیش‌گوست.

مراجع

[1] A. Orso and G. Rothermel, “Software Testing: A Research Travelogue (2000 – 2014),” In *Proceedings of the on Future of Software Engineering*, pp. 117–132. ACM, 2014.

[2] S. Anand, E. K. Burke, T. Y. Chen, J. Clark, M. B. Cohen, W. GriesKamp, M. Harman, M. J. Harrold, and P. McMinn, “An Orchestrated Survey of Methodologies for Automated Software Test Case Generation,” *J. Syst. Softw.*, vol. 86, no. 8, pp. 1978–2001, 2013.

[3] H. Kaur and G. Gupta, “Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and TestComplete,” *J. Eng. Res. Appl.*, vol. 3, no. 5, pp. 1739–1743, 2013.

[4] S. R. Shahamiri, W. M. N. W. Kadir, and S. Ibrahim, “A Single-network ANN-based Oracle to Verify Logical Software Modules,” *ICSTE 2010 - 2010 2nd Int. Conf. Softw. Technol. Eng. Proc.*, vol. 2, pp. 272–276. IEEE, 2010.

[5] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, “The Oracle Problem in Software Testing: A Survey,” *IEEE Trans. Softw. Eng.*, vol. 41, no. 5, pp. 507–525, 2015.

[6] M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, “A Comprehensive Survey of Trends in Oracles for Software Testing,”*Univ. Sheffield, Dep. Comput. Sci. Tech. Rep. CS-13-01,* pp. 1–32, 2013.

[7] Vineeta, A. Singhal, and A. Bansal, “Generation of Test Oracles Using Neural Network and Decision Tree Model,” 2*014 5th Int. Conf. - Conflu. Next Gener. Inf. Technol. Summit*, pp. 313–318, 2014.

[8] M. E. Yousif, S. R. Shahamiri, and M. B. Mustafa, “Test Oracles Based on Artificial Neural Networks and Info Fuzzy Networks: A Comparative Study,” *Proc. 2015 10th IEEE Conf. Ind. Electron. Appl. ICIEA 2015*, pp. 467–471, 2015.

[9] S. R. Shahamiri, W. M. N. W. Kadir, S. Ibrahim, and S. Z. M. Hashim, “An Automated Framework for Software Test Oracle,” *Inf. Softw. Technol.*, vol. 53, no. 7, pp. 774–788, 2011.

[10] P. A. Nardi and E. F. Damasceno, “Survey on Test Oracles,”*J. Adv. Theor. Appl. Informatics (ISSN 2447-5033)*, vol. 1, no. 1, pp. 50–59, 2015.

[11] B. P. Lamancha, M. Polo, D. Caivano, M. Piattini, and G. Visaggio, “Automated Generation of Test Oracles Using a Model-driven Approach,” *Inf. Softw. Technol.*, vol. 55, no. 2, pp. 301–319, 2013.