누구를 위한 정적 분석 도구인가

임세창

정적 분석 도구는 개발자와 인공지능 모두에게 필요하다. 인공지능 에이전트가 주도하는 코드 생성 시대에도, 기존 개발자 친화적인 도구의 원칙을 기반으로 하되, 인공지능 친화적인 데이터 표준화를 통해 코드 품질과 보안을 강화해야 한다. 결국, 인간과 인공지능 개발자를 구분하지 않고 모두에게 필요한 정적 분석 도구가 미래 소프트웨어 개발의 핵심이 될 것이다.

정적 분석 도구 (static analysis tool)는 점점 개발자들이 친숙하게 접할 수 있도록 그들에게 스며들고 있다. 하지만, 수많은 인공지능 (artifical intelligence) 관련 서비스들이 새롭게 개발되는 현대 사회에서 지금의 정적 분석 도구는 충분할까? 코파일럿 (Copliot)의 등장과 함께 수 많은 코드들이 새롭게 생성되고, 이에 따라서 취약한 코드들도 동시에 생성된다. 프로그램의 명세만 자세하게 짜주면 알아서 개발을 해주는 Devin AI와 같은 인공지능 에이전트도 생산성을 크게 높여주지만, 동시에 취약한 코드를 생성할 위험도 내포하고 있다. 이와 같은 인공지능 에이전트들의 결과물을 더 신뢰할 수 있게 하려면, 단순히 에이전트들의 발전만으로는 부족하며, 그들이 생성해낸 코드에 대한 정적 분석 도구와 기존 코드를 분석하여 더 나은 코드를 만들어낼 수 있는 도구의 발전도 함께 이루어져야 할 것이다.

아무리 좋은 도구를 만들더라도 사용하지 않으면 의미가 없다. 마찬가지로 아무리 우수한 정적 분석 도구라도 개발자가 실제로 활용하지 않으면 그 가치는 떨어진다. 구글의 Tricorder와 페이스북의 Zoncolan, Infer와 같은 정적 분석 도구들이 오탐 (false positive)을 줄이고, 코드 리뷰 (code review) 과정에 자연스럽게 정적 분석 결과를 통합하도록 노력했던 것도 바로 이러한 이유 때문이다.

구글과 페이스북의 정적 분석 도구는 단순히 코드의 오류를 지적하는 것에 그치지 않고, 개발자가 실제로 문제를 해결할 수 있도록 구체적인 분석 결과와 수정 제안을 제공했다. 구글의 Error Prone과 Tricorder는 컴파일 타임에 변수 사용 오류, 불필요한 코드, 스타일 위반 등 다양한 문제를 탐지하고, 자동 수정(diff) 제안을 통해 개발자가 빠르게 문제를 해결할 수 있도록 했다. 이들 도구는 문제의 유형과 심각도를 수치화하여, 개발자들이 어떤 문제에 우선적으로 대응해야 하는지 판단할 수 있도록 도왔다. 페이스북의 Infer와 Zoncolan은 인터프로시저 분석을 통해 복잡한 메모리 안전 문제, 동시성 오류 등의 버그 경로와 원인을 상세히 분석하여 제공한다. 또한, 이들은 각 버그의 발생 빈도와 수정 이력 등 통계적 데이터를 함께 제공해, 개발자들이 코드의 위험 요소를 체계적으로 관리할 수 있도록 지원한다. 분석 결과는 표준화된 데이터 형식이나 코드 리뷰 도구 내 자동 주석 형태로 전달되어, AI 기반 도구나 다른 자동화 시스템과의 연계에도 용이하다. 이렇듯, 구글과 페이스북의 정적 분석 도구는 개발자에게 필요한 정보들을 분석을 통해 지원하며 코드 품질과 보안을 획기적으로 향상시키는 데 기여해왔다.

만약 가까운 미래 혹은 먼 미래에 인공지능 개발자가 대부분의 코드를 작성하게 된다면, 기존의 구글과 페이스 북이 개발한 코드 리뷰 단계에서의 정적 분석 결과 통합 방식은 인공지능이 생성한 코드에는 적합하지 않을 수 있다. 기존 도구들은 인간 개발자가 작성하고 검토하는 환경에 최적화되어 있기 때문에, 인공지능이 생산하는 코드의 특성과 요구에 맞지 않을 수 있다. 인공지능이 생성한 코드를 코드리뷰해서 문제가 발생했을 때, 그것을 고치는 것 또한 인공지능이라면 애초에 인공지능 개발자에게 적합하도록 정적 분석 결과를 제공해주면, 더욱 품질이 좋은 코드를 생성했을 것이다.

그렇다면, 인공지능 개발자에게 적합한 정적 분석 도구는 무엇일까? 인간 개발자에게 제공되는 정적 분석 결과와는 다른 결과를 보여주는 것이 효과적일까? 아니면 인간 개발자에게 효과적인 정적 분석 도구가 인공지능 개발자에게도 동일하게 효과적일까? 아직은 명확하게 알기 어렵지만, 개발자에게 친숙한 정적 분석 도구를 만들어야 한다는 사실은 변하지 않을 것이다.

점점 인간 개발자와 인공지능 개발자 간의 경계가 모호해지고 있다. 오히려 평균 이하의 실력을 가진 개발자는 인공지능 개발자보다 못할 수도 있다. 이러한 상황에서는 정적 분석 기술이

인간뿐만 아니라 인공지능에게도 더욱 필요한 기술일 수 있다고 생각한다. 결국, 인간 개발자와 인공지능 개발자를 구분하지 않고 모두에게 필요한 정적 분석 도구를 만드는 것이 중요하다.