

## Sichere Produktlinien: Herausforderungen für Syntax- und Typ-Prüfungen

Christian Kästner  
Universität Magdeburg  
ckaestne@ovgu.de

Sven Apel  
Universität Passau  
apel@uni-passau.de

Gunter Saake  
Universität Magdeburg  
saake@ovgu.de

Softwareproduktlinien stellen neue Herausforderungen an Entwicklungswerkzeuge und Programmiersprachen. Während in der klassischen Anwendungsentwicklung genau ein Programm entwickelt wird, entwickelt man in einer Softwareproduktlinie gleich eine Vielzahl verwandter Programme in einer Domäne aus gemeinsamen, wiederverwendbaren Artefakten. Die Gemeinsamkeiten und Unterschiede der Programme einer Software-Produktlinie werden durch Features modelliert. Durch Kombination von Features können (potentiell tausende) maßgeschneiderte Programme spezifiziert und entsprechend generiert werden. Typische Implementierungsansätze die eine automatisierte Generierung von Programmen aus einer Produktlinie erlauben sind Frameworks, Präprozessoren (*#ifdef*), Generatoren oder spezielle Sprachen der Aspekt- oder Feature-Orientierten Programmierung.

Software-Produktlinien erlauben die Maßschneidung von Programmen auf bestimmte Anwendungsszenarien bei hoher Wiederverwendung, und versprechen somit schnellere Entwicklung von maßgeschneiderten Lösungen bei geringeren Kosten. Diesen Vorteilen stehen aber auch neue Herausforderungen gegenüber: eines der wichtigsten Probleme bei der Implementierung von Softwareproduktlinien ist, dass Fehler sehr schwierig zu finden sind wenn sie nur in einzelnen generierten Programmen mit bestimmten Features oder Feature-Kombinationen auftauchen. So schleichen sich Syntax-Fehler (z. B. vergessene schließende Klammer), Typ-Fehler (z. B. ins leere zeigende Methodenaufrufe oder fehlende Klassen), und auch Semantik-Fehler (z. B. fehlerhaftes Verhalten bei bestimmten Featurekombinationen) bei vielen Implementierungsansätzen leicht in einem generierten Programm ein. Oft werden diese Fehler aber erst bemerkt wenn ein Kunde tatsächlich – ggf. Monate nach der ursprünglichen Entwicklung – eine problematische Feature-Kombination anfragt. Das Generieren und isolierte Prüfen aller potentiellen Programme skaliert für Produktlinien nicht.

In dem Vortrag geben wir einen Überblick zu aktuellen Forschungen zum Erkennen von Syntax- und Typ-Fehlern in Produktlinien, und zwar für die gesamte Produktlinie anstatt nur für einzelne Programme:

- Syntax-Fehler können etwa beim Präprozessoreinsatz vermieden werden, wenn man gewisse, leicht überprüfbare Regeln einhält (und dass sogar sprachübergreifend).
- Typ-Fehler können mit einem speziellen Typsystem für Softwareproduktlinien die gesamte Produktlinie statt einzelner Programme prüfen. Dazu haben wir mit FFJ und CFJ zwei auf Featherweight Java basierende Calculi entwickelt (und entsprechende Tools für Java implementiert), die zum Quelltext auch Informationen zu Features der Produktlinie berücksichtigen und somit *garantieren* können dass aus einer wohlgetypten Produktlinie nur wohlgetypte Programme generiert werden können.

Zu diesen Konzepten stellen wir konkrete Tools und Sprachen zur Produktlinienentwicklung vor, welche diese Prüfungen unterstützen, sowie Erfahrungen aus verschiedenen Fallstudien.

Verwandte Publikationen:

- Christian Kästner and Sven Apel. *Type-checking Software Product Lines - A Formal Approach*. In Proc. International Conference on Automated Software Engineering (ASE), pages 258-267. IEEE Computer Society, September 2008.
- Christian Kästner, Sven Apel, Salvador Trujillo, Martin Kuhlemann, and Don Batory. *Guaranteeing Syntactic Correctness for all Product Line Variants: A Language-Independent Approach*. In Proc. TOOLS EUROPE. Springer, June 2009.
- Sven Apel, Christian Kästner, and Christian Lengauer. *Feature Featherweight Java: A Calculus for Feature-Oriented Programming and Stepwise Refinement*. In Proc. International Conference on Generative Programming and Component Engineering (GPCE), pages 101-112. ACM Press, October 2008.