Streszczenie

Rozprawa poświęcona jest badaniu *specyfikacji architekturalnych* — formalizmu, który służy do opisu modularnej struktury procesu budowy oprogramowania. Badania te pozostają w bliskim związku z językiem budowania specyfikacji Casl.

W pierwszej części pracy definiujemy i analizujemy pojęcie specyfikacji architekturalnej. Rozważamy między innymi różne odmiany semantyki tych specyfikacji. Badamy także stosunek między formalizacją procesu budowania oprogramowania dokonaną w sposób pośredni, na modelach (algebrach) danej logiki, a formalizacją dokonaną bezpośrednio na programach. Wprowadzamy pojęcie w pełni statycznej poprawności, które można stosować do języków specyfikacji mających własność amalgamacji. W pełni statyczna poprawność zapewnia, że proces budowania oprogramowania poprawny na poziomie modeli jest też poprawny na poziomie programów.

Druga część rozprawy poświęcona jest *weryfikacji* specyfikacji architekturalnych, tzn. dowodzeniu, że opisany przez taką specyfikację proces budowy oprogramowania jest poprawny. Wprowadzamy poprawny i (względnie) zupełny rachunek logiczny służący weryfikacji w pełni statycznych specyfikacji architekturalnych. Ponieważ język Casl nie ma własności amalgamacji, przedstawiamy także dwie poprawne i zupełne metody weryfikacji dla języków bez tej własności. Obie metody opierają się na reprezentacji interesującego nas języka specyfikacji w innym języku, posiadającym własność amalgamacji i spełniającym pewne dalsze warunki. W pierwszej metodzie zastępujemy pojęcie w pełni statycznej poprawności pojęciem tzw. rozszerzonej statycznej poprawności. Stosując tę metodę do języka Casl, korzystamy z jego reprezentacji w języku EnrichedCasl i dowodzimy, że język ten ma wymagane własności. Dowód ten wymaga m.in. uogólnienia wyników pochodzących z teorii amalgamacji dla półgrup na operady i multikategorie. W metodzie drugiej nie wymagamy spełnienia żadnych dodatkowych warunków statycznych; stosując tę metodę do języka Casl, korzystamy z jego reprezentacji w jego zubożonej o podrodzaje wersji.