CONTEXT-AWARE PROGRAMMING LANGUAGES

TOMAS PETRICEK

The development of programming languages needs to reflect important changes in the way programs execute. In recent years, this has included e. g. the development of parallel programming models (in reaction to the multi-core revolution) or improvements in data access technologies. This thesis is a response to another such revolution – the diversification of devices and systems where programs run.

The key point made by this thesis is the realization that an execution environment or a *context* is fundamental for writing modern applications and that programming languages should provide abstractions for programming with context and verifying how it is accessed.

We identify a number of program properties that were not connected before, but model some notion of context. Our examples include tracking different execution platforms (and their versions) in cross-platform development, resources available in different execution environments (e. g. GPS sensor on a phone and database on the server), but also more traditional notions such as variable usage (e. g. in liveness analaysis and linear logics) or past values in stream-based data-flow programming.

Our first contribution is the discovery of the connection between the above examples and their novel presentation in the form of calculi (*coeffect systems*). The presented type systems and formal semantics highlight the relationship between different notions of context. Our second and third contributions are two unified coeffect calculi that capture commonalities in the presented examples. In particular, our *flat coeffect calculus* models languages with contextual properties of the execution environment and our *structural coeffect calculus* models languages where the contextual properties are attached to the variable usage. We close the thesis with a *unified coeffect calculus* that captures the two notions using a single parameterized system.

Although the focus of this thesis is on the syntactic properties of the presented systems, we also discuss their category-theoretical motivation. We introduce the notion of an *indexed* comonad (based on dualisation of the well-known monad structure) and show how they provide semantics of the two coeffect calculi.