



Pattern Guards and Transformational Patterns

Martin Erwig and Peyton Jones Simon

Abstract

We propose three extensions to patterns and pattern matching in Haskell. The first, *pattern guards*, allows the guards of a guarded equation to match patterns and bind variables, as well as to test boolean condition. For this we introduce a natural generalisation of guard *expressions* to guard *qualifiers*.

A frequently-occurring special case is that a function should be applied to a matched value, and the result of this is to be matched against another pattern. For this we introduce a syntactic abbreviation, *transformational patterns*, that is particularly useful when dealing with views.

These proposals can be implemented with very modest syntactic and implementation cost. They are upward compatible with Haskell; all existing programs will continue to work.

We also offer a third, much more speculative proposal, which provides the transformational-pattern construct with additional power to explicitly catch pattern match failure.

We demonstrate the usefulness of the proposed extension by several examples, in particular, we compare our proposal with views, and we also discuss the use of the new patterns in combination with equational reasoning.
