Categorical design patterns

FOP Away Day, 17 Jan 2007
Ondřej Rypáček
University of Nottingham, UK
Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson, Vlissides) describe the “good” designs in OOP informal, ambiguous

Decorator Pattern: . . . The decorator conforms to the interface component it decorates so that its presence transparent to the component’s clients. The decorator forwards requests to the component may perform additional actions before or after forwarding. . .
Composite pattern
Formal Design Patterns

formal objects in the language
support reasoning about programs
replace inheritance and lots of hand-coding by formally
defined refinement steps
Category of “simple objects”

**Objects:** parametric object types (signatures)

**Arrows:** freely generated from *constructors*, *method calls*, *pairs*, *composition*

Objects interpreted directly, not via functional models and $\text{Set}$ [Reichel, Jacobs, Pierce, Hoffman]

Thanks to *Command* and *Visitor* patterns, the category has *exponents* and *co-products* (weakly) *terminal co-algebras* correspond to abstract object types and abstract methods
“Decorator is a singleton
Composite”
Composite pattern – formally

\[ 0 \xrightarrow{m} F \]
Composite pattern – formally

\[ C \cdot O \xrightarrow{C \cdot m} C \cdot F \cdot O \xrightarrow{\cdot \ast} F \cdot C \cdot O \]

\[ O \xrightarrow{m} F \cdot O \]
Composite pattern – formally
Decorator pattern – formally

\( C = I \)

\[
\begin{array}{c}
I_0 \xrightarrow{I_m} I \xrightarrow{F_0} F.I.O \\
\downarrow e \quad \downarrow \quad \downarrow F.c \\
0 \xrightarrow{m} F.O
\end{array}
\]
Conclusion and further steps

The approach is very promising
We already have some new results
natural interpretation of terminal co-algebras as abstract object-types
natural zips correspond to rearrangement of inputs and outputs in an object
discovered a relation between composite, decorator and adapter
formalised the relation of Composite and initial algebras – recursive structure traversals

Future: lot of work and more results