Marktoberdorf 2004

Towards Trusted Components

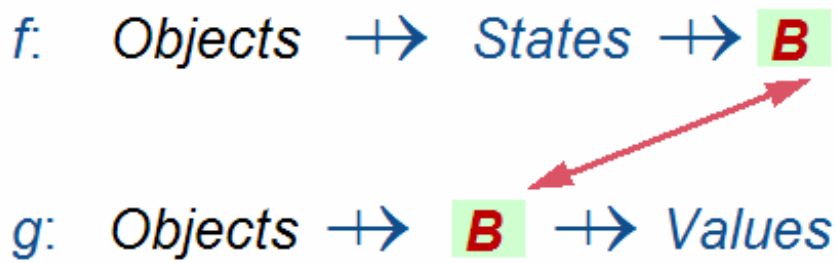
Bertrand Meyer

ETH, Zürich & Eiffel Software, California

Lesson 4: Proving classes

Example operator: rightmost composition

Consider functions



Then $f \square g$ is meaningless, but the following is defined:

function obj | $f\ (obj) \square g\ (obj)$

This will be written

$f \blacksquare g$

State-based composition

Consider functions

$$f: \quad A \xrightarrow{\sim} [States \xrightarrow{\sim} \mathbf{B}]$$

$$g: \quad \mathbf{B} \xrightarrow{\sim} [States \xrightarrow{\sim} C]$$

Then $f \square g$ is meaningless, but can compose f and g applied to a given state s.

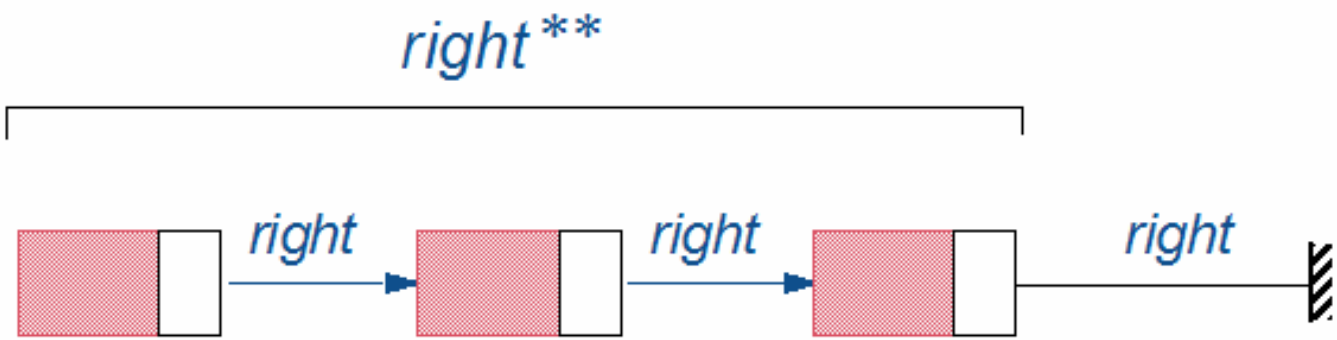
$f \cdot g$ is that composition, with signature

$$A \xrightarrow{\sim} [States \xrightarrow{\sim} \mathbf{C}]$$

and value

function x | [function s | [g ([$f(x)$] (s))] (s)]

Sequence closure



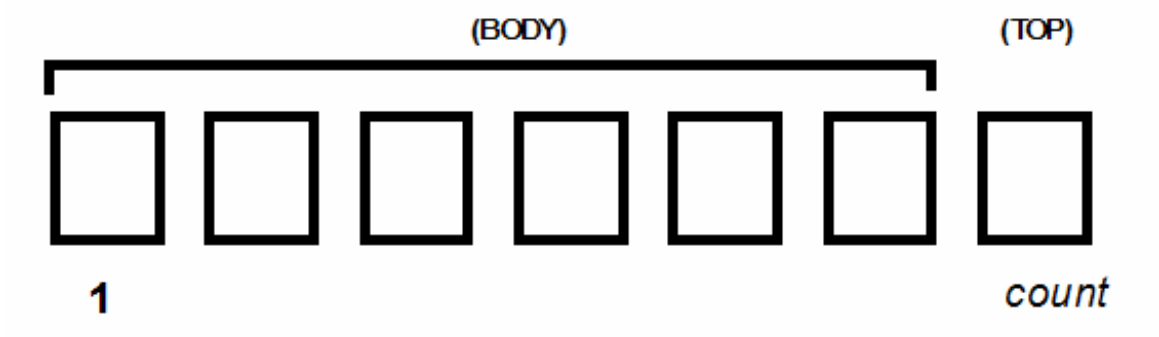
Using a model

Model library: *SET, RELATION, FUNCTION, TOTAL_FUNCTION...*

Totally applicative: functions only, no side effects, no assignments

In e.g. class *LIST[G]* and its descendants:

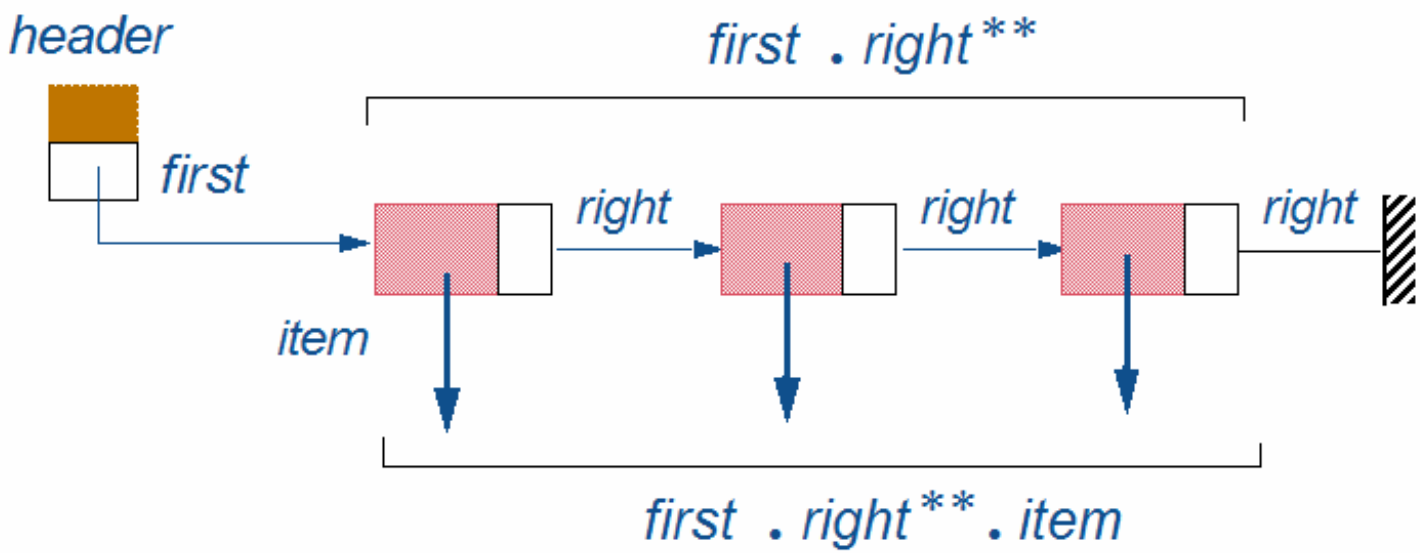
model: SEQUENCE[G]



Modeling linked lists

Composition and sequence closure:

Class invariant: *model = first . right** . item*



Modeling assignment

Function substitution:

$$f := g$$

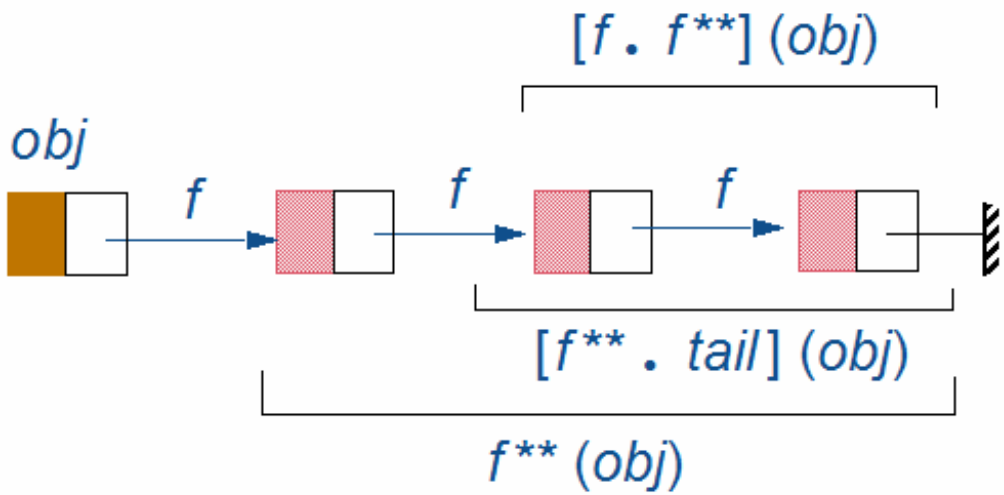
Denotes function in *Objects* $\rightarrow$ *States* $\rightarrow$ *States* such that, for any *obj*, *f* (*obj*) in resulting state is *g* (*obj*) (or undefined if *obj* not in domain of *g*).

Characteristic properties:

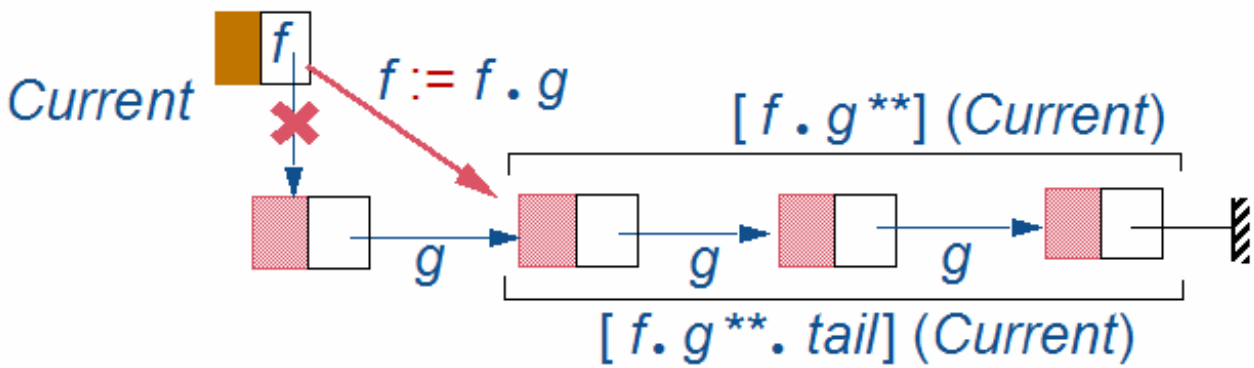
$$[f := g] \blacksquare f = g$$
$$[f := g] \blacksquare h = h \qquad \text{-- For } h \text{ other than } f$$

Static modeling: properties of operators

$f . f^{**} = f^{**} . tail$



$[f := f . g] \blacksquare [f . g^{**}] = f . g^{**} . tail$



The process

For each class in a group (one deferred, some effective):

- Devise a model
- Build a static theory
- Extend the contracts

In deferred class:

- Prove that abstract contracts imply model contracts

In each effective class:

- Make sure all loops have invariants
- Translate the class to mathematical form
- Prove that implementations satisfy model contracts