Programmer's task

Informally:

Given a requirements specification
produce a module that correctly implements it

Semantically:

Given a requirements specification SP
build a model $M \in |\mathbf{Mod}(Sig[SP])|$ such that
$M \in Mod[SP]$

Key idea

$$\boxed{SP \rightsquigarrow M}$$

Never in a single jump!

Rather: proceed step by step, adding gradually more and more detail and incorporating more and more design and implementation decisions, until a specification is obtained that is easy to implement directly

$$\boxed{SP_0 \leadsto SP_1 \leadsto \cdots \leadsto SP_n}$$

Refinement step

$$\boxed{SP' \leadsto SP}$$

Means:

$$Sig[SP'] = Sig[SP] \text{ and } Mod[SP] \subseteq Mod[SP']$$

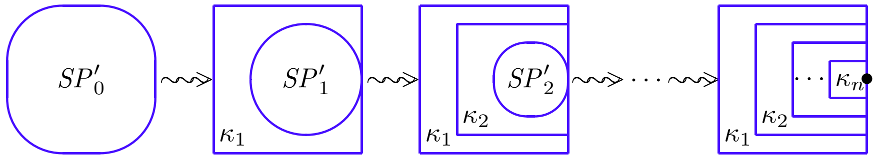
So:

- preserve the static interface (by preserving the signature)
- incorporate further details (by narrowing the class of models)

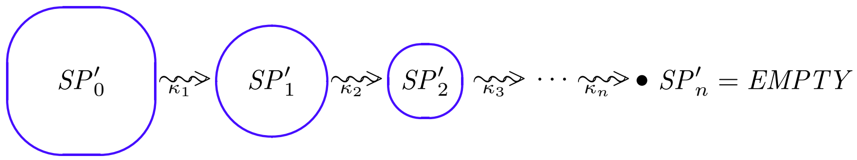
Fact:

$$\frac{SP_0 \leadsto SP_1 \leadsto \cdots \leadsto SP_n \qquad M \in Mod[SP_n]}{M \in Mod[SP_0]}$$

In practice, some parts will get fixed:



Keep them apart from whatever is really left for implementation:



Constructor refinement step

$$\boxed{SP' \underset{\kappa}{\rightsquigarrow} SP}$$

Means:

$$\kappa(Mod[SP]) \subseteq Mod[SP']$$

where

$$\kappa \colon |\mathbf{Mod}(Sig[SP])| \to |\mathbf{Mod}(Sig[SP'])|$$

is a *constructor*:

Intuitively: *parametrized module* (*functor* of STANDARD ML)

Semantically: function between model classes

Development process

Fact:

$$\frac{SP_0 \underset{\kappa_1}{\rightsquigarrow} SP_1 \underset{\kappa_2}{\rightsquigarrow} \cdots \underset{\kappa_n}{\rightsquigarrow} SP_n = EMPTY}{\kappa_1(\kappa_2(\ldots\kappa_n(empty)\ldots)) \in Mod[SP_0]}$$

Methodological issues:

- *top-down* vs. *bottom-up* vs. *middle-out* development?
- *modular decomposition* (designing modular structure)

Compositionality

Simple refinement

$$\frac{SP'' \rightsquigarrow SP' \qquad SP' \rightsquigarrow SP}{SP'' \rightsquigarrow SP}$$

??? $$\frac{SP'_1 \rightsquigarrow SP_1 \quad \cdots \quad SP'_n \rightsquigarrow SP_n}{\textbf{sbo}(SP'_1, \ldots, SP'_n) \rightsquigarrow \textbf{sbo}(SP_1, \ldots, SP_n)}$$???

$$\frac{SP'' \underset{\kappa'}{\rightsquigarrow} SP' \qquad SP' \underset{\kappa}{\rightsquigarrow} SP}{SP'' \underset{\kappa;\kappa'}{\rightsquigarrow} SP}$$

Constructor refinement

NO $$\frac{SP'_1 \underset{\kappa_1}{\rightsquigarrow} SP_1 \quad \cdots \quad SP'_n \underset{\kappa_n}{\rightsquigarrow} SP_n}{\textbf{sbo}(SP'_1, \ldots, SP'_n) \underset{???}{\rightsquigarrow} \textbf{sbo}(SP_1, \ldots, SP_n)}$$ NO