Dynamic dispatch may be seen as an abstract type of objects supporting two methods, creation of an object of a class, and sending a message to an object to obtain a result. The textbook describes an example with two classes of complex number, \texttt{cart} and \texttt{pol}, and two methods, \texttt{dist} and \texttt{quad}.

A dynamic dispatch scenario with classes \(C\) and methods \(D\) specified by instance types \(\tau_c\) for each \(c \in C\), and result types \(\rho_d\) for each \(d \in D\), may be organized as implementation of the following existential type:

\[
\tau_{dd} \triangleq \exists (t_{obj} . (\texttt{new} \hookrightarrow (\tau^c \rightarrow t_{obj})_{c \in C}, \texttt{snd} \hookrightarrow (t_{obj} \rightarrow \rho_d)_{d \in D})).
\]  

Given a package, call it \(x\), of this type, a client may open \(x\) to gain access to the object creation and message send operations as follows:

\[
\text{open } x \text{ as } t_{obj} \text{ with } (\texttt{new} \hookrightarrow \texttt{new}, \texttt{snd} \hookrightarrow \texttt{snd}) \text{ in } e_{\text{client}}
\]  

Within with expression \(e_{\text{client}}\) define

\[
\texttt{new}[c](e^c) \triangleq \texttt{new} \cdot c(e^c) \quad \texttt{snd}[e](d) \triangleq \texttt{snd} \cdot d(e)
\]

wherein \(e^c : \tau^c\) is instance data appropriate to class \(c\) and \(e : t_{obj}\). A new instance of class \(c\) is created by the \texttt{new} operation applied to an argument of type \(\tau^c\). It creates an abstract object of type \(t_{obj}\) to which one may send a message \(d\) to obtain a result of type \(\rho_d\). Importantly, the types \(\tau^c\) and \(\rho_d\) are independent of the abstract type \(t_{obj}\); their values are meaningful outside of the object abstraction. Moreover, the statics of the \texttt{open} expression ensures that the type of the client also be meaningful apart from \(t_{obj}\) so as to ensure that abstraction is not violated.

For example, in the case of the abstract type of complex numbers, write

\[
z \triangleq \texttt{new}[\texttt{cart}](\langle x \hookrightarrow x, y \hookrightarrow y \rangle)
= \texttt{new} \cdot \texttt{cart}(\langle x \hookrightarrow x, y \hookrightarrow y \rangle)
\]

to create a complex number \(z\) with rectangular coordinates \((x, y)\). Then write

\[
u \triangleq \texttt{snd}[\texttt{dist}](z)
= \texttt{snd} \cdot \texttt{dist}(z)
\]
to compute the squared distance of \(z\) from the origin, namely \(x^2 + y^2\).

Two natural implementations of the type \(\tau_{dd}\) arise, one by taking an object to be a tuple of methods, one for each method \(d \in D\), and one taking an object to be an instance datum labeled...
with some class $c \in C$. In the former case creating an object requires some work, but sending a message is simply a projection. In the latter case creating an object is simply an injection, but sending a message requires a case analysis. More precisely, two packages of type $\tau_{dd}$ are given by

$$\text{pack } \tau^I_{\text{obj}} \text{ with } (\text{new} \leftrightarrow e^I_{\text{new}}, \text{snd} \leftrightarrow e^I_{\text{snd}}) \text{ as } \tau_{dd}$$

and

$$\text{pack } \tau^II_{\text{obj}} \text{ with } (\text{new} \leftrightarrow e^II_{\text{new}}, \text{snd} \leftrightarrow e^II_{\text{snd}}) \text{ as } \tau_{dd}$$

whose components are defined as follows:

$$\tau^I_{\text{obj}} \triangleq (d \mapsto \rho_d)_{d \in D}$$
$$e^I_{\text{new}} \triangleq (\lambda (x^c : \tau^c) \langle d \mapsto e_{DM} \cdot c \cdot d(x^c) \rangle_{d \in D})_{c \in C}$$
$$e^I_{\text{snd}} \triangleq (\lambda (x : t_{\text{obj}}) x \cdot d)_{d \in D}$$

and

$$\tau^II_{\text{obj}} \triangleq [c \mapsto \tau^c]_{c \in C}$$
$$e^II_{\text{new}} \triangleq (\lambda (x^c : \tau^c) c \cdot x^c)_{c \in C}$$
$$e^II_{\text{snd}} \triangleq (\lambda (x : t_{\text{obj}}) \text{ case } x \{ c \cdot x^c \mapsto e_{DM} \cdot c \cdot d(x^c) \})_{c \in \rho_d, d \in D}.$$
Because the dynamics is deterministic, and from its definition, the relation $R$ respects evaluation in that $R(e^I, e^\Pi)$ if $e^I \mapsto^* e$, $e^\Pi \mapsto^* e'$ and $R(e, e')$.

This relation is preserved by the \texttt{new} and \texttt{snd} operations. More precisely, interpreting the type $t_{\text{obj}}$ by the relation $R$,

1. If $e_\text{c}^I =_{e_\text{c}} e_\text{c}^\Pi$, then $\text{new}^I \cdot c(e_\text{c}^I) =_{e_\text{c}} \text{new}^\Pi \cdot c(e_\text{c}^\Pi)$, i.e., $R(\text{new}^I \cdot c(e_\text{c}^I), \text{new}^\Pi \cdot c(e_\text{c}^\Pi))$.

2. If $e^I =_{e_\text{c}} e^\Pi$, i.e., $R(e^I, e^\Pi)$, then $\text{snd}^I \cdot d(e^I) =_{e_\text{c}} \text{snd}^\Pi \cdot d(e^\Pi)$

The variables $\text{new}^I$ and $\text{new}^\Pi$, and similarly annotated versions of $\text{snd}$, indicate the implementations in question.

Let us consider the verifications required.

1. By definition
   \begin{align*}
   \text{new}^I \cdot c(e_\text{c}^I) & \mapsto^* \text{new}^I(e_\text{c}^I) \mapsto^* \langle d \mapsto \text{DM} \cdot c \cdot d(e_\text{c}^I) \rangle_{d \in D},
   
   \text{and, similarly,}
   
   \text{new}^\Pi \cdot c(e_\text{c}^\Pi) & \mapsto^* c \cdot e_\text{c}^\Pi.
   \end{align*}

   For these to be related by $R$, it suffices to show for all $d \in D$,

   \[
   e_{\text{DM}} \cdot c \cdot d(e_\text{c}^I) =_{e_\text{c}} e_{\text{DM}} \cdot c \cdot d(e_\text{c}^\Pi).
   \]

   Now, by the parametricity theorem, the $(c, d)$ entry of the dispatch matrix is related to itself by equality at type $\tau_c \rightarrow \rho_d$. Because it is assumed that $e_\text{c}^I =_{e_\text{c}} e_\text{c}^\Pi$, the desired equation follows directly from the definition of equality at a function type.

2. By definition of the implementation
   \begin{align*}
   & \text{snd}^I \cdot d(e^I) \mapsto^* \text{snd}^I \cdot d(e^I) \mapsto^* \text{snd}^I \cdot d(e^I) \\
   \text{and}
   
   & \text{snd}^\Pi \cdot d(e^\Pi) \mapsto^* \text{snd}^\Pi \cdot d(e^\Pi) \mapsto^* \text{case} e^\Pi \{ c \cdot x^c \mapsto e_{\text{DM}} \cdot c \cdot d(x^c) \}_{c \in \rho_d}
   \end{align*}

   Because it is assumed that $R(e^I, e^\Pi)$, it follows that

   \[
   e_{\text{snd}}^I \cdot d(e^I) =_{e_\text{c}} e_{\text{d}}^I,
   \]

   and

   \[
   \text{case} e^\Pi \{ c \cdot x^c \mapsto e_{\text{DM}} \cdot c \cdot d(x^c) \}_{c \in \rho_d} =_{e_\text{c}} \text{case} c \cdot e_{\text{c}}^\Pi \{ c \cdot x^c \mapsto e_{\text{DM}} \cdot c \cdot d(x^c) \}_{c \in \rho_d},
   \]

   which completes the proof.

References