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, `cart` and `pol`, and two methods, `dist` and `quad`.

A dynamic dispatch scenario with classes `C` and methods `D` specified by instance types `τ_c` for each `c ∈ C`, and result types `ρ_d` for each `d ∈ D`, may be organized as implementation of the following existential type:

\[ τ_{dd} ≜ \exists (t_{obj} \cdot \langle \text{new} ↦ τ_c \rightarrow t_{obj} \rangle_{c ∈ C}, \text{snd} ↦ τ_d \rightarrow ρ_d \rangle_{d ∈ D}) \]  

(1)

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:

\[
\text{open } x \text{ as } t_{obj} \text{ with } \langle \text{new} ↦ \text{new}, \text{snd} ↦ \text{snd} \rangle \text{ in } e_{client}
\]

(2)

Within with expression `e_{client}` define

\[
\text{new}[c](e^c) ≜ \text{new} \cdot c(e^c) \quad \text{snd}[e](d) ≜ \text{snd} \cdot d(e)
\]

wherein `e^c : τ^c` is instance data appropriate to class `c` and `e : t_{obj}`. A new instance of class `c` is created by the `new` operation applied to an argument of type `τ^c`. It creates an abstract object of type `t_{obj}` to which one may send a message `d` to obtain a result of type `ρ_d`. Importantly, the types `τ^c` and `ρ_d` are independent of the abstract type `t_{obj}`; their values are meaningful outside of the object abstraction. Moreover, the statics of the `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

\[
z ≜ \text{new}[\text{cart}](\langle x ↦ x, y ↦ y \rangle)
= \text{new} \cdot \text{cart}(\langle x ↦ x, y ↦ y \rangle)
\]

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

\[
u ≜ \text{snd}[\text{dist}](z)
= \text{snd} \cdot \text{dist}(z)
\]

to compute the squared distance of `z` from the origin, namely `x^2 + y^2`.  

*Copyright © Robert Harper. All Rights Reserved.*
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

$$\text{pack } \tau_{obj}^I \text{ with } (\text{new} \hookrightarrow e_{\text{new}}^I, \text{snd} \hookrightarrow e_{\text{snd}}^I) \text{ as } \tau_{dd}$$

and

$$\text{pack } \tau_{obj}^II \text{ with } (\text{new} \hookrightarrow e_{\text{new}}^{II}, \text{snd} \hookrightarrow e_{\text{snd}}^{II}) \text{ as } \tau_{dd}$$

whose components are defined as follows:

$$\tau_{obj}^I \triangleq \langle d \hookrightarrow \rho_d \rangle_{d \in D}$$
$$e_{\text{new}}^I \triangleq \langle \chi (x^c : \tau^c) \langle d \hookrightarrow e_{DM} \cdot c \cdot d(x^c) \rangle_{d \in D} \rangle_{c \in C}$$
$$e_{\text{snd}}^I \triangleq \langle \chi (x^c : t_{obj}) x \cdot d \rangle_{d \in D}$$

and

$$\tau_{obj}^{II} \triangleq \langle c \hookrightarrow \tau^c \rangle_{c \in C}$$
$$e_{\text{new}}^{II} \triangleq \langle \chi (x^c : \tau^c) c \cdot x^c \rangle_{c \in C}$$
$$e_{\text{snd}}^{II} \triangleq \langle \chi (x : t_{obj}) \text{case } x \{ c \cdot x^c \hookrightarrow e_{DM} \cdot c \cdot d(x^c) \} \rangle_{c \in \rho_d} \langle d \hookrightarrow \rangle_{d \in D}.$$
to that instance. This suggests defining the binary relation $R$ between the two implementation types as follows:

$$R(e^I, e^{II}) \iff e^I \mapsto^* (e^I_d)_{d \in D}, \ e^{II} \mapsto^* c \cdot e^{II}_c, \ \text{and for all } d \in D, \ e^I_d = \rho_d \ e_{DM} \cdot c \cdot d(e^{II}_c).$$

Because the dynamics is deterministic, and from its definition, the relation $R$ respects evaluation in that $R(e^I, e^{II})$ iff $e^I \mapsto^* e, \ e^{II} \mapsto^* e'$ and $R(e, e')$.

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

1. If $e^I_c = \tau_c e^{II}_c$, then $new^I \cdot c(e^I_c) = \iota_{obj} new^{II} \cdot c(e^{II}_c)$, i.e., $R(new^I \cdot c(e^I_c), new^{II} \cdot c(e^{II}_c))$.

2. If $e^I = \iota_{obj} e^{II}$, i.e., $R(e^I, e^{II})$, then $snd^I \cdot d(e^I) = \rho_d \ snd^{II} \cdot d(e^{II})$.

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

Let us consider the verifications required.

1. By definition

$$new^I \cdot c(e^I_c) \mapsto^* e^I_{new}(e^I_c) \mapsto^* (d \mapsto e_{DM} \cdot c \cdot d(e^I_c))_{d \in D},$$

and, similarly,

$$new^{II} \cdot c(e^{II}_c) \mapsto^* c \cdot e^{II}_c.$$

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

$$e_{DM} \cdot c \cdot d(e^I_c) = \rho_d \ e_{DM} \cdot c \cdot d(e^{II}_c).$$

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 $e^I_c = \tau_c e^{II}_c$, the desired equation follows directly from the definition of equality at a function type.

2. By definition of the implementation

$$snd^I \cdot d(e^I) \mapsto^* e^I_{snd} \cdot d(e^I) \mapsto^* e^I_{snd} \cdot d(e^I)$$

and

$$snd^{II} \cdot d(e^{II}) \mapsto^* e^{II}_{snd} \cdot d(e^{II}) \mapsto^* \text{case } e^{II} \{ c \cdot x^c \mapsto e_{DM} \cdot c \cdot d(x^c) \}_{c \in \rho_d}$$

Because it is assumed that $R(e^I, e^{II})$, it follows that

$$e^I_{snd} \cdot d(e^I) \mapsto^* (e^I_d)_{d \in D} \cdot d \mapsto e^I_d$$

and

$$\text{case } e^{II} \{ c \cdot x^c \mapsto e_{DM} \cdot c \cdot d(x^c) \}_{c \in \rho_d} \mapsto^* \text{case } c \cdot e^I_c \{ c \cdot x^c \mapsto e_{DM} \cdot c \cdot d(x^c) \}_{c \in \rho_d} \mapsto e_{DM} \cdot c \cdot d(e^{II}_c)$$

and

$$e^I_d = \rho_d \ e_{DM} \cdot c \cdot d(e^{II}_c),$$

which completes the proof.

References