We give an operational semantics that relates terms to results in stores. The operational semantics is based on a global store. We say that a term b reduces to a result v to mean that, operationally, b yields v. Object terms reduce to object results consisting of sequences of store locations, one location for each object component. In order to stay close to standard implementation techniques, we avoid using formal substitutions during reduction. We describe a semantics based on stacks and closures. A stack associates variables with results; a closure is a pair of a method together with a stack that is used for the reduction of the method body. A store maps locations to method closures.

The operational semantics is expressed in terms of a relation that relates a store σ, a stack S, a term b, a result v, and another store σ'. This relation is written:

$$\sigma \bullet S \vdash b \rightsquigarrow v \bullet \sigma'$$

and it means that with the store σ and the stack S, the term b reduces to a result v, yielding an updated store σ' in the process; the stack does not change.

10.5.1 Reduction Rules

We now list the entities used in the operational semantics, the judgments that relate them to terms, and the reduction rules that manipulate them. We represent stacks and stores as finite sequences. In particular, the sequence $\iota_i \mapsto m_i \ ^{i \in 1..n}$ is the store that maps the location ι_i to the closure m_i, for $i \in 1..n$. We let $\sigma.\iota_j \leftarrow m$ denote the result of storing m in the location ι_j of σ, so if $\sigma \equiv \iota_i \mapsto m_i \ ^{i \in 1..n}$ and $j \in 1..n$ then $\sigma.\iota_j \leftarrow m \equiv \iota_j \mapsto m, \iota_i \mapsto m_i \ ^{i \in 1..n-\{j\}}$.

Operational semantics

ι	store location	(e.g., an integer)
$v ::= [l_i = \iota_i \ ^{i \in 1..n}]$	result	(l_i distinct)
$\sigma ::= \iota_i \mapsto \langle \varsigma(x_i) b_i, S_i \rangle \ ^{i \in 1..n}$	store	(ι_i distinct)
$S ::= x_i \mapsto v_i \ ^{i \in 1..n}$	stack	(x_i distinct)
$\sigma \vdash \diamond$	well-formed store judgment	
$\sigma \bullet S \vdash \diamond$	well-formed stack judgment	
$\sigma \bullet S \vdash a \rightsquigarrow v \bullet \sigma'$	term reduction judgment	

$$\frac{}{\emptyset \vdash \diamond}\ (\text{Store } \emptyset) \qquad \frac{\sigma \bullet S \vdash \diamond \quad \iota \notin dom(\sigma)}{\sigma, \iota \mapsto \langle \varsigma(x) b, S \rangle \vdash \diamond}\ (\text{Store } \iota)$$

$$\frac{\sigma \vdash \diamond}{\sigma \bullet \emptyset \vdash \diamond}\ (\text{Stack } \emptyset) \qquad \frac{\sigma \bullet S \vdash \diamond \quad x \notin dom(S) \quad \forall i \in 1..n}{\sigma \bullet (S, x \mapsto [l_i = \iota_i \ ^{i \in 1..n}]) \vdash \diamond}\ (\text{Stack } x) \ (l_i, \iota_i \text{ distinct})$$