

0.0.1 Associate At

The Primitive *associateAt* establishes a relationship between $k?_j$ and $v?$ at the nesting $k?_i \dots k?_{j-1}$ within Map $m!$

$$\frac{\begin{array}{l} k? = \langle k?_i \dots k?_j \rangle \\ (k?_j, m?_{k?}) \in m? \vee (k?_j, m?_{k?}) \notin m? \\ (k?_j, m?_{k?}) \notin m! \iff m?_{k?} \neq v? \\ (k?_j, v?) \in m! \end{array}}{m! = \text{associateAt}(m?, k?, v?)}$$

This implies that any existing mapping at $k?_j \in m?$ will be overwritten by *associateAt* but an existing mapping is not a precondition. The following helper Operation *getFirstKey* is introduced to establish navigation into a nested Map given a Collection of Keys.

$$\frac{\begin{array}{l} \text{GetFirstKey}[KV, \text{Collection}] \\ m?: KV \\ k?: \text{Collection} \\ v!: V \\ \text{getFirstKey}_- : KV \times \text{Collection} \rightarrow V \end{array}}{v! = \text{getFirstKey}(m?, k?) \bullet v! = \text{atKey}(m?, \text{head}(k?))}$$

This allows for the navigation into a nested Map to be defined as $\langle \text{getFirstKey}_-, \text{recur}_- \rangle^{\#k?-1}$ which represents a step down for each $k \in (k? \setminus k?_j)$. Once at $k?_{j-1}$, the mapped value v_{j-1} has $(k?_j, v?)$ added to it. This update is localized within $m?$ and all other mappings within $m?$ are left alone.

$$\frac{\begin{array}{l} \text{AssociateAt}[KV, \text{Collection}, V] \\ \text{GetFirstKey}, \text{Recur} \\ m?, m!: KV \\ k?: \text{Collection} \bullet \forall k?_n \in k?_{\langle i \dots j \rangle} \mid k_n : K \\ v?: V \\ \text{associateAt}_- : KV \times \text{Collection} \times V \rightarrow KV \end{array}}{\begin{array}{l} \text{associateAt} = \langle \langle \text{getFirstKey}_-, \text{recur}_- \rangle^{\#k?-1}, \langle \text{associateAt}_- \rangle^{\#k} \rangle \\ m! = \text{associateAt}(m?, k?, v?) \bullet \\ \quad \forall n : i \dots j - 1 \in \text{dom } k? \bullet j = \text{first}(\text{last}(k?)) \Rightarrow \text{first}(j, k?_j) \mid \exists_1 v_n \bullet \\ \quad \text{let } c_n == \text{tail}(k?)^{n-i} \\ \quad \quad v_i == \text{getFirstKey}(m?, c_n) \Rightarrow \\ \quad \quad \quad c_n = k? \iff n = i \bullet v_i = \text{atKey}(m?, \text{head}(k?)) \\ \quad \quad v_n = \text{recur}(v_i, c_n, \text{getFirstKey}_-)^{j-1} \\ \quad \quad v_{j-1} == \text{getFirstKey}(v_n, (j-1 \upharpoonright k?)) \iff n = j-2 \\ \quad \quad v_j == \text{associate}(v_{j-1}, \text{last}(k?), v?) \Rightarrow \langle \langle k?_j \mapsto v? \rangle \rangle \cup v_{j-1} \triangleleft k?_j \\ = (v_j \cup v_{n-\text{succ}(1)} \triangleleft k?_{n-\text{succ}(0)})^{j-1} \bullet n \leq j-1 \Rightarrow \\ \quad v_{j-4} \cup (v_{j-3} \cup (v_j \cup v_{j-2} \triangleleft k?_{j-1}) \triangleleft k?_{j-2}) \triangleleft k?_{j-3} \bullet \\ m! = \text{associate}(m?, k?_i, \text{associate}(v_i, k?_n, \text{associate}(v_n, k?_{j-1}, \text{associate}(v_{j-1}, k?_j, v?)))) \end{array}}$$

In the schema above, the localization of the change and the retention of the other mappings is indicated via

$$(v_j \cup v_{n-succ(1)} \triangleleft k?_{n-succ(0)})^{j-1} \bullet n \leq j-1 \Rightarrow \\ v_{j-4} \cup (v_{j-3} \cup (v_j \cup v_{j-2} \triangleleft k?_{j-1}) \triangleleft k?_{j-2}) \triangleleft k?_{j-3}$$

and is the reason why $\langle associate_ \rangle^{\#k}$ is included in the definition of $associateAt$ and is equivalent to

$$associate(m?, k?_i, associate(v_i, k?_n, associate(v_n, k?_{j-1}, associate(v_{j-1}, k?_j, v?))))$$

which gracefully walks into and back out of a KV regardless of $k?_n \in m?_{k?_{n-1}}$.

$$\begin{aligned} M &= \langle\langle k_i \mapsto v_i, k_n \mapsto \langle\langle k_{ni} \mapsto v_{ni}, k_{nj} \mapsto v_{nj} \rangle\rangle \rangle\rangle \\ associateAt(M, \langle k_n, k_{nn} \rangle, v?) &= \langle\langle k_i \mapsto v_i, k_n \mapsto \langle\langle k_{ni} \mapsto v_{ni}, k_{nj} \mapsto v_{nj}, k_{nn} \mapsto v? \rangle\rangle \rangle\rangle \\ associateAt(M, \langle k_j, k_{ji} \rangle, v?) &= \langle\langle k_i \mapsto v_i, k_n \mapsto \langle\langle k_{ni} \mapsto v_{ni}, k_{nj} \mapsto v_{nj} \rangle\rangle, k_j \mapsto \langle\langle k_{ji} \mapsto v? \rangle\rangle \rangle\rangle \\ associateAt(M, \langle k_i \rangle, v?) &= \langle\langle k_i \mapsto v?, k_n \mapsto \langle\langle k_{ni} \mapsto v_{ni}, k_{nj} \mapsto v_{nj} \rangle\rangle \rangle\rangle \\ associateAt(M, \langle k_i, k_{ii} \rangle, v?) &= \langle\langle k_i \mapsto \langle\langle k_{ii} \mapsto v? \rangle\rangle, k_n \mapsto \langle\langle k_{ni} \mapsto v_{ni}, k_{nj} \mapsto v_{nj} \rangle\rangle \rangle\rangle \end{aligned}$$

The last example demonstrates what happens when the value at some key is not a Map.

$$\begin{aligned} atKey(v_i, k_{ii}) = \emptyset &\iff k_{ii} \notin \text{dom } v_i \bullet \\ associateAt(M, \langle k_i, k_{ii} \rangle, v?) &\Rightarrow \\ associate(M, k_i, associate(\emptyset, k_{ii}, v?)) & \end{aligned}$$