```
(defn sample& [a dist k]
  (let [;; reuse previous value,
        ;; or sample from prior
        x (or (get-cache a)
              (sample dist))]
    ;; add to log-weight when reused
    (when (get-cache a)
      (add-log-weight! (observe dist x)))
    ;; store value and its log prob in trace
    (store-in-trace! a x dist)
    ;; continue with value x
    (k x)))

(defn observe& [dist value k]
  ;; Compute and record the log weight
  (add-log-weight! (observe dist value))
  ;; Call the continuation with no arguments
  (k))
```