$$
\begin{array}{ll}
\varsigma \in \Sigma = \mathsf{Call} \times \mathit{BEnv} \times \mathit{VEnv} \times \mathit{Time} & \hat{\varsigma} \in \hat{\Sigma} = \mathsf{Call} \times \widehat{\mathit{BEnv}} \times \widehat{\mathit{VEnv}} \times \widehat{\mathit{Time}} \\
\beta \in \mathit{BEnv} = \mathsf{Var} \rightharpoonup \mathit{Bind} & \hat{\beta} \in \widehat{\mathit{BEnv}} = \mathsf{Var} \rightharpoonup \widehat{\mathit{Bind}} \\
\mathit{ve} \in \mathit{VEnv} = \mathit{Bind} \rightharpoonup D & \widehat{\mathit{ve}} \in \widehat{\mathit{VEnv}} = \widehat{\mathit{Bind}} \to \hat{D} \\
d \in D = \mathit{Val} & \hat{d} \in \hat{D} = \mathcal{P}(\widehat{\mathit{Val}}) \\
\mathit{val} \in \mathit{Val} = \mathit{Clo} & \widehat{\mathit{val}} \in \widehat{\mathit{Val}} = \widehat{\mathit{Clo}} \\
\mathit{clo} \in \mathit{Clo} = \mathsf{Lam} \times \mathit{BEnv} & \widehat{\mathit{clo}} \in \widehat{\mathit{Clo}} = \mathsf{Lam} \times \widehat{\mathit{BEnv}} \\
b \in \mathit{Bind} \text{ is an } \textbf{infinite} \text{ set of bindings} & \hat{b} \in \widehat{\mathit{Bind}} \text{ is a } \textbf{finite} \text{ set of bindings} \\
t \in \mathit{Time} \text{ is an } \textbf{infinite} \text{ set of times} & \hat{t} \in \widehat{\mathit{Time}} \text{ is a } \textbf{finite} \text{ set of times}
\end{array}
$$

Fig. 1. State-space for the lambda calculus: Concrete (left) and abstract (right).

we can define the single concrete transition rule for CPS:

$$
\begin{aligned}
([\![(f\ e_1 \ldots e_n)^\ell]\!], \beta, \mathit{ve}, t) \Rightarrow (\mathit{call}, \beta'', \mathit{ve}', t'), \text{ where:}& \\
d_i &= \mathcal{E}(e_i, \beta, \mathit{ve}) \\
d_0 &= ([\![(\lambda^{\ell'}\ (v_1 \ldots v_n)\ \mathit{call})]\!], \beta') \\
t' &= \mathit{tick}(\mathit{call}, t) \\
b_i &= \mathit{alloc}(v_i, t') \\
\beta'' &= \beta'[v_i \mapsto b_i] \\
\mathit{ve}' &= \mathit{ve}[b_i \mapsto d_i].
\end{aligned}
$$

With the help of an abstract evaluator, $\hat{\mathcal{E}} : \mathsf{Exp} \times \widehat{\mathit{BEnv}} \times \widehat{\mathit{VEnv}} \to \hat{D}$:

$$
\begin{aligned}
\hat{\mathcal{E}}\ (v, \hat{\beta}, \widehat{\mathit{ve}}) &= \widehat{\mathit{ve}}(\hat{\beta}(v)) \\
\hat{\mathcal{E}}\ (\mathit{lam}, \hat{\beta}, \widehat{\mathit{ve}}) &= \left\{(\mathit{lam}, \hat{\beta})\right\},
\end{aligned}
$$

we can define an analogous transition rule for the abstract semantics:

$$
\begin{aligned}
([\![(f\ e_1 \ldots e_n)^\ell]\!], \hat{\beta}, \widehat{\mathit{ve}}, \hat{t}) \leadsto (\mathit{call}, \hat{\beta}'', \widehat{\mathit{ve}}', \hat{t}'), \text{ where:}& \\
\hat{d}_i &= \hat{\mathcal{E}}(e_i, \hat{\beta}, \widehat{\mathit{ve}}) \\
\hat{d}_0 &\ni ([\![(\lambda^{\ell'}\ (v_1 \ldots v_n)\ \mathit{call})]\!], \hat{\beta}') \\
\hat{t}' &= \widehat{\mathit{tick}}(\mathit{call}, \hat{t}) \\
\hat{b}_i &= \widehat{\mathit{alloc}}(v_i, \hat{t}') \\
\hat{\beta}'' &= \hat{\beta}'[v_i \mapsto \hat{b}_i] \\
\widehat{\mathit{ve}}' &= \widehat{\mathit{ve}} \sqcup [\hat{b}_i \mapsto \hat{d}_i].
\end{aligned}
$$