Tutorial: Small-step CFA

Matt Might
University of Utah
matt.might.net

Why small steps?

"Read my diss."

-Olin

“Fix Chapter 8.”

-Olin

I’m not smart enough.

The Law of Jones

Jones

Jones, 2006

Jones, 2006-25

(Jones, 1981)

Today

- Small steps
- Using CPS
- Using ANF

Today

- *k*-CFA
- Advances

Small-step analysis?

Small-step semantics

Small-step semantics

Small-step semantics

- Convert program e into machine state s_0

$$
\begin{tikzcd}
e \arrow[d] \\
s_0
\end{tikzcd}
$$

Small-step semantics

- Convert program e into machine state s_0
- Transition from state s_n to state s_{n+1}

$$\begin{tikzcd}
e \arrow[d] \\
s_0 \arrow[r] & s_1 \arrow[r] & s_2 \arrow[r] & s_3 \arrow[r] & s_4 \arrow[r] & \ldots
\end{tikzcd}$$

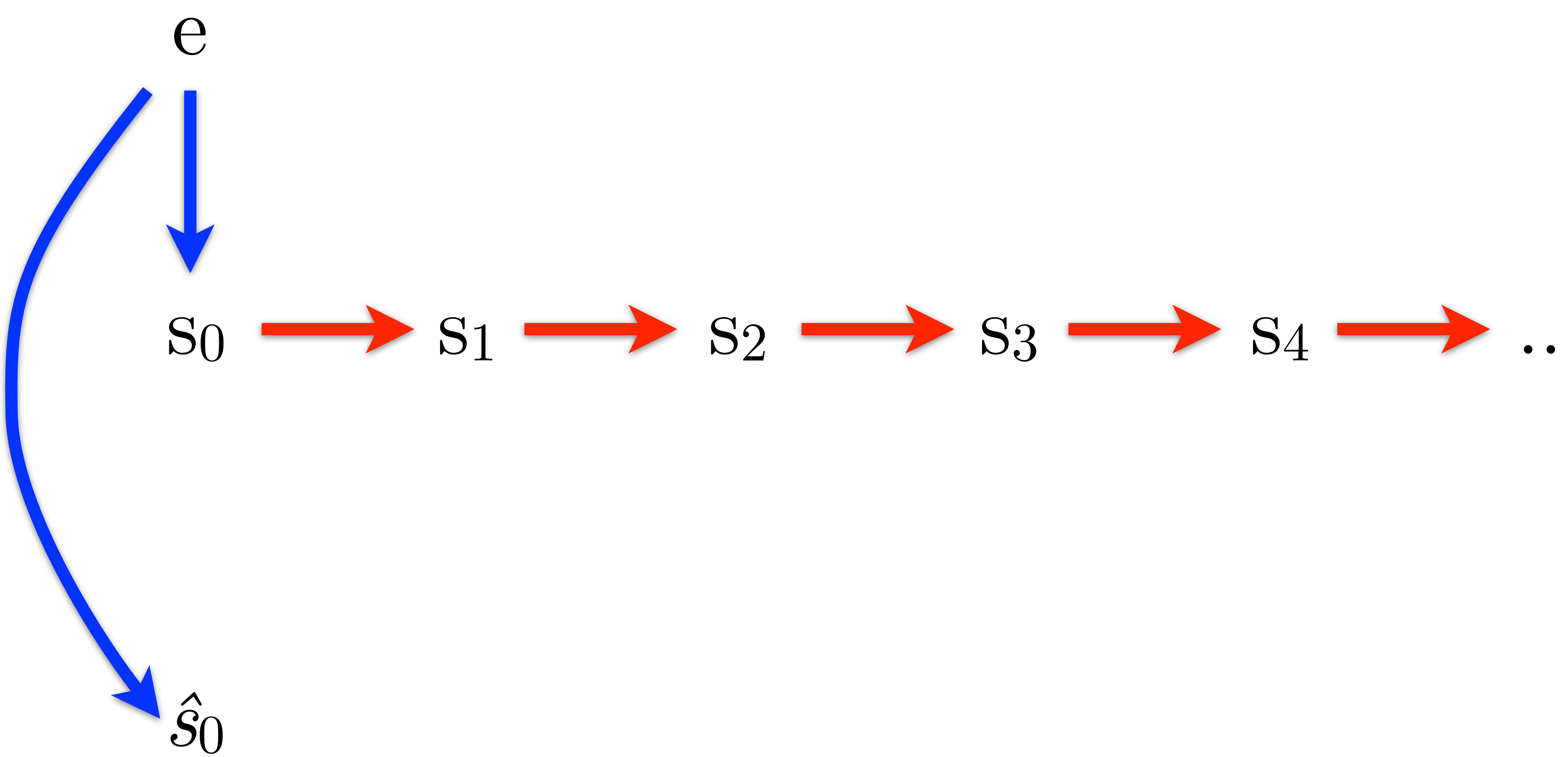
Abstract semantics

$$\begin{tikzcd}
e \arrow[d] \\
s_0 \arrow[r] & s_1 \arrow[r] & s_2 \arrow[r] & s_3 \arrow[r] & s_4 \arrow[r] & \dots
\end{tikzcd}$$

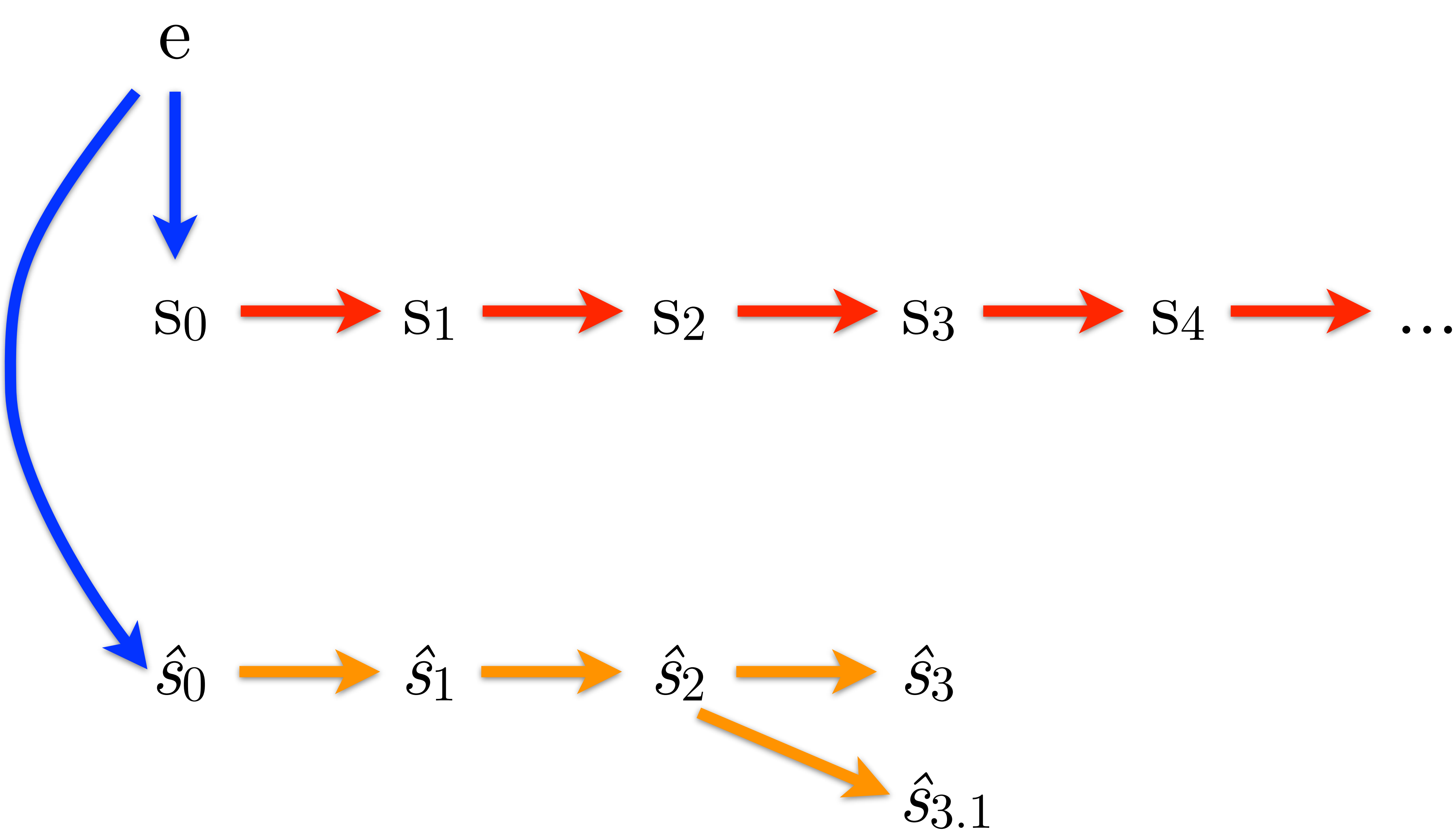
Abstract semantics

$$\begin{tikzcd}
\mathrm{e} \arrow[d] & & & & & \\
\mathrm{s}_0 \arrow[r] & \mathrm{s}_1 \arrow[r] & \mathrm{s}_2 \arrow[r] & \mathrm{s}_3 \arrow[r] & \mathrm{s}_4 \arrow[r] & \ldots
\end{tikzcd}$$

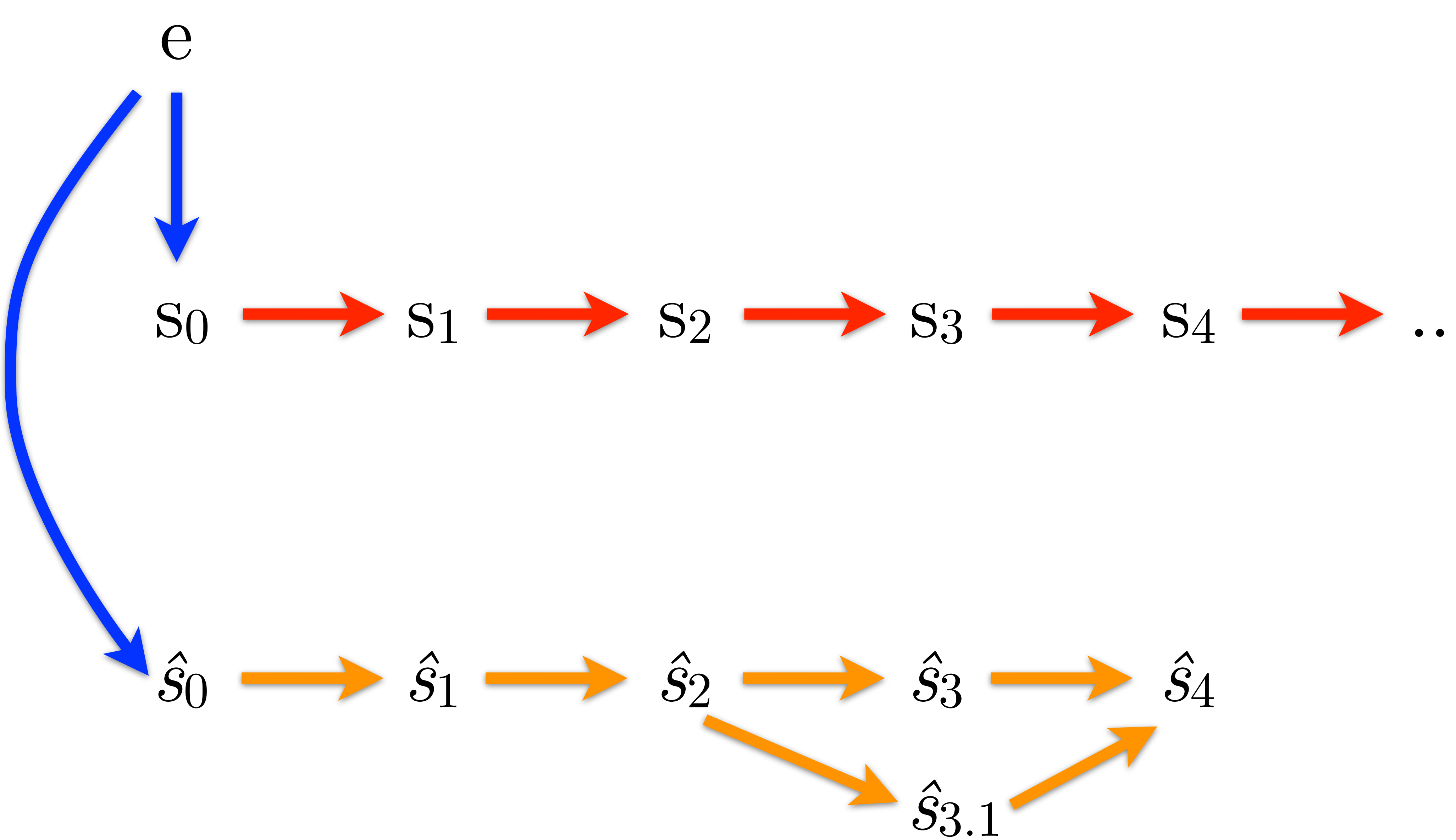
Abstract semantics



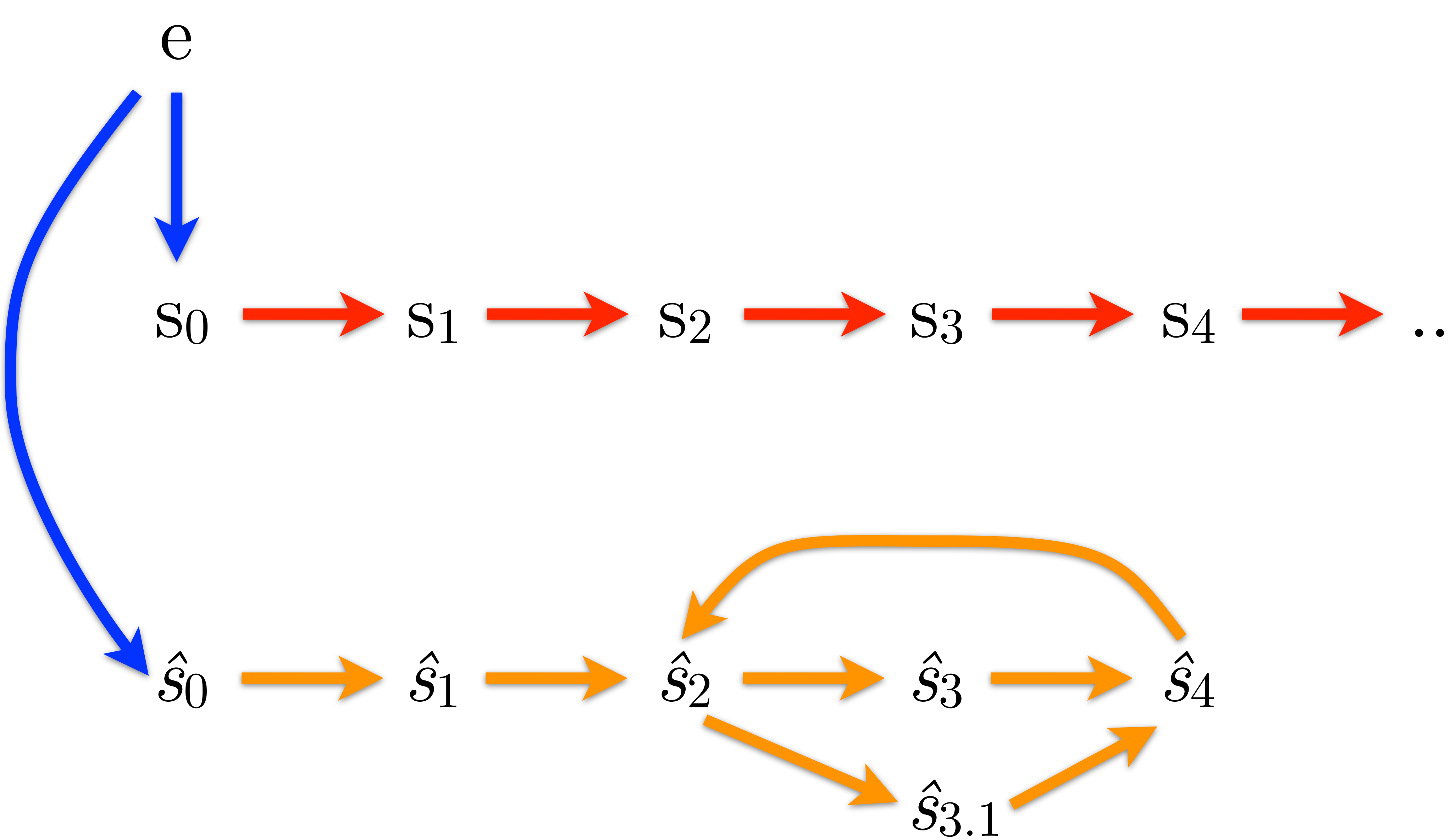
Abstract semantics



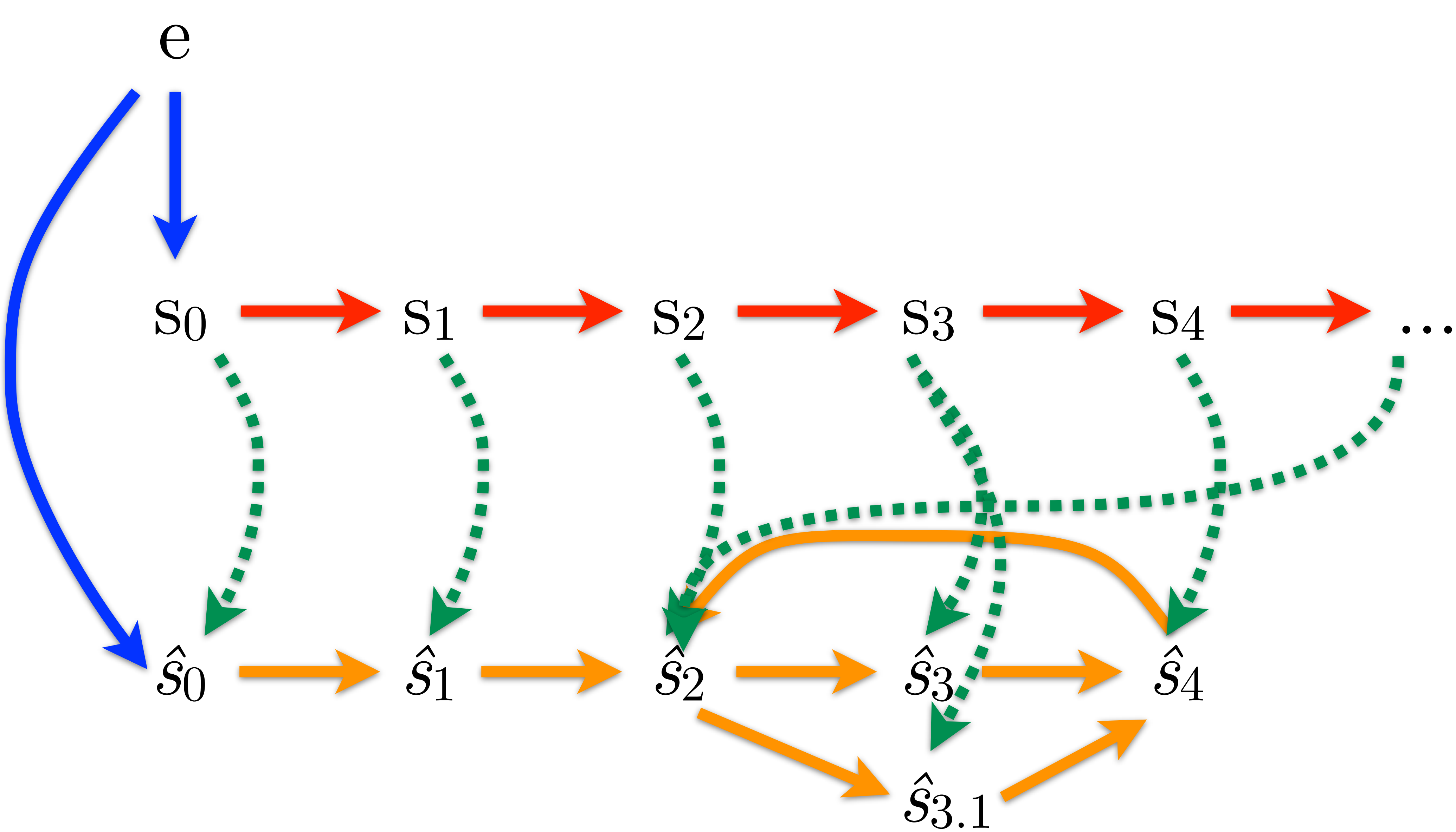
Abstract semantics



Abstract semantics

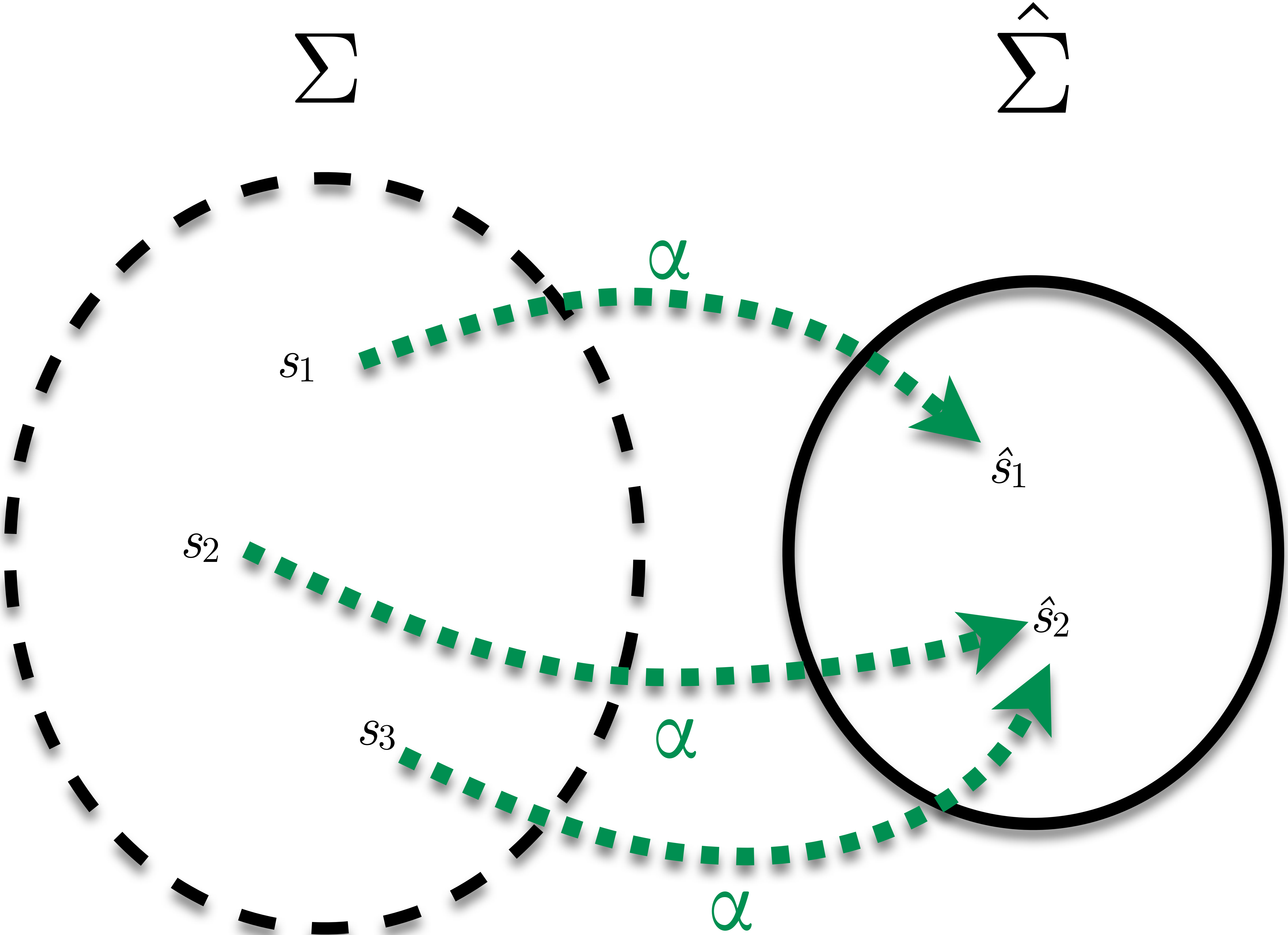


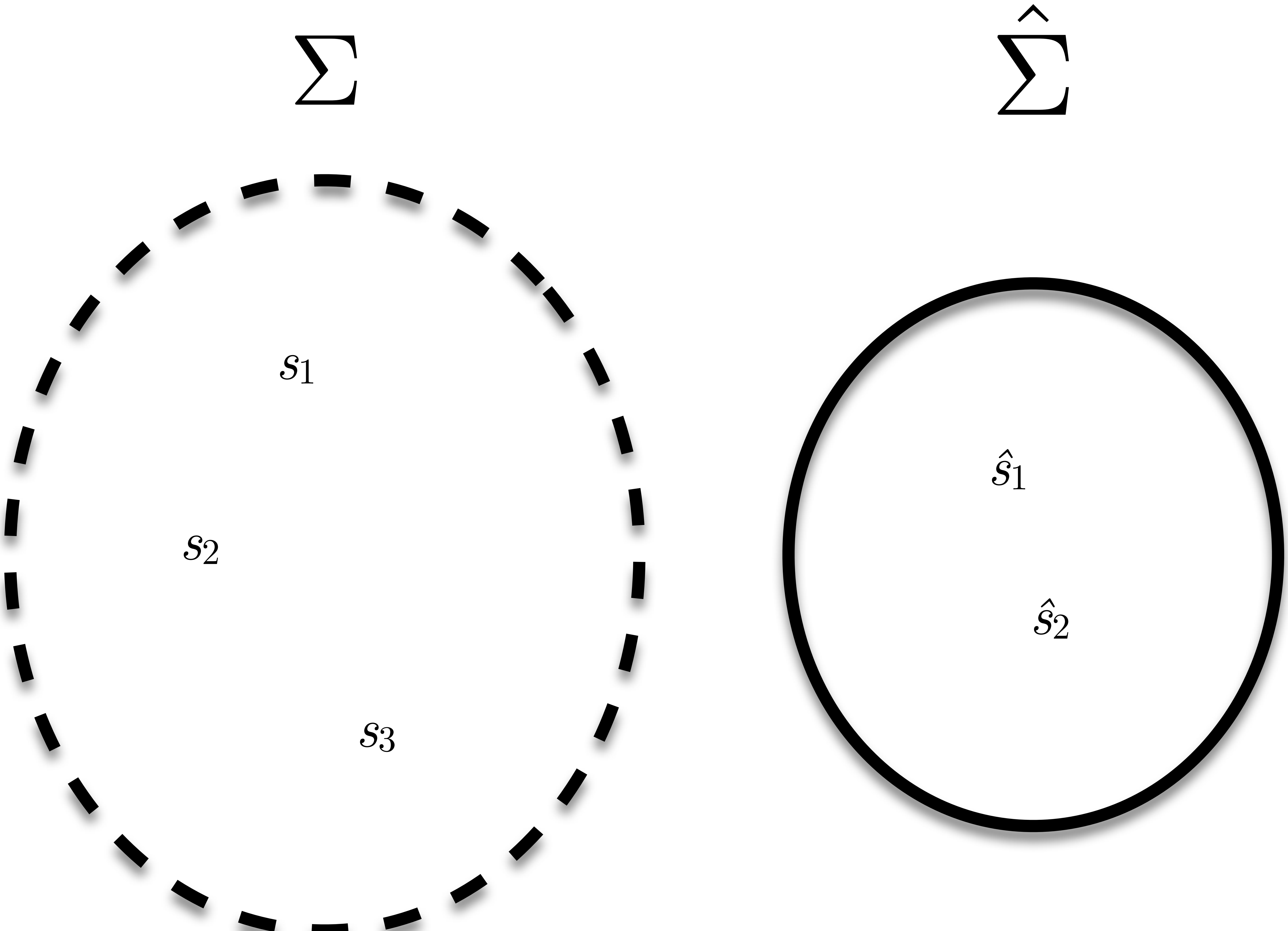
Abstract semantics

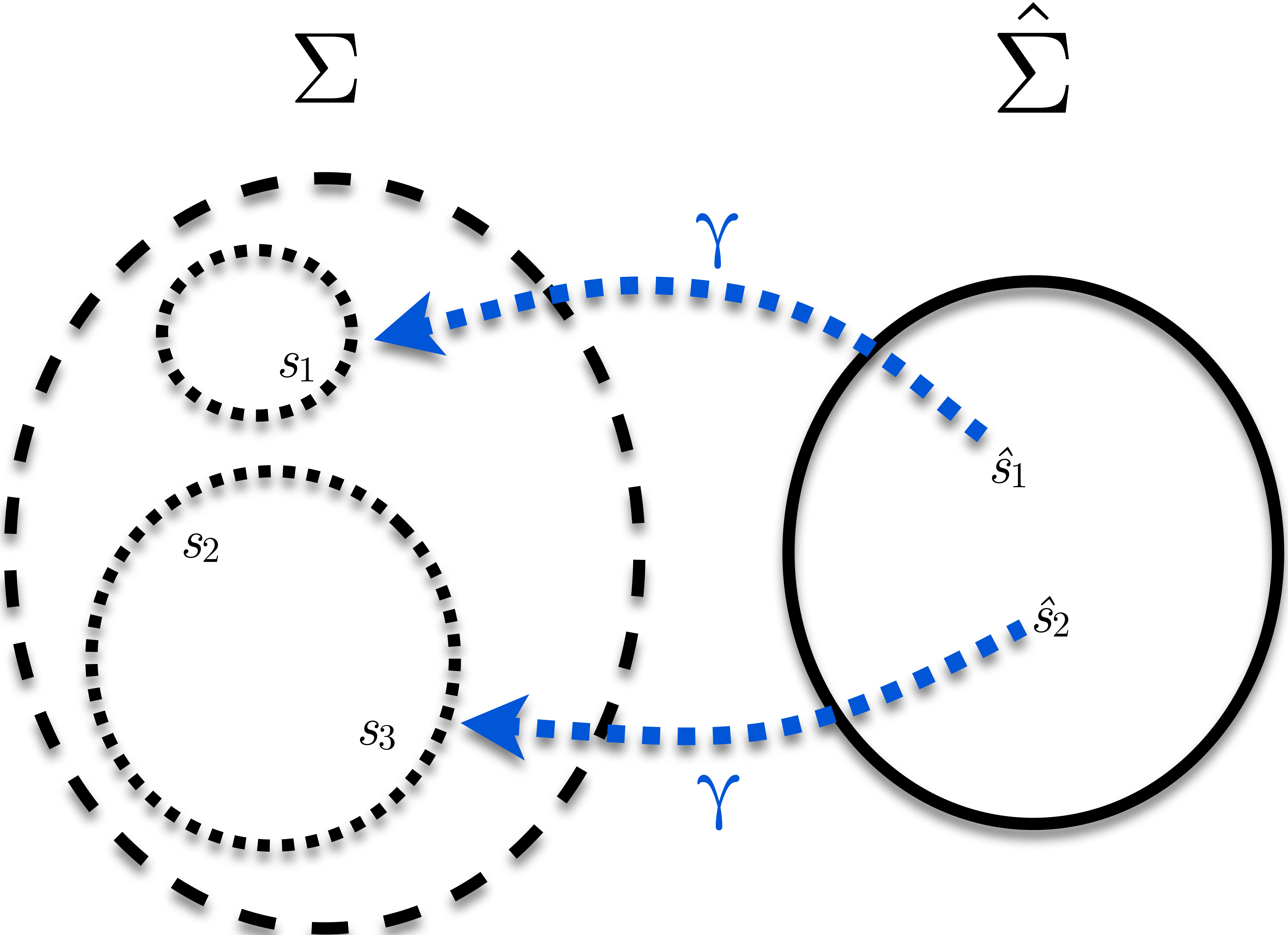


Theorem: The abstract simulates the concrete.

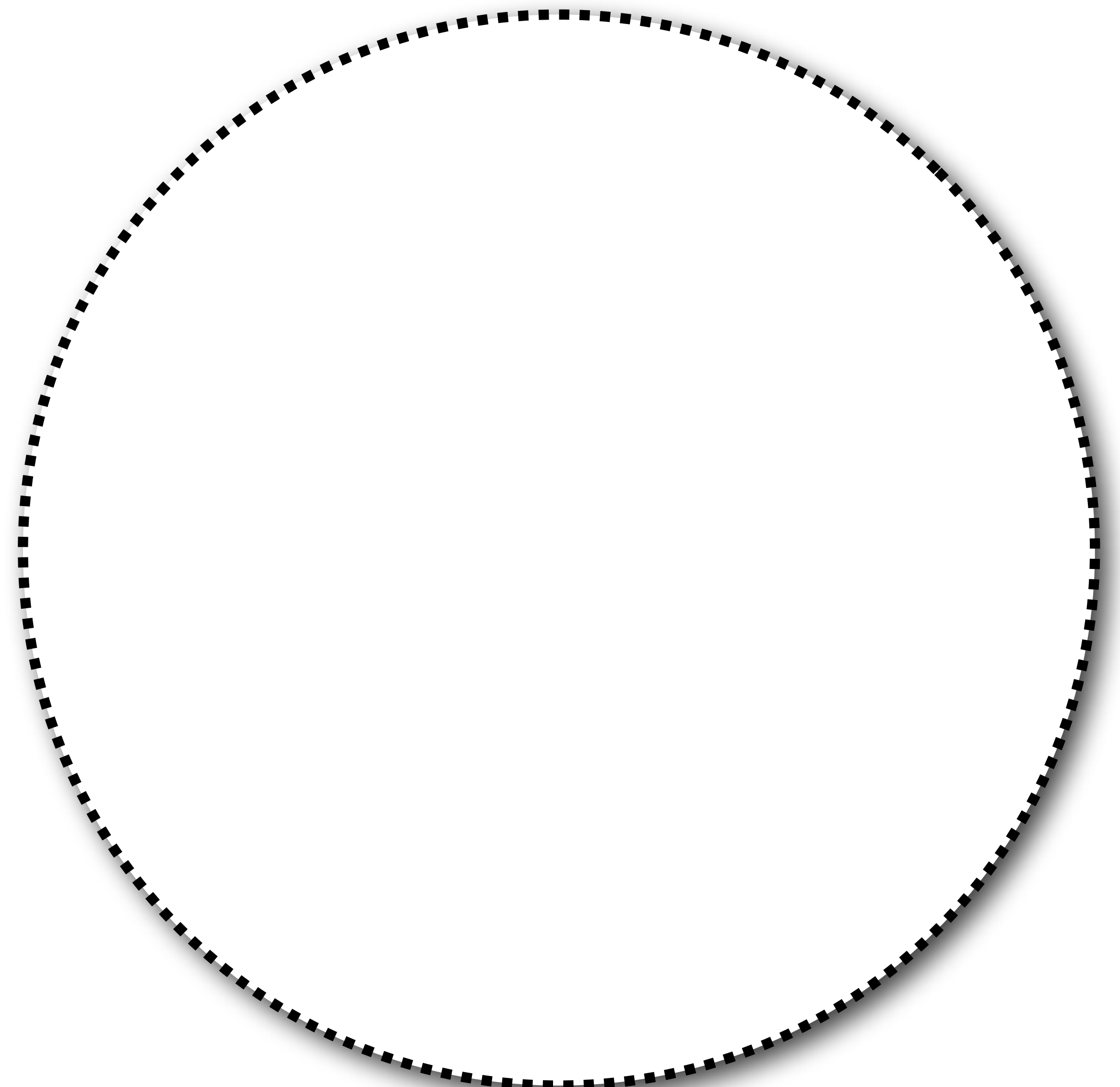
Σ $\hat{\Sigma}$


Σ
$\hat{\Sigma}$
α
s_1
$\hat{s}_1$
s_2
$\hat{s}_2$
α
s_3
α


Σ
$\hat{\Sigma}$
s_1
s_2
s_3
$\hat{s}_1$
$\hat{s}_2$

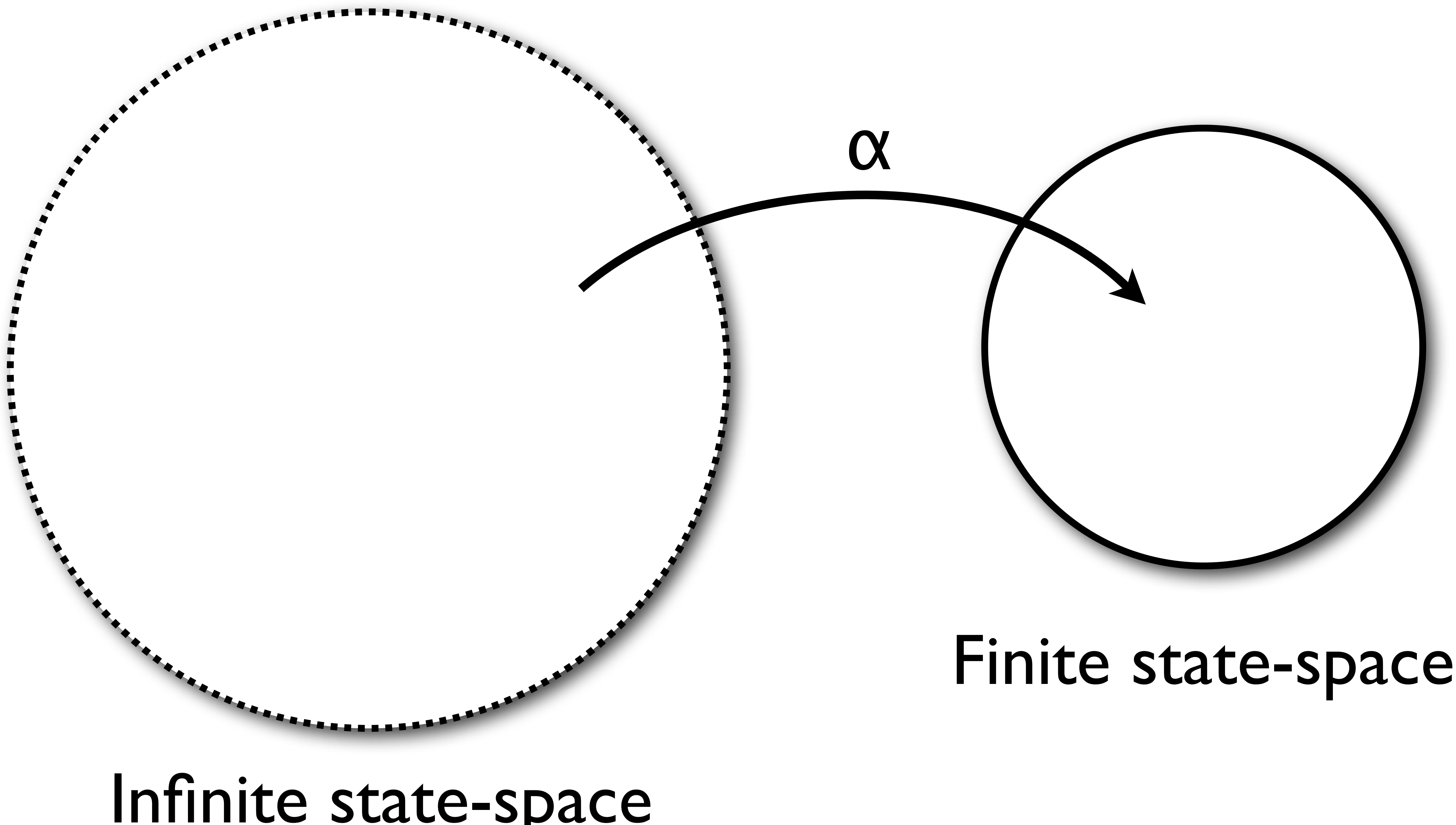

Σ
$\hat{\Sigma}$
γ
s_1
$\hat{s}_1$
s_2
$\hat{s}_2$
s_3
γ

Small-step analysis...



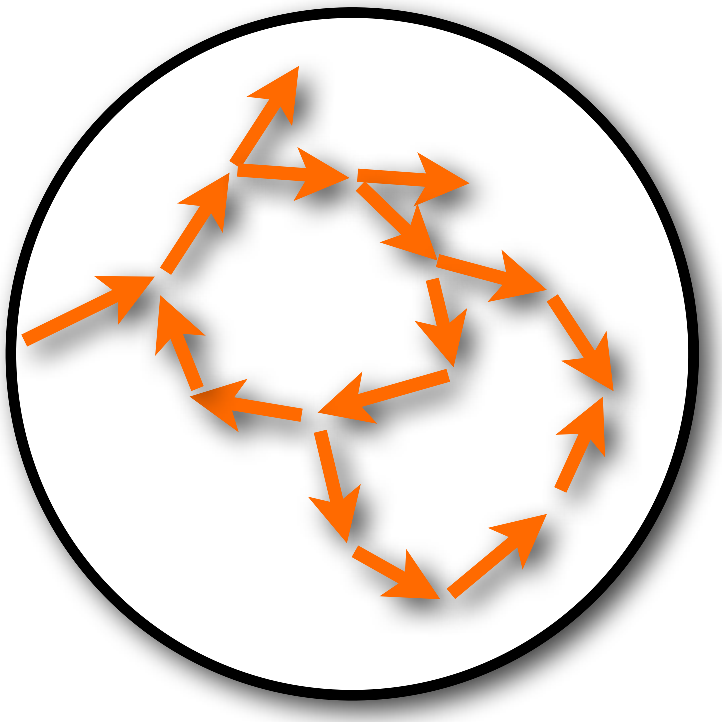
Infinite state-space

Small-step analysis...



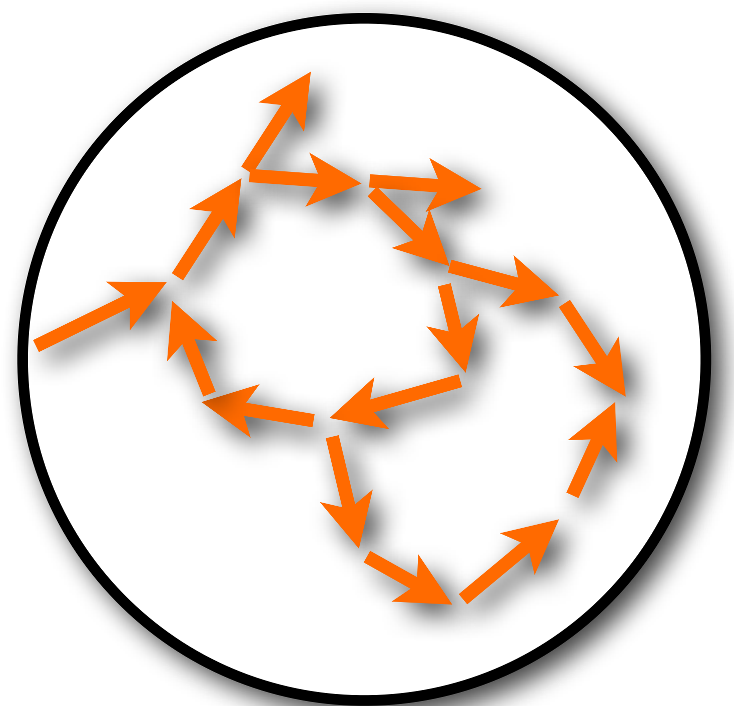
...is bounded graph search.

...is bounded graph search.



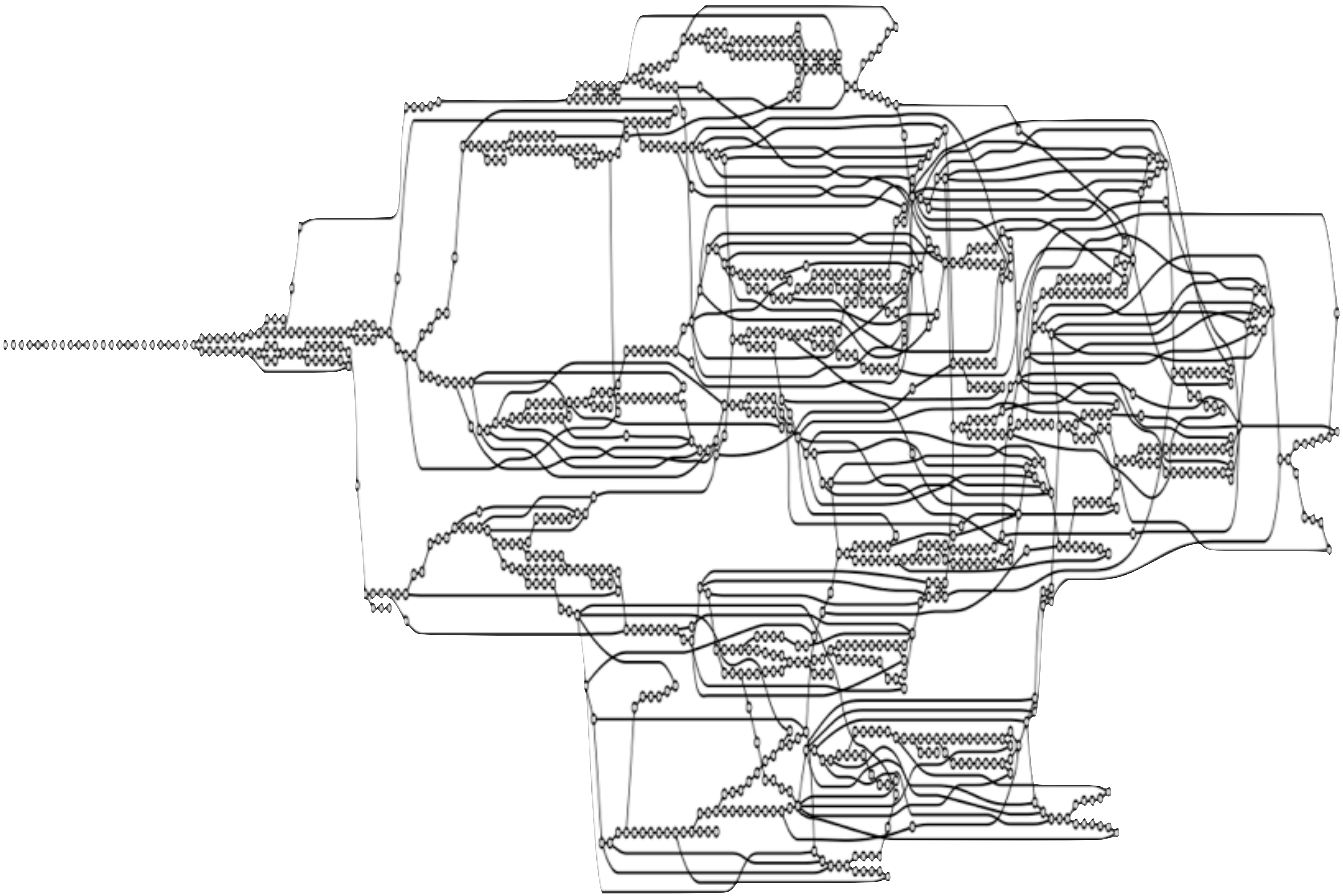
Finite state-space

...is bounded graph search.*



Finite state-space

* Unless you do pushdown analysis.



Components of small-step

$$\Sigma$$

$$(\Rightarrow) \subseteq \Sigma \times \Sigma$$

$$\Sigma$$

$$\Sigma$$

$$\alpha : \Sigma \to \hat{\Sigma}$$

$$\hat{\Sigma}$$

$$\hat{\Sigma}$$

$$(\leadsto) \subseteq \hat{\Sigma} \times \hat{\Sigma}$$

$$(\rightsquigarrow) \subseteq \hat{\Sigma} \times \hat{\Sigma}$$

$$(\Rrightarrow) \subseteq \Sigma \times \Sigma$$

$$(\rightsquigarrow) \subseteq \hat{\Sigma} \times \hat{\Sigma}$$

$$\begin{tikzcd}
\Sigma \arrow[r, Rightarrow] \arrow[d, "R"'] & \Sigma \arrow[d, "R"] \\
\hat{\Sigma} \arrow[r, rightsquigarrow, "\exists"'] & \hat{\Sigma}
\end{tikzcd}$$

$$R(\varsigma, \hat{\varsigma}) \text{ iff } \alpha(\varsigma) \sqsubseteq \hat{\varsigma}$$

$$(\sqsubseteq) \subseteq \hat{\Sigma} \times \hat{\Sigma}$$

$$\hat{L} = \mathcal{P}(\hat{\boldsymbol{\Sigma}})$$

$$\hat{f} : \hat{L} \to \hat{L}$$

$$\hat{f}(\hat{S}) = \{\hat{\varsigma}_0\} \cup \{\hat{\varsigma}' \mid \hat{\varsigma} \rightsquigarrow \hat{\varsigma}' \text{ and } \hat{\varsigma} \in \hat{S}\}$$

$$\mathrm{lfp}(\hat{f})$$

$$\operatorname{lfp}(\hat{f}) = \{\hat{\varsigma}' \mid \hat{\varsigma}_0 \Rightarrow^* \hat{\varsigma}'\}$$

$$\hat{V} = \{\hat{\varsigma}' \mid \hat{\varsigma}_0 \Rightarrow^* \hat{\varsigma}'\}$$

CPS: Small steps for free

Map check

- Develop CPS
- Abstract CPS
- Shivers's *k*-CFA

$$
\begin{aligned}
v \in \mathsf{Var} &\text{ is a set of variables} \\
lam \in \mathsf{Lam} &::= (\lambda\ (v_1 \ldots v_n)\ call) \\
f, æ \in \mathsf{AExp} &::= v \mid lam \\
call \in \mathsf{Call} &::= (f\ æ_1 \ldots æ_n)
\end{aligned}
$$

$$
\begin{aligned}
\Sigma &= \mathsf{Call} \times Env \\
Env &= \mathsf{Var} \rightarrow Clo \\
Clo &= \mathsf{Lam} \times Env
\end{aligned}
$$

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \\
\rho \in \mathit{Env} &= \mathsf{Var} \to \mathit{Clo} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env}
\end{aligned}
$$

$$\mathcal{A} : \mathsf{AExp} \times Env \to Clo$$

$$\mathcal{A} : \mathsf{AExp} \times Env \rightarrow Clo$$

$$\mathcal{A}(v, \rho) = \rho(v)$$
$$\mathcal{A}(lam, \rho) = (lam, \rho)$$

$$\mathcal{I} : \mathsf{Call} \to \Sigma$$

$$\mathcal{I} : \mathsf{Call} \to \Sigma$$

$$\mathcal{I}(call) = (call, [])$$

$$
\begin{aligned}
([\![\texttt{(}f\ \text{æ}_1 \ldots \text{æ}_n\texttt{)}]\!], \rho) &\Rightarrow (call, \rho''), \text{ where} \\
([\![\texttt{(λ (}v_1 \ldots v_n\texttt{)}\ call\texttt{)}]\!], \rho') &= \mathcal{A}(f, \rho) \\
\rho'' &= \rho'[v_i \mapsto \mathcal{A}(\text{æ}_i, \rho)]
\end{aligned}
$$

Structural abstraction?

$$\varsigma \in \Sigma = \mathsf{Call} \times Env$$

$$\rho \in Env = \mathsf{Var} \to Clo$$

$$clo \in Clo = \mathsf{Lam} \times Env$$

$$\hat{\varsigma} \in \widehat{\Sigma} = \mathsf{Call} \times \widehat{Env}$$

$$\hat{\rho} \in \widehat{Env} = \mathsf{Var} \rightarrow \widehat{Clo}$$

$$\widehat{clo} \in \widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$

$$\begin{aligned} \varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \\ \rho \in \mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Clo} \\ \mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \end{aligned}$$

$$\begin{aligned} Env &= \mathsf{Var} \to Clo \\ Clo &= \mathsf{Lam} \times Env \end{aligned}$$

$$Env = \mathsf{Var} \to Clo$$

$$Clo = \mathsf{Lam} \times Env$$

∞

How to cut the knot?

Scott & Strachey, 1966

Store-passing transform.

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times Env \\
\rho \in Env &= \mathsf{Var} \to Clo \\
clo \in Clo &= \mathsf{Lam} \times Env
\end{aligned}
$$

$$\begin{aligned} \varsigma \in \Sigma &= \mathsf{Call} \times Env \\ \rho \in Env &= \mathsf{Var} \to Clo \\ clo \in Clo &= \mathsf{Lam} \times Env \end{aligned}$$

$$\varsigma \in \Sigma = \mathsf{Call} \times \mathit{Env} \times \mathit{Store}$$

$$\rho \in \mathit{Env} = \mathsf{Var} \rightarrow \mathit{Clo}$$

$$\mathit{clo} \in \mathit{Clo} = \mathsf{Lam} \times \mathit{Env}$$

$$\varsigma \in \Sigma = \mathsf{Call} \times Env \times Store$$

$$\rho \in Env = \mathsf{Var} \rightarrow Clo$$

$$clo \in Clo = \mathsf{Lam} \times Env$$

$$\sigma \in Store = Addr \rightarrow$$

$$a \in Addr \text{ is an infinite set.}$$

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightarrow \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo} \\
a \in \mathit{Addr} & \text{ is an infinite set.}
\end{aligned}
$$

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightharpoonup \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightharpoonup \mathit{Clo} \\
a \in \mathit{Addr} & \text{ is an infinite set.}
\end{aligned}
$$

$$\mathcal{A} : \mathsf{AExp} \times Env \to Clo$$

$$\mathcal{A} : \mathsf{AExp} \times Env \times Store \rightarrow Clo$$

$$\mathcal{A} : \mathsf{AExp} \times Env \times Store \rightarrow Clo$$

$$\mathcal{A}(v, \rho, \sigma) = \sigma(\rho(v))$$
$$\mathcal{A}(lam, \rho, \sigma) = (lam, \rho)$$

$$
\begin{aligned}
(\llbracket (f\ æ_1 \ldots æ_n) \rrbracket, \rho, \sigma) &\Rightarrow (call, \rho'', \sigma'), \text{ where} \\
(\llbracket (\lambda\ (v_1 \ldots v_n)\ call) \rrbracket, \rho') &= \mathcal{A}(f, \rho, \sigma) \\
\rho'' &= \rho'[v_i \mapsto a_i] \\
\sigma' &= \sigma[a_i \mapsto \mathcal{A}(æ_i, \rho, \sigma)] \\
a_i &= alloc(v_i, \sigma)
\end{aligned}
$$

Abstraction

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightharpoonup \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightharpoonup \mathit{Clo} \\
a \in \mathit{Addr} & \text{ is an infinite set.}
\end{aligned}
$$

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightharpoonup \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightharpoonup \mathit{Clo} \\
a \in \mathit{Addr} & \text{ is an infinite set.}
\end{aligned}
$$

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo}
\end{aligned}
$$

$a \in \mathit{Addr}$ is a finite set.

$$\varsigma \in \Sigma = \mathsf{Call} \times Env \times Store$$

$$\rho \in Env = \mathsf{Var} \rightarrow Addr$$

$$clo \in Clo = \mathsf{Lam} \times Env$$

$$\sigma \in Store = Addr \rightarrow \mathcal{P}(\ Clo\ \ )$$

$a \in Addr$ is a finite set.

$$\hat{\varsigma} \in \hat{\Sigma} = \mathsf{Call} \times \widehat{Env} \times \widehat{Store}$$

$$\hat{\rho} \in \widehat{Env} = \mathsf{Var} \rightarrow \widehat{Addr}$$

$$\widehat{clo} \in \widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$

$$\hat{\sigma} \in \widehat{Store} = \widehat{Addr} \rightarrow \mathcal{P}(\widehat{Clo})$$

$$\hat{a} \in \widehat{Addr} \text{ is a finite set.}$$

$$\alpha : \Sigma \to \hat{\Sigma}$$

$$\alpha(call, \rho, \sigma) =$$

$$\alpha(call, \rho, \sigma) =$$

$$\alpha(call, \rho, \sigma) = (\quad , \quad , \quad)$$

$$\alpha(call, \rho, \sigma) = (call, \quad \rho \ , \quad \sigma \ )$$

$$\alpha(call, \rho, \sigma) = (call, \alpha(\rho), \quad \sigma \ )$$

$$\alpha(call, \rho, \sigma) = (call, \alpha(\rho), \alpha(\sigma))$$

$$\alpha(call, \rho, \sigma) = (call, \alpha(\rho), \alpha(\sigma))$$

$$\alpha(\rho) = \lambda v.\alpha(\rho(v))$$

$$\alpha(\sigma) = \lambda \hat{a}. \bigsqcup_{\alpha(a)=\hat{a}} \{\alpha(\sigma(a))\}$$

$$\alpha(lam, \rho) = \{(lam, \alpha(\rho))\}$$

$\alpha(a)$ is to be continued...

$$
\begin{aligned}
(\llbracket (f\ \text{æ}_1 \ldots \text{æ}_n) \rrbracket, \hat{\rho}, \hat{\sigma}) &\leadsto (call, \hat{\rho}'', \hat{\sigma}'), \text{ where} \\
(\llbracket (\lambda\ (v_1 \ldots v_n)\ \ call) \rrbracket, \hat{\rho}') &\in \hat{\mathcal{A}}(f, \hat{\rho}, \hat{\sigma}) \\
\hat{\rho}'' &= \hat{\rho}'[v_i \mapsto \hat{a}_i] \\
\hat{\sigma}' &= \hat{\sigma} \sqcup [\hat{a}_i \mapsto \hat{\mathcal{A}}(\text{æ}_i, \hat{\rho}, \hat{\sigma})] \\
\hat{a}_i &= \widehat{alloc}(v_i, \hat{\sigma})
\end{aligned}
$$

$$\hat{\mathcal{A}} : \mathsf{AExp} \times \widehat{Env} \times \widehat{Store} \to \mathcal{P}\left(\widehat{Clo}\right)$$

$$\hat{\mathcal{A}} : \mathsf{AExp} \times \widehat{Env} \times \widehat{Store} \to \mathcal{P}\left(\widehat{Clo}\right)$$

$$\hat{\mathcal{A}}(v, \hat{\rho}, \hat{\sigma}) = \hat{\sigma}(\hat{\rho}(v))$$

$$\hat{\mathcal{A}}(lam, \hat{\rho}, \hat{\sigma}) = \{(lam, \hat{\rho})\}$$

Allocation

$$
\begin{aligned}
([\![(f\ æ_1 \ldots æ_n)]\!], \hat{\rho}, \hat{\sigma}) &\Rightarrow (call, \hat{\rho}'', \hat{\sigma}'), \text{ where} \\
([\![(\lambda\ (v_1 \ldots v_n)\ \ call)]\!], \hat{\rho}') &= \hat{\mathcal{A}}(f, \hat{\rho}, \hat{\sigma}) \\
\hat{\rho}'' &= \hat{\rho}'[v_i \mapsto \hat{a}_i] \\
\hat{\sigma}' &= \hat{\sigma} \sqcup [\hat{a}_i \mapsto \hat{\mathcal{A}}(æ_i, \hat{\rho}, \hat{\sigma})] \\
\hat{a}_i &= \widehat{alloc}(v_i, \hat{\sigma})
\end{aligned}
$$

$$(\llbracket(f\ æ_1 \ldots æ_n)\rrbracket, \hat{\rho}, \hat{\sigma}) \Rightarrow (call, \hat{\rho}'', \hat{\sigma}'), \text{ where}$$

$$(\llbracket(\lambda\ (v_1 \ldots v_n)\ \ call)\rrbracket, \hat{\rho}') = \hat{\mathcal{A}}(f, \hat{\rho}, \hat{\sigma})$$

$$\hat{\rho}'' = \hat{\rho}'[v_i \mapsto \hat{a}_i]$$

$$\hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_i \mapsto \hat{\mathcal{A}}(æ_i, \hat{\rho}, \hat{\sigma})]$$

$$\hat{a}_i = \boxed{\widehat{alloc}(v_i, \hat{\sigma})}$$

Determines polyvariance

But with what?

$$
\begin{aligned}
(\llbracket(f\ æ_1 \ldots æ_n)\rrbracket, \hat{\rho}, \hat{\sigma}) &\Rightarrow (call, \hat{\rho}'', \hat{\sigma}'), \text{ where} \\
(\llbracket(\lambda\ (v_1 \ldots v_n)\ \ call)\rrbracket, \hat{\rho}') &= \hat{\mathcal{A}}(f, \hat{\rho}, \hat{\sigma}) \\
\hat{\rho}'' &= \hat{\rho}'[v_i \mapsto \hat{a}_i] \\
\hat{\sigma}' &= \hat{\sigma} \sqcup [\hat{a}_i \mapsto \hat{\mathcal{A}}(æ_i, \hat{\rho}, \hat{\sigma})] \\
\hat{a}_i &= \boxed{\widehat{alloc}(v_i, \hat{\sigma})}
\end{aligned}
$$

Need enriched concrete

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \\
\rho \in \mathit{Env} &= \mathsf{Var} \to \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \to \mathit{Clo} \\
a \in \mathit{Addr} & \text{ is an infinite set}
\end{aligned}
$$

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \times \mathit{Time} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo} \\
a \in \mathit{Addr} & \text{ is an infinite set} \\
t \in \mathit{Time} & \text{ is an infinite set of times.}
\end{aligned}
$$

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \times \mathit{Time} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo} \\
a \in \mathit{Addr} &\text{ is an infinite set} \\
t \in \mathit{Time} &\text{ is an infinite set of times.}
\end{aligned}
$$

(You might call them contours or histories.)

$$
\begin{aligned}
\overbrace{([\![(f\ æ_1 \ldots æ_n)]\!], \rho, \sigma, t)}^{\varsigma} &\Rightarrow (call, \rho'', \sigma', t'), \text{ where} \\
([\![(\lambda\ (v_1 \ldots v_n)\ call)]\!], \rho') &= clo \\
clo &= \mathcal{A}(f, \rho, \sigma) \\
\rho'' &= \rho'[v_i \mapsto a_i] \\
\sigma' &= \sigma[a_i \mapsto \mathcal{A}(æ_i, \rho, \sigma)] \\
t' &= tick(clo, \varsigma) \\
a_i &= alloc(v_i, t', clo, \varsigma)
\end{aligned}
$$

Abstraction

$$
\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \times \mathit{Time} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo}
\end{aligned}
$$

$a \in \mathit{Addr}$ is an infinite set

$t \in \mathit{Time}$ is an infinite set of times.

$$\begin{aligned}
\varsigma \in \Sigma &= \mathsf{Call} \times \mathit{Env} \times \mathit{Store} \times \mathit{Time} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \rightarrow \mathit{Clo} \\
a \in \mathit{Addr} &\text{ is a} \qquad \text{finite set} \\
t \in \mathit{Time} &\text{ is a} \qquad \text{finite set of times.}
\end{aligned}$$

$$\hat{\varsigma} \in \hat{\Sigma} = \mathsf{Call} \times \widehat{Env} \times \widehat{Store} \times \widehat{Time}$$

$$\hat{\rho} \in \widehat{Env} = \mathsf{Var} \rightarrow Addr$$

$$\widehat{clo} \in \widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$

$$\hat{\sigma} \in \widehat{Store} = \widehat{Addr} \rightarrow \mathcal{P}\left(\widehat{Clo}\right)$$

$$\hat{a} \in \widehat{Addr} \text{ is a finite set.}$$

$$\hat{t} \in \widehat{Time} \text{ is a finite set.}$$

$$\alpha(call, \rho, \sigma, t) = (call, \alpha(\rho), \alpha(\sigma), \alpha(t))$$

$$\alpha(\rho) = \lambda v.\alpha(\rho(v))$$

$$\alpha(\sigma) = \lambda \hat{a}. \bigsqcup_{\alpha(a)=\hat{a}} \{\alpha(\sigma(a))\}$$

$$\alpha(lam, \rho) = \{(lam, \alpha(\rho))\}$$

$\alpha(a)$ determines polyvariance

$\alpha(t)$ determines context-sensitivity

$$
\begin{aligned}
\overbrace{(\llbracket (f\ \text{æ}_1 \ldots \text{æ}_n) \rrbracket, \hat{\rho}, \hat{\sigma}, \hat{t})}^{\hat{\varsigma}} &\leadsto (call, \hat{\rho}'', \hat{\sigma}', \hat{t}'), \text{ where} \\
\widehat{clo} &\in \hat{\mathcal{A}}(f, \hat{\rho}, \hat{\sigma}) \\
(\llbracket (\lambda\ (v_1 \ldots v_n)\ call) \rrbracket, \hat{\rho}') &= \widehat{clo} \\
\hat{\rho}'' &= \hat{\rho}'[v_i \mapsto \hat{a}_i] \\
\hat{\sigma}' &= \hat{\sigma} \sqcup [\hat{a}_i \mapsto \hat{\mathcal{A}}(\text{æ}_i, \hat{\rho}, \hat{\sigma})] \\
t' &= \widehat{tick}(\widehat{clo}, \hat{\varsigma}) \\
\hat{a}_i &= \widehat{alloc}(v_i, \hat{t}', \widehat{clo}, \hat{\varsigma})
\end{aligned}
$$

$$\overbrace{(\llbracket(f\ æ_1 \ldots æ_n)\rrbracket, \hat{\rho}, \hat{\sigma}, \hat{t})}^{\hat{\varsigma}} \leadsto (call, \hat{\rho}'', \hat{\sigma}', \hat{t}'), \text{ where}$$

$$\widehat{clo} \in \hat{\mathcal{A}}(f, \hat{\rho}, \hat{\sigma})$$

$$(\llbracket(\lambda\ (v_1 \ldots v_n)\ call)\rrbracket, \hat{\rho}') = \widehat{clo}$$

$$\hat{\rho}'' = \hat{\rho}'[v_i \mapsto \hat{a}_i]$$

$$\hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a}_i \mapsto \hat{\mathcal{A}}(æ_i, \hat{\rho}, \hat{\sigma})]$$

Context-sensitivity: $t' = \widehat{tick}(\widehat{clo}, \hat{\varsigma})$

Polyvariance: $\hat{a}_i = \widehat{alloc}(v_i, \hat{t}', \widehat{clo}, \hat{\varsigma})$

k-CFA

Allocation strategy

$$Time = \mathsf{Call}^*$$

$$Time = \mathsf{Call}^*$$

$$tick(clo, (call, \ldots, t)) = call : t$$

$$Addr = \mathsf{Var} \times Time$$

$$Addr = \mathsf{Var} \times Time$$

$$alloc(v_i, t, clo, \varsigma) = (v_i, t)$$

$$\widehat{Time} = \mathsf{Call}^{k}$$

$$\widehat{Time} = \mathsf{Call}^k$$

$$\widehat{tick}(\widehat{clo}, (call, \ldots, \hat{t})) = \lfloor call : \hat{t} \rfloor_k$$

$$\widehat{Addr} = \mathsf{Var} \times \widehat{Time}$$

$$\widehat{Addr} = \mathsf{Var} \times \widehat{Time}$$

$$\widehat{alloc}(v_i, \hat{t}, \widehat{clo}, \hat{\varsigma}) = (v_i, \hat{t})$$

Complexity

$$|\hat{\Sigma}| = $$

$$\times$$

$$\times$$

$$\times$$

$$|\hat{\Sigma}| = |\mathsf{Call}|$$

$$\times$$

$$\times$$

$$\times$$

$$
\begin{aligned}
|\hat{\Sigma}| = {} & |\mathsf{Call}| \\
& \times \left(|\mathsf{Var}| \times |\,\mathsf{Call}|^{k}\right)^{|\mathsf{Var}|} \\
& \times \\
& \times
\end{aligned}
$$

$$
\begin{aligned}
|\hat{\Sigma}| = {} & |\mathsf{Call}| \\
& \times \left(|\mathsf{Var}| \times |\,\mathsf{Call}|^{k}\right)^{|\mathsf{Var}|} \\
& \times \left(2^{|\mathsf{Lam}| \times (|\mathsf{Var}| \times |\mathsf{Call}|^{k})^{|\mathsf{Var}|}}\right)^{|\mathsf{Var}| \times |\mathsf{Call}|^{k}} \\
& \times
\end{aligned}
$$

$$
\begin{aligned}
|\hat{\Sigma}| = {} & |\mathsf{Call}| \\
& \times (|\mathsf{Var}| \times |\ \mathsf{Call}|^{k})^{|\mathsf{Var}|} \\
& \times \left(2^{|\mathsf{Lam}| \times (|\mathsf{Var}| \times |\mathsf{Call}|^{k})^{|\mathsf{Var}|}}\right)^{|\mathsf{Var}| \times |\mathsf{Call}|^{k}} \\
& \times |\mathsf{Call}|^{k}
\end{aligned}
$$

Widening/Re-abstraction

$$\hat{f} : \hat{L} \to \hat{L}$$

$$\begin{aligned}\hat{L} &= \mathcal{P}(\hat{\Sigma}) \\ &= \\ &\geq \\ &= \end{aligned}$$

$$
\begin{aligned}
\hat{L} &= \mathcal{P}(\hat{\Sigma}) \\
&= \\
&\geq \\
&=
\end{aligned}
$$

$$
\begin{aligned}
\hat{L} &= \mathcal{P}(\hat{\Sigma}) \\
&= \mathcal{P}(\mathsf{Call} \times \widehat{Env} \times \widehat{Store} \times \widehat{Time}) \\
&\geq \\
&=
\end{aligned}
$$

$$
\begin{aligned}
\hat{L} &= \mathcal{P}(\hat{\Sigma}) \\
&= \mathcal{P}(\mathsf{Call} \times \widehat{Env} \times \widehat{Store} \times \widehat{Time}) \\
&\geq \mathcal{P}(\mathsf{Call} \times \widehat{Env} \times \widehat{Time}) \times \widehat{Store} \\
&=
\end{aligned}
$$

$$
\begin{aligned}
\hat{L} &= \mathcal{P}(\hat{\Sigma}) \\
&= \mathcal{P}(\mathsf{Call} \times \widehat{Env} \times \widehat{Store} \times \widehat{Time}) \\
&\geq \mathcal{P}(\mathsf{Call} \times \widehat{Env} \times \widehat{Time}) \times \widehat{Store} \\
&= \hat{L}'
\end{aligned}
$$

$$\alpha' : \hat{L} \to \hat{L}'$$

$$
\begin{aligned}
&\alpha'\{(call_1, \hat{\rho}_1, \hat{\sigma}_1), \ldots, (call_2, \hat{\rho}_2, \hat{\sigma}_2)\} \\
&= (\{(call_1, \hat{\rho}_1), \ldots, (call_n, \hat{\rho}_n)\}, \bigsqcup_i \hat{\sigma}_i)
\end{aligned}
$$

Complexity: *k*-CFA

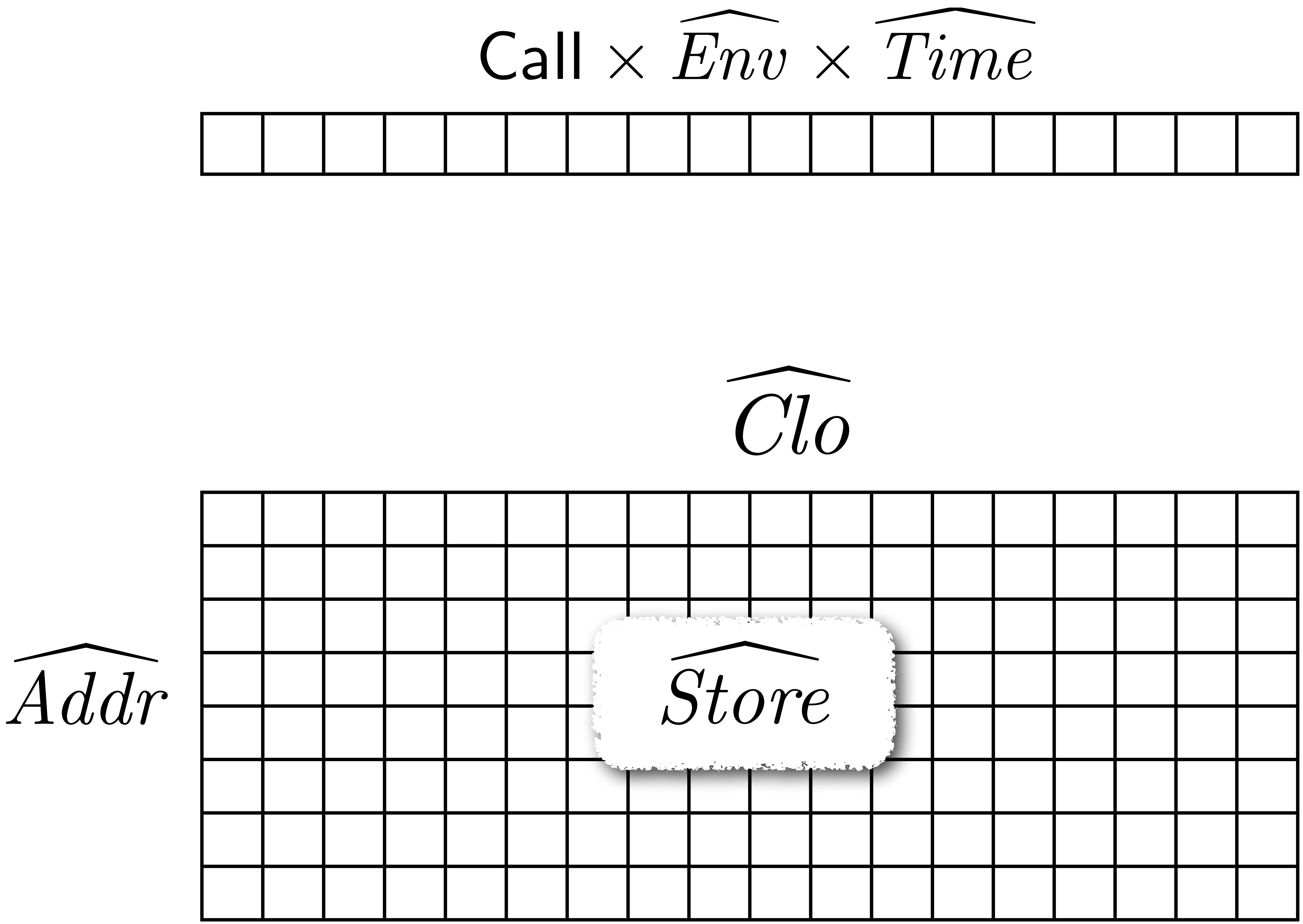
$\hat{f}$ is monotonic

$$\hat{f} : \hat{L} \to \hat{L}$$

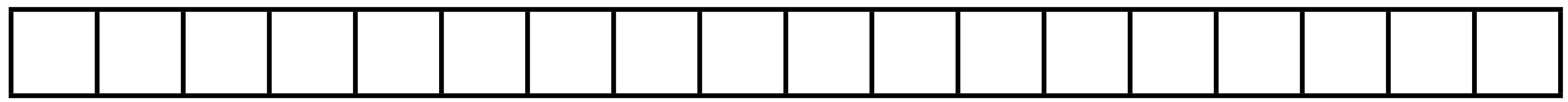
$$\hat{L} = \mathcal{P}(\mathsf{Call} \times \widehat{Env} \times \widehat{Time}) \times \widehat{Store}$$

$$\mathcal{P}(\mathsf{Call} \times \widehat{Env} \times \widehat{Time})$$

$$\widehat{Store} = \widehat{Addr} \to \mathcal{P}(\widehat{Clo})$$

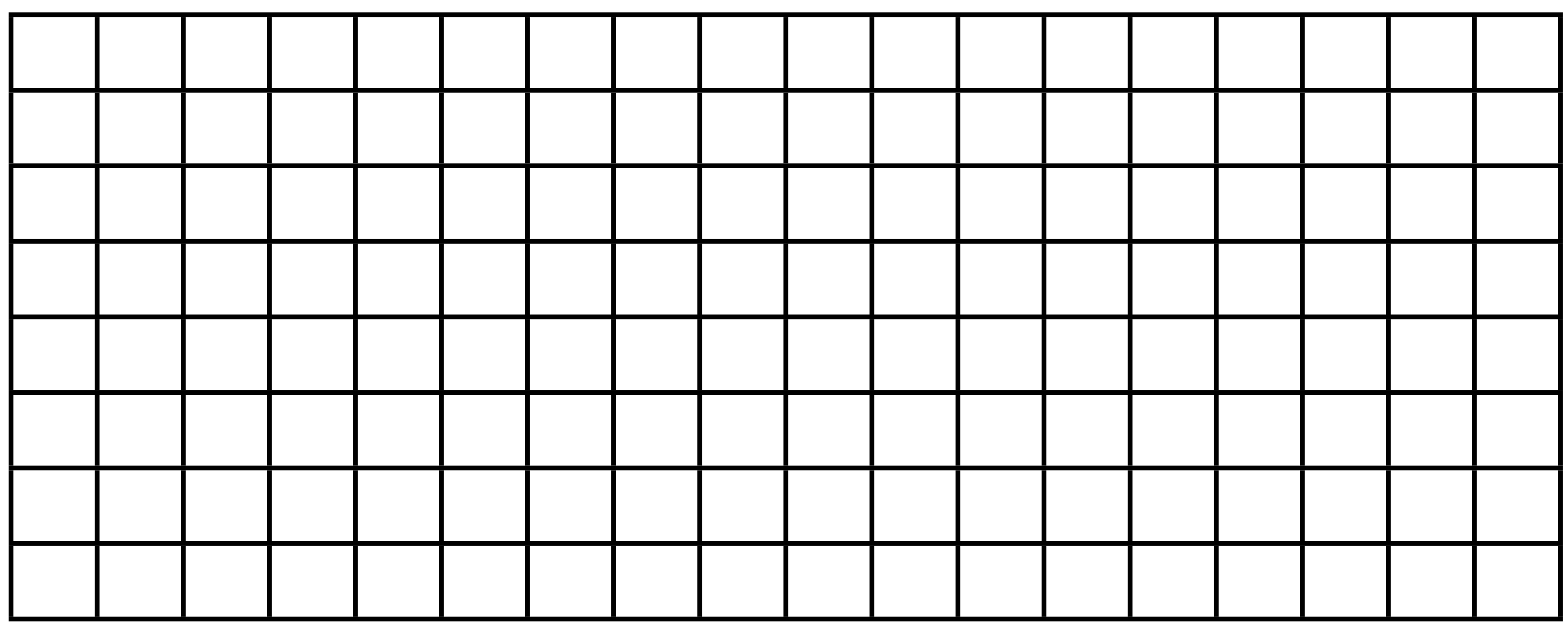

Call $\times$ $\widehat{Env}$ $\times$ $\widehat{Time}$
$\widehat{Clo}$
$\widehat{Addr}$
$\widehat{Store}$

$\mathsf{Call} \times \widehat{Env} \times \widehat{Time}$



$\widehat{Clo}$

$\widehat{Addr}$



$$\mathsf{Call} \times \widehat{Env} \times \widehat{Time}$$

		✓								✓						✓	

$$\widehat{Clo}$$

$\widehat{Addr}$

																	✓
	✓					✓											
														✓			
					✓					✓							
		✓															
											✓						✓
				✓													

$$|\mathsf{Call} \times \widehat{Env} \times \widehat{Time}\,|$$

+

$$|\,\widehat{Clo}\,| \times |\widehat{Addr}|$$

number of passes

$$|\mathsf{Call} \times \widehat{Env} \times \widehat{Time}\,|$$

+

$$|\,\widehat{Clo}\,| \times |\widehat{Addr}|$$

number of passes

$$|\mathsf{Call}| \times |\mathsf{Call}|^{k \times |\mathsf{Var}|} \times |\mathsf{Call}|^{k}$$

+

$$|\mathsf{Var}| \times |\mathsf{Call}|^{k} \times |\mathsf{Lam}| \times |\mathsf{Call}|^{k \times |\mathsf{Var}|}$$

number of passes

$$|\mathsf{Call}| \times |\mathsf{Call}|^{k \times |\mathsf{Var}|} \times |\mathsf{Call}|^{k}$$

cost per pass

Complexity: 0CFA

$$O(|\mathsf{Call}| \cdot |\mathsf{Var}| \cdot |\mathsf{Lam}|)$$

Language features

Side effects

$$
\begin{aligned}
\textsf{Call} ::= {} & \ldots \\
\mid {} & (\texttt{set!-then}\ v\ \text{\ae}\ call)
\end{aligned}
$$

$$\overbrace{(\llbracket \texttt{(set!-then}\ v\ \text{æ}\ call\texttt{)} \rrbracket, \rho, \sigma, t)}^{\varsigma} \Rightarrow (call, \rho, \sigma', t'), \text{ where}$$

$$\begin{aligned} \sigma' &= \sigma[a \mapsto \mathcal{A}(\text{æ}, \rho, \sigma)] \\ t' &= tick_{\texttt{set}}(\varsigma) \\ a &= \rho(v) \end{aligned}$$

$$\overbrace{(\llbracket \texttt{(set!-then}\ v\ \textit{æ}\ \mathit{call}\texttt{)} \rrbracket, \hat{\rho}, \hat{\sigma}, \hat{t})}^{\hat{\varsigma}} \Rightarrow (\mathit{call}, \hat{\rho}, \hat{\sigma}', \hat{t}'), \text{ where}$$

$$\hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a} \mapsto \hat{\mathcal{A}}(\textit{æ}, \hat{\rho}, \hat{\sigma})]$$

$$\hat{t}' = \widehat{tick}_{\texttt{set}}(\hat{\varsigma})$$

$$\hat{a} = \hat{\rho}(v)$$

Recursion

$$\text{Call} ::= \ldots$$
$$\mid \ \texttt{(letrec ((}v_1\ lam_1\texttt{)} \cdots \texttt{(}v_n\ lam_n\texttt{))}\ call\texttt{)}$$

$$\overbrace{(\llbracket\texttt{(letrec (}\cdots\texttt{(}v_i\ lam_i\texttt{)}\cdots\texttt{)}\ call\texttt{)}\rrbracket, \rho, \sigma, t)}^{\varsigma} \Rightarrow (call, \rho', \sigma', t'), \text{ where}$$

$$\begin{aligned}
t' &= tick_{\texttt{letrec}}(\varsigma) \\
a_i &= alloc_{\texttt{letrec}}(v_i, lam_i, t', \varsigma) \\
\rho' &= \rho[v_i \mapsto a_i] \\
clo_i &= \mathcal{A}(lam_i, \rho', \sigma) \\
\sigma' &= \sigma[a_i \mapsto clo'_i]
\end{aligned}$$

$$\overbrace{([\![\texttt{(letrec (}\cdots\texttt{(}v_i\ lam_i\texttt{)}\cdots\texttt{)}\ call\texttt{)}]\!], \hat{\rho}, \hat{\sigma}, \hat{t})}^{\hat{\varsigma}} \leadsto (call, \hat{\rho}', \hat{\sigma}', \hat{t}'), \text{ where}$$

$$\begin{aligned}
\hat{t}' &= \widehat{tick}_{\texttt{letrec}}(\hat{\varsigma}) \\
\hat{a}_i &= \widehat{alloc}_{\texttt{letrec}}(v_i, lam_i, \hat{t}', \hat{\varsigma}) \\
\hat{\rho}' &= \hat{\rho}[v_i \mapsto \hat{a}_i] \\
\widehat{clo}_i &= \hat{\mathcal{A}}(lam_i, \hat{\rho}', \hat{\sigma}) \\
\hat{\sigma}' &= \hat{\sigma} \sqcup [\hat{a}_i \mapsto \{\widehat{clo}_i\}]
\end{aligned}$$

Advanced techniques

Garbage collection

Steps

- ?
- Kill zombies.
- Profit.

$$\varsigma \longrightarrow \varsigma' \longrightarrow \varsigma'' \longrightarrow \varsigma''' \longrightarrow \varsigma''''$$

$$
\begin{tikzcd}
\varsigma \arrow[r] & {} \arrow[d, "\Gamma"] & & & \\
& \varsigma' \arrow[r] & {} \arrow[d, "\Gamma"] & & \\
& & \varsigma'' \arrow[r] & {} \arrow[d, "\Gamma"] & \\
& & & \varsigma''' \arrow[r] & {} \arrow[d, "\Gamma"] \\
& & & & \varsigma''''
\end{tikzcd}
$$

$$\Gamma : \Sigma \to \Sigma$$

$$\Gamma\overbrace{(call, \rho, \sigma)}^{\varsigma} = (call, \rho, \sigma | Reachable(\varsigma))$$

$$Reachable\overbrace{(call, \rho, \sigma)}^{\varsigma} = \left\{a' : a_0 \in Root(\varsigma) \text{ and } a_0 \xmapsto[\sigma]{*} a'\right\}$$

$$Root : \Sigma \to \mathcal{P}(Addr)$$

$$Root(call, \rho, \sigma) = range(\rho)$$

$$Reachable\overbrace{(call, \rho, \sigma)}^{\varsigma} = \left\{a' : a_0 \in Root(\varsigma) \text{ and } a_0 \xmapsto[\sigma]{*} a'\right\}$$

$$a \underset{\sigma}{\longmapsto} a' \text{ iff } (lam, \rho) = \sigma(a) \text{ and } a' \in range(\rho)$$

Abstract GC

$$\hat{\Gamma} : \hat{\Sigma} \to \hat{\Sigma}$$

$$\hat{\Gamma}\overbrace{(call, \hat{\rho}, \hat{\sigma})}^{\hat{\varsigma}} = (call, \hat{\rho}, \hat{\sigma} | Reachable(\hat{\varsigma}))$$

$$Reachable\overbrace{(call, \hat{\rho}, \hat{\sigma})}^{\hat{\varsigma}} = \left\{ \hat{a}' : \hat{a}_0 \in Root(\hat{\varsigma}) \text{ and } \hat{a}_0 \xmapsto[\hat{\sigma}]{*} \hat{a}' \right\}$$

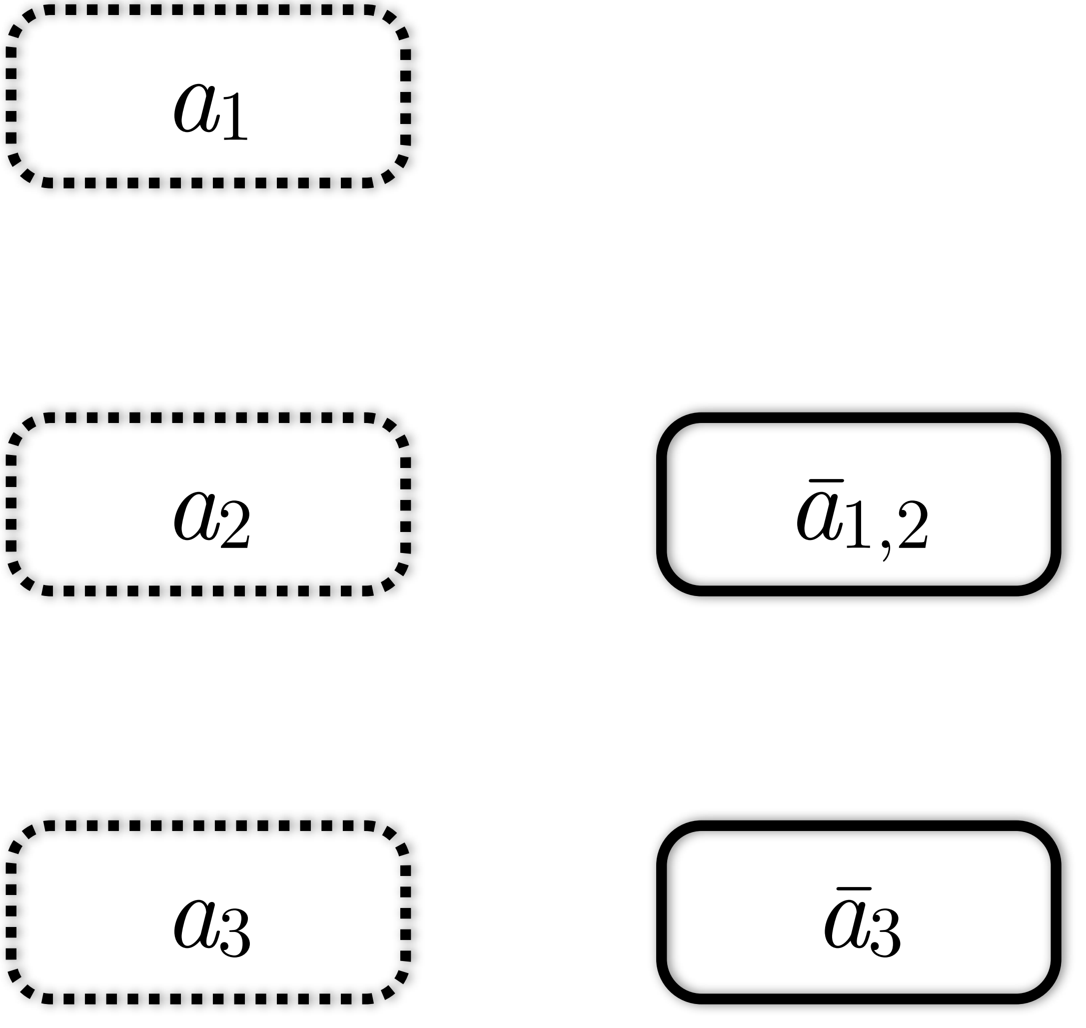
$$Root : \hat{\Sigma} \rightarrow \mathcal{P}(\widehat{Addr})$$

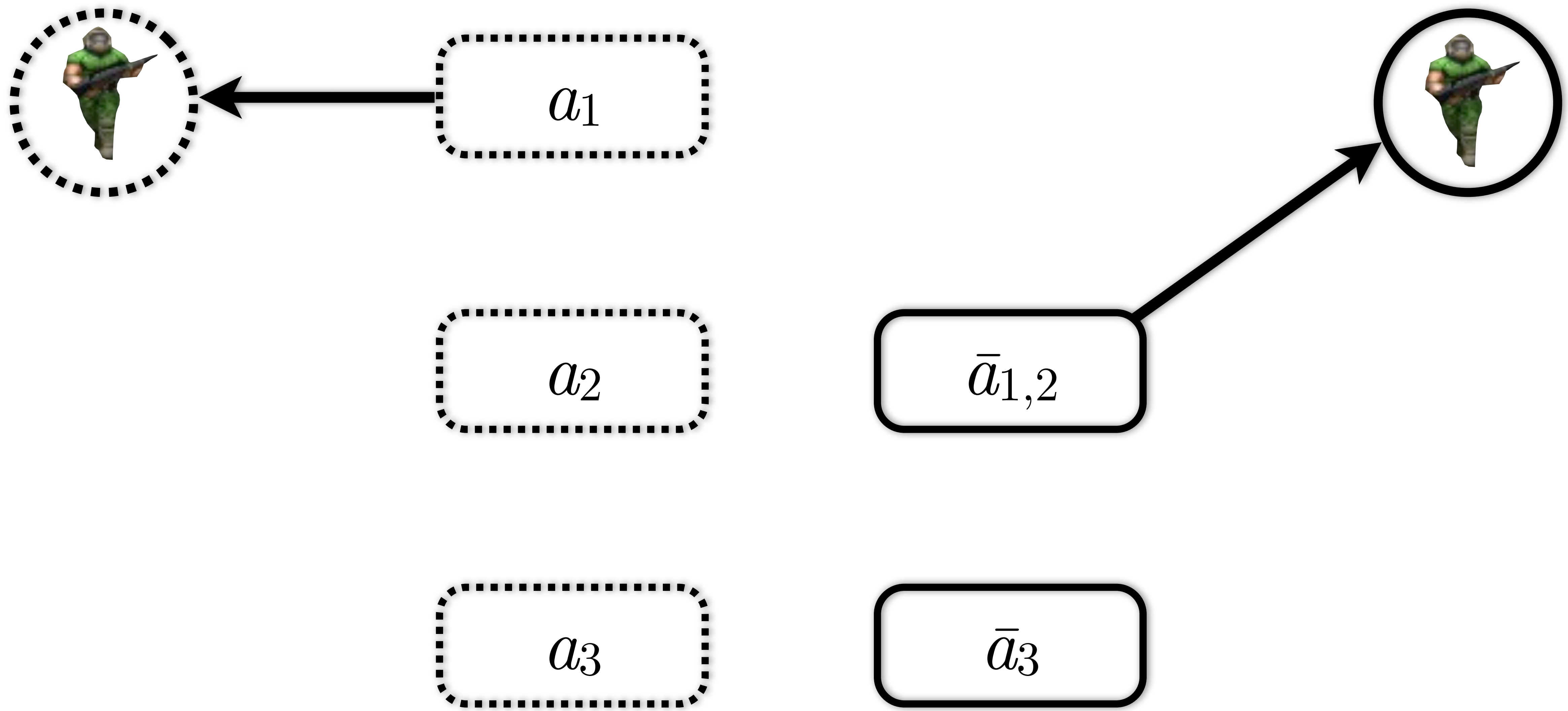
$$Root : \hat{\Sigma} \to \mathcal{P}(\widehat{Addr})$$

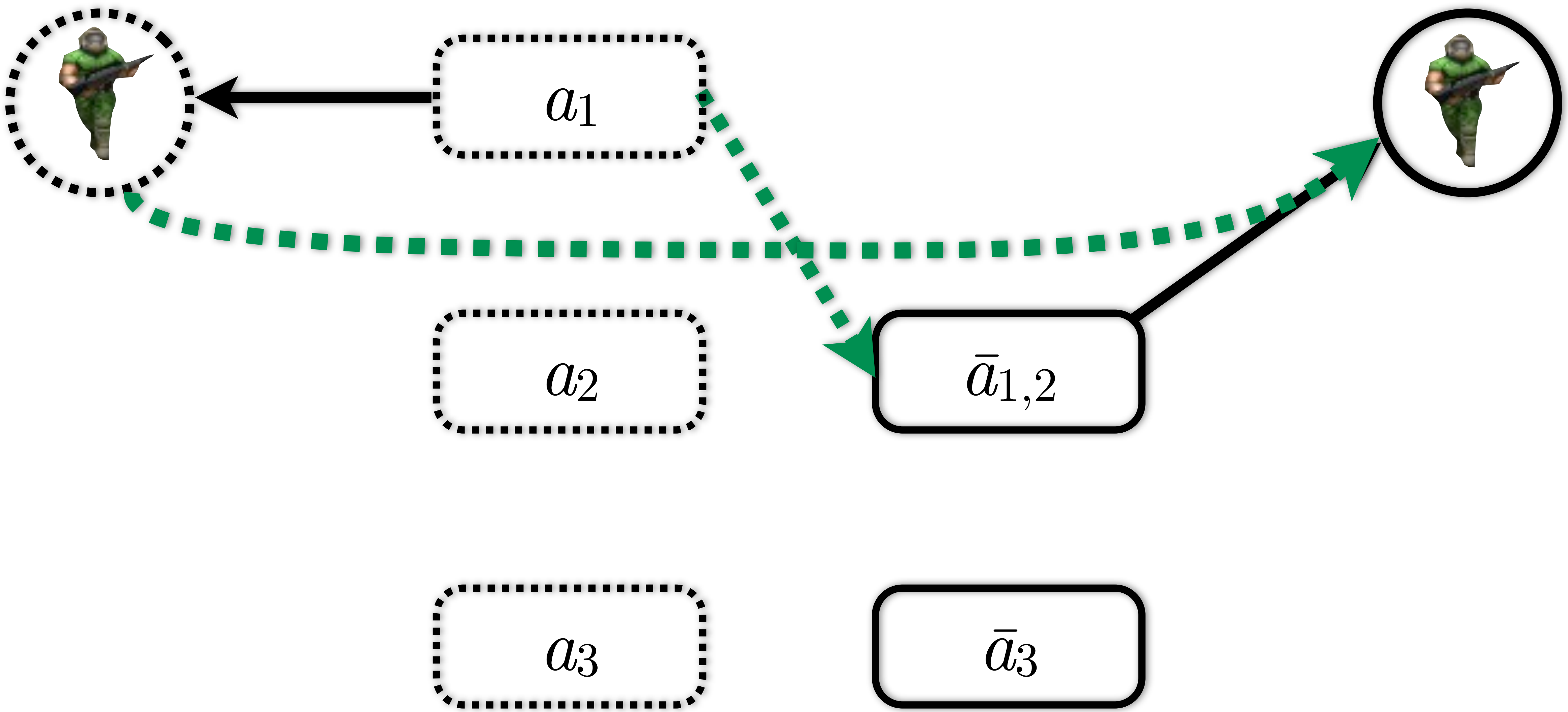
$$Root(call, \hat{\rho}, \hat{\sigma}) = range(\hat{\rho})$$

$$\hat{a} \underset{\hat{\sigma}}{\longmapsto} \hat{a}' \text{ iff there exists } (lam, \hat{\rho}) \in \sigma(\hat{a}) \text{ such that } \hat{a}' \in range(\hat{\rho})$$

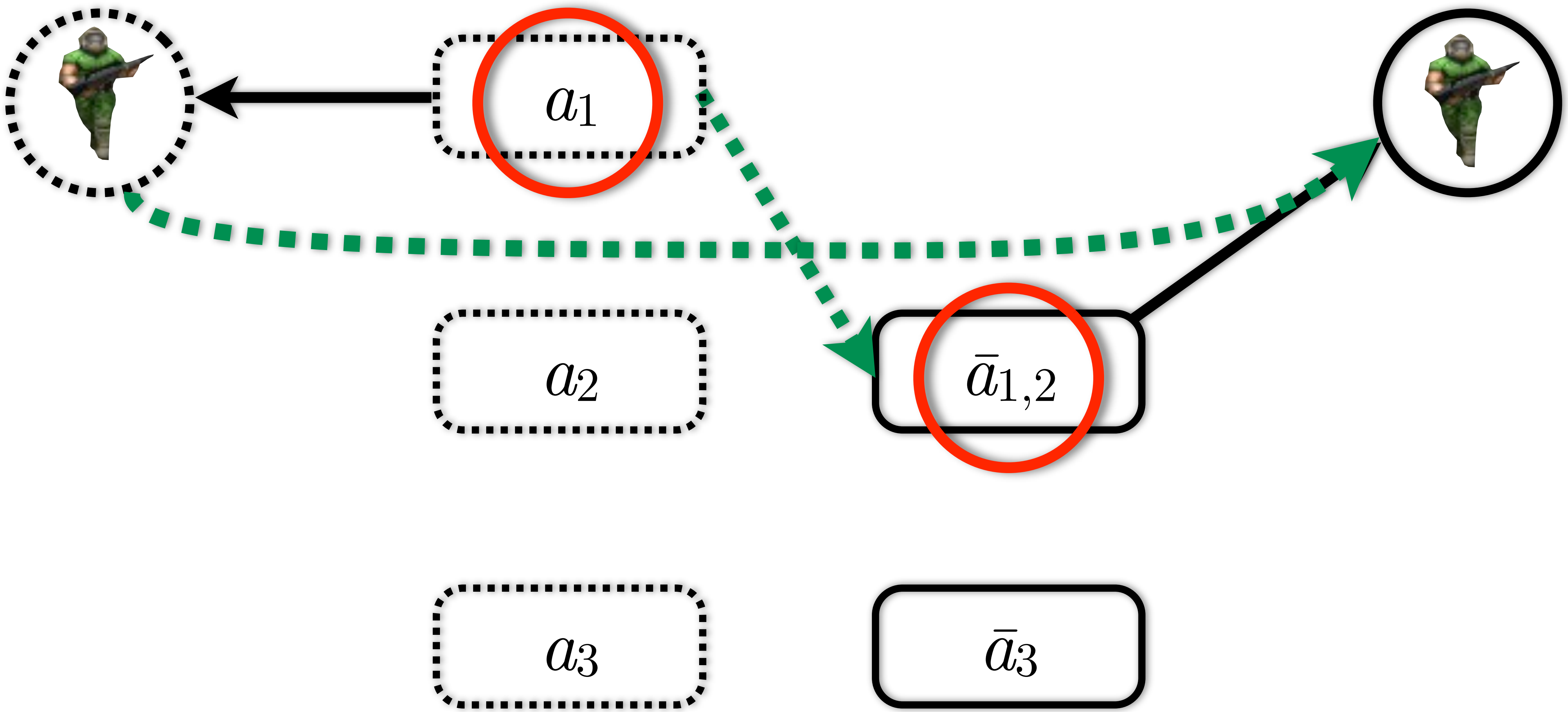
Killing zombies


a_1
a_2
$\bar{a}_{1,2}$
a_3
$\bar{a}_3$

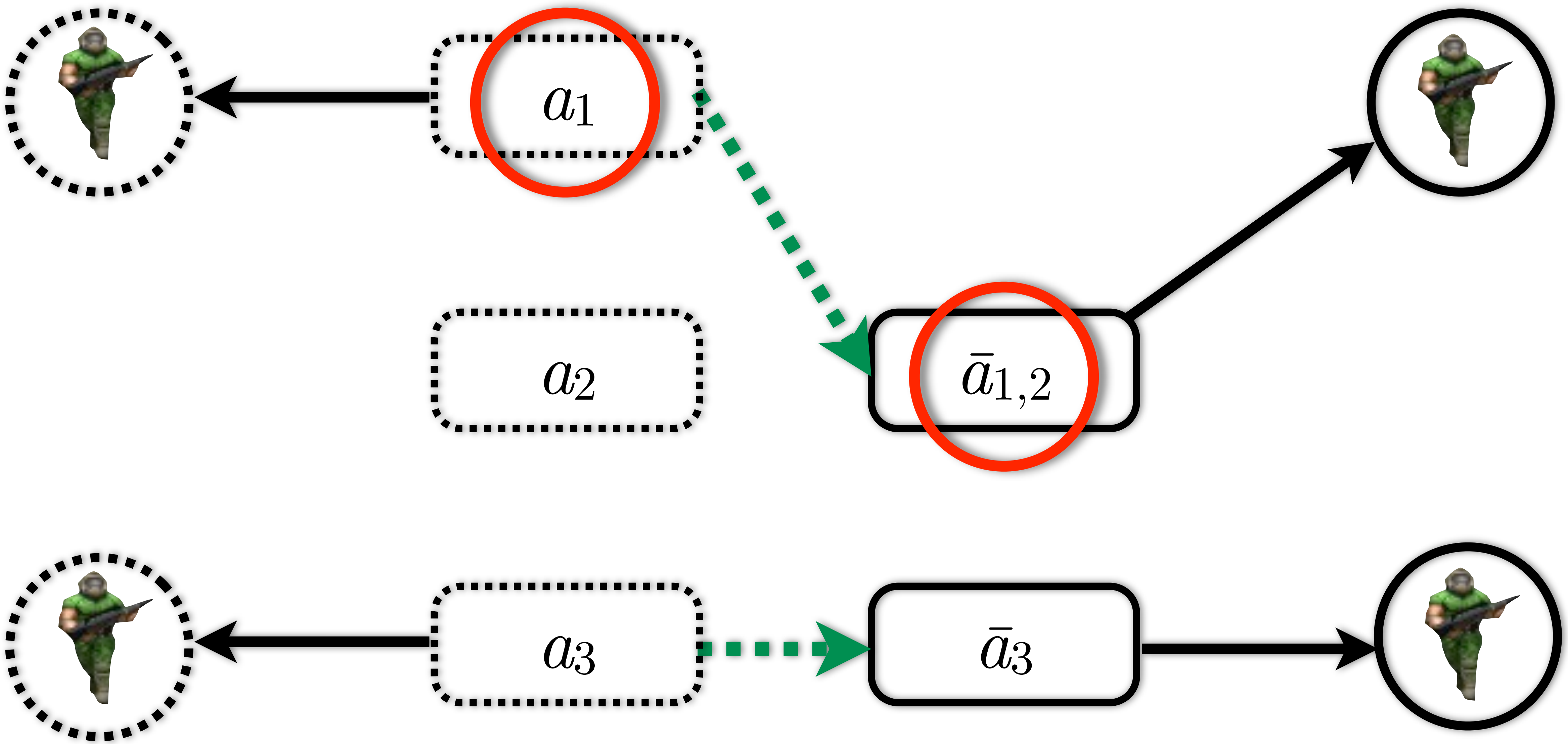

a_1
a_2
$\bar{a}_{1,2}$
a_3
$\bar{a}_3$



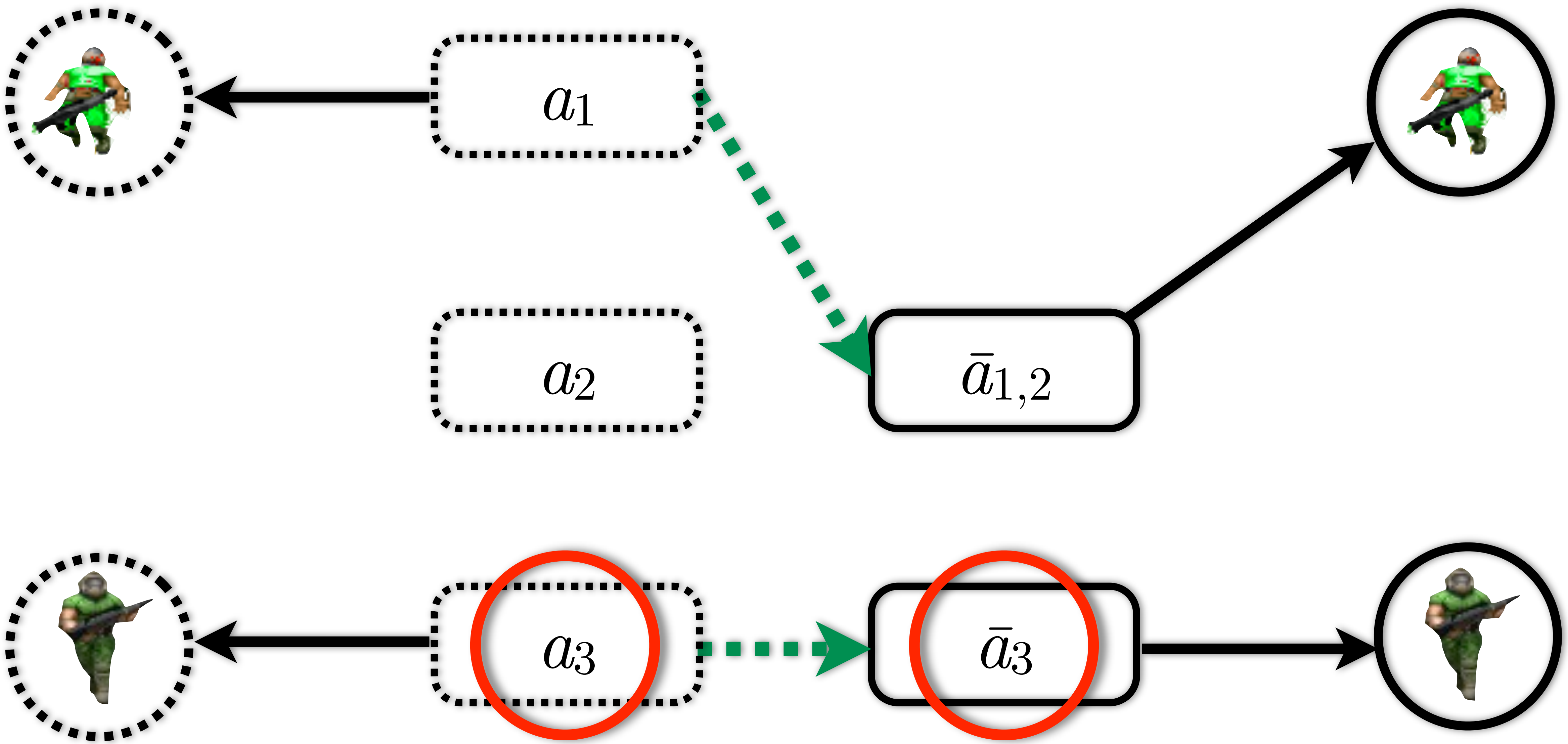
a_1
a_2
a_3
$\bar{a}_{1,2}$
$\bar{a}_3$



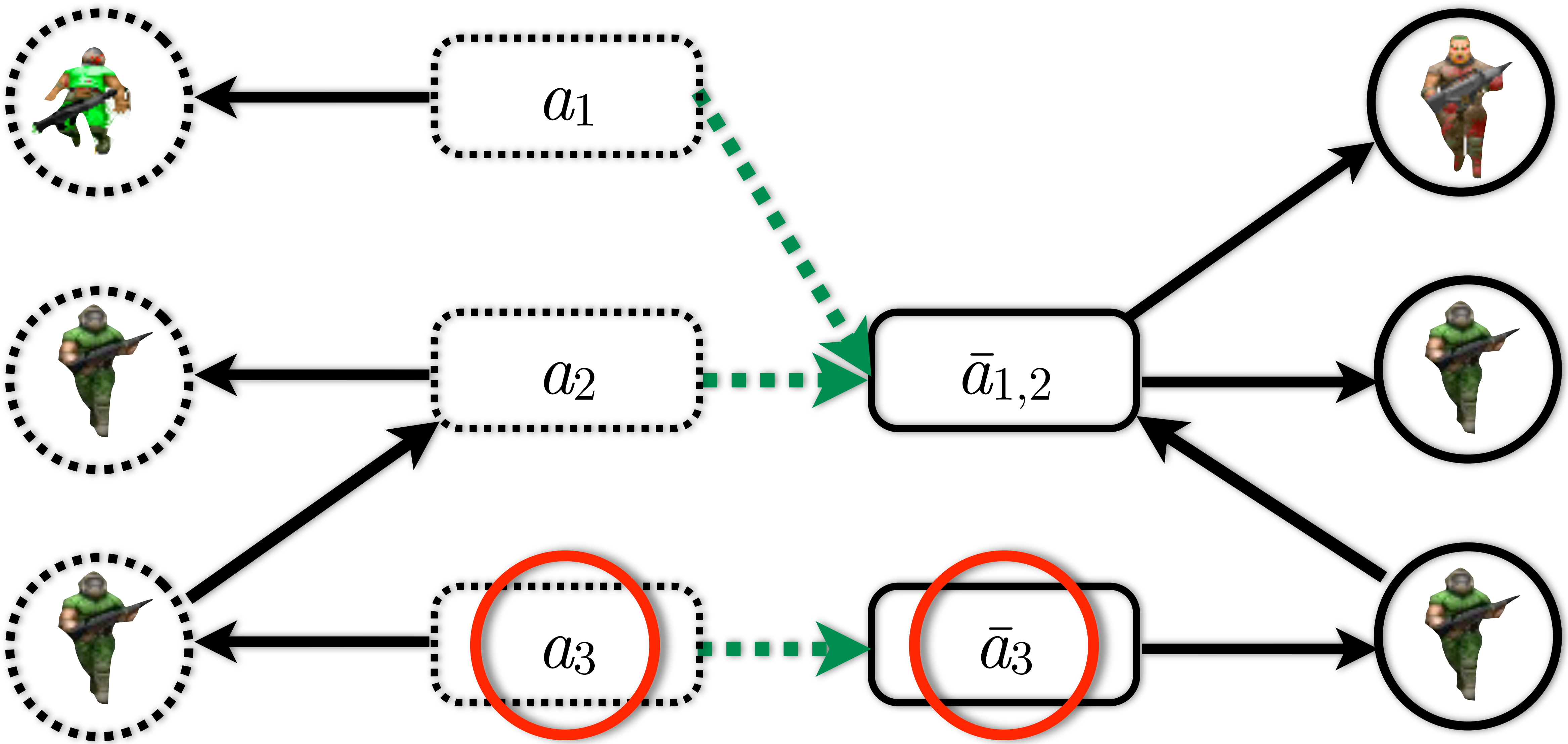
a_1
a_2
a_3
$\bar{a}_{1,2}$
$\bar{a}_3$

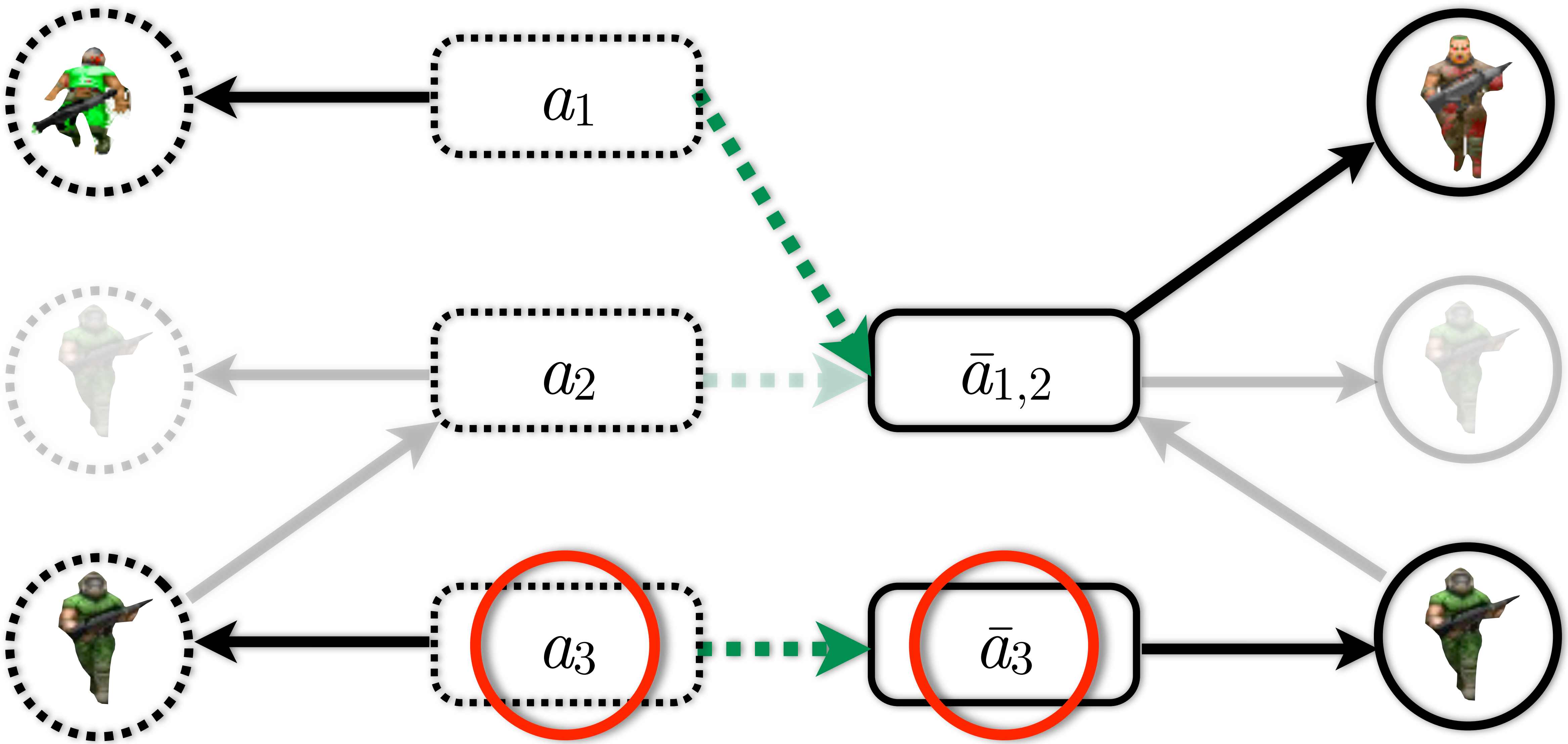


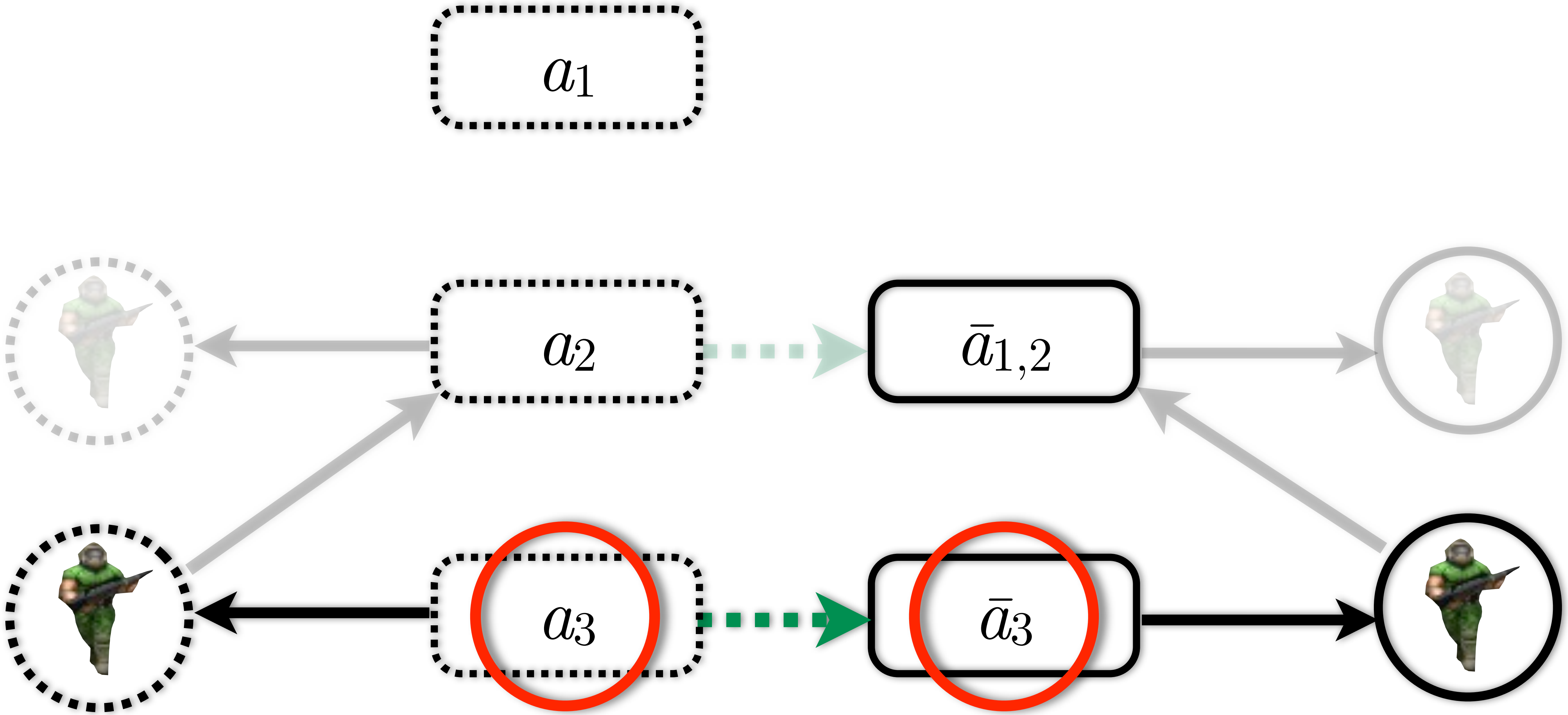
a_1
a_2
$\bar{a}_{1,2}$
a_3
$\bar{a}_3$



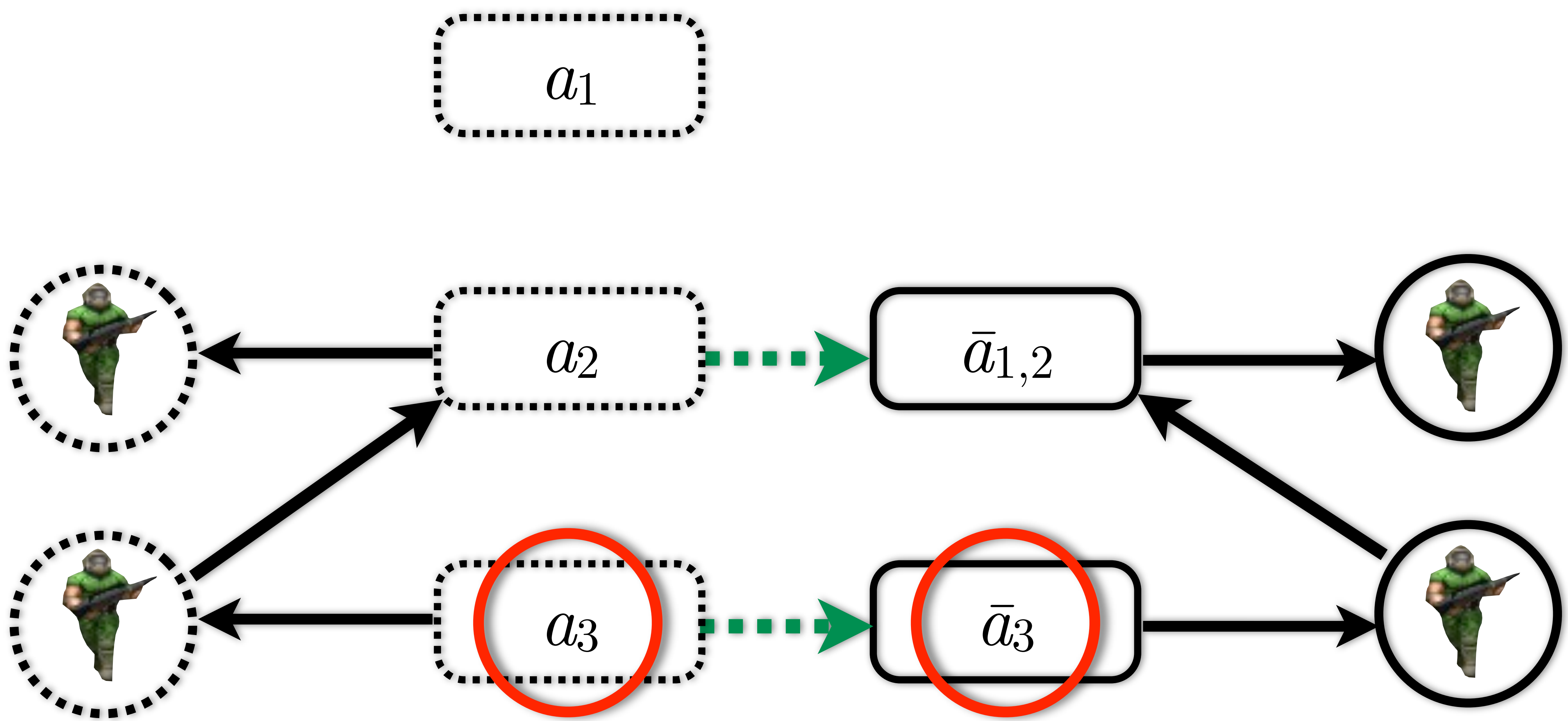
a_1
a_2
$\bar{a}_{1,2}$
a_3
$\bar{a}_3$


a_1
a_2
a_3
$\bar{a}_{1,2}$
$\bar{a}_3$

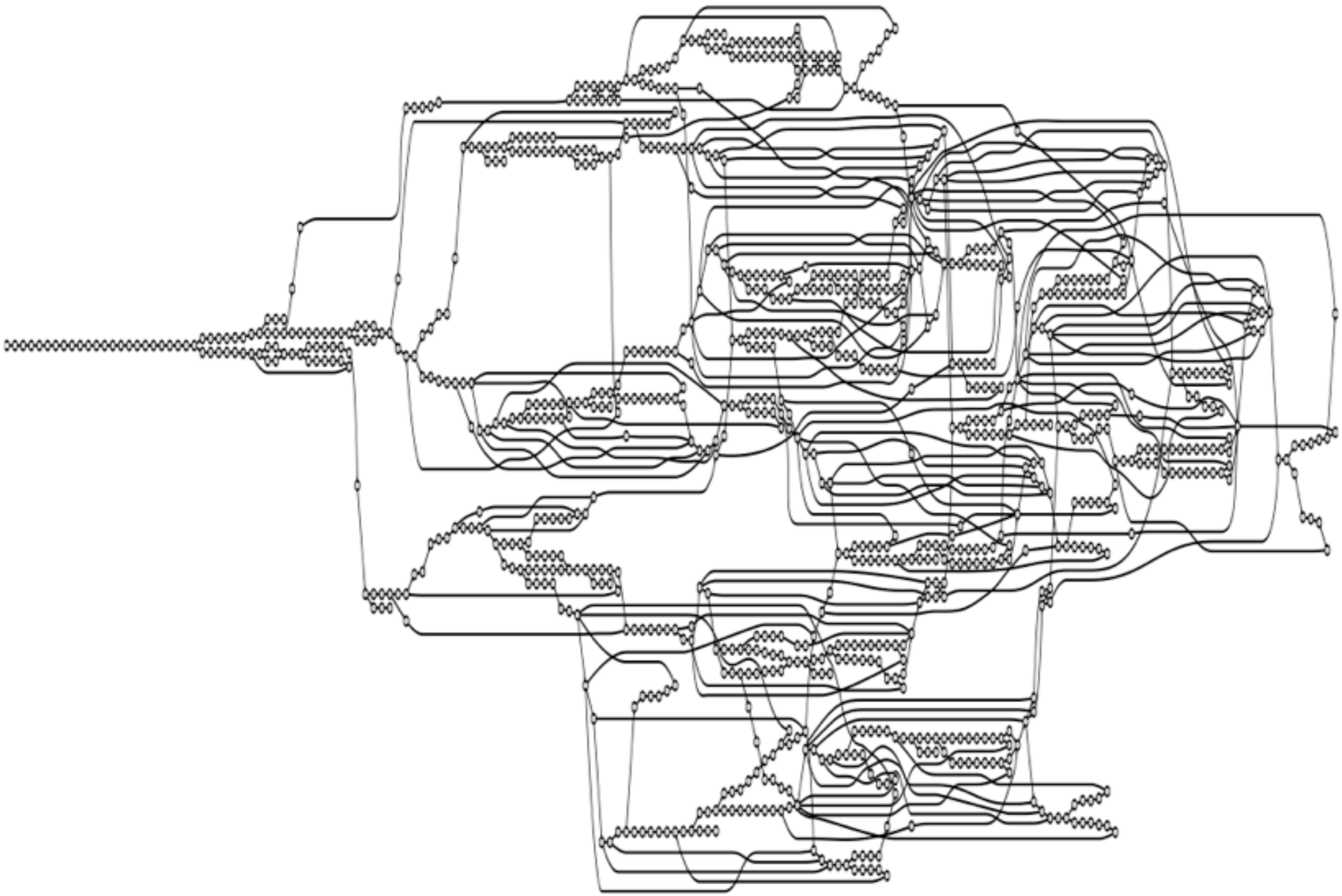

a_1
a_2
a_3
$\bar{a}_{1,2}$
$\bar{a}_3$

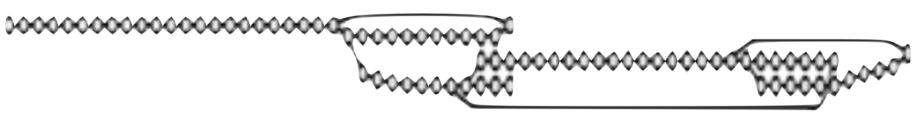


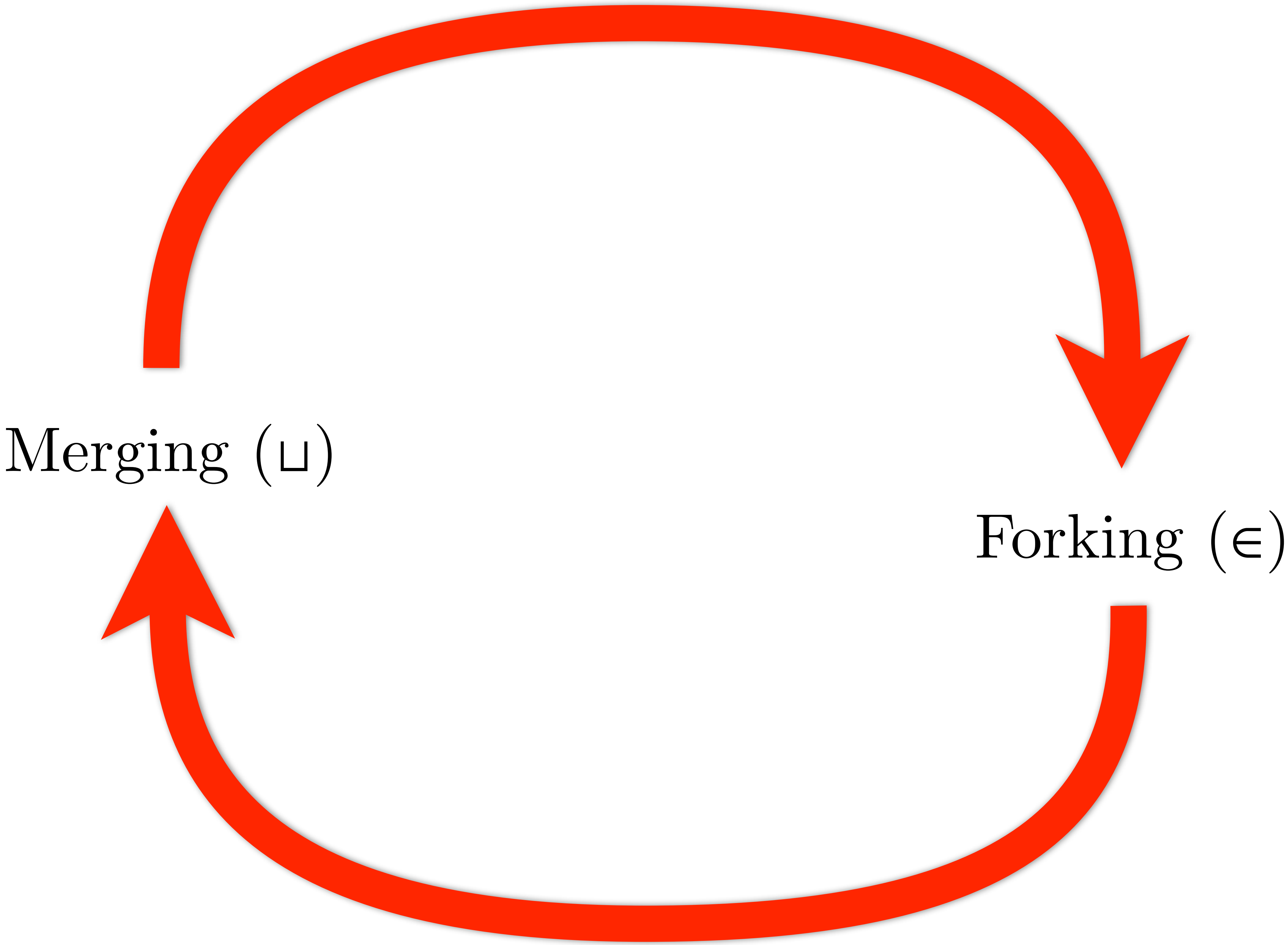
a_1
a_2
$\bar{a}_{1,2}$
a_3
$\bar{a}_3$

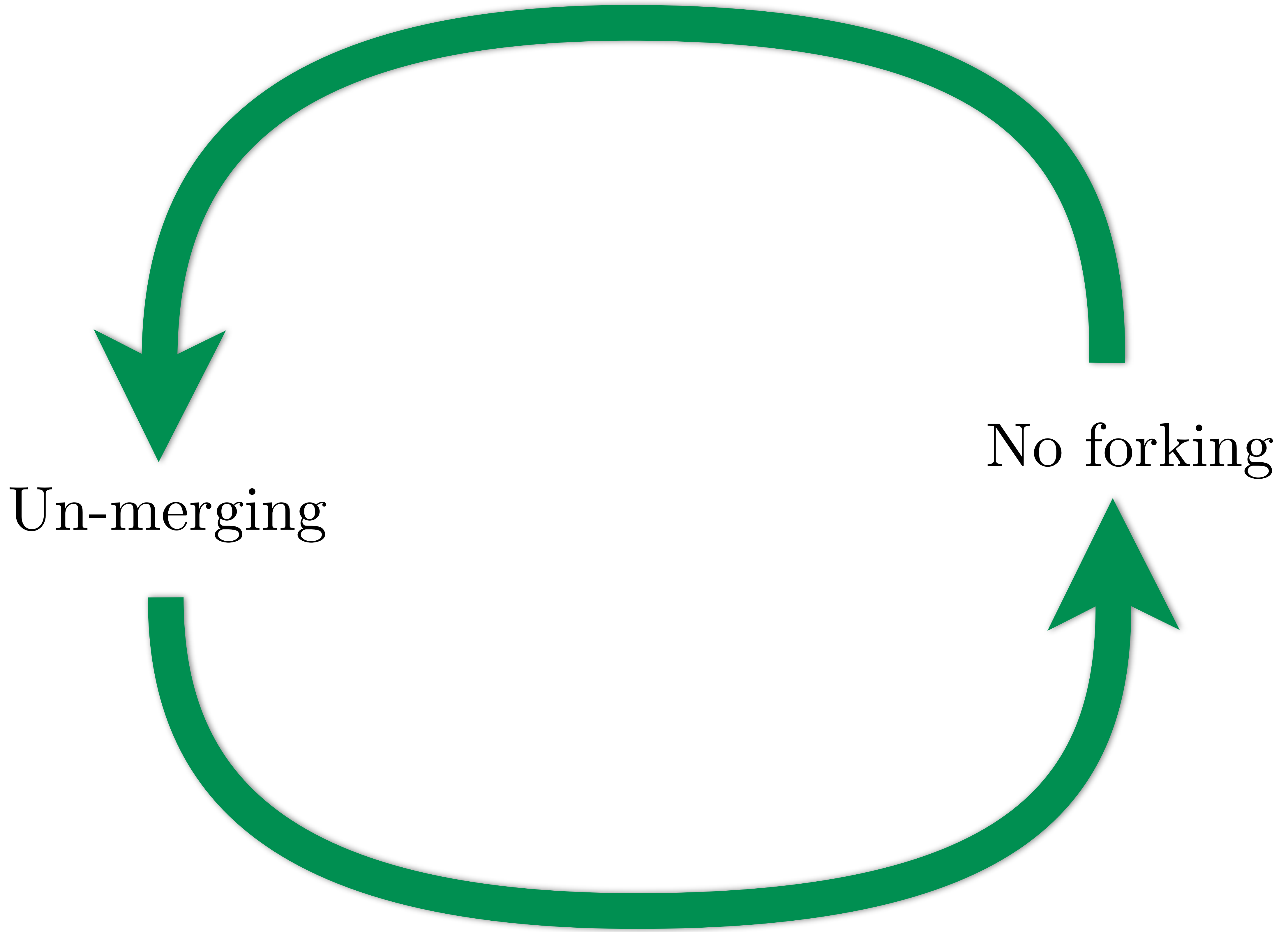


a_1
a_2
$\bar{a}_{1,2}$
a_3
$\bar{a}_3$






Merging ($\sqcup$)
Forking ($\in$)


Un-merging
No forking

Abstract counting

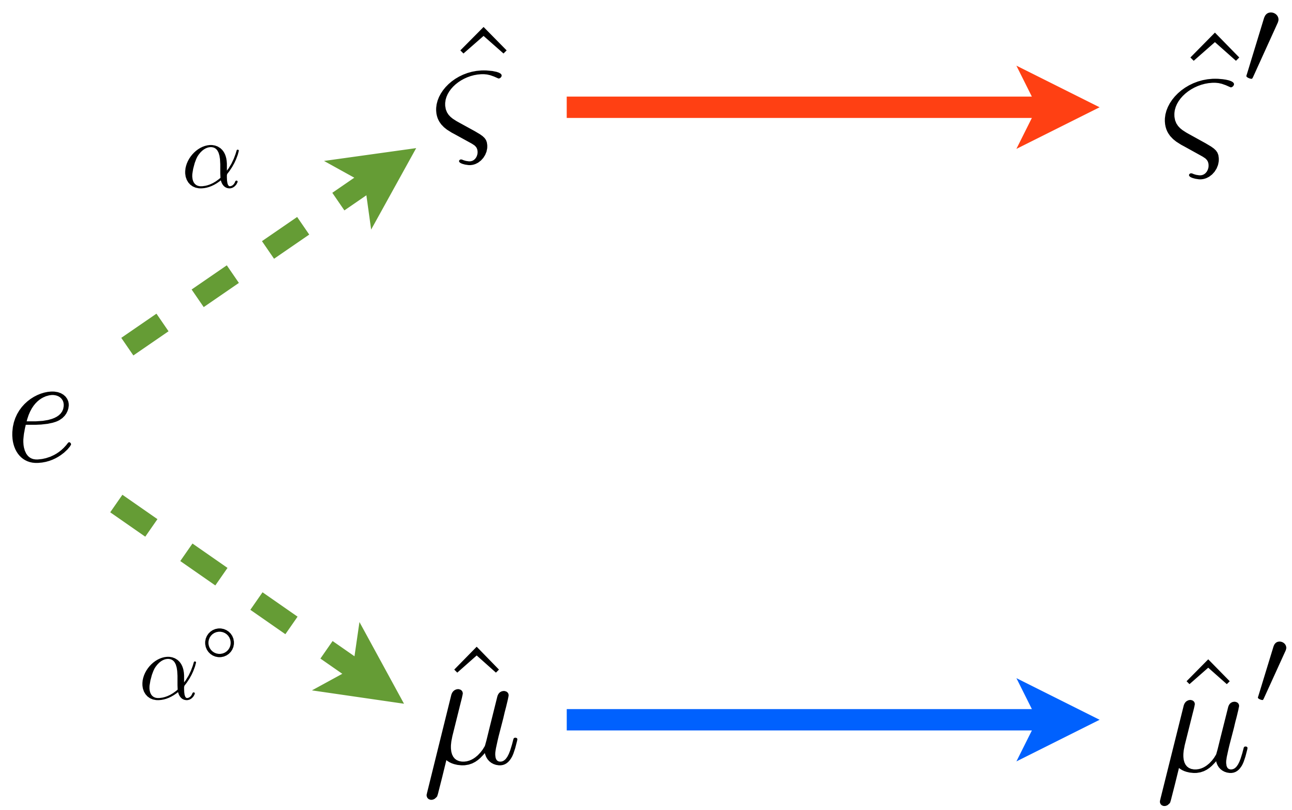
Want strong updates

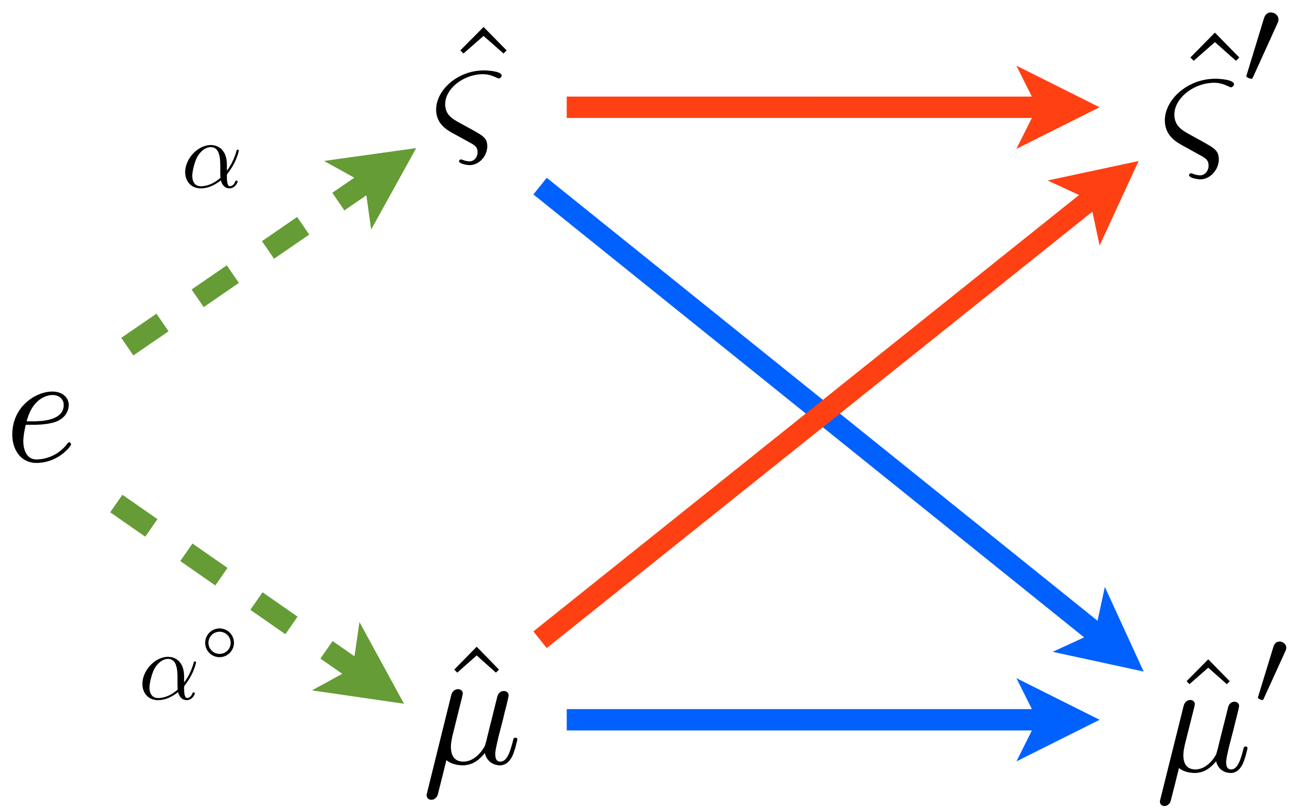
A second abstraction

A direct product

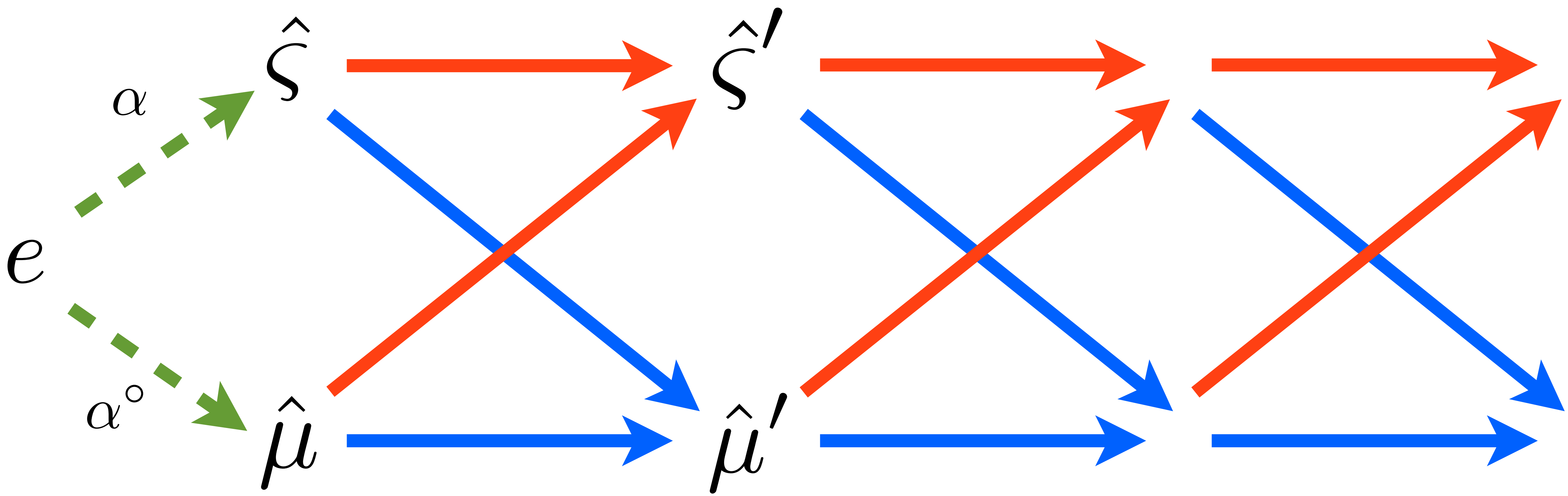
e

$$\begin{tikzcd} & \hat{\varsigma} \\ e \arrow[ur, dashed, "\alpha"] \arrow[dr, dashed, "\alpha^\circ"'] & \\ & \hat{\mu} \end{tikzcd}$$


e
α
α°
$\hat{\varsigma}$
$\hat{\varsigma}'$
$\hat{\mu}$
$\hat{\mu}'$



$\hat{\varsigma}$
$\hat{\varsigma}'$
α
e
α°
$\hat{\mu}$
$\hat{\mu}'$


e
α
α°
$\hat{\varsigma}$
$\hat{\mu}$
$\hat{\varsigma}'$
$\hat{\mu}'$

$$\alpha^\circ : \Sigma \to \hat{M}$$

$$\hat{\mu} \in \hat{M} = \widehat{Addr} \to \{0, 1, \infty\}$$

$$\alpha^{\circ}(call, \rho, \sigma) = \lambda\hat{a}.asize(dom(\sigma) \cap \gamma(\hat{a}))$$

$$\alpha^\circ(call, \rho, \sigma) = \lambda\hat{a}.asize(dom(\sigma) \cap \gamma(\hat{a}))$$

$$asize\,\{\} = 0$$

$$asize\,\{x\} = 1$$

$$asize\,\{x_1, x_2, \ldots\} = \infty$$

$$\frac{\hat{\mu}(\hat{\boldsymbol{a}}) = \infty}{(\llbracket \texttt{(set!-then } v\ \text{æ}\ \mathit{call}\texttt{)} \rrbracket, \hat{\rho}, \hat{\sigma}, \hat{t}, \hat{\mu}) \leadsto (\mathit{call}, \hat{\rho}, \hat{\sigma}', \hat{t}', \hat{\mu})}, \text{ where}$$

$$\hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a} \mapsto \hat{\mathcal{A}}(\text{æ}, \hat{\rho}, \hat{\sigma})]$$

$$\hat{t}' = \widehat{tick}_{\texttt{set}}(\hat{\varsigma})$$

$$\hat{a} = \hat{\rho}(v)$$

$$\frac{\hat{\mu}(\hat{\boldsymbol{a}}) = 1}{(\llbracket \texttt{(set!-then}\ v\ æ\ call\texttt{)} \rrbracket, \hat{\rho}, \hat{\sigma}, \hat{t}, \hat{\mu}) \leadsto (call, \hat{\rho}, \hat{\sigma}', \hat{t}', \hat{\mu})}, \text{ where}$$

$$\hat{\sigma}' = \hat{\sigma} \sqcup [\hat{a} \mapsto \hat{\mathcal{A}}(æ, \hat{\rho}, \hat{\sigma})]$$

$$\hat{t}' = \widehat{tick}_{\texttt{set}}(\hat{\varsigma})$$

$$\hat{a} = \hat{\rho}(v)$$

$$\frac{\hat{\mu}(\hat{\boldsymbol{a}}) = \ 1}{([\![\texttt{(set!-then}\ v\ æ\ call\texttt{)}]\!], \hat{\rho}, \hat{\sigma}, \hat{t}, \hat{\mu}) \leadsto (call, \hat{\rho}, \hat{\sigma}', \hat{t}', \hat{\mu})}, \text{ where}$$

$$\begin{aligned}
\hat{\sigma}' &= \hat{\sigma} \quad [\hat{a} \mapsto \hat{\mathcal{A}}(æ, \hat{\rho}, \hat{\sigma})] \\
\hat{t}' &= \widehat{tick}_{\texttt{set}}(\hat{\varsigma}) \\
\hat{a} &= \hat{\rho}(v)
\end{aligned}$$

ANF

$$
\begin{aligned}
e \in \mathsf{Exp} &::= (\texttt{let}\ ((v\ call))\ e) \\
&\quad\mid\ call \\
&\quad\mid\ æ \\
f, æ \in \mathsf{AExp} &::= v \mid lam \\
lam \in \mathsf{Lam} &::= (\lambda\ (v)\ e) \\
call \in \mathsf{Call} &::= (f\ æ) \\
v \in \mathsf{Var} &\text{ is a set of identifiers}
\end{aligned}
$$

CESK

CESK*

CESK*

$$\Sigma = \mathsf{Exp} \times Env \times Store \times Kont$$

C E S K

$$\Sigma = \mathsf{Exp} \times \mathit{Env} \times \mathit{Store} \times \mathit{Kont}$$

state-space

$$\Sigma = \mathsf{Exp} \times Env \times Store \times Kont$$

$$\Sigma = \mathsf{Exp} \times \mathit{Env} \times \mathit{Store} \times \mathit{Kont}$$

$$\mathit{Store} = \mathit{Addr} \rightarrow \mathit{Clo}$$

$$\begin{aligned} \mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\ \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\ \kappa \in \mathit{Kont} &::= \mathbf{letk}(v, e, \rho, \kappa) \\ &\mid \ \mathbf{halt} \end{aligned}$$

$$\widehat{\Sigma} = \mathsf{Exp} \times \widehat{Env} \times \widehat{Store} \times \widehat{Kont}$$

$$\widehat{Store} = \widehat{Addr} \rightarrow \widehat{Clo}$$

$$\widehat{Env} = \mathsf{Var} \rightarrow \widehat{Addr}$$

$$\widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$

$$\begin{aligned} \kappa \in \widehat{Kont} ::= & \ \mathbf{letk}(v, e, \hat{\rho}, \hat{\kappa}) \\ \mid & \ \mathbf{halt} \end{aligned}$$

$$\widehat{\Sigma} = \mathsf{Exp} \times \widehat{Env} \times \widehat{Store} \times \widehat{Kont}$$

$$\widehat{Store} = \widehat{Addr} \to \widehat{Clo}$$

$$\widehat{Env} = \mathsf{Var} \to \widehat{Addr}$$

$$\widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$

$$\boxed{\kappa \in \widehat{Kont} ::= \mathbf{letk}(v, e, \hat{\rho}, \hat{\kappa})}$$

$$\mid \ \mathbf{halt}$$

$$\Sigma = \mathsf{Exp} \times Env \times Store \times Kont$$

$$\boxed{\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, \kappa)}$$

$$\kappa \in \mathit{Kont} ::= \mathbf{letk}(v, e, \rho, \kappa)$$

$$\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, \kappa)$$

$$\hat{\kappa} \in \widehat{Kont} ::= \mathbf{letk}(v, e, \hat{\rho}, \hat{\kappa})$$

Store-allocate $Kont$

$$Store = Addr \rightarrow Clo$$

$$Store = Addr \to Clo + Kont$$

$$\Sigma = \mathsf{Exp} \times Env \times Store \times Kont$$

$$\Sigma = \mathsf{Exp} \times Env \times Store \times Addr$$

$$\kappa \in \mathit{Kont} ::= \mathbf{letk}(v, e, \rho, \kappa)$$

$$\kappa \in \mathit{Kont} ::= \mathbf{letk}(v, e, \rho, a)$$

$$\kappa \in Kont ::= \mathbf{letk}(v, e, \rho, a)$$

$$
\begin{aligned}
([\![(f\ \text{æ})]\!], \rho, \sigma, a_\kappa) &\Rightarrow (e, \rho', \sigma', a_\kappa), \text{ where} \\
([\![(\lambda\ (v_1 \ldots v_n)\ call)]\!], \rho') &= \mathcal{A}(f, \rho, \sigma) \\
\rho'' &= \rho'[v_i \mapsto a_i] \\
\sigma' &= \sigma[a_i \mapsto \mathcal{A}(\text{æ}_i, \rho, \sigma)] \\
a_i &= alloc(v_i, \ldots)
\end{aligned}
$$

$$
\begin{aligned}
(\text{æ}, \rho, \sigma, a_\kappa) &\Rightarrow (e, \rho'', \sigma', a'_\kappa), \text{ where} \\
\mathbf{letk}(v, e, \rho', a'_\kappa) &= \sigma(a_k) \\
\rho'' &= \rho'[v \mapsto a] \\
\sigma' &= \sigma[a \mapsto \mathcal{A}(\text{æ}, \rho, \sigma)] \\
a &= alloc(v, \ldots)
\end{aligned}
$$

$$
\begin{aligned}
(\llbracket \texttt{(let ((}v\ call\texttt{))}\ e\texttt{)} \rrbracket, \rho, \sigma, a_\kappa) \Rightarrow (call, \rho, \sigma', a'_\kappa), \text{ where} \\
\sigma' &= \sigma[a'_\kappa \mapsto \mathbf{letk}(v, e, \rho, a_\kappa)] \\
a'_\kappa &= alloc(\ldots)
\end{aligned}
$$

Abstract semantics

$$
\begin{aligned}
\hat{\varsigma} \in \hat{\Sigma} &= \mathsf{Exp} \times \widehat{Env} \times \widehat{Store} \times \widehat{Addr} \\
\hat{\rho} \in \widehat{Env} &= \mathsf{Var} \rightharpoonup \widehat{Addr} \\
\hat{\sigma} \in \widehat{Store} &= \widehat{Addr} \to \mathcal{P}(\widehat{Clo} + \widehat{Kont}) \\
\widehat{clo} \in \widehat{Clo} &= \mathsf{Lam} \times \widehat{Env} \\
\hat{\kappa} \in \widehat{Kont} &::= \mathbf{letk}(v, e, \hat{\rho}, \hat{a}) \\
&\mid \quad \mathbf{halt} \\
\hat{a} \in \widehat{Addr} & \text{ is a finite set of addresses}
\end{aligned}
$$

Multithreading

CESK

(Felleisen & Friedman, 1986)

P(CEK)S

P(CEK*)S

P(CEK*)S

$$\Sigma = \mathsf{Exp} \times Env \times Store \times Kont$$

$$\mathsf{Exp} \times Env \times Kont$$

$$Thread = \mathsf{Exp} \times Env \times Kont \times TID$$

$$\Sigma = Thread \times Store$$

$$\Sigma = \mathcal{P}(Thread) \times Store$$

$\mathcal{P}(Thread)$

$$\mathcal{P}(Thread) = \mathcal{P}(\mathsf{Exp} \times Env \times Addr \times TID)$$

$$
\begin{aligned}
\mathcal{P}(Thread) &= \mathcal{P}(\mathsf{Exp} \times Env \times Addr \times TID) \\
&\cong TID \to \mathcal{P}(\mathsf{Exp} \times Env \times Addr)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}(Thread) &= \mathcal{P}(\mathsf{Exp} \times Env \times Addr \times TID) \\
&\cong TID \to \mathcal{P}(\mathsf{Exp} \times Env \times Addr) \\
&\approx TID \to \mathsf{Exp} \times Env \times Addr
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}(Thread) &= \mathcal{P}(\mathsf{Exp} \times Env \times Addr \times TID) \\
&\cong TID \to \mathcal{P}(\mathsf{Exp} \times Env \times Addr) \\
&\approx TID \to \mathsf{Exp} \times Env \times Addr \\
&= Threads
\end{aligned}
$$

$$\Sigma = Threads \times Store$$

$$
\begin{aligned}
\Sigma &= \mathit{Threads} \times \mathit{Store} \\
\mathit{Threads} &= \mathit{TID} \to \mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \\
\mathit{Store} &= \mathit{Addr} \to \mathit{Clo} + \mathit{Kont}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Env} &= \mathsf{Var} \to \mathit{Addr} \\
\mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\mathit{Kont} &::= \mathbf{letk}(v, e, \rho, a) \\
&\quad\mid\ \mathbf{halt}
\end{aligned}
$$

$$
\begin{aligned}
\Sigma &= \mathit{Threads} \times \mathit{Store} \\
\mathit{Threads} &= \mathit{TID} \rightharpoonup \mathsf{Exp} \times \mathit{Env} \times \mathit{Addr} \\
\mathit{Store} &= \mathit{Addr} \rightharpoonup \mathit{Clo} + \mathit{Kont}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Env} &= \mathsf{Var} \rightharpoonup \mathit{Addr} \\
\mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\mathit{Kont} &::= \mathbf{letk}(v, e, \rho, a) \\
&\mid \ \mathbf{halt}
\end{aligned}
$$

$$
\begin{aligned}
\Sigma &= \mathit{Threads} \times \mathit{Store} \\
\mathit{Threads} &= \mathit{TID} \rightarrow \mathcal{P}(\mathsf{Exp} \times \mathit{Env} \times \mathit{Addr}) \\
\mathit{Store} &= \mathit{Addr} \rightarrow \mathcal{P}(\mathit{Clo} + \mathit{Kont})
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Env} &= \mathsf{Var} \rightarrow \mathit{Addr} \\
\mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\mathit{Kont} &::= \mathbf{letk}(v, e, \rho, a) \\
&\quad \mid \ \mathbf{halt}
\end{aligned}
$$

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store}$$
$$\widehat{Threads} = \widehat{TID} \to \mathcal{P}(\mathsf{Exp} \times \widehat{Env} \times \widehat{Addr})$$
$$\widehat{Store} = \widehat{Addr} \to \mathcal{P}(\widehat{Clo} + \widehat{Kont})$$

$$\widehat{Env} = \mathsf{Var} \to \widehat{Addr}$$
$$\widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$
$$\widehat{Kont} ::= \mathbf{letk}(v, e, \hat{\rho}, \hat{a})$$
$$\mid \ \mathbf{halt}$$

$$\widehat{TCount} = \widehat{TID} \to \{0, 1, \infty\}$$

$$\hat{\Sigma} = \widehat{Threads} \times \widehat{Store} \times \widehat{TCount}$$

Pushdown

$$
\begin{aligned}
c \in \mathit{Conf} &= \mathsf{Exp} \times \mathit{Env} \times \mathit{Store} \times \mathit{Kont} \\
\rho \in \mathit{Env} &= \mathsf{Var} \rightharpoonup \mathit{Addr} \\
\sigma \in \mathit{Store} &= \mathit{Addr} \to \mathit{Clo} \\
\mathit{clo} \in \mathit{Clo} &= \mathsf{Lam} \times \mathit{Env} \\
\kappa \in \mathit{Kont} &= \mathit{Frame}^{*} \\
\phi \in \mathit{Frame} &= \mathsf{Var} \times \mathsf{Exp} \times \mathit{Env} \\
a \in \mathit{Addr} & \text{ is an infinite set of addresses}
\end{aligned}
$$

$$\hat{c} \in \widehat{Conf} = \mathsf{Exp} \times \widehat{Env} \times \widehat{Store} \times \widehat{Kont}$$

$$\hat{\rho} \in \widehat{Env} = \mathsf{Var} \rightharpoonup \widehat{Addr}$$

$$\hat{\sigma} \in \widehat{Store} = \widehat{Addr} \to \mathcal{P}\left(\widehat{Clo}\right)$$

$$\widehat{clo} \in \widehat{Clo} = \mathsf{Lam} \times \widehat{Env}$$

$$\hat{\kappa} \in \widehat{Kont} = \widehat{Frame}^{*}$$

$$\hat{\phi} \in \widehat{Frame} = \mathsf{Var} \times \mathsf{Exp} \times \widehat{Env}$$

$\hat{a} \in \widehat{Addr}$ is a *finite* set of addresses control states stack

$$\hat{c} \in \widehat{Conf} = \mathsf{Exp} \times \widehat{Env} \times \widehat{Store} \times \widehat{Kont}$$

$$\hat{c} \in \widehat{Conf} = \overbrace{\mathsf{Exp} \times \widehat{Env} \times \widehat{Store}}^{\text{control states}} \times \overbrace{\widehat{Kont}}^{\text{stack}}$$

Control-state reachability

Thanks!