$$\frac{\langle c_0, 6 \rangle \overset{\lambda}{\longrightarrow} 6'}{\langle c_0; c_1, 6 \rangle \overset{\lambda}{\longrightarrow} \langle c_1, 6' \rangle}$$

$$\frac{\langle c_0, 6 \rangle \overset{\lambda}{\longrightarrow} \langle c_0', 6' \rangle}{\langle c_0; c_1, 6 \rangle \overset{\lambda}{\longrightarrow} \langle c_0'; c_1, 6' \rangle}$$

$$\frac{\langle c_0, 6 \rangle \overset{\lambda}{\longrightarrow} \langle abort, 6' \rangle}{\langle c_0; c_1, 6 \rangle \overset{\lambda}{\longrightarrow} \langle abort, 6' \rangle}$$

$$\frac{\langle c, [6 \mid v: [\![e]\!]6] \rangle \overset{\lambda}{\longrightarrow} 6'}{\langle \text{newvar } v := e \text{ in } c, 6 \rangle \overset{\lambda}{\longrightarrow} [6' \mid v: 6(v)]}$$

$$\frac{\langle c, [6 \mid v: [\![e]\!]6] \rangle \overset{\lambda}{\longrightarrow} \langle abort, 6' \rangle}{\langle \text{newvar } v := e \text{ in } c, 6 \rangle \overset{\lambda}{\longrightarrow} \langle abort, [6' \mid v: 6(v)] \rangle}$$

$$\frac{\langle c, [6 \mid v: [\![e]\!]6] \rangle \overset{\lambda}{\longrightarrow} \langle c', 6' \rangle}{\langle \text{newvar } v := e \text{ in } c, 6 \rangle \overset{\lambda}{\longrightarrow} \langle \text{newvar } v := 6(v) \text{ in } c', [6' \mid v: 6(v)] \rangle}$$

Whenever an old rule contains a premise, we copy the rule and put λ above $\rightarrow$ in the premise and the conclusion, so that the label λ gets propagated from the execution of a subcommand to that of the original command.

④ This operational semantics corresponds to the denotational semantics that we studied. ~~Instead~~ The correspondence is formalised by the function F in p134 of the textbook. Intuitively, F runs a configuration ~~until~~ until it finishes, or outputs a number, or waits for an input. F then returns what it gets when it completes this execution. ~~to the end~~ In a sense, the correspondence says that the denotational semantics comes from the operational semantics after unobservable intermediate states are abstracted away. For detail, look at the textbook.