

③ In this chapter, we will primarily study continuations as a new language feature (2nd point) and as a tool in the semantics. But we will say a few words on the cps transformation (3rd point).

④ Another big part of Chap 12 is a semantic version of defunctionalisation, a technique to replace higher-order functions by records. This is one of the key techniques in compilation. Also, a large number of PL researchers, especially those working on object-oriented languages, use defunctionalised denotational semantics.

2. Continuation Semantics.

① One way to understand the idea of continuation is to rewrite the semantics of the eager functional programming language in the previous chapter, using continuation. In this setting, continuations are continuous functions from V to V_* .

$$V_{\text{cont}} \stackrel{\text{def}}{=} [V \rightarrow_c V_*]$$

They represent the rest of computation. If we provide the value a of the current computation step to a continuation k , the computation will perform all the remaining (represented by $k(a)$)

computation steps and output the final result, which is the value of $k(a)$.

③ Let's define this continuation semantics more formally. Recall the semantic domains and predomains that we used in the semantics of the eager functional language in the previous chapter.

$$V_* = (V + \{\text{err}, \text{typerr}\})_{\perp}$$

$$V \stackrel{\phi}{\cong} V_{\text{int}} + V_{\text{bool}} + V_{\text{fun}} + V_{\text{tuple}} + V_{\text{alt}}$$

$$V_{\text{int}} = \mathbb{Z} \quad V_{\text{bool}} = \mathbb{B} \quad V_{\text{fun}} = V \rightarrow_c V_*$$

$$V_{\text{tuple}} = \bigcup_{n \geq 0} V^n \quad V_{\text{alt}} = \mathbb{N} \times V$$

$$\llbracket - \rrbracket \in [\langle \text{exp} \rangle \rightarrow_c [V \xrightarrow{\langle \text{var} \rangle} V_*]]$$

③ The key idea of continuation semantics is to add a new input to $\llbracket - \rrbracket$ that represents continuation, and also to add a new parameter to each function that again represents continuation. This means the following two changes:

$$V_{\text{cont}} = V \rightarrow_c V_*$$

$$V_{\text{fun}} = V \rightarrow_c [V_{\text{cont}} \rightarrow_c V_*]$$

new continuation parameters.

$$\llbracket - \rrbracket \in [\langle \text{exp} \rangle \rightarrow_c [V \xrightarrow{\langle \text{var} \rangle} [V_{\text{cont}} \rightarrow_c V_*]]]$$

The remaining parts of the semantic predomains and domains are defined in the same way as before.

④ Altering V_{fun} and the form of $\llbracket - \rrbracket$ has a huge impact on the defining clauses of $\llbracket - \rrbracket$. When defining $\llbracket - \rrbracket$, we now have to specify how a given continuation is used and modified. Observe this change in the following definition of $\llbracket - \rrbracket$: