## *Traduction d'un programme fouine en fouine CPS*

Dans la suite, on notera $\lambda x.\ e$ au lieu de `fun` $x$ `->` $e$, pour abréger les notations...
On notera en cyan les variables fraîches.

- $[\![n]\!] := \lambda k.\ (\mathit{fst}\ k)\ n$
- $[\![b]\!] := \lambda k.\ (\mathit{fst}\ k)\ b$
- $[\![()]\!] := \lambda k.\ (\mathit{fst}\ k)\ ()$
- $[\![x]\!] := \lambda k.\ (\mathit{fst}\ k)\ x$
- $[\![\lambda x.\ e]\!] := \lambda k.\ (\mathit{fst}\ k)\ (\lambda x.\ [\![e]\!])$
- $[\![e_1\ e_2]\!] = \lambda k.\ [\![e_2]\!]\ (\lambda v.\ [\![e_1]\!]\ (\lambda f.\ f\ v\ k, \mathit{snd}\ k), \mathit{snd}\ k)$
- $[\![e_1 \circledast e_2]\!] := \lambda k.\ [\![e_2]\!]\ (\lambda v_2.\ [\![e_1]\!]\ (\lambda v_1.\ (\mathit{fst}\ k)\ (v_1 \circledast v_2)), \mathit{snd}\ k), \mathit{snd}\ k)$
- $[\![\texttt{if}\ b\ \texttt{then}\ e_1\ \texttt{else}\ e_2]\!] := \lambda k.\ [\![b]\!]\ (\lambda v.\ \texttt{if}\ v\ \texttt{then}\ [\![e_1]\!]\ k\ \texttt{else}\ [\![e_2]\!]\ k, \mathit{snd}\ k)$
- $[\![\circledast\ e]\!] := \lambda k.\ [\![e]\!]\ (\lambda v.\ (\mathit{fst}\ k)\ (\circledast\ v), \mathit{snd}\ k)$
- $[\![e_1\ ;\ e_2]\!] := \lambda k.\ [\![e_1]\!]\ (\lambda\_.\ [\![e_2]\!]\ k, \mathit{snd}\ k)$
- $[\![\mathrm{C}(e_1, \ldots, e_n)]\!] := \lambda k.\ [\![e_n]\!]\ (\lambda v_n.\ \ldots\ ([\![e_1]\!]\ (\lambda v_1.\ (\mathit{fst}\ k)\ \ \mathrm{C}(v_1, \ldots, v_n), \mathit{snd}\ k)\ldots), \mathit{snd}\ k)$
- $[\![\texttt{while}\ b\ \texttt{do}\ e]\!] := \texttt{let rec}\ \mathit{boucle}\ k =$
  $[\![b]\!]\ (\lambda v.$
  $\texttt{if}\ v\ \texttt{then}\ [\![e]\!]\ (\lambda\_.\ \mathit{boucle}\ k, \mathit{snd}\ k)$
  $\texttt{else}\ (\mathit{fst}\ k)\ ()$
  $\texttt{in}\ \mathit{boucle}$
- $[\![\texttt{let rec}\ f = e\ \texttt{in}\ e']\!] := \lambda k.\ \texttt{let rec}\ f = [\![e]\!]\ \texttt{in}\ [\![e']\!]\ k$
- $[\![\texttt{for}\ i = e_1\ \texttt{to}\ e_2\ \texttt{do}\ e_3\ \texttt{done}]\!] := \lambda k.\ [\![e_1]\!]\ (\lambda v_1.\ [\![e_2]\!]\ (\lambda v_2.$
  $\texttt{let rec}\ \mathit{boucle}\ i\ k =$
  $\texttt{if}\ i \leq v_2\ \texttt{then}\ [\![e_3]\!]\ (\lambda\_\ .\ \mathit{boucle}\ (i+1)\ k, \mathit{snd}\ k)$
  $\texttt{else}\ (\mathit{fst}\ k)\ ()$
  $\texttt{in}\ \mathit{boucle}\ v_1))$
- $[\![\texttt{for}\ i = e_1\ \texttt{downto}\ e_2\ \texttt{do}\ e_3\ \texttt{done}]\!] := \lambda k.\ [\![e_1]\!]\ (\lambda v_1.\ [\![e_2]\!]\ (\lambda v_2.$
  $\texttt{let rec}\ \mathit{boucle}\ i\ k =$
  $\texttt{if}\ i \geq v_1\ \texttt{then}\ [\![e_3]\!]\ (\lambda\_\ .\ \mathit{boucle}\ (i-1)\ k, \mathit{snd}\ k)$
  $\texttt{else}\ (\mathit{fst}\ k)\ ()$
  $\texttt{in}\ \mathit{boucle}\ v_2))$
- $[\![\texttt{match}\ e\ \texttt{with}\ p_1\ \texttt{when}\ e'_1\ \texttt{->}\ e_1 \mid \cdots \mid p_n\ \texttt{when}\ e'_n\ \texttt{->}\ e_n]\!] := \lambda k.\ [\![e]\!]\ (\cdots(\lambda \mathit{match}_{\text{next}}.\ \lambda v.$
  $\texttt{match}\ v\ \texttt{with}$
  $\mid p_1\ \texttt{->}\ [\![e'_1]\!]\ (\lambda v'.$
  $\texttt{if}\ v'\ \texttt{then}\ [\![e_1]\!]\ k$
  $\texttt{else}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\mid \_\ \texttt{->}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $(\lambda \mathit{match}_{\text{next}}.\ \lambda v.$
  $\texttt{match}\ v\ \texttt{with}$
  $\mid p_2\ \texttt{->}\ [\![e'_2]\!]\ (\lambda v'.$
  $\texttt{if}\ v'\ \texttt{then}\ [\![e_2]\!]\ k$
  $\texttt{else}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\mid \_\ \texttt{->}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\vdots$
  $(\lambda \mathit{match}_{\text{next}}.\ \lambda v.$
  $\texttt{match}\ v\ \texttt{with}$
  $\mid p_n\ \texttt{->}\ [\![e'_n]\!]\ (\lambda v'.$
  $\texttt{if}\ v'\ \texttt{then}\ [\![e_n]\!]\ k$
  $\texttt{else}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\mid \_\ \texttt{->}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $(\lambda\_.\ (\mathit{snd}\ k)\ \mathbf{MatchError})\cdots)$
- $[\![\texttt{try}\ e\ \texttt{with}\ p_1\ \texttt{when}\ e'_1\ \texttt{->}\ e_1 \mid \cdots \mid p_n\ \texttt{when}\ e'_n\ \texttt{->}\ e_n]\!] := \lambda k.\ [\![e]\!]\ (\mathit{fst}\ k, \lambda v.$
  $(\cdots(\lambda \mathit{match}_{\text{next}}.\ \lambda v.$
  $\texttt{match}\ v\ \texttt{with}$
  $\mid p_1\ \texttt{->}\ [\![e'_1]\!]\ (\lambda v'.$
  $\texttt{if}\ v'\ \texttt{then}\ [\![e_1]\!]\ k$
  $\texttt{else}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\mid \_\ \texttt{->}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $(\lambda \mathit{match}_{\text{next}}.\ \lambda v.$
  $\texttt{match}\ v\ \texttt{with}$
  $\mid p_2\ \texttt{->}\ [\![e'_2]\!]\ (\lambda v'.$
  $\texttt{if}\ v'\ \texttt{then}\ [\![e_2]\!]\ k$
  $\texttt{else}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\mid \_\ \texttt{->}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\vdots$
  $(\lambda \mathit{match}_{\text{next}}.\ \lambda v.$
  $\texttt{match}\ v\ \texttt{with}$
  $\mid p_n\ \texttt{->}\ [\![e'_n]\!]\ (\lambda v'.$
  $\texttt{if}\ v'\ \texttt{then}\ [\![e_n]\!]\ k$
  $\texttt{else}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $\mid \_\ \texttt{->}\ \mathit{match}_{\text{next}}\ v,$
  $\mathit{snd}\ k)$
  $(\mathit{snd}\ k)\ )\cdots)$
- $[\![\texttt{raise}\ e]\!] := \lambda k.\ [\![e]\!]\ (\mathit{snd}\ k, \mathit{snd}\ k)$

On définit

- $\mathit{id} := \lambda x.\ x$
- $\mathit{fst} := \lambda (x, y).\ x$
- $\mathit{snd} := \lambda (x, y).\ y$