# PFPL Syntax Master Chart*

Robert Harper

December 16, 2022

Function types:

| | |
|---|---|
| $\texttt{fun}(\tau_1\,;\tau_2)$ | $\tau_1 \to \tau_2$ |
| $\texttt{lam}[\tau_1\,;\tau_2](x\,.\,e)$ | $\lambda(x\,.\,e)$ |
| $\texttt{ap}[\tau_1\,;\tau_2](e_1\,;e_2)$ | $\texttt{ap}(e_1\,;e_2)$ |

Product types:[1]

| | | |
|---|---|---|
| $\texttt{unit}$ | $\mathbf{1}$ | |
| $\texttt{null}$ | $\langle\rangle$ | |
| $\texttt{prod}(\tau_1\,;\tau_2)$ | $\tau_1 \times \tau_2$ | |
| $\texttt{proj}\langle i\rangle[\tau_1\,;\tau_2](e)$ | $e \cdot i$ | $(i = \texttt{1}, \texttt{2})$ |
| $\texttt{pair}[\tau_1\,;\tau_2](e_1\,;e_2)$ | $\langle e_1, e_2\rangle$ | |
| $\texttt{vprod}\langle I\rangle(\tau_I)$ | $\times_{i\in I}(i \hookrightarrow \tau_i)$ | |
| $\texttt{vtuple}\langle I\rangle[\tau_I](e_I)$ | $\langle i \hookrightarrow e_i \mid i \in I\rangle$ | |
| $\texttt{vproj}\langle i\rangle\langle I\rangle[\tau_I](e)$ | $e \cdot i$ | $(i \in I)$ |

* © 2022 Robert Harper. All Rights Reserved.

[1] Variadic operators are implicitly indexed by finite sets $I$ of labels. Then $\tau_I$ stands for the finite map $i \hookrightarrow \tau_i \mid i \in I$ and $e_I$ stands for $i \hookrightarrow e_i \mid i \in I$.

Sum types:

| | | |
|---|---|---|
| $\texttt{void}$ | $\mathbf{0}$ | |
| $\texttt{absurd}[\rho](e)$ | $\texttt{absurd}(e)$ | |
| $\texttt{sum}(\tau_1\,;\tau_2)$ | $\tau_1 + \tau_2$ | |
| $\texttt{inj}\langle i\rangle[\tau_1\,;\tau_2](e)$ | $i \cdot e$ | $(i = 1, 2)$ |
| $\texttt{case}[\tau_1\,;\tau_2\,;\rho](e\,;x\,.\,e_1\,;x\,.\,e_2)$ | $\texttt{case}\ e\,\{\,x\,.\,e_1 \mid x\,.\,e_2\,\}$ | |
| $\texttt{bool}$ | $\mathbf{2}$ | |
| $\texttt{true}$ | $\texttt{true}$ | |
| $\texttt{false}$ | $\texttt{false}$ | |
| $\texttt{if}[\rho](e\,;e_1\,;e_2)$ | $\texttt{if}(e\,;e_1\,;e_2)$ | |
| $\texttt{vsum}\langle I\rangle(\tau_I)$ | $+_{i\in I}(i \hookrightarrow \tau_i)$ | |
| $\texttt{vinj}\langle i\rangle\langle I\rangle[\tau_I](e)$ | $i \cdot e$ | $(i \in I)$ |
| $\texttt{vcase}\langle I\rangle[\tau_I\,;\rho](e\,;x\,.\,e'_I)$ | $\texttt{vcase}\ e\,\{\,i \hookrightarrow x\,.\,e'_i \mid i \in I\,\}$ | |

Inductive and coinductive types:

| | |
|---|---|
| $\texttt{ind}(t\,.\,\tau)$ | $\texttt{ind}(t\,.\,\tau)$ |
| $\texttt{in}[t\,.\,\tau](e)$ | $\texttt{in}(e)$ |
| $\texttt{rec}[t\,.\,\tau\,;\rho](e\,;x\,.\,e')$ | $\texttt{rec}(e\,;x\,.\,e')$ |
| $\texttt{nat}$ | $\texttt{nat}$ |
| $\texttt{zero}$ | $\texttt{zero}$ |
| $\texttt{succ}(e)$ | $\texttt{succ}(e)$ |
| $\texttt{natit}[\rho](e\,;e_0\,;x\,.\,e_1)$ | $\texttt{natit}\ e\,\{\,e_0 \mid x\,.\,e_1\}$ |
| $\texttt{ifz}[\rho](e\,;e_0\,;x\,.\,e_1)$ | $\texttt{ifz}\ e\,\{\,e_0 \mid x\,.\,e_1\}$ |
| $\texttt{list}(\tau)$ | $\tau\ \texttt{list}$ |
| $\texttt{nil}[\tau]$ | $\texttt{nil}$ |
| $\texttt{cons}[\tau](e_1\,;e_2)$ | $\texttt{cons}(e_1\,;e_2)$ |
| $\texttt{listrec}[\rho\,;\tau](e\,;e_1\,;x,y\,.\,e_2)$ | $\texttt{listrec}\ e\,\{\,e_1 \mid x,y\,.\,e_2\}$ |
| $\texttt{listcase}[\rho\,;\tau](e\,;e_1\,;x,y\,.\,e_2)$ | $\texttt{listcase}\ e\,\{\,e_1 \mid x,y\,.\,e_2\}$ |
| $\texttt{coi}(t\,.\,\tau)$ | $\texttt{coi}(t\,.\,\tau)$ |
| $\texttt{out}[t\,.\,\tau](e)$ | $\texttt{out}(e)$ |
| $\texttt{gen}[t\,.\,\tau\,;\sigma](e\,;x\,.\,e')$ | $\texttt{gen}(e\,;x\,.\,e')$ |
| $\texttt{conat}$ | $\texttt{conat}$ |
| $\texttt{stream}(\tau)$ | $\tau\ \texttt{stream}$ |

Polymorphic types:

| | |
|---|---|
| $\texttt{All}(t\,.\,\tau)$ | $\forall(t\,.\,\tau)$ |
| $\texttt{Lam}[t\,.\,\tau](t\,.\,e)$ | $\Lambda(t\,.\,e)$ |
| $\texttt{Ap}[t\,.\,\tau](e\,;\sigma)$ | $\texttt{Ap}(e\,;\sigma)$ |
| $\texttt{Some}(t\,.\,\tau)$ | $\exists(t\,.\,\tau)$ |
| $\texttt{Pack}[t\,.\,\tau](\rho\,;e)$ | $\texttt{Pack}(\rho\,;e)$ |
| $\texttt{Open}[t\,.\,\tau\,;\rho](e\,;t,x\,.\,e')$ | $\texttt{Open}(e\,;t,x\,.\,e')$ |

Continuation types:

| | |
|---|---|
| $\texttt{ans}$ | $\texttt{ans}$ |
| $\texttt{yes}$ | $\texttt{yes}$ |
| $\texttt{no}$ | $\texttt{no}$ |
| $\texttt{cont}(\tau)$ | $\tau\,\texttt{cont}$ |
| $\texttt{cont}[\tau](k)$ | $\texttt{cont}(k)$ |
| $\texttt{letcc}[\tau](x\,.\,e)$ | $\texttt{letcc}(x\,.\,e)$ |
| $\texttt{throw}[\tau\,;\rho](e_1\,;e_2)$ | $\texttt{throw}(e_1\,;e_2)$ |
| $\texttt{emp}[\tau]$ | $\bullet$ |
| $\texttt{ext}[\tau_1\,;\tau_2](k\,;f)$ | $k\;\texttt{;}\;f$ |

Recursive types:

| | |
|---|---|
| $\texttt{parr}(\tau_1\,;\tau_2)$ | $\tau_1 \rightharpoonup \tau_2$ |
| $\texttt{fun}[\tau_1\,;\tau_2](f,x\,.\,e)$ | $\texttt{fun}(f,x\,.\,e)$ |
| $\texttt{ap}[\tau_1\,;\tau_2](e_1\,;e_2)$ | $\texttt{ap}(e_1\,;e_2)$ |
| $\texttt{fix}[\tau](x\,.\,e)$ | $\texttt{fix}(x\,.\,e)$ |
| $\texttt{rec}(t\,.\,\tau)$ | $\texttt{rec}(t\,.\,\tau)$ |
| $\texttt{fold}[t\,.\,\tau](e)$ | $\texttt{fold}(e)$ |
| $\texttt{unfold}[t\,.\,\tau](e)$ | $\texttt{unfold}(e)$ |
| $\texttt{self}(\tau)$ | $\texttt{self}(\tau)$ |
| $\texttt{roll}[\tau](e)$ | $\texttt{roll}(e)$ |
| $\texttt{self}[\tau](x\,.\,e)$ | $\texttt{self}(x\,.\,e)$ |
| $\texttt{lam}(x\,.\,M)$ | $\lambda(x\,.\,M)$ |
| $\texttt{app}(M_1\,;M_2)$ | $M_1(M_2)$ |
| $\texttt{I}$ | $\texttt{I}$ |
| $\texttt{K}$ | $\texttt{K}$ |
| $\texttt{S}$ | $\texttt{S}$ |
| $\texttt{B}$ | $\texttt{B}$ |

Commands:[2]

| | |
|---|---|
| $\mathtt{cmd}(\tau)$ | $\tau\ \mathtt{cmd}$ |
| $\mathtt{cmd}[\tau](m)$ | $\mathtt{cmd}(m)$ |
| $\mathtt{ret}[\tau](e)$ | $\mathtt{ret}(e)$ |
| $\mathtt{bnd}[\tau\,;\rho](e\,;x\,.\,m)$ | $\mathtt{bnd}\ x \leftarrow e\ \mathtt{;}\ m$ |
| $\mathtt{dcl}[\tau\,;\rho](e\,;a\,.\,m)$ | $\mathtt{dcl}\ a\ \mathtt{:=}\ e\ \mathtt{;}\ m$ |
| $\mathtt{ref}\langle a\rangle$ | $\mathtt{\&}\,a$ |
| $\mathtt{get}\langle a\rangle$ | $\mathtt{!}\,a$ |
| $\mathtt{getref}[\tau](e)$ | $\mathtt{*}e$ |
| $\mathtt{set}\langle a\rangle(e)$ | $a\ \mathtt{:=}\ e$ |
| $\mathtt{setref}[\tau](e_1\,;e_2)$ | $e_1\ \mathtt{*=}\ e_2$ |

Polarized types:

| | |
|---|---|
| $\mathtt{F}(\tau^{+})$ | $\uparrow(\tau^{+})$ |
| $\mathtt{ret}[\tau^{+}](v)$ | $\mathtt{ret}(v)$ |
| $\mathtt{let}[\tau^{-}\,;\rho^{-}](e\,;x\,.\,e')$ | $\mathtt{let}(e\,;x\,.\,e')$ |
| $\mathtt{U}(\tau^{-})$ | $\downarrow(\tau^{-})$ |
| $\mathtt{thunk}[\tau^{-}](e)$ | $\mathtt{thunk}(e)$ |
| $\mathtt{force}(v)$ | $\mathtt{force}(v)$ |

Parallel types:

| | |
|---|---|
| $\mathtt{tens}(\tau_1\,;\tau_2)$ | $\tau_1 \otimes \tau_2$ |
| $\mathtt{with}[\tau_1\,;\tau_2](v_1\,;v_2)$ | $v_1 \otimes v_2$ |
| $\mathtt{split}[\tau_1\,;\tau_2\,;\rho](v\,;x_1,x_2\,.\,e)$ | $\mathtt{split}\,v\,\{x_1,x_2\,.\,e\}$ |
| $\mathtt{and}(\tau_1\,;\tau_2)$ | $\tau_1\ \&\ \tau_2$ |
| $\mathtt{both}(e_1\,;e_2)$ | $e_1\ \&\ e_2$ |
| $\mathtt{par}(e\,;x,y\,.\,e')$ | $\mathtt{par}(e\,;x,y\,.\,e')$ |

[2] Symbols $a$ are constants of sort `loc`.

Modules:

| | |
|---|---|
| $\mathtt{st}$ | $\mathtt{st}$ |
| $\mathtt{type}$ | $\mathtt{type}$ |
| $\mathtt{val}(\tau)$ | $\mathtt{val}(\tau)$ |
| $\mathtt{Ext}(S\,;M)$ | $\{S \mid M\}$ |
| $\mathtt{in}[S\,;M](M')$ | $\mathtt{in}(M')$ |
| $\mathtt{out}[S\,;M](M')$ | $\mathtt{out}(M')$ |
| $\mathtt{Comp}(S)$ | $S\,\mathtt{Comp}$ |
| $\mathtt{Pi}(S_1\,;X\,.\,S_2)$ | $X{:}S_1 \to S_2$ |
| $\mathtt{fun}[S_1](X\,.\,M_2)$ | $\mathtt{fun}\,X : S_1\;\mathtt{in}\;M_2$ |
| $\mathtt{inst}[S_1\,;X\,.\,S_2](M_1\,;M_2)$ | $M_1(M_2)$ |
| $\mathtt{Sig}(S_1\,;X\,.\,S_2)$ | $X{:}S_1 \times S_2$ |
| $\mathtt{str}[S_1\,;X\,.\,S_2](M_1\,;M_2)$ | $\mathtt{str}\,M_1\;\mathtt{in}\;M_2$ |
| $\mathtt{fst}[S_1\,;X\,.\,S_2](M)$ | $M \cdot \mathtt{1}$ |
| $\mathtt{snd}[S_1\,;X\,.\,S_2](M)$ | $M \cdot \mathtt{2}$ |

Processes:

| | |
|---|---|
| $\mathtt{one}$ | $1$ |
| $\mathtt{par}(p_1\,;p_2)$ | $p_1 \otimes p_2$ |
| $\mathtt{null}$ | $0$ |
| $\mathtt{or}(p_1\,;p_2)$ | $p_1 \oplus p_2$ |
| $\mathtt{que}\langle a\rangle(p)$ | $?\langle a\rangle(p)$ |
| $\mathtt{sig}\langle a\rangle(p)$ | $!\langle a\rangle(p)$ |
| $\mathtt{rcv}\langle a\rangle(x\,.\,p)$ | $?\langle a\rangle(x\,.\,p)$ |
| $\mathtt{snd}\langle a\rangle(e\,;p)$ | $\mathtt{!}\langle a\rangle(e\,;p)$ |
| $\mathtt{asnd}\langle a\rangle(e)$ | $!\langle a\rangle(e)$ |
| $\mathtt{sync}(\varepsilon)$ | $\mathtt{sync}(\varepsilon)$ |
| $\mathtt{new}[\tau](a\,.\,p)$ | $\nu(a\,.\,p)$ |
| $\mathtt{sil}$ | $\epsilon$ |
| $\mathtt{sig}\langle a\rangle$ | $a\,!$ |
| $\mathtt{que}\langle a\rangle$ | $a\,?$ |
| $\mathtt{snd}\langle a\rangle(e)$ | $a\,!\,e$ |
| $\mathtt{rcv}\langle a\rangle(e)$ | $a\,?\,e$ |
| $\mathtt{any}(e)$ | $?\,e$ |
| $\mathtt{emit}(e)$ | $!\,e$ |