This document doesn't add anything new yet.

Introduce an abstract machine semantics? uscheme in chapter 3.

First, talk about syntactic forms and forms of judgement.

1 for verse, one for PM.

Forms of judgement

First, the base translation:

Translation from P to V-style if-then-else:

case `v` *of* $\_ \to e$
$\triangleq$
`if (True) e e`

TODO FIX if true e e to if true e impossible

$$\textsc{Verse-IfTrueBindings}\ \frac{}{\textsc{Verse-If}(\texttt{True}\ e\ e) \rightarrowtail \{\}}$$

$$\textsc{Verse-IfTrueEval}\ \frac{}{\textsc{Verse-If}(\texttt{True}\ e\ e) \rightarrowtail e}$$

Maybe want these as Verse-IfLiteralBindings?

$$\textsc{TranslateWildcardBindings}\ \frac{\textsc{Verse-If}(\texttt{True}\ e\ e) \rightarrowtail \{\}}{\textsc{case}(\textsc{Wildcard}, v) \rightarrowtail \{\}}$$

$$\textsc{TranslateWildcardEval}\ \frac{}{\textsc{case}(\textsc{Wildcard}, v, e) \rightarrowtail \textsc{Verse-If}(\texttt{True}\ e\ e)}$$

$$\textsc{TranslateWildcardEval'}\ \frac{\textsc{Verse-If}(\texttt{True}\ e\ e) \rightarrowtail e}{\textsc{case}(\textsc{Wildcard}, v, e) \rightarrowtail e}$$

Moving on to variables:

case `v` *of* $x \to e$
$\triangleq$
`if` $(\exists x.\ x = v)$ `e e`

$$\textsc{Verse-IfBindings}\ \frac{}{\textsc{Verse-If}((\exists x.\ x = v) e\ e) \rightarrowtail e\{x \longmapsto v\}}$$

$$\textsc{Verse-IfEval}\ \frac{}{\textsc{Verse-If}(\texttt{True}\ e\ e) \rightarrowtail e}$$

$$\textsc{TranslateVarBindings}\ \frac{\textsc{Verse-If}((\exists x.\ x = v)\ e\ e) \rightarrowtail \{x \longmapsto v\}}{\textsc{case}(x, v, e) \rightarrowtail \{x \longmapsto v\}}$$

$$\textsc{TranslateVarEval}\ \frac{\textsc{Verse-If}((\exists x.\ x = v)\ e\ e) \rightarrowtail e\{x \longmapsto v\}}{\textsc{case}(x, v, e) \rightarrowtail e\{x \longmapsto v\}}$$