$t ::=$		
	x	*variable*
	$\lambda x{:}T.\ t$	*abstraction*
	$\lambda X{<:}T.\ t$	*type abstraction*
	$t\ t$	*application*
	$t\ T$	*type application*
	$\textit{new}\ C$	*constructor call*
	$t\ \textit{match}\{\overline{x{:}C \Rightarrow t}\}\ \textit{or}\ t$	*match expr.*
$v ::=$		
	$\lambda x{:}T.\ t$	*abstraction*
	$\lambda X{<:}T.\ t$	*type abstraction*
	$\textit{new}\ C$	*constructor call*

$T ::=$		
	X	*type variable*
	$T \to T$	*type of functions*
	$\forall X{<:}T.\ T$	*universal type*
	Top	*maximum type*
	C	*class*
	$\{\textit{new}\ C\}$	*constructor singleton*
	$T\ \textit{match}\{\overline{T \Rightarrow T}\}\ \textit{or}\ T$	*match type*
$\Gamma ::=$		
	$\emptyset$	*empty context*
	$\Gamma, x{:}T$	*term binding*
	$\Gamma, X{<:}T$	*type binding*