The Definition of CHARS is interesting. It defines a 'token' EOF and then all applications of the function Other to the type ORDINARY. This is, by specification of the Z notation an injective function.

$$
\begin{array}{l}
[\mathit{ORDINARY}] \\
\mathit{CHARS} ::= \mathit{EOF} \mid \mathit{Other}\langle\!\langle \mathit{ORDINARY} \rangle\!\rangle \\
\mathit{STATE} ::= \mathit{READ} \mid \mathit{WRITE}
\end{array}
$$

File

$$
\begin{array}{|l}
\mathit{content} : \mathrm{seq}\, \mathit{CHARS} \\
\mathit{state} : \mathit{STATE} \\
\mathit{position} : \mathbb{N} \\
\hline
\mathit{last\ content} = \mathit{EOF} \\
\#(\mathit{content} \upharpoonright \{\mathit{EOF}\}) = 1
\end{array}
$$

FileInit

$$
\begin{array}{|l}
\mathit{File}' \\
\hline
\mathit{content}' = \langle\rangle \\
\mathit{state}' = \mathit{WRITE}
\end{array}
$$

Overwrite

$$
\begin{array}{|l}
\Delta \mathit{File} \\
\mathit{in}? : \mathrm{seq}\, \mathit{CHARS} \\
\hline
\mathit{state} = \mathit{WRITE} \\
\mathit{content}' = \mathit{in}? \\
\mathit{state}' = \mathit{state}
\end{array}
$$

Append

$$
\begin{array}{|l}
\Delta \mathit{File} \\
\mathit{in}? : \mathit{ORDINARY} \\
\hline
\mathit{state} = \mathit{WRITE} \\
\mathit{content}' = \mathit{content} \frown \langle \mathit{Other}(\mathit{in}?) \rangle \\
\mathit{state}' = \mathit{state}
\end{array}
$$

If a user tries to reopen an open file it will not reset the read position.

Open

$$
\begin{array}{|l}
\Delta \mathit{File} \\
\hline
\mathit{state} = \mathit{WRITE} \Rightarrow \mathit{state}' = \mathit{READ} \land \mathit{position}' = 0 \\
\mathit{state} = \mathit{READ} \Rightarrow \mathit{state}' = \mathit{state} \land \mathit{position}' = \mathit{position} \\
\mathit{content}' = \mathit{content}
\end{array}
$$

Close

$$
\begin{array}{|l}
\Delta \mathit{File} \\
\hline
\mathit{state}' = \mathit{WRITE} \\
\mathit{content}' = \mathit{content}
\end{array}
$$

I think this should work, however if there is an EOF (somehow) in the middle of the file it will not read beyond there. Using the domain of the sequence (1,2,3...n) to do this is more elegant.

ReadChar

$$\Delta File$$
$$out! : CHARS$$

$$state = READ$$
$$content' = content$$
$$state' = state$$
$$\textbf{if}\ content\ position = EOF\ \textbf{then}\ out! = EOF \land position' = position$$
$$\qquad\qquad \textbf{else}\ out! = content\ position \land position' = position + 1$$