MODULE *Md2LaTeXAlgorithms*

EXTENDS *Md2LaTeXSystemDesign*, *Functions*

At run time / compile time, the preferences file is parsed,
which yields a dictionary (in Python) / *HashMap* (in *Java*) object,
namely 'preferences_as_dict'.
We specify the parsing process.

VARIABLE *preferences_as_dict*

If it is No, then no setting.
So, current key is off *preferences_as_dict*.

First, filter:

$$filteredKeys \triangleq \{$$
$$key \in \text{DOMAIN } preferences :$$
$$\wedge\ isFollowingYesOrNoPolicy(preferences[key])$$
$$\wedge\ preferences[key][Y\_N] \notin JSON\_NO$$
$$\}$$

Next, stir up:

$$parsing \triangleq [key \in filteredKeys \mapsto preferences[key]]$$

* *
Initial state
* *

$$InitAlgorithms \triangleq$$
$$\wedge\ InitSystemDesign$$
$$\wedge\ preferences\_as\_dict = preferences$$

* *
Next state
* *

$$NextAlgorithms \triangleq$$
$$\wedge\ NextSystemDesign$$
$$\wedge\ preferences\_as\_dict' = parsing$$

* *
Invariant
IsParsingOK = TRUE *if.f* the parsing outputs a dictionary that:
i. is compatible with the *YesOrNo* policy, *i.e* every subrecord is so;
ii.is 'lean', in the sense that no "turned off" option
- see *Md2LaTeXSystemDesign* - keeps existing in the dictionary
* *

* *
This is actually repeating what is done with *Md2LaTeXSystemDesign*,

but this time, it is an invariant!

* *

$$
\begin{array}{l}
\mathit{IsParsingOK} \triangleq \\
\quad \lor \mathit{InitAlgorithms} \\
\quad \lor \ \forall\, \mathit{key} \in \text{DOMAIN } \mathit{preferences\_as\_dict} : \\
\qquad \land \mathit{isFollowingYesOrNoPolicy}(\mathit{preferences\_as\_dict}[\mathit{key}]) \\
\qquad \land \ \mathit{XOR}(\ \text{either:} \\
\qquad\qquad \land \neg \mathit{isFollowingYesOrNoPolicy}(\mathit{preferences\_as\_dict}[\mathit{key}]) \\
\qquad\qquad \land \mathit{isCompatibleWithYesOrNoPolicy}(\mathit{preferences\_as\_dict}[\mathit{key}]), \\
\qquad\qquad \text{either:} \\
\qquad\qquad \land \mathit{isFollowingYesOrNoPolicy}(\mathit{preferences\_as\_dict}[\mathit{key}]) \\
\qquad\qquad \land \mathit{preferences\_as\_dict}[\mathit{key}][\mathit{Y\_N}] \in \mathit{JSON\_YES})
\end{array}
$$