Assignment 3

Ruofei HUANG(z5141448) Anqi ZHU(z5141541)

May 27, 2018

1 Task 1

1.1 Type *word*

This definiation of wrod is basicly from requirement of assignment 3 We say two words v, w that v is an absolute prefix of w as $v < w$ which is define as $v \leq w \wedge v \neq w$.

1.2 abstract Data Type *Dict*

According to the specified problem statement in the assignment, we could describe the syntactic data type *Dict* as below. The encapsulated state is a dictionary word set W.

$$
\begin{aligned}
&Dict = (W = \phi,\\
&\begin{pmatrix}
\textbf{proc } addword^{Dict}(\textbf{word } w) \cdot b, W : [\ \textsc{true}, b = b_0 \wedge W = W_0 \cup \{w\}] \\
\textbf{func } checkword^{Dict}(\textbf{word } w) : \mathbb{B} \cdot \textbf{var } b \cdot\ b, W : [\ \textsc{true}, b = (w \in W_0)];\ \textbf{return } b \\
\textbf{proc } delword^{Dict}(\textbf{word } w) \cdot b, W : [w \in W, b = b_0 \wedge W = W_0 \backslash \{w\}]
\end{pmatrix})
\end{aligned}
$$

2 Task 2

Now we would like to refine *Dict* to a second data type *DictA* where we replace W with a trie t,the corresponding trie domain $D = \textbf{dom}(t)$. It represents the set of all tries according to the domain. We shall use this definition later in our refinement.

2.1 Datat Invariant

$$\forall w \in \textbf{dom}(t), t(w) = 1, w' \leq w\,(w' \in \textbf{dom}(t))$$

2.2 Data Type Refinement

This suggests we should first build up a inductively defined predicate to ensure the provable relations between $DictA$ and $Dict$.

$$r = (W = \{w \in \mathbf{dom}(t) | t(w) = 1\})$$

which we can translate into a function from concrete to abstract values:

$$f(t) = \{w \in \mathbf{dom}(t) | t(w) = 1\}$$

With that in mind we propose the initialisation predicate $init^{DictA} = (i = 0)$ and operations given as follows.

$$\begin{array}{l}
\mathbf{proc}\ addword^{DictA}(\mathbf{word}\ w) \cdot b, t : \\
\quad [\ \text{TRUE}, b = b_0 \land t = t_0 \cup \{w \mapsto 1\} \cup \{w' < w \land w' \notin \mathbf{dom}(t) | w' \mapsto 0\}] \\
\mathbf{func}\ checkword^{DictA}(\mathbf{word}\ w)\mathbb{B} \cdot \mathbf{var}\ b \cdot b, t : \\
\quad [\text{TRUE}, b = (w \in \mathbf{dom}(t)) \land t = t_0];\ \mathbf{return}\ b \\
\mathbf{proc}\ delword^{DictA}(\mathbf{word}\ w) \cdot b, t : \\
\quad [\text{TRUE}, b = b_0 \land (w \notin \mathbf{dom}(t) \lor t := t : w \mapsto 0)]
\end{array}$$

2.3 Proof of Refinement

We need start the proof with the initialisation:

$$\begin{array}{ll}
 & init^{DictA} \Rightarrow init^{Dict}[^{f(t)}/_{W}] \\
\Leftrightarrow & \qquad \langle \text{Definition of } init^{DictA} \text{ and } init^{Dict} \rangle \\
 & \forall w \in \mathbf{dom}(t)\ (t(w) = 0) \Rightarrow W = \phi
\end{array}$$

Since all our precondition of concrete is trivial which all of them are TRUE, we don't need to proof the condition (3_f).But condition (4_f) must be checked for all three operations. For the *addword* we proof:

$$sss$$

For the *checkword* we proof:

$$sss$$

For the *delword* we proof:

$$sss$$