437 (brackets) You are given a text t of characters drawn from the alphabet "x", "(", ")", "[", "]" . Write a program to determine if t has its brackets properly paired and nested.

After trying the question, scroll down to the solution.

§ Introduce variable n: *nat* , and a stack. Define predicate P: $text \to bin$ so that $P\ s$ means that s has its brackets properly paired and nested. Here are its axioms.

$$P\ “”$$
$$P\ “x”$$
$$P\ a\ =\ P\ (“(”;\ a;\ “)”)$$
$$P\ a\ =\ P\ (“[”;\ a;\ “]”)$$
$$P\ a \land P\ z\ \Rightarrow\ P\ (a;\ z)$$

Now we need two specifications.

$$R\ =\ (isempty' \land n'{=}{\leftrightarrow}t\ \ =\ \ P\ t)$$
$$Q\ =\ \text{(defined later)}$$

Here are the refinements.

$R\ \Leftarrow\ \ mkempty.\ \ n{:=}\ 0.\ \ Q$

$Q\ \Leftarrow\ \ $ **if** $n{=}{\leftrightarrow}t$ **then** *ok*
else if $t_n{=}“x”$ **then** $n{:=}\ n{+}1.\ \ Q$
else if $t_n{=}“(”$ **then** *push* “)”. $n{:=}\ n{+}1.\ \ Q$
else if $t_n{=}“[”$ **then** *push* “]”. $n{:=}\ n{+}1.\ \ Q$
else if *isempty* **then** *ok*
else if $t_n{=}top$ **then** *pop*. $n{:=}\ n{+}1.\ \ Q$
else *ok* **fi fi fi fi fi fi**

I have used a stack, and for the purpose of executing the program, the stack can be implemented any way that is correct. But for the purpose of defining specification Q , I implement it as follows. Let s be a text-valued implementer's variable.

$$mkempty\ \ =\ \ s{:=}\ nil$$
$$isempty\ \ =\ \ {\leftrightarrow}s = 0$$
$$push\ =\ \langle c{:}\ char \cdot\ s{:=}\ c;s\rangle$$
$$top\ =\ s_0$$
$$pop\ \ =\ \ s{:=}\ s_{1;..{\leftrightarrow}s}$$

Now I can define specification Q .

$$Q\ \ =\ \ P\ (t_{0;..n}\ ;\ s) \Rightarrow R$$

And finally we can prove the refinements. UNFINISHED