194 (pattern search) Let *subject* and *pattern* be two texts. Write a program to do the following. If *pattern* occurs somewhere within *subject* , natural variable h is assigned to indicate the beginning of its first occurrence

(a) using any string operators given in Section 2.2.

(b) using string indexing and string length, but no other string operators.

After trying the question, scroll down to the solution.

§ It might be best to strengthen the specification to provide an indication if *pattern* does not occur anywhere in *subject*, but I'll stick with the question as asked.

(a) using any string operators given in Section 2.2.

§ Define specifications P and Q as follows.

$$\begin{array}{lll} P \;= & & \leftrightarrow pattern \le \leftrightarrow subject \\ & \wedge & (\exists i\colon 0,..\#subject-\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) \\ & \Rightarrow & subject_{h';..h'+\leftrightarrow pattern} = pattern \\ & \wedge & \neg(\exists i\colon 0,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) \\ & \wedge & t' \le t + \leftrightarrow subject - \leftrightarrow pattern \end{array}$$

P says if there is room for *pattern* in *subject*, and if *pattern* does occur somewhere in there, then make h' be the starting index of its first occurrence, and the time is bounded by the length of *subject* minus the length of *pattern*.

$$\begin{array}{lll} Q \;= & & h + \leftrightarrow pattern \le \leftrightarrow subject \\ & \wedge & (\exists i\colon h,..\#subject-\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) \\ & \Rightarrow & subject_{h';..h'+\leftrightarrow pattern} = pattern \\ & \wedge & \neg(\exists i\colon h,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) \\ & \wedge & t' \le t + \leftrightarrow subject - \leftrightarrow pattern - h \end{array}$$

Q says if there is room for *pattern* in *subject* starting at index h, and if *pattern* does occur somewhere in there, then make h' be the starting index of its first occurrence, and the time is bounded by the length of *subject* minus the length of *pattern* minus h.

The refinements, including recursive time, are as follows.

$$\begin{array}{ll} P \Leftarrow & h:= 0.\ Q \\ Q \Leftarrow & \textbf{if}\ h + \leftrightarrow pattern > \leftrightarrow subject\ \textbf{then}\ ok \\ & \textbf{else if}\ subject_{h;..h+\leftrightarrow pattern} = pattern\ \textbf{then}\ ok \\ & \quad \textbf{else}\ h:= h+1.\ t:= t+1.\ Q\ \textbf{fi fi} \end{array}$$

The proofs are as follows. First the P refinement.

$h:= 0.\ Q$ — expand Q, substitution law

$= \;\; P$

Now the Q refinement by cases. There are three cases. First case:

$Q \;\Leftarrow\; h + \leftrightarrow pattern > \leftrightarrow subject \;\wedge\; ok$ — replace Q and ok

$$\begin{array}{llll} = & (& & h + \leftrightarrow pattern \le \leftrightarrow subject \\ & & \wedge & (\exists i\colon h,..\#subject-\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) \\ & & \Rightarrow & subject_{h';..h'+\leftrightarrow pattern} = pattern \\ & & \wedge & \neg(\exists i\colon h,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) \\ & & \wedge & t' \le t + \leftrightarrow subject - \leftrightarrow pattern - h) \\ & \multicolumn{3}{l}{\Leftarrow\; h + \leftrightarrow pattern > \leftrightarrow subject \;\wedge\; h'=h \;\wedge\; t'=t} \end{array}$$

use antecedent as context in consequent

$$\begin{array}{llll} = & (& & \bot \\ & & \wedge & (\exists i\colon h,..\#subject-\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) \\ & & \Rightarrow & subject_{h;..h+\leftrightarrow pattern} = pattern \\ & & \wedge & \neg\bot \\ & & \wedge & t \le t + \leftrightarrow subject - \leftrightarrow pattern - h) \\ & \multicolumn{3}{l}{\Leftarrow\; h + \leftrightarrow pattern > \leftrightarrow subject \;\wedge\; h'=h \;\wedge\; t'=t} \end{array}$$

base

$$= \quad \top$$

Middle case:

$$Q \quad \Leftarrow \quad h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \ \land \ subject_{h;..h+{\leftrightarrow}pattern} = pattern \ \land \ ok$$

replace Q and ok

$$\begin{array}{ll}
= & (\quad h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \\
& \quad \land \ (\exists i\colon h,..\#subject{-}\#pattern\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \Rightarrow \quad subject_{h';..h'+{\leftrightarrow}pattern} = pattern \\
& \quad \land \ \neg(\exists i\colon h,..h'\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \quad \land \ t' \le t + {\leftrightarrow}subject - {\leftrightarrow}pattern - h) \\
& \Leftarrow \ h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \ \land \ subject_{h;..h+{\leftrightarrow}pattern} = pattern \ \land \ h'=h \ \land \ t'=t
\end{array}$$

use antecedent as context in consequent

$$\begin{array}{ll}
= & (\quad \top \\
& \quad \land \ (\exists i\colon h,..\#subject{-}\#pattern\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \Rightarrow \quad \top \\
& \quad \land \ \neg(\exists i\colon h,..h\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \quad \land \ t \le t + {\leftrightarrow}subject - {\leftrightarrow}pattern - h) \\
& \Leftarrow \ h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \ \land \ subject_{h;..h+{\leftrightarrow}pattern} = pattern \ \land \ h'=h \ \land \ t'=t
\end{array}$$

$h,..h$ is $null$ and $t \le t +$ (nonnegative)

$$\begin{array}{ll}
= & (\quad \top \\
& \quad \land \ (\exists i\colon h,..\#subject{-}\#pattern\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \Rightarrow \ \top \land \top \land \top \\
& \Leftarrow \ h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \ \land \ subject_{h;..h+{\leftrightarrow}pattern} = pattern \ \land \ h'=h \ \land \ t'=t
\end{array}$$

base

$$= \quad \top$$

Last case:

$$\begin{array}{l}
Q \quad \Leftarrow \quad h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \ \land \ subject_{h;..h+{\leftrightarrow}pattern} \ne pattern \\
\qquad \land \ (h{:=}\ h{+}1.\ \ t{:=}\ t{+}1.\ \ Q)
\end{array}$$

--------- replace Q twice and substitution law twice

$$\begin{array}{ll}
= & (\quad h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \\
& \quad \land \ (\exists i\colon h,..\#subject{-}\#pattern\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \Rightarrow \quad subject_{h';..h'+{\leftrightarrow}pattern} = pattern \\
& \quad \land \ \neg(\exists i\colon h,..h'\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \quad \land \ t' \le t + {\leftrightarrow}subject - {\leftrightarrow}pattern - h) \\
& \Leftarrow \quad h + {\leftrightarrow}pattern \le {\leftrightarrow}subject \\
& \quad \land \ subject_{h;..h+{\leftrightarrow}pattern} \ne pattern \\
& \quad \land \ (\quad h + 1 + {\leftrightarrow}pattern \le {\leftrightarrow}subject \\
& \qquad \land \ (\exists i\colon h{+}1,..\#subject{-}\#pattern\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \quad \Rightarrow \quad subject_{h';..h'+{\leftrightarrow}pattern} = pattern \\
& \qquad \land \ \neg(\exists i\colon h{+}1,..h'\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) \\
& \qquad \land \ t' \le t + 1 + {\leftrightarrow}subject - {\leftrightarrow}pattern - h - 1)
\end{array}$$

in last line simplify $1{-}1$; portation

$$\begin{array}{lll}
= & \quad h + {\leftrightarrow}pattern \le {\leftrightarrow}subject & (0) \\
& \land \ subject_{h;..h+{\leftrightarrow}pattern} \ne pattern & (1) \\
& \land \ (\quad h + 1 + {\leftrightarrow}pattern \le {\leftrightarrow}subject & (2) \\
& \qquad \land \ (\exists i\colon h{+}1,..\#subject{-}\#pattern\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) & (3) \\
& \quad \Rightarrow \quad subject_{h';..h'+{\leftrightarrow}pattern} = pattern & (4) \\
& \qquad \land \ \neg(\exists i\colon h{+}1,..h'\cdot \ subject_{i;..i+{\leftrightarrow}pattern} = pattern) & (5) \\
& \qquad \land \ t' \le t + {\leftrightarrow}subject - {\leftrightarrow}pattern - h) & (6) \\
& \land \ \ h + {\leftrightarrow}pattern \le {\leftrightarrow}subject & (7)
\end{array}$$

$$
\begin{array}{lll}
& \quad\land\ \ (\exists i\text{: } h,..\#subject–\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) & (8)\\
& \quad\land\ \ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h & (9)\\
& \Rightarrow\quad subject_{h';..h'+\leftrightarrow pattern} = pattern & (10)\\
& \quad\land\ \neg(\exists i\text{: } h,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) & (11)\\
& \quad\land\ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h & (12)
\end{array}
$$

Line (1) is context for line (5) so line (5) can say $\neg(\exists i\text{: } h,..h'\cdot\ ...)$.
Line (1) is also context for line (8) so line (8) can say $(\exists i\text{: } h{+}1,..\#subject–\#pattern\cdot\ ...)$.
Line (7) duplicates line (0).

$$
\begin{array}{lll}
= & \qquad h + \leftrightarrow pattern \le \leftrightarrow subject & (13)\\
& \quad\land\ subject_{h;..h+\leftrightarrow pattern} \ne pattern & (14)\\
& \quad\land\ (\qquad h + 1 + \leftrightarrow pattern \le \leftrightarrow subject & (15)\\
& \qquad\qquad\land\ (\exists i\text{: } h{+}1,..\#subject–\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) & (16)\\
& \qquad\quad\Rightarrow\quad subject_{h';..h'+\leftrightarrow pattern} = pattern & (17)\\
& \qquad\qquad\land\ \neg(\exists i\text{: } h,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) & (18)\\
& \qquad\qquad\land\ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h) & (19)\\
& \quad\land\ \ (\exists i\text{: } h{+}1,..\#subject–\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) & (20)\\
& \quad\land\ \ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h & (21)\\
& \Rightarrow\quad subject_{h';..h'+\leftrightarrow pattern} = pattern & (22)\\
& \quad\land\ \neg(\exists i\text{: } h,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern) & (23)\\
& \quad\land\ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h & (24)
\end{array}
$$

Line (20) is context for line (16).
The domain in line (20) also implies line (15).

$$
\begin{array}{ll}
= & \qquad h + \leftrightarrow pattern \le \leftrightarrow subject\\
& \quad\land\ subject_{h;..h+\leftrightarrow pattern} \ne pattern\\
& \quad\land\ (\qquad \top\\
& \qquad\quad\land\ \top\\
& \qquad\Rightarrow\quad subject_{h';..h'+\leftrightarrow pattern} = pattern\\
& \qquad\quad\land\ \neg(\exists i\text{: } h,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern)\\
& \qquad\quad\land\ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h)\\
& \quad\land\ \ (\exists i\text{: } h{+}1,..\#subject–\#pattern\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern)\\
& \quad\land\ \ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h\\
& \Rightarrow\quad subject_{h';..h'+\leftrightarrow pattern} = pattern\\
& \quad\land\ \neg(\exists i\text{: } h,..h'\cdot\ subject_{i;..i+\leftrightarrow pattern} = pattern)\\
& \quad\land\ t' \le t + \leftrightarrow subject–\leftrightarrow pattern–h \qquad\qquad \text{base and specialization}\\
\\
= & \top
\end{array}
$$

(b) using string indexing and string length, but no other string operators.

§ The program in part (a) has only one string comparison, namely

$$subject_{h;..h+\leftrightarrow pattern} = pattern$$

To replace it with string indexing, introduce binary variable m (for <u>m</u>atch), and natural variable n. We need two more specifications.

$$R \ =\ h + \leftrightarrow pattern \le \leftrightarrow subject \ \Rightarrow\ m'{=}(subject_{h;..h+\leftrightarrow pattern} = pattern) \ \land\ h'{=}h$$

$$S \ =\ \quad h \le n \le h{+}\leftrightarrow pattern \le \leftrightarrow subject$$
$$\qquad \Rightarrow\ m'{=}(subject_{n;..h+\leftrightarrow pattern} = pattern_{n–h;..\leftrightarrow pattern}) \land h'{=}h$$

Now the refinements.

$$R \ \Leftarrow\ n{:=}\ h.\ S$$

$S \ \Leftarrow$ **if** $n = h + \leftrightarrow pattern$ **then** $m{:=}\ \top$
else if $subject_n{=}pattern_{n–h}$ **then** $n{:=}\ n{+}1.\ S$
else $m{:=}\ \bot$ **fi fi**

And the proofs. First the R refinement.

$$n:= h.\ S \qquad \text{expand } S \text{, substitution law, simplify}$$
$$= \quad R$$

Now the S refinement.
NOT YET DONE

Finally, recursive time, which counts the time for the string comparison.
NOT YET DONE

Now we put it all together, as follows.

$P \Leftarrow h:= 0.\ Q$
$Q \Leftarrow$ **if** $h + \leftrightarrow pattern > \leftrightarrow subject$ **then** ok
else R. **if** m **then** ok
else $h:= h+1.\ Q$ **fi fi**

We can optimize a little, by redefining S, and re-refining as follows.

$P \Leftarrow h:= 0.\ Q$
$Q \Leftarrow$ **if** $h + \leftrightarrow pattern > \leftrightarrow subject$ **then** ok
else $n:= h.\ S$ **fi**
$S \Leftarrow$ **if** $n = h + \leftrightarrow pattern$ **then** ok
else if $subject_n = pattern_{n-h}$ **then** $n:= n+1.\ S$
else $h:= h+1.\ Q$ **fi fi**