Quiz Section 9: Points – Solutions

In this section, we will be working with a data structure called a LocTree. For more information on the data structures and definitions used, please refer to the resource found here: https://courses.cs.washington.edu/courses/cse331/25wi/homework/hw9def.pdf.

Task 1 – Regions in Hot Pursuit!

We have a function overlap : (Region, Region) → Bool that returns true if two regions overlap. Two regions overlap if they share any area in common. Write an expression that returns true if 2 regions R_1 and R_2 overlap.

$$((R_1.\text{x1} \leqslant R_2.\text{x2}) \text{ and } (R_1.\text{x2} \geqslant R_2.\text{x1}) \text{ and } (R_1.\text{y1} \leqslant R_2.\text{y2}) \text{ and } (R_1.\text{y2} \geqslant R_2.\text{y1}))$$

Task 2 – Tree-search takes root!

(a) Define a function findAll : (LocTree, Region) → List⟨Location⟩ that returns a list of all locations in the LocTree that fall within the given region. The order of the locations in the list does not matter but there should be no duplicate entries. Assume we have the following function contains : (Region, Location) → Bool that returns true if the location is within the region.

$$\begin{aligned}
\text{findAll}(\text{empty}, R) &:= [] \\
\text{findAll}(\text{single(s)}, R) &:= [\text{s}] && \text{if contains}(R, s) \\
\text{findAll}(\text{single(s)}, R) &:= [] && \text{if not contains}(R, s) \\
\text{findAll}(\text{split}(m, \text{nw}, \text{ne}, \text{sw}, \text{se}), R) &:= \text{findAll(nw, R)} \\
&\quad + \text{findAll(ne, R)} \\
&\quad\quad + \text{findAll(sw, R)} \\
&\quad\quad\quad + \text{findAll(se, R)}
\end{aligned}$$

(b) Improve the algorithm by excluding any quadrants that do not overlap with the region passed in. This will avoid traversing any subtrees that cannot contain any locations in the region. We can do this by defining an improved function fa : (LocTree, Region, Region) → List⟨Location⟩ that takes in an additional Region parameter. The second Region parameter is the region that we are looking for locations in. The first Region parameter is a region containing all the points in the tree. It will use this region parameter to avoid recursing into quadrants of split nodes when they cannot contain a location in the second Region parameter.

$$
\begin{aligned}
\mathsf{fa}(\mathsf{empty}, S, R) &:= [] \\
\mathsf{fa}(\mathsf{single(s)}, S, R) &:= [\mathsf{s}] && \text{if } \mathsf{contains}(R, s) \\
\mathsf{fa}(\mathsf{single(s)}, S, R) &:= [] && \text{if not } \mathsf{contains}(R, s) \\
\mathsf{fa}(\mathsf{split}(m, \mathsf{nw}, \mathsf{ne}, \mathsf{sw}, \mathsf{se}), S, R) &:= [] && \text{if not } \mathsf{overlap}(S, R) \\
\mathsf{fa}(\mathsf{split}(m, \mathsf{nw}, \mathsf{ne}, \mathsf{sw}, \mathsf{se}), S, R) &:= \mathsf{fa}(\mathsf{nw}, \mathsf{NW}(m, S), R) \\
&\quad + \mathsf{fa}(\mathsf{ne}, \mathsf{NE}(m, S), R) \\
&\quad\quad + \mathsf{fa}(\mathsf{sw}, \mathsf{SW}(m, S), R) \\
&\quad\quad\quad + \mathsf{fa}(\mathsf{se}, \mathsf{SE}(m, S), R) && \text{if } \mathsf{overlap}(S, R)
\end{aligned}
$$

(c) Prove that if region S contains all locations in the tree T, then fa$(T, S, R) =$ findAll(T, R). Your proof should be by structural induction on T.

Feel free to use the fact that, if S contains all the locations in split$(m, \text{nw}, \text{ne}, \text{sw}, \text{se})$, then NW$(m, S)$ contains all the locations in nw and likewise for ne, sw, and se. (This follows from the representation invariant for split nodes and the definitions of these functions.)

Define $P(T)$ to be the claim that, if the region S contains all locations in tree T, then fa$(T, S, R) =$ findAll(T, R). We will prove that this holds for all trees of locations T by structural induction.

Base Cases. We start with the empty node. We can see that

$$
\begin{aligned}
\mathsf{fa}(\mathsf{empty}, S, R) &= [] && \text{def of fa} \\
&= \mathsf{findAll}(\mathsf{empty}, R) && \text{def of findAll}
\end{aligned}
$$

For any single(s), note that, if S does not contain s, then P is vacuously true, so we continue under the assumption that S contains s.

We continue by cases. Suppose, first, that contains(R, s) is true. In that case, we can see that

$$
\begin{aligned}
\mathsf{fa}(\mathsf{single}(s), S, R) &= [\mathsf{s}] && \text{def of fa (since contains}(R, s)) \\
&= \mathsf{findAll}(\mathsf{single}(s), R) && \text{def of findAll (since contains}(R, s))
\end{aligned}
$$

Now, suppose that contains(R, s) is false. In that case, we see that

$$
\begin{aligned}
\mathsf{fa}(\mathsf{single}(s), S, R) &= [] && \text{def of fa (since not contains}(R, s)) \\
&= \mathsf{findAll}(\mathsf{single}(s), R) && \text{def of findAll (since not contains}(R, s))
\end{aligned}
$$

These two cases are exhaustive, so we have completed the proof for single(s).

Inductive Hypothesis. Suppose that $P(\text{nw})$, $P(\text{ne})$, $P(\text{sw})$, and $P(\text{se})$ holds for some trees nw, ne, sw, and se.

Inductive Step. We must prove that $P(\text{split}(m, \text{nw}, \text{ne}, \text{sw}, \text{se}))$ holds.

Again, note that, if S does not contain m, then P is vacuously true, so we continue under the assumption that S contains m.

If $\mathsf{overlap}(S, R)$ is false, then we know that no location in the tree can be in the region R. Since S contains all locations in the tree, we know that $\mathsf{findAll}(\mathsf{T}, R) = []$, matching the definition of $\mathsf{fa}(\mathsf{T}, S, R)$. Thus we can continue under the assumption that $\mathsf{overlap}(S, R)$ is true.

As was noted above, we then know that $\mathsf{NE}(m, \mathsf{ne})$ contains all locations in the tree ne. Call this fact "NE". Thus, we can see that

$$\mathsf{fa}(\mathsf{ne}, \mathsf{NE}(m, S), R) = \mathsf{findAll}(\mathsf{ne}, R) \quad \text{Ind. Hyp (since NE)}$$

We can apply this same argument to NW, SW, and SE which gives us:

$$\begin{aligned}
\mathsf{fa}(\mathsf{nw}, \mathsf{NW}(m, S), R) &= \mathsf{findAll}(\mathsf{nw}, R) && \text{Ind. Hyp (since NW)} \\
\mathsf{fa}(\mathsf{sw}, \mathsf{SW}(m, S), R) &= \mathsf{findAll}(\mathsf{sw}, R) && \text{Ind. Hyp (since SW)} \\
\mathsf{fa}(\mathsf{se}, \mathsf{SE}(m, S), R) &= \mathsf{findAll}(\mathsf{se}, R) && \text{Ind. Hyp (since SE)}
\end{aligned}$$

Substituting these into the definition of $\mathsf{fa}(\mathsf{split}(m, \mathsf{nw}, \mathsf{ne}, \mathsf{sw}, \mathsf{se}), S, R)$ we can calculate:

$$\begin{aligned}
\mathsf{fa}(\mathsf{split}(m, \mathsf{nw}, \mathsf{ne}, \mathsf{sw}, \mathsf{se}), S, R) &= \mathsf{fa}(\mathsf{nw}, \mathsf{NW}(m, S), R) \\
&\qquad + \mathsf{fa}(\mathsf{ne}, \mathsf{NE}(m, S), R) \\
&\qquad\quad + \mathsf{fa}(\mathsf{sw}, \mathsf{SW}(m, S), R) \\
&\qquad\qquad + \mathsf{fa}(\mathsf{se}, \mathsf{SE}(m, S), R) && \text{Def of fa} \\
&= \mathsf{findAll}(\mathsf{nw}, R) \\
&\qquad + \mathsf{findAll}(\mathsf{ne}, R) \\
&\qquad\quad + \mathsf{findAll}(\mathsf{sw}, R) \\
&\qquad\qquad + \mathsf{findAll}(\mathsf{se}, R) && \text{Substituting from above} \\
&= \mathsf{findAll}(\mathsf{split}(m, \mathsf{nw}, \mathsf{ne}, \mathsf{sw}, \mathsf{se}), R) && \text{Def of findAll}
\end{aligned}$$

Thus we have proven the claim that $P(\mathsf{split}(m, \mathsf{nw}, \mathsf{ne}, \mathsf{sw}, \mathsf{se}))$ holds.

Conclusion. $P(T)$ holds for all trees T by structural induction.