Recitation 14

PASL

14.1 Announcements

- *PASLLab* is due **Thursday night**.
- We will likely be having a final review sometime on Wednesday, Dec 16. Keep your ears open for more details.
- The final exam is on Thursday, Dec 17, 5:30-8:30pm.

14.2 map_flatten

Let's create a new file in the *PASLLab* top directory called rec14.hpp, and write a few functions. To manipulate sparrays, we'll write the line #include "sparray.hpp" at the top of rec14.hpp.

Task 14.1. *Using PASL, implement the function*

```
template <class Map_func, class Size_func>
sparray map_flatten(const Map_func& f,
                    const Size_func& g,
                    const sparray& xs);
```

where, at a high-level, the goal is to compute

$$\texttt{flatten}\left\langle f(x) : x \in xs\right\rangle.$$

You should assume that the function arguments are typed as follows, where f(xs[i]) *is a pointer to the front of an array of length* g(xs[i]).

$$\begin{aligned} f &: \texttt{value\_type} \rightarrow \texttt{value\_type}* \\ g &: \texttt{value\_type} \rightarrow \texttt{long} \end{aligned}$$

14.3 `inject`

The sequence function `inject` has always seemed to be shrouded in mystery. Let's see how the magic really works!

Task 14.2. *Using PASL, implement the function*

```
sparray inject(const sparray& xs,
               const sparray& indices,
               const sparray& updates);
```

which returns the result of injecting into `xs`. We require that `indices` and `updates` be the same length, such that for each i, we attempt to write `updates[i]` at position `indices[i]` in `xs`. Note that you should not destructively modify `xs`.
If there are multiple updates specified at the same position, then all except the last should be ignored. (We want to match the behavior of `inject` as specified in the 15210 Library.)