Programming Massively Parallel Hardware

Assignment 1

Viktor Hansen

Task 1

The following solution makes use of the syntax in the Bird paper. We wish to show that the equivalence $(\odot\backslash) \circ (f\star) \equiv (\odot\backslash) \circ (((\odot\backslash) \circ (f\star))\star) \circ d_p$ holds:

Proof. Starting from the lefthandside, composing with $id \equiv (+\!\!+\backslash) \circ d_p$ gives

$$\begin{aligned}(\odot\backslash) \circ (f\star) &\equiv (\odot\backslash) \circ (f\star) \circ id \\ &\equiv (\odot\backslash) \circ (f\star) \circ (+\!\!+\backslash) \circ d_p\end{aligned}$$

Applying the second list-homomorphism promotion on the second and third terms yields

$$\equiv (\odot\backslash) \circ (+\!\!+\backslash) \circ ((f\star)\star) \circ d_p$$

Next, applying the third list-homomorphism promotion on the two first terms gives

$$\equiv (\odot\backslash) \circ ((\odot\backslash)\star) \circ ((f\star)\star) \circ d_p$$

Finally, the first list-homomorphism promotion is applied on the two map-terms, yielding the right-hand side

$$\equiv (\odot\backslash) \circ (((\odot\backslash) \circ (f\star))\star) \circ d_p$$

□

Task 2

Refer to `code/task2/LongestSatSgm.hs` for the solution.

Task 3

Refer to `code/task3/PrimesQuickSort.hs` for the solution. The solution could probably be nicer, but it exhibits the correct semantics and does not have nested parallel constructs.

Task 4

Refer to `code/task4/task4.cu` and `code/task4/Makefile` for the solution. The outputs of invoking `make run` is shown in Listing 1. The CUDA kernel was invoked with a block size of 256 threads and performed significantly better than the serial function executed on the CPU when time spent copying memory to and from the device was not taken into account. Regardless of the choice of array-size, the penalty incurred by copying memory to the GPGPU, however, seemed proportional to the time spent performing the computation on the CPU. Had the kernel been more computationally intensive, more time would probably have been spent on actual work relative to the penalty incurred by copying memory. The nature of the computation should thus preferably entail that the time spent on copying memory to and from the GPGPU should be small when compared to kernel execution time.

```
jqp221@gpu01-diku-apl:~> make run
./a.out
VALID
GPU took 2762 microseconds (2.76ms)
-- of which  2638 microseconds (2.64ms) was spent copying memory
-- of which  124 microseconds (0.12ms) was spent in the kernel
CPU took a total of  3112 microseconds (3.11ms)
jqp221@gpu01-diku-apl:~>
```

Listing 1: Output of `make run` invoking the kernel on an array of size 753411.