

# Raport tygodniowy 13-17 lipca

Jakub Mendyk

18 lipca 2020

- Zaimplementowałem restartującego planistę.
- Uniezależniłem planistów wstrzymującego i restartującego od typów związanych z przykładem obliczeń w arkuszu kalkulacyjnym. Robiąc to doskwierał mi odrobinę brak ad-hoc polimorfizmu np. typów klas z Haskella. Z tego powodu struktura zawierająca funkcję porównującą klucze musi być jawnie podana planiście.
- Zaimplementowałem rebuilder używający brudnego bitu. Dodanie wsparcia dla rebuildersów w planistach było dobrą okazją by uporządkować ich implementacje. Niestety chwilami problemy z dedukcją typów dawały o sobie znać, stąd implementacje obu planistów muszą nieestety mieć jawnie podane typy wszystkich argumentów. Postanowiłem, że rebuilder będzie podawany planiście jako argument. Wydaje mi się, że łatwo można by zamienić go w efekt uboczny i może warto byłoby zobaczyć czy da to ciekawe wyniki (choć wątpię).
- Zacząłem implementować operacje na śladach/okruszkach (traces) konstruktywnych i weryfikujących. Nie zaimplementowałem jeszcze jednak żadnego rebuildera ich używającego.
- Znaczną część czwartku przeznaczyłem na myślenie nad treścią tekstu pracy. Znalazłem ciekawy artykuł z MIMUW ze wskazówkami jak przygotowywać pracę licencjacką. Po syntezie tego tekstu oraz własnych przemyśleń, wytworzył mi się następujący konspekt treści pracy:
  1. **Wstęp** Krótko czym są efekty uboczne, podanie i opisanie systemów budowania jako przykładu systemu silnie opartego o efekty. Zarys treści „Build Systems à la Carte”. Powiedzenie, że monady są znany sposobem radzenia sobie z efektami ubocznymi ale jeśli chcemy używać wielu efektów, to monady muszą być przygotowane by układać je w stos (bez konkretów). Efekty algebraiczne i uchwyt jako nowy pomysł ujarzmnienia efektów ubocznych bez wad monad. Opowiedzenie, że w pracy wzięto wyniki z „Build Systems à la Carte”, zaadoptowano rozwiązanie do systemów algebraicznych. Krótki opis wniosków i co właściwie z tego wyszło.
  2. **O efektach (algebraicznych) teoretycznie** Efekty uboczne, problemy jakie stwarzają – brak referential transparency. Haskell i monady, problemy z ich łączeniem, potrzeba wykorzystywania transformerów. Efekty algebraiczne i uchwyt – pomysł, spojrzenie jak na rozszerzenie wyjątków oraz wywołania systemowe w systemach operacyjnych, czemu „algebraiczne”?, łatwość składania/kompozycji.
  3. **O systemach budowania (i ich klasyfikacji)** Wprowadzenie czym są, przykłady, Excel jako system budowania. Wprowadzenie do „Build Systems à la Carte” – punkt wyjścia, obserwacje, przedstawiona klasyfikacja i podział schedulers VS rebuilders. Opisy schedulerów i rebuildersów. Jak `Applicative` odpowiada zależnościom statycznym, a `Monad` dynamicznym.

4. **Efekty algebraiczne i uchwyt w praktyce** Omówienie Helium, przykłady efektów i ich implementacji – wyjątek, niedeterminizm, stan – oraz różnych uchwytów i pokazanie jak wygodnie umożliwiają one zmianę co chcemy uzyskać (np. wszystkie sukcesy lub jeden sukces). Przykłady łączenia efektów.
5. **Implementacja i klasyfikacja systemów budowania z użyciem efektów algebraicznych i uchwytów** Fetch jako efekt, Store jako efekt, w formie rozgrzewki demonstracja i implementacja systemu busy. Dalej prezentacja, omówienie i porównanie systemów ze wstrzymywaniem oraz restartem w oryginalnej pracy oraz implementacji z efektami algebraicznymi. Problem z reprezentacją planisty topologicznego – brak odpowiednika dla `Applicative`, może przedstawienie jakiejś protezy rozwiązania np. zadanie mówi jawnie jakich podzadań potrzebuje, a `fetch` rzuca wyjątkiem gdy podzadanie nie było zadeklarowane jako potrzebne.
6. **Podsumowanie i wnioski** Co się udało, a co nie. Jakie mocne oraz słabe strony efektów algebraicznych i uchwytów oraz monad zostały uwydatnione w implementacjach z ich użyciem. Może jakieś propozycje co dalej?

Zgodnie z sugestią autorów wspomnianego wyżej artykułu o pisaniu prac, przemyślałem kto jest docelowym odbiorcą (poza recenzentem) mojego tekstu. Przyjąłem, że jest to osoba która ma wiedzę z zakresu programowania funkcyjnego w stopniu dostatecznym by samodzielnie przeczytać oraz zrozumieć „Build Systems à la Carte”, jednak nie wie lub słyszała tylko niewiele czym są efekty algebraiczne. W zależności od tego, jak długi wyjdzie opis wyników w rozdziale 5 będę modyfikował długości rozdziałów 2–4 (z czego głównie 4), tak aby całość pracy osiągnęła satysfakcjonującą objętość oraz ilość informacji.

To kończy raport na ten tydzień. W tygodniu 20–24 lipca mam zaplanowanych trochę odpoczynku i aktywności poza domem, stąd zapewne mniej uda mi się zrobić niż w poprzednich dwóch tygodniach. W połowie sierpnia wybieram się ze znajomymi na 6 dniowy spływ kajakowy, prócz tego nie planuję do września innych form odpoczynku. Na pisanie pracy pozostają więc co najmniej 4 tygodnie – wydaje mi się, że jest to wystarczająco dużo czasu by dobrze przygotować pracę.

Sugerując się artykułem z MIMUW, nastawiam się że pisanie pracy będę prowadził „spiralnie”. Myślę więc, że w nadchodzącym tygodniu zacznę pisać właściwy tekst pracy zaczynając od wstępu i skupiając się na rozdziałach, których treść najmniej zależy od uzyskanych wyników.

Moje pytania na ten tydzień:

- Co sądzi Pan o zaproponowanym układzie rozdziałów oraz ogólnej treści pracy. Co dałoby się ulepszyć? Jaka jest oczekiwana długość i/lub gęstość treści w pracy licencjackiej? Czy opisany przeze mnie konspekt wydaje się być zgodny z tymi wymaganiami?
- Czy istnieje możliwość zobaczenia historycznych prac licencjackich absolwentów naszego instytutu (prócz przeszukiwania repozytoriów studentów na GitHubie)? Chciałbym zobaczyć jak wyglądają inne prace, które dotyczyły tematyki podobnej do tematu pracy. Na GitHubie znalazłem prace kilku absolwentów ale były one wszystkie mocno techniczne – opisywały raczej implementacje usługi informatycznej lub aplikacji.
- Czy przygotowane przeze mnie implementacje schedulerów są bliskie Pana wizji, gdy proponował mi Pan ten temat? Mam na myśli, czy moje implementacje nie są zbyt podobne do tych opisanych w „Build Systems à la Carte”? Czy myślał Pan może o jakimś istotnie innym sposobie wykorzystania efektów algebraicznych i uchwytów do implementacji systemów budowania?