

Recenzja Haskell FFT

<https://github.com/adamszczerba/haskell-fft/>

Michał Osadnik, Tomasz Czajęcki

Kod

Aplikacja została napisana semantycznie poprawnym Haskelllem. Autorzy nie stronią od monadycznej obsługi interakcji z użytkownikiem i poprawnie dbają o czystość i prostotę deklaratywnych funkcji pomocniczych i wewnętrznych.

GUI

GUI łączy matematyczne podłoże aplikacji z interakcją z użytkownikiem przy pomocy prezentacji graficznej. Obsługa trybu graficznego została napisana z wykorzystaniem notacji `do`, co sprawia, że kod jest czytelny.

Dodatkowo nie brakuje w tym module komentarzy tłumaczących jakie zastosowanie ma dana funkcja, co znacznie ułatwia zrozumienie kodu.

Sound

Nie jesteśmy przekonani co do nazwy tego modułu. Nawet po przeczytaniu kilku tekstów związanych z FFT, nie jest dla nas oczywisty wybór nazwy "Sound". Natomiast jego zakres odpowiedzialności jest dla nas jasny.

Smutny odrobinę jest fakt, że sama Szybka Transformacja Fouriera jest obliczana z użyciem narzędzia do obliczania... Szybkiej Transformacji Fouriera, zwłaszcza że jej implementacja nie jest niewiarygodnie skomplikowana, a interesująca.

Queue

Kolejka jest zaimplementowana poprawnie, kod jest przejrzysty. Wszystkie funkcje zdają się mieć poprawny opis i istotny cel. Jednakże nie znajdujemy celu w jakim ten kod został napisany. Problem implementacji kolejki nie był istotą działania programu ani nie jest potężnym zadaniem algorytmicznym i równie dobrze można było użyć gotowej struktury.

Co więcej, kod chyba nie jest zbyt optymalny. Niektóre operacje są degradowalne do $O(n)$, choć efektywniejsze implementacje (np. kopiec binarny MIN lub tablica) gwarantują niższą klasę złożoności i zapewne zostały użyte w `Data.Queue`.

Config

Trafnym rozwiązaniem jest stworzenie pliku konfiguracyjnego związanego z GUI, gdyż w prosty sposób osoba niezaznajomiona z kodem może zmienić podstawowe parametry konfiguracyjne mając gwarancję (względnie - nadzieję), że zmiana ta zostanie zastosowana wszędzie tam gdzie to konieczne.

Komentarze, dokumentacja

Projekt zawiera komentarze dokumentacyjne, które zwięźle tłumaczą zastosowanie poszczególnych funkcji. Za drobny mankament można uznać brak obecności głównych komentarzy modułowych, które tłumaczyłyby zastosowanie danego modułu (chlubnym wyjątkiem jest `Config.hs`).

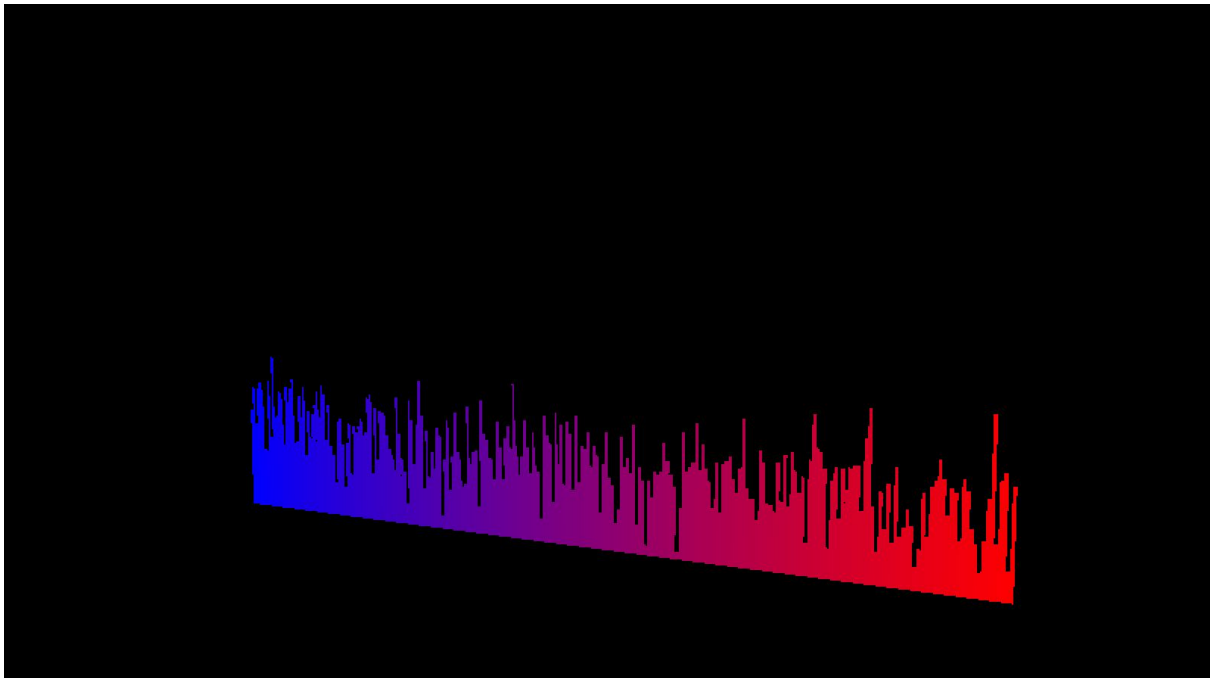
W projekcie brak również pliku `README.md`, który opisywałby projekt, wskazywał sugerowaną metodę uruchomienia itd.

Testy

Projekt zawiera testy realizowane przy pomocy biblioteki `QuickCheck` oraz jednostkowe, jednak skupiają się one jedynie wokół modułu `Queue`, nie dowodząc poprawności działania rdzenia aplikacji.

Działanie

Program zgodnie z oczekiwaniami wyświetla okno graficzne z podglądem wykresu. Niestety ani w `README` (którego nie było), ani w komentarzach dokumentacyjnych nie znaleźliśmy informacji o tym jakiego inputu oczekuje program, co zmusiło nas do przeanalizowania funkcji `runApplication :: IO ()` oraz skontaktowania się z autorami.



VCS

Za istotny element analizy kodu uważamy jego rozwój, który tutaj pozostaje dla nas owiany tajemnicą, gdyż repozytorium zawiera tylko jednego commita o nazwie "init". O jakości, zaletach czy problematyce często świadczą wybory podejmowane przez jego twórców, których tutaj nie możemy przeanalizować, więc nasza ocena siłą rzeczy pozostaje niepełna.