

Recenzja I – <https://github.com/arkgil/FP-game-of-life>

Projekt został napisany w innym języku niż haskell, dlatego zrobiliśmy review dla testów, które zostały napisane w haskellu.

W pliku MainTestSuite.hs znajdował się ciekawy według nas podział testów. Spodobał nam się zwłaszcza podział na grupy testów, ponieważ powoduje on lepszą czytelność (zwłaszcza jeżeli testów byłoby więcej).

W pliku MonadTests.hs znaleźliśmy testy parametryczne (QuickCheck) dla praw monad, które były przedstawione na wykładzie. Spodobało nam się generowanie losowych funkcji do testów (z odpowiednimi własnościami, tzn. shrinkable, showable), co zostało osiągnięte przez wykorzystanie Fun.

W pliku FunctorTests.hs znaleźliśmy testy parametryczne dla praw Functorów. Podobnie jak w MonadTests.hs koledzy skorzystali tutaj z generowania losowych funkcji.

W pliku ApplicativeTests.hs znaleźliśmy testy jednostkowe z użyciem HUnita dla praw dotyczących funktorów aplikatywnych. Testy były napisane dla list. Spodobało nam się wykorzystanie (@=?) do tworzenia asercji.

Podsumowując, koledzy napisali proste testy dla praw funktorów, funktorów aplikatywnych i monad. Wykorzystali kilka ciekawych i przydatnych według nas rzeczy, które staraliśmy się wymienić w powyższych akapitach.

Recenzja II – <https://git.iiet.pl/zagura/haskell-projekt>

Ponieważ przy pierwszym review udało nam się skomentować tylko testy, postanowiliśmy dokonać review drugiego projektu. Tym razem tematyką projektu był prosty serwer HTTP.

W pliku Main.hs dokonywane jest pobranie argumentów i wywołanie funkcji serve z odpowiednimi argumentami. Zwróciliśmy tutaj uwagę na rozwiązanie przypadku, gdy użytkownik nie poda żadnej ścieżki.

W pliku NetUtils.hs do sprawdzania poprawności portu użyto wyrażeń regularnych, dzięki czemu mogliśmy się przekonać, że proste wyrażenia regularne w haskellu nie różnią się od tych znanych nam z innych języków programowania.

W pliku Socket.hs przeprowadzana jest obsługa socketów. Mogliśmy porównać obsługę socketów w haskellu z obsługą socketów w znanych nam językach (javie, C). Jest ona podobna do C, dokonujemy po kolei operacji tworzenia socketu, związania adresu i portu z gniazdem (bind), nasłuchiwanie na zgłoszenia (listen) oraz akceptowania zgłoszeń (accept).

W pliku HTTP.hs dokonywana jest obsługa zapytań HTTP. Tutaj ponownie mogliśmy porównać haskella z innymi znanymi nam językami, tym razem pod kątem obsługi zapytań HTTP.

Cały projekt był przejrzystie napisany. Komentarze oraz dokumentacja pozwoliły nam dość szybko zrozumieć działanie całego projektu. Ciekawa była możliwość porównania haskella do innych języków programowania pod kilkoma aspektami.

Testy parametryczne (QuickCheck) skupiały się na funkcjach sprawdzających poprawność portu oraz funkcji zamieniającej port w postaci napisu na port w postaci liczby całkowitej.

Testy jednostkowe skupiały się na funkcji sprawdzającej poprawność portu, funkcji zamieniającej numer portu z postaci napisu na postać liczby całkowitej, oraz funkcji zamieniającej port z postaci PortNumber do postaci liczby całkowitej.

Wydaje się, że oba rodzaje testów pokrywały niemal tą samą część projektu.