

Weryfikacja Synchronizacji

Zuzanna Santorowska

May 2020

Treść Zadania

Zgodnie z zapowiedzia na wykładzie, ćwiczenie 5 będzie dotyczyło weryfikacji rozwiązań synchronizacyjnych z zadania nr 3 lub nr 4. Ćwiczenie to jest wykonywana zamiast ćwiczenia "gospodarka pamięci". Podaje Państwu adresy preprocesora weryfikatora Dedan i preprocesora Rybu:

<http://staff.ii.pw.edu.pl/dedan/>

<https://zyla.neutrino.re/rybu/>

Program Dedan jest ponadto na stronie przedmiotu. Zasady:

1. należy zweryfikować w programach Rybu/Dedan rozwiązania zadań z synchronizacji,
2. Za weryfikację z użyciem preprocesora Rybu i programu Dedan można otrzymać maksimum punktów (4), bez preprocesora Rybu (sam program Dedan) - 3 pkt.
3. Po ukończeniu ćwiczenia należy na adres wbd@ii.pw.edu.pl wysłać dokumentację weryfikacji: treść zadania z synchronizacji, rozwiązanie w języku C++, zapis w programie Rybu, wynik weryfikacji. Jeżeli weryfikacja pozwoliła wykryć błąd, to należy jeszcze dołączyć kontrprzykład z programu Dedan (ilustrujący błąd) i poprawione rozwiązanie.
4. Dokumentację należy spakować w pliku o nazwie `SOI.A_nazwisko.imie.ZIP`. W temacie maila należy umieścić `SOI.A_nazwisko.imie`.
5. Przesłanie dokumentacji jest warunkiem zaliczenia ćwiczenia.
6. Ćwiczenie ocenia prowadzący laboratorium. Dokumentację należy wysłać do Pana Daszczuka po uzyskaniu oceny.

Powodzenia!

Uruchamianie

Program napisany jest w języku Haskell. Przy użyciu narzędzia stack kompilacja jest prosta:

```
stack build
```

Program przyjmuje plik wejściowy poprzez standardowe wejście i wypisuje wynikowy model na standardowy wyjście. Chcac skompilować przykład kolos.txt należy wpisać:

```
stack exec rybu < examples/kolos.txt > kolos.dedan.txt
```

Po czym plik kolos.dedan.txt należy załadować do programu Dedan.