

# Tema 1 Paradigme de Programare

## 1. Descrierea scheletului de cod

- NetworkData.hs
  - definitiile tipurilor de date care trebuie folosite in implementare
  - definitiile sunt suficiente pentru implementarea cerintelor
  - acest fisier NU trebuie modificat
- NetworkFunction.hs
  - definitiile tipurilor care reflecta procesarea din retea
  - definitiile sunt suficiente si fisierul NU trebuie modificat
- Utils.hs
  - contine definitii pentru cel mai general flow compact, cat si pentru cel mai general flow
  - aici puteti adauga orice functii ajutatoare de care aveti nevoie
- TestCompactFlow.hs
  - fisierul de testare pentru CompactFlow.hs
  - functia "test" produce o lista cu testele rulate: indexul testului, rezultatul si punctajul maxim acordat pentru acel test
  - functia "score" intoarce punctajul obtinut pe tema (maxim 20pt)
  - in timpul lucrului la tema e indicat sa comentati/stergeti orice portiune (teste) din acest fisier, concentrandu-va asupra testelor care verifica portiunea la care lucrati acum.
  - acest fisier o sa fie folosit si in punctarea temei (nu vor exista alte teste)
  - pentru obtinerea unui punctaj partial (teme incomplete):
    - fisierul TREBUIE sa compileze
    - punctajul este cel returnat de score
    - modificarea testelor astfel incat sa treaca atrage punctaj 0 pe intreaga tema
- TestFlow.hs
  - fisierul de testare pentru Flow.hs
  - inainte de a rula testele pentru Flow.hs, trebuie implementata functionalitatea pentru CompactFlow
  - functia "test" produce o lista cu testele rulate: indexul testului, rezultatul si punctajul maxim acordat pentru acel test
  - functia "score" intoarce punctajul obtinut pe tema (maxim 35pt)
  - in timpul lucrului la tema e indicat sa comentati/stergeti orice portiune (teste) din acest fisier, concentrandu-va asupra testelor care verifica portiunea la care lucrati acum.
  - acest fisier o sa fie folosit si in punctarea temei (nu vor exista alte teste)
  - pentru obtinerea unui punctaj partial (teme incomplete):
    - fisierul TREBUIE sa compileze

- punctajul este cel returnat de score
  - modificarea testelor astfel incat sa treaca atrage punctaj 0 pe intreaga tema
- TestReachability.hs
  - fisierul de testare pentru Reachability.hs
  - inainte de a rula testele pentru Reachability.hs, trebuie implementata functionalitatea pentru CompactFlow si Flow
  - functia "test" produce o lista cu testele rulate: indexul testului, rezultatul si punctajul maxim acordat pentru acel test
  - functia "score" intoarce punctajul obtinut pe tema (maxim 45pt)
  - in timpul lucrului la tema e indicat sa comentati/stergeti orice portiune (teste) din acest fisier, concentrandu-va asupra testelor care verifica portiunea la care lucrati acum.
  - acest fisier o sa fie folosit si in punctarea temei (nu vor exista alte teste)
  - pentru obtinerea unui punctaj partial (teme incomplete):
    - fisierul TREBUIE sa compileze
    - punctajul este cel returnat de score
    - modificarea testelor astfel incat sa treaca atrage punctaj 0 pe intreaga tema
- CompactFlow.hs
  - aici se va inrola CompactFlow la Eq si FlowLike
- Flow.hs
  - aici se va inrola Flow la Eq si FlowLike
- Reachability.hs
  - aici va trebui implementata functia "reachability"
- Elements.hs
  - contine definitiile tipurilor de date pentru cele trei tipuri de elemente: wire (fir), Filter, Rewriter
  - pentru fiecare tip de date in parte trebuie sa se faca inrolarea la clasa Element
  - clasa element defineste functia de conversie de la Wire, Filter sau Rewriter la perechea (Match, Modify) corespunzatoare (i.e. ofera implementarea elementului specificat printr-unul dintre constructori)
  - Hint: pentru inrolarea elementelor puteti profita de functia "fuse" definita deja in NetworkFunction.hs

## 2. Precizari generale

- Orice modificare a definitiilor tipurilor de date specificate atrage punctaj 0. Definitiiile existente sunt suficiente pentru finalizarea temei, orice adaugari/modificari atrag complicatii nenecesare.
- Tema consta in efectuarea:
  - inrolarii Flow si CompactFlow la Eq si FlowLike

- inrolarii Wire, Rewriter si Filter la Element
- implementarii algoritmului "reachability" (conform enuntului)
- Respectati declaratiile functiilor care trebuie implementate (tipurile acestora) pentru a nu avea probleme la rularea testelor.
- Orice implementare voit incorecta care aduce punctaj pe anumite teste atrage punctaj nul pe intreaga tema. Nu rulati testele care testeaza functionalitate pe care nu ati implementat-o.
- Puteti folosi orice biblioteca sau feature al limbajului, nu exista restrictii.

### 3. Trimitere si testare tema

- Trebuie sa va asigurati inainte de deadline ca nu exista probleme de compilare.
- Codul care nu compileaza NU este punctat.
- Punctajul total este suma punctajelor intoarse de functia "score" din cele trei fisiere de test.
- Documentarea codului se va face ad-hoc, prin comentarii - fisierele Readme sau orice altceva, mai putin fisierele \*.hs nu sunt luate in considerare.
- Punctajul maxim este de 100 de puncte - obtinut in intregime ca rezultat al testerului.
- Tema va fi trimisa sub forma unei arhive "zip" cu formatul:  
     nume\_prenume\_3XYZ.zip  
     de exemplu: popovici\_matei\_321CB.zip sau popovici\_matei\_342C1.zip
- Din cauza volumului, temele sunt corectate automat, daca nu se respecta conventiile, studentul este responsabil.

### 4. Resurse

- Haskell in general (<http://learnyouahaskell.com/chapters>)
  - Module in Haskell (<http://learnyouahaskell.com/modules>)
  - Operatori constructori de date (:> , :!?:, :!?!: etc)
- <http://learnyouahaskell.com/making-our-own-types-and-typeclasses#recursive-data-structures>