

# Studie van Declaratieve Talen

## Prolog 3

### Gequoteerde Oefening

#### Richtlijnen

Los onderstaande opgave zelfstandig op (samenwerking is **niet** toegestaan!), en stuur ze binnen de 24 uren na de zitting in.

Zorg ervoor dat je programma correct is. Indien dit niet zo is, moet er in commentaar bijstaan wat er niet werkt. Zorg er voor dat je programma ook strikt de specificatie volgt zoals gevraagd (bvb. niet afwijken van de opgegeven predikaatnamen). Schrijf je oplossing in 1 enkel bestand dat je `oplossing.pl` noemt. Voeg vooraan het bestand de volgende lijnen toe:

```
% Voornaam Naam  
% Richting
```

Waarbij je jouw eigen gegevens invult. Als je oplossing klaar is, controleer dan eerst of de predikaatnamen overeenstemmen met de opgave. Stuur ze vervolgens via Toledo in (onderdeel Toetsen en opdrachten). Opgelet: je kan slechts één keer een oplossing indienen. **Vergeet niet op Submit te klikken om je oplossing effectief door te sturen.**

Indien je vragen hebt over deze opgave kan je mailen naar

`anton.dries@cs.kuleuven.be`.

## Opgave

Eén van de taken bij het analyseren van informatienetwerken (bvb. sociale netwerken) is het bepalen van “belangrijke” knopen in het netwerk. Een vaak gebruikt hulpmiddel daarbij zijn de zogenaamde *centrality measures*.

In deze opgave willen we de *closeness centrality*<sup>1</sup> van een knoop evalueren. Deze wordt berekend als de gemiddelde afstand tussen deze knoop en alle *andere* knopen in het netwerk. Deze wordt dus berekend als:

$$C(v) = \frac{\sum_{t \in V \setminus v} d(v, t)}{n - 1}$$

We nemen  $n$  als het aantal knopen in het netwerk. (We gaan dus uit van een volledig verbonden grafe.) Als afstandsmaat  $d(v, t)$  gebruiken we de lengte van het kortste pad tussen  $v$  en  $t$  in de (gerichte) grafe.

**Opgave** Schrijf een predikaat `closeness(Knoop,Waarde)` dat de closeness centrality berekent voor een gegeven knoop, bijvoorbeeld:

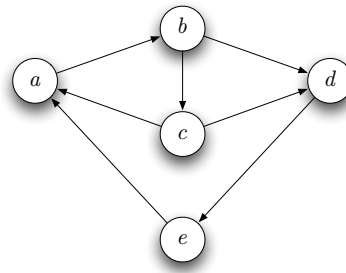
```
?- closeness(a,Res).  
Res = 2.
```

```
?- closeness(b,Res).  
Res = 1.5.
```

```
?- closeness(c,Res).  
Res = 1.5.
```

```
?- closeness(d,Res).  
Res = 2.5.
```

```
?- closeness(e,Res).  
Res = 2.25.
```



Bijvoorbeeld:

$$\begin{aligned} C(a) &= \frac{d(a, b) + d(a, c) + d(a, d) + d(a, e)}{4} \\ &= \frac{1 + 2 + 2 + 3}{4} = 2 \end{aligned}$$

**Opmerkingen** De bestaande predikaten `findall/3`, `sort/2` en `sumlist/2` kunnen nuttig zijn bij het uitwerken van deze opgave. De berekening van het kortste pad mag naïef gebeuren, dat wil zeggen, door eerst alle paden te genereren en dan het kortste pad te selecteren. Jullie mogen hierbij vertrekken van onderstaand predikaat (pas dit bijvoorbeeld aan zodat de lengte van het pad wordt teruggegeven). We gaan uit van een gerichte grafe, gedefinieerd als feiten `edge(a,b)`.

```
path(A,B,_) :- edge(A,B).  
path(A,B,P) :-  
    edge(A,C),  
    \+ member(C,P),  
    path(C,B,[C|P]).
```

Veel succes!

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Centrality#Closeness\\_centrality](http://en.wikipedia.org/wiki/Centrality#Closeness_centrality)