COMPSCI 752

Student ID: 437681250

Name: Shuo Mao

1. Write down the formal property graph model (V,E,η,λ,ν) for the property graph in Figure 1.

- V: in graph model it repersends as finite obecjt, called vectors. In figure 1: it includes `trek` and `hut`.
  Vector in figure1: 1; 2; 3.
- E: in graph model it repersends as finite object, called edges. In figure 1: it repersends the relationship between each hut.
  Edges in figure1: 10; 11; 12.
- η: E → V × V. A function mapping each edges to pair vectors. In figure 1: it repersends as the hut connections. Edge relationship
  η: 12 -> (3,1); 10 -> (1,2); 11 -> (2,3).
- λ: V ∪ E → P(L).A function assigning each oject with a finite label. In Figure 1: labels are `hut`, `trek` and `section`.Labelling
  λ: 1 ->{trek,hut}; 2 -> {hut}; 3 -> {hut}; 10 -> {section}; 11 -> {section};12 -> {section}
- ν:(V ∪ E) × K → N. a partial function assigning values for properties to objects. In Figure 1: values includes: `name`, `location`,`#beds`,`shower`, `distance`.
  Value assignmen: (1,name) -> 'kepler'; (1,location) -> 'Fiordland`; (1,#beds) -> '25'; (1,shower) -> 'yes'; (2,#beds) ->'20'; (2,shower) -> 'yes'; (3,#beds) ->'20'; (3,shower) -> 'yes'

2.Write the following English language query in regular property graph logic: Return the trek nodes that have 2 or 3 sections

```
:sectionTrek(x,y) in a <-:section(x,y)/:section(x,y)as t |
                        :section(x,y)/:section+(x,y) as t, :trek(t).
result (x) <- :section(x,x).
```

3.Write the following English language query in regular property graph logic: Find all pairs of hut nodes, each having showers and at least 15 beds, and directly connected by a section

```
:qualifyingHut(h) IN a ← :Hut(h), h.beds >= 15, h.shower = 'yes'.
:connectedHuts(h1, h2) b ← :section(h1, h2), :qualifyingHut(h1), :qualifyingHut(h2).
result(x,y) <- :connectedHuts(x,y)
```

4.Write the following English language query in regular property graph logic:Return all trek nodes that have some section, each at most 25kms long and only connecting huts with showers available.

```
:section25 <- :section(x,y) as w, w.distance <= '25'.
result(x,y) <- :section25(x,y), x.shower = 'yes', y.shower = 'yes'
```

5.Write down the queries from Questions 2, 3, and 4 in regular property graph algebra.

question 2:
⋈/∅ src1 (⋈/Φ,x trg1,trg1 (:section/:section | :section/:section+))
where
Φ is: (λ(trg1) = :Trek)

question3:
⋈/∅ src1 (⋈/Φ,x trg1,trg2 (:section))
where
Φ is: (λ(trg1) = :Hut) ∧ (trg1.showers = yes) ∧ (trg1.beds >= 15) ∧ (λ(trg2) = :Hut) ∧ (trg2.showers = yes) ∧ (trg2.beds >= 15)
x is an arbitrary context identifier

question 4:
⋈/∅ src1 (⋈/Φ,x trg1,trg1 (:section))
where
Φ is: (λ(trg1) = :Hut) ∧ (trg1.showers = yes) ∧ (:section.distance <= 25)
x is an arbitrary context identifier