Програмски парадигми

Домашна задача 2

Стефан Милев | 206055

1.

а)

Дефиниран е предикатот generate_rodeni_razlicen_grad за да се генерираат сите лица за кои важат условите дека се деца и родени во градови различни од градовите на родителите. Потоа, предикатот rodeni_razlicen_grad ги собира сите генерирани лица и брои колку има со помош на вградениот предикат length.

б)

Дефиниран е предикатот generate_predci кој ги генерира сите лица кои се предци на лицето со шифрата која е предадена како прв аргумент на предикатот. Во предикатот прво се земаат сите лица во парови со ист пол, па потоа преку предикатот diff_ignore_year проверува дали родендените на двајцата се разделени за најмногу една недела, па потоа проверува дали лицето е предок преку предикатот ancestor, или ги генерира сите предци доколку аргументот не е поставен. Предикатот parent проверува само дали второто лице е родител на првото лице, додека предикатот ancestor проверува дали постои лице или повеќе лица кои родители меѓусебно и се наоѓаат меѓу двете лица за кои не интересира, или пак се директно дете и родител, со користење на предикатот parent. Предикатот diff_less_than_week е за проверување дали родендените се разделени за една недела или помалку, додека предикатот diff_ignore_year работи на начин што прво се пресметува редниот број на денот во годината за родендените, па потоа се зема разликата меѓу нив со апсолутна вредност, па со помош на diff_less_than_week, се проверува дали разликата е помала од 7 дена, или пак поголема од 365 - 7 дена, т.е. се наоѓаат на почеток и крај на годините, соодветно.

2.

а)

Предикатот generate_calls е за генерирање на сите повици според примач на повикот, додека со предикатот count_calls се бројат овие повици. Предикатот generate_call_counts служи за генерирање на бројот на повици за секој број. Предикатот najbroj го наоѓа најголемиот број на повици од сите, со тоа што го бара бројот на повици кој е еднаков на максималниот број на повици.

б)

Предикатот count го пресметува бројот на елементи кои се наоѓаат во листата во вториот аргумент со првиот аргумент, и тој број го сместува во третиот аргумент. Предикатот total_calls_duration го пресметува времетраењето на сите повици од бројот во првиот аргумент, во третиот аргумент. Предикатот count_members брои колку пати се појавува првиот аргумент во вториот аргумент кој е во форма на листа од листи, и бројот го поставува во третиот аргумент. Предикатот count_sms прима праќач во првиот аргумент и примач во вториот аргумент и пресметува колку пораки биле испратени во соодветната насока, и тој број го поставува во третиот аргумент. Предикатот outgoing исто така прима праќач и примач во првиот и вториот аргумент, соодветно, и ја пресметува вкупната „вредност“ (траење на повици + СМС пораки), и ја поставува во третиот аргумент. Предикатот total_duration е за пресметување на вкупното времетраење меѓусебно, и го поставува во третиот аргумент. Предикатот generate_durations ги генерира сите вкупни времетраења за сите телефонски броеви, па потоа преикатот omilen ги собира сите вкупни времетраења, и го наоѓа времетраењето кое е еднакво со максималното, па го поставува во вториот аргумент.

3.

а)

Предикатот generate_clients е за генерирање на клиенти кои имаат услуги со почетна или крајна локација еднаква на локацијата дадена во првиот аргумент. Потоа, предикатот izboj_lokacija ги брои клиентите кои го исполнуваат условот да имаат услуги со почетна или крајна локација еднаква на таа дадена во првиот аргумент, и бројката ја сместува во вториот аргумент.

б)

Предикатот direct_distance го калкулира двонасочното растојание меѓу две локации во третиот аргумент. Предикатот generate_service_distances е за генерирање на сите дистанци за дадената рута во аргументите, и ја поставува во третиот аргумент, и го

користи предикатот generate_service_distances_helper за заобиколување на проблемот на бескрајна рекурзија која може да се предизвика доколку има пат во кој локации се повторуваат, т.е. има лупа. Прво проверува дали двете локации се директно поврзани, па доколку не се, бара дистанци преку локација во средина, па ги собира дистанците. Предикатот minimum_distance служи за наоѓање на најмалата дистанца од сите кои се наоѓаат во графот. Предикатот total_distances е за да се калкулира вкупната дистанца од сите рути во првиот аргумент, и вкупната дистанца ја поставува во вториот аргумент. Преку предикатот generate_total_distances се генерираат сите вкупни дистанци, т.е. збировите на сите дистанци на сите услуги за сите клиенти. Потоа, предикатот najmnogu_kilometri ги собира сите генерирани дистанци и го наоѓа клиентот кој има вкупна дистанца еднаква на максималната.

в)

Предикатот price_service го користи предикатот minimum_distance (од барањето под б) за да ја најде вкупната цена на една услуга (должина * цена по км). Предикатот generate_vehicle_price ги генерира сите цени на услуги за возилото предадено во првиот аргумент. Предикатот generate_service_vehicle ги генерира сите возила кои што се појавуваат во услугите на сите клиенти. Предикатот total_price е за да се калкулира вкупната цена на сите услуги за возилото од првиот аргумент, и вкупната цена ја поставува во вториот аргумент. Предикатот generate_total_price е за генерирање на сите вкупни цени на сите возила. Предикатот najmnogu_zarabotil ги пресметува сите вкупни цени и го наоѓа возилото со вкупна цена еднаква на максималната.