

SYSTEMY EKSPERTOWE

ZAJĘCIA 5-6

Zad. 1

Mamy graf skierowany opisany w instancjach ramy

WIERZCHOLEK

SLOT NAZWA (STRING)

MULTISLOT SASIEDZI (STRING)

Dodatkowo mamy ramę

PYTANIE

SLOT START (STRING)

SLOT META (STRING)

SLOT ODPOWIEDZ (STRING DEFAULT NIE)

Dla każdej instancji w ramie PYTANIE należy do slotu ODPOWIEDZ wpisać wartość TAK, jeśli istnieje droga pomiędzy wierzchołkami pobranymi ze slotów START i META.

```
(
deftemplate wierzcholek
(slot nazwa (type string))
(multislot sasiedzi (type string))
)
```

```
(
deftemplate pytanie
(slot start (type string))
(slot meta (type string))
(slot odpowiedz (type string) (default NIE))
)
```

```
(def facts graf
(
wierzcholek
(nazwa "A")
(sasiedzi "B" "C")      )
(
wierzcholek
(nazwa "B")
(sasiedzi "C" "D")      )
(
wierzcholek
(nazwa "C")
(sasiedzi "A" "B" "D")  )
(
wierzcholek
(nazwa "D")
(sasiedzi "B" "C")      )
(
wierzcholek
(nazwa "E")
(sasiedzi "A" "B" "C" "D")  )
(
pytanie
(start "D")
(meta "E")  )
)
```

```
(reset)
(facts)
```

```
(
deftemplate polaczenie
(declare (ordered TRUE))
)
```

```

(
defrule regula_1
  (declare (salience 0))
  (wierzcholek (nazwa ?n) (sasiedzi $? ?sasiad $?))
=>
  (assert (polaczenie ?n ?sasiad))
  (printout t "Jest polaczenie pomiedzy " ?n " a " ?sasiad crlf)
)

(defrule regula_2
  (declare (salience -1))
  (polaczenie ?x1 ?x2 )
  (polaczenie ?x2 ?x3)
  (test (neq ?x1 ?x3))
=>
  (assert (polaczenie ?x1 ?x3))
  (printout t "Jest polaczenie przechodzace przez inne wierzcholki pomiedzy " ?x1 " a
" ?x3 crlf)
)

(defrule regula_3
  (declare (salience -2))
  ?p <- (pytanie (start ?s) (meta ?m))
  (polaczenie ?s ?m)
=>
  (modify ?p (odpowiedz "TAK"))
  (printout t "Jest polaczenie start = " ?s ", meta = " ?m crlf)
)

(run)

(facts)

```

Zamienić pytanie na: START = A, META = E. Sprawdzić wynik działania.

Zad. 2

Mamy zdefiniowane następujące ramy:

ZAMOWIENIE

SLOT PRODUKT (STRING)
 SLOT LICZBA_JEDNOSTEK (FLOAT)
 SLOT WARTOSC_CALKOWITA (FLOAT)
 SLOT OCENA (STRING)

PRODUKT

SLOT NAZWA (STRING)
 SLOT CENA_JEDNOSTKOWA (FLOAT)

Napisać:

1. Regułę obliczającą całkowitą wartość zamówienia na podstawie liczby jednostek oraz ceny jednostkowej (ma wstawiać wynik do slotu WARTOSC_CALKOWITA)
2. Regułę wstawiającą do slotu OCENA napis WYSOKA, jeżeli wartość całkowita zamówienia jest większa od 200 lub liczba jednostek jest większa od 400.

Która z tych reguł odpali się jako pierwsza ?

```

(
deftemplate zamowienie
(slot produkt (type string))
(slot liczba_jednostek (type float))
(slot wartosc_calkowita (type float))
(slot ocena (type string))
)

(
deftemplate produkt
(slot nazwa (type string))
(slot cena_jednostkowa (type float))
)

```

```
(deffacts zamowienia
(
zamowienie
(produkt "produkt.1")
(liczba_jednostek 10)  )
(
produkt
(nazwa "produkt.1")
(cena_jednostkowa 40)  )
)

(reset)
(facts)

(
defrule regula_1
  ?z <- (zamowienie (produkt ?p)
         (liczba_jednostek ?l)
         (wartosc_calkowita ?w&~nil))
=>
  (produkt (nazwa ?p) (cena_jednostkowa ?c))
  (bind ?wartosc (* ?l ?c))
  (modify ?z (wartosc_calkowita ?wartosc))
  (printout t "Wartosc zamowienia to: " ?wartosc crlf)
)

(
defrule regula_2
  ?z <- (zamowienie (liczba_jednostek ?l)
         (wartosc_calkowita ?w&~nil))
=>
  (test (or (> ?l 400) (> ?w 200)))
  (modify ?z (ocena "WYSOKA"))
  (printout t "Wysoka ocena zamowienia!!!" crlf)
)

(run)

(facts)
```

Zad. 3

Dane są ramy:

STUDENT

SLOT ID_STUDENTA (STRING)
SLOT IMIE (STRING)
SLOT NAZWISKO (STRING)
SLOT STYPENDIUM (FLOAT)
SLOT SREDNIA (FLOAT)
SLOT ILE_ZALICZEN (FLOAT)
SLOT WSZYSTKIE_ZALICZENIA (STRING DEFAULT NIE)

ZALICZENIE

SLOT STUDENT (STRING)
SLOT PRZEDMIOT (STRING)
SLOT OCENA (FLOAT)

1. Obliczyć średnią ocen studenta i wpisać do slotu SREDNIA
2. Przydzielić STYPENDIUM, jeśli są wszystkie zaliczenia

ŚREDNIA	STYPENDIUM
< 4.0	0
<4.0; 4.25)	200
<4.25; 4.5)	300
<4.5; 4.75)	400
<4.75; 5.0>	500

```
(
deftemplate student
(slot id_studenta (type string))
(slot imie (type string))
(slot nazwisko (type string))
(slot stypendium (type float))
(slot srednia (type float))
(slot ile_zaliczen (type float))
(slot wszystkie_zaliczenia (type string) (default NIE))
)
```

```
(
deftemplate zaliczenie
(slot student (type string))
(slot przedmiot (type string))
(slot ocena (type float))
)
```

```
(deffacts studenci
(
student
(id_studenta "id.1")
(imie "imie.1")
(nazwisko "nazwisko.1") )
(
student
(id_studenta "id.2")
(imie "imie.2")
(nazwisko "nazwisko.2") )
(
zaliczenie
(student "id.1")
(przedmiot "p.1")
(ocena 3.0) )
(
zaliczenie
(student "id.1")
(przedmiot "p.2")
(ocena 4.0) )
(
zaliczenie
(student "id.1")
)
```

```

(przedmiot "p.3")
(ocena 3.5) )
(
zaliczenie
(student "id.1")
(przedmiot "p.4")
(ocena 2.0) )
(
zaliczenie
(student "id.2")
(przedmiot "p.1")
(ocena 4.5) )
(
zaliczenie
(student "id.2")
(przedmiot "p.2")
(ocena 4.0) )
(
zaliczenie
(student "id.2")
(przedmiot "p.3")
(ocena 4.5) )
(
zaliczenie
(student "id.2")
(przedmiot "p.4")
(ocena 5.0) )
(
zaliczenie
(student "id.2")
(przedmiot "p.5")
(ocena 4.5) )
)

(reset)
(facts)

(
deftemplate zal
(declare (ordered TRUE))
)

(
deftemplate srednia_studenta
(declare (ordered TRUE))
)

(
deftemplate nie_wszystkie
(declare (ordered TRUE))
)

(
defrule regula_1
  (declare (salience 0))
  (zaliczenie (student ?s)
    (przedmiot ?p)
    (ocena ?o))
=>
  (assert (zal ?s ?p ?o))
)

(
defrule regula_2
  (declare (salience -1))
  (zal ?s ?p ?o&:(< ?o 3))
=>
  (assert (nie_wszystkie ?s))
)

```

```

(
defrule regula_3
  (declare (salience -2))
  ?s <- (student (id_studenta ?id))
  (not (exists (nie_wszystkie ?id)))
=>
  (modify ?s (wszystkie_zaliczenia TAK))
  (printout t "Student " ?id " ma wszystkie zaliczenia" crlf)
)

(
defrule regula_4
  (declare (salience -3))
  ?s <- (student (id_studenta ?id) (srednia ?sr&nil) (ile_zaliczen ?ile&nil))
=>
  (modify ?s (srednia 0) (ile_zaliczen 0))
)

(
defrule regula_5
  (declare (salience -4))
  ?s <- (student (id_studenta ?id) (srednia ?sr&~nil) (ile_zaliczen ?ile&~nil))
  ?z <- (zal ?id ?p ?o)
=>
  (retract ?z)
  (modify ?s (srednia (+ ?sr ?o)) (ile_zaliczen (+ ?ile 1)))
)

(
defrule regula_6
  (declare (salience -5))
  ?s <- (student (id_studenta ?id) (srednia ?sr&~nil) (ile_zaliczen ?ile&~nil))
  (not (exists (srednia_studenta ?id ?x)))
=>
  (bind ?srednia (/ ?sr ?ile))
  (assert (srednia_studenta ?id ?srednia))
)

(
defrule regula_7
  (declare (salience -6))
  ?s <- (student (id_studenta ?id))
  (srednia_studenta ?id ?srednia)
=>
  (modify ?s (srednia ?srednia))
)

(
defrule regula_8
  (declare (salience -7))
  ?s <- (student (id_studenta ?id) (srednia ?sr&~nil) (wszystkie_zaliczenia TAK))
  (test (< ?sr 4.0) )
=>
  (modify ?s (stypendium 0))
)

(
defrule regula_9
  (declare (salience -8))
  ?s <- (student (id_studenta ?id) (srednia ?sr&~nil) (wszystkie_zaliczenia TAK))
  (test
  (and (>= ?sr 4.0) (< ?sr 4.25)))
=>
  (modify ?s (stypendium 200))
)

```

```

(
defrule regula_10
  (declare (salience -9))
  ?s <- (student (id_studenta ?id) (srednia ?sr&~nil) (wszystkie_zaliczenia TAK))
  (test
  (and (>= ?sr 4.25) (< ?sr 4.5)))
=>
  (modify ?s (stypendium 300))
)

(
defrule regula_11
  (declare (salience -10))
  ?s <- (student (id_studenta ?id) (srednia ?sr&~nil) (wszystkie_zaliczenia TAK))
  (test
  (and (>= ?sr 4.5) (< ?sr 4.75)))
=>
  (modify ?s (stypendium 400))
)

(
defrule regula_12
  (declare (salience -11))
  ?s <- (student (id_studenta ?id) (srednia ?sr&~nil) (wszystkie_zaliczenia TAK))
  (test
  (and (>= ?sr 4.75)))
=>
  (modify ?s (stypendium 500))
)

(run)

(facts)

```

Zad. 4

Dane są ramy:

PLYTA

SLOT WYKONAWCA (STRING)
 SLOT CZAS_TRWANIA (FLOAT DEFAULT 0)
 MULTISLOT UTWORY (STRING)

UTWOR

SLOT TYTUL (STRING)
 SLOT CZAS (FLOAT)

Obliczyć łączny czas trwania płyty (wstawić odpowiednią wartość do slotu CZAS_TRWANIA).

Zad. 5

W sklepie internetowym mamy zdefiniowane ramy:

ZAMOWIENIA

SLOT ID_KLIENTA (STRING)
 SLOT KWOTA_ZAMOWIENIA (FLOAT)
 SLOT DO_ZAPLATY (FLOAT)

KLIENT

SLOT ID_KLIENTA (STRING)
 SLOT ODLEGLOSC (FLOAT)
 SLOT WYDATKI (FLOAT)

Rama KLIENT zawiera informacje o kupującym w naszym sklepie. Odległość między siedzibą firmy i miejscowością klienta wpisana jest w slot ODLEGLOSC. WYDATKI to kwota jaką do tej pory klient wydał w naszym sklepie. Zamówienie jest aktualnie realizowanym zamówieniem. KWOTA_ZAMOWIENIA zawiera sumę cen produktów, które klient wybrał.

System powinien obliczyć DO_ZAPLATY biorąc pod uwagę następujące zasady:

- Do zamówienia doliczany jest koszt transportu:
 - 7 zł jeżeli odległość \leq 300 km,
 - 11 zł jeżeli odległość $>$ 300 km,
 - jeżeli kwota zamówienia przekroczyła 200 zł to koszt transportu nie jest doliczany.
- Klient, który do tej pory wydał u nas więcej niż 1000 zł ma 5% zniżki, a więcej niż 5000 zł ma 10% zniżki.

Zad. 6

Dane są ramy:

OSOBA

SLOT ID_OSOBY (STRING)
SLOT IMIE (STRING)
SLOT NAZWISKO (STRING)
MULTISLOT ZAINTERESOWANIA (STRING)
SLOT ZASOBY (FLOAT)

TARGI

SLOT NAZWA (STRING)
MULTISLOT DZIEDZINY (STRING)
SLOT KOSZT (FLOAT)

Dla każdej osoby znajdź najdroższe i najtańsze targi odpowiadające zainteresowaniom tejże osoby. Targi są brane pod uwagę jedynie jeśli wszystkie wartości z multislota DZIEDZINY znajdują się w multislocie ZAINTERESOWANIA. Wypisz ile będzie kosztował udział w targach oraz ile zasobów zostanie.

Jeśli koszt uczestnictwa w wybranych targach jest wyższy niż posiadane zasoby wypisz że osoba nie może uczestniczyć w targach.