Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский

политехнический университет»

(ПНИПУ)

Кафедра вычислительной математики и механики

Лабораторная работа № 1

по дисциплине: «ИНТЕЛЛЕКТУАЛЬНЫЕ ИСИТ»

по теме: «Проектирование и реализация продукционной базы знаний»

Выполнил:

Студент группы ИСТ-19-1бзу

Соболь Е. В.

Пермь

2022 г.

В системах знаний основанных на правилах представлены не с помощью декларативного представления знаний, а в форме многочисленных правил, которые указывают, какие заключения должны быть сделаны или не сделаны в различных ситуациях. Каждое правило обозначается именем. Вслед за именем находится часть IF правила. Участок правила между частями IF и THEN правила упоминается под разными именами, такими как антецедент, условная часть, и т.п. Вслед за частью THEN правила находится список действий, которые должны быть выполнены после запуска правила. Эта часть правила называется консеквентом, или правой частью.

Составим правила для нашей продукционной базы знаний.

IF (engine-starts = None) and (repair = None) THEN engine-starts ("Запускается ли двигатель (да - True / нет - False)?")

IF (engine-starts = True) and (repair = None) THEN runs-normally ("Двигатель работает нормально (да - True / нет - False)?")

IF (engine-starts = False) and (repair = None) THEN engine-rotates ("Вращается ли двигатель (да - True / нет - False)? ")

IF (runs-normally = False) and (repair = None) THEN engine-sluggish ("Работает ли двигатель вяло (да - True / нет - False)?")

IF (runs-normally = Fasle) and (repair = None) THEN engine-misfires ("Имеет ли двигатель пропуски зажигания (да - True / нет - False)?")

IF (runs-normally = Fasle) and (repair = None THEN engine-knocks ("Стук в двигателе (да - True / нет - False)?")

IF (runs-normally = Fasle) and (repair = None) THEN engine-output-low ("Мощность двигателя низкая (да - True / нет - False)?")

IF (engine-starts = False) and (engine-rotates = True) and (repair = None) THEN tank-has-gas ("Есть ли в баке бензин (да - True / нет - False)?")

IF (engine-rotates = False) and (repair = None) THEN battery-has-charge ("Аккумулятор заряжен (да - True / нет - False)?")

IF (engine-starts = False) and (engine-rotates = True) or (engine-output-low = True) and (repair = None) THEN point-surface-state ("Каково состояние контактов на катушке? (normal - нормальный / burned - сгоревшие / contaminated - загрязненный)?")

IF (engine-starts = False) and (engine-rotates = False) and (battery-has-charge = True) and (repair = None) THEN conductivity-test-positive ("Проверка проводимости катушки зажигания положительна (да - True / нет - False)?")

IF (runs-normally = True) and (repair = None) THEN repair ("Ремонт не требуется.")

IF (engine-sluggish = True) and (repair = None) THEN repair ("Очистите топливопровод.")

IF (engine-misfires = True) and (repair = None) THEN repair ("Регулировка точечного зазора.")

IF (engine-knocks = True) and (repair = None) THEN repair ("Регулировка тайминтга двигателя")

IF (tank-has-gas = False) and (repair = None) THEN repair ("Добавьте газ.")

IF (battery-has-charge = False) and (repair = None) THEN repair ("Зарядите аккумулятор.")

IF point-surface-state = 'burned') and (repair = None) THEN repair ("Замените контакты.")

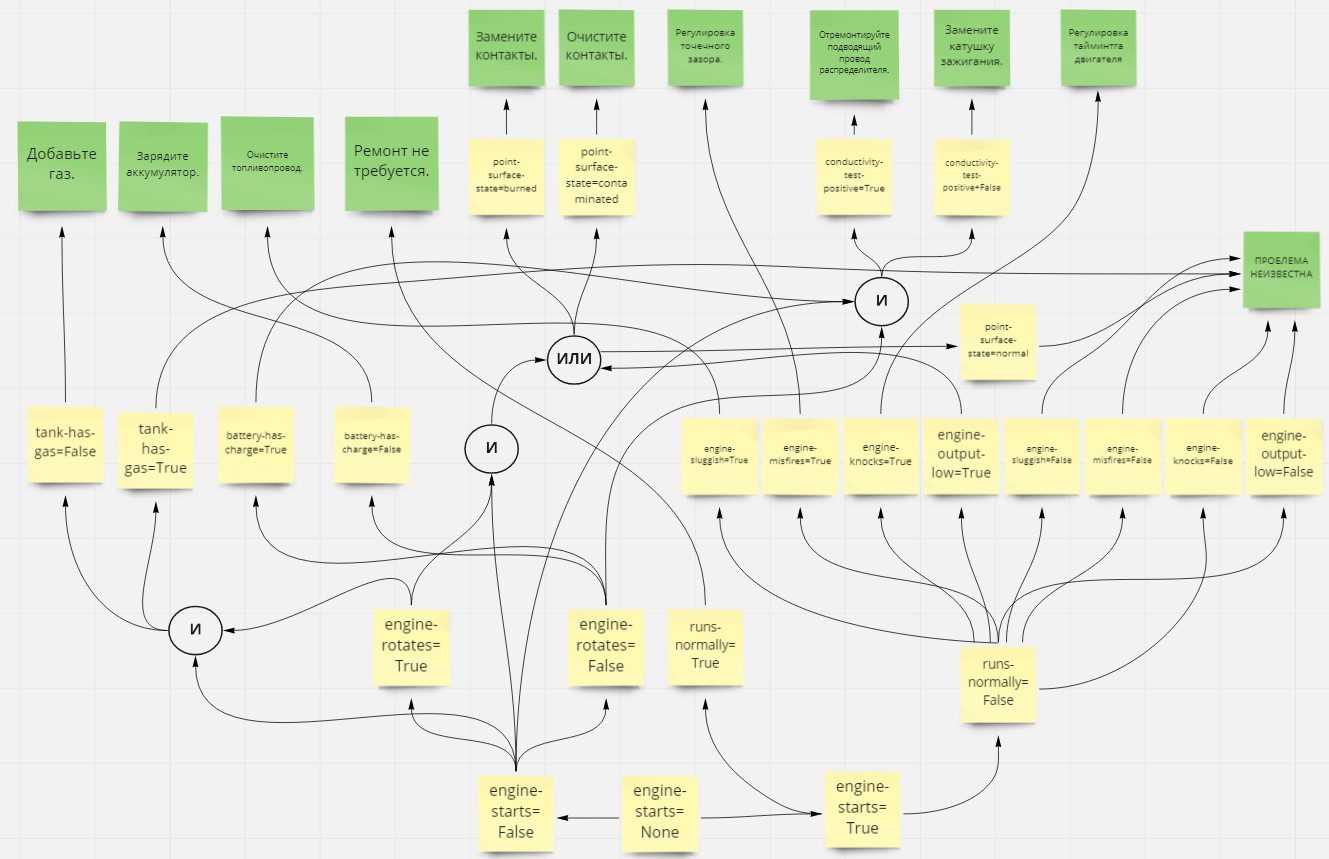
IF point-surface-state = 'contaminated') and (repair = None) THEN repair ("Очистите контакты.")

IF (conductivity-test-positive = True) and (repair = None) THEN repair ("Отремонтируйте подводящий провод распределителя.")

IF (conductivity-test-positive = False) and (repair = None) THEN repair ("Замените катушку зажигания.")

IF (repair = None) THEN repair ("Обратитесь в автосервис, для диагностики!")

Визуально оформленный граф И/ИЛИ:



Реализую её в виде двух файлов:

Первый текстовый файл rules.txt хранит правила:

determine-engine-state||(current\_dict['engine-starts'] == None) and (current\_dict['repair'] == None)||engine-starts

determine-runs-normally||(current\_dict['engine-starts'] == True) and (current\_dict['repair'] == None)||runs-normally

determine-rotation-state||(current\_dict['engine-starts'] == False) and (current\_dict['repair'] == None)||engine-rotates

determine-sluggishness||(current\_dict['runs-normally'] == False) and (current\_dict['repair'] == None)||engine-sluggish

determine-misfiring||(current\_dict['runs-normally'] == False) and (current\_dict['repair'] == None)||engine-misfires

determine-knocking||(current\_dict['runs-normally'] == False) and (current\_dict['repair'] == None)||engine-knocks

determine-low-output||(current\_dict['runs-normally'] == False) and (current\_dict['repair'] == None)||engine-output-low

determine-gas-level||(current\_dict['engine-starts'] == False) and (current\_dict['engine-rotates'] == True) and (current\_dict['repair'] == None)||tank-has-gas

determine-battery-state||(current\_dict['engine-rotates'] == False) and (current\_dict['repair'] == None)||battery-has-charge

determine-point-surface-state||((current\_dict['engine-starts'] == False) and (current\_dict['engine-rotates'] == True)) or ((current\_dict['engine-output-low'] == True) and (current\_dict['repair'] == None))||point-surface-state

defrule determine-conductivity-test||(current\_dict['engine-starts'] == False) and (current\_dict['engine-rotates'] == False) and (current\_dict['battery-has-charge'] == True) and (current\_dict['repair'] == None)||conductivity-test-positive

defrule normal-engine-state-conclusions||(current\_dict['runs-normally'] == True) and (current\_dict['repair'] == None)||"Ремонт не требуется."

defrule engine-sluggish||(current\_dict['engine-sluggish'] == True) and (current\_dict['repair'] == None)||"Очистите топливопровод."

defrule engine-misfires||(current\_dict['engine-misfires'] == True) and (current\_dict['repair'] == None)||"Регулировка точечного зазора."

defrule engine-knocks||(current\_dict['engine-knocks'] == True) and (current\_dict['repair'] == None)||"Регулировка тайминтга двигателя"

defrule tank-out-of-gas||(current\_dict['tank-has-gas'] == False) and (current\_dict['repair'] == None)||"Добавьте газ."

defrule battery-dead||(current\_dict['battery-has-charge'] == False) and (current\_dict['repair'] == None)||"Зарядите аккумулятор."

defrule point-surface-state-burned||(current\_dict['point-surface-state'] == 'burned') and (current\_dict['repair'] == None)||"Замените контакты."

defrule point-surface-state-contaminated||(current\_dict['point-surface-state'] == 'contaminated') and (current\_dict['repair'] == None)||"Очистите контакты."

defrule conductivity-test-positive-yes||(current\_dict['conductivity-test-positive'] == True) and (current\_dict['repair'] == None)||"Отремонтируйте подводящий провод распределителя."

defrule conductivity-test-positive-no||(current\_dict['conductivity-test-positive'] == False) and (current\_dict['repair'] == None)||"Замените катушку зажигания."

defrule no-repairs||current\_dict['repair'] == None||"Обратитесь в автосервис, для диагностики!"

Второй текстовый файл params.txt хранит промежуточные звенья графа:

engine-starts||"Запускается ли двигатель (да - True / нет - False)?"

runs-normally||"Двигатель работает нормально (да - True / нет - False)?"

engine-rotates||"Вращается ли двигатель (да - True / нет - False)? "

engine-sluggish||"Работает ли двигатель вяло (да - True / нет - False)?"

engine-misfires||"Имеет ли двигатель пропуски зажигания (да - True / нет - False)?"

engine-knocks||"Стук в двигателе (да - True / нет - False)?"

engine-output-low||"Мощность двигателя низкая (да - True / нет - False)?"

tank-has-gas||"Есть ли в баке бензин (да - True / нет - False)?"

battery-has-charge||"Аккумулятор заряжен (да - True / нет - False)?"

point-surface-state||"Каково состояние контактов на катушке? (normal - нормальный / burned - сгоревшие / contaminated - загрязненный)?"||[normal,burned,contaminated]

conductivity-test-positive||"Проверка проводимости катушки зажигания положительна (да - True / нет - False)?"