

Универзитет у Београду  
Електротехнички факултет

РАЧУНАРСТВО У БИО-МЕДИЦИНИ

## Извештај пројектног задатка

Ментор  
др Бранко Маровић

Студент  
Никола Пејић  
3106/2018

Београд, 2019



## Поставка проблема

Пројекат представља прву фазу у систему за дијагностиковање пацијента и одређивања одговарајућих корака у његовом лечењу. Прва фаза има за циљ да користећи медицинске податке о пацијенту (нпр. крвне слике) утврди да ли пацијент болује од неке болести, односно да постави дијагнозу пацијенту. Зарад лакше интеракције са системом, одлучено је да се он има REST интерфејс.

Интеракција са системом реализује се кроз следеће кораке:

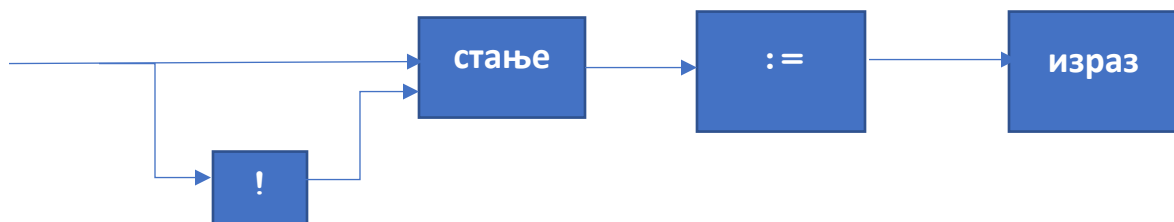
- Лекар напише сет правила на основу којих се одређује стање пацијента (поставља дијагнозу).
- При покретању, сервис прочита спецификације правила, парсира их и направи модел на основу којег касније поставља дијагнозу.
- Корисник интерагује са REST интерфејсом и може да
  - Затражи сва стања која систем може да одреди, у којем случају добија списак стања.
  - Затражи списак параметара које треба да достави да би се извршила евалуација одређеног стања. Као одговор добија списак параметара.
  - Затражи евалуацију одређеног стања или свих стања (уколико се затражи евалуација стања „all“). Као одговор добија списак стања која су могла бити одређена на основу достављених података, као и списак додатних података који су потребни да би се извршила евалуација осталих стања.

## Имплементација решења

Решење се састоји из три дела – парсера, евалуатора израза и REST интерфејса. Решење је имплементирано у програмском језику Java. У процесу имплементације парсера коришћене су и `jflex` (лексер) и `cup` (генератор LALR парсера) технологије, док је за имплементацију REST интерфејса коришћен `JerseyEmbeddedHTTPServerCrunchify` сервер.

### Парсирање спецификације

За потребе парсирање спецификације правила написан је једноставан парсер. Спецификација се састоји од произвољног броја линија, које могу бити празне или садржати правило и/или коментар. Коментар почиње са две косе црте (`//`) и наставља се до краја реда. Свако правило специфицирано је у једном реду и његова структура приказана је у дијаграму испод.



Слика 1 - приказ конструкције једног правила

Уколико се испред стања стави узвичник, така се то стање означава као излазно стање, и оно ће бити приказано у листингу свих стања које систем може да идентификује. Тиме је омогућено да се, у случају комплексног правила за дефинисање стања, дефинишу помоћна стања, чиме би спецификација излазног стања постала прегледнија. Ова особина такође омогућава да се уведу и симболичке константе које се не би приказивале кориснику (једноставно се специфицира `granica := 200` и касније се ова симболичка константа користи у другим изразима као да је параметар). Сами називи стања и симболичких константи представљају низ алфанумеричких карактера који почиње словом.

Након имена стања следи знак доделе (`:=`), након којег следи израз. Израз представља било који аритметичко-логичко-релациони израз (произвољне дужине) у којем су операнди друга стања или литерали. Оператори и резервисане речи које се могу користити у изразима дати су у наставку.

Табела 1 - Преглед оператора и резервисаних речи

Аритметички оператори	Логички оператори	Релациони оператори	Резервисане речи	Остали оператори
<b>+</b>	<b>AND</b>	<b>&lt;</b>	<b>true</b>	<b>:=</b>
<b>-</b>	<b>OR</b>	<b>&lt;=</b>	<b>false</b>	<b>!</b>
<b>*</b>		<b>&gt;</b>		
<b>/</b>		<b>&gt;=</b>		
		<b>==</b>		
		<b>!=</b>		

```
// Example rules.
```

```
a := 2
```

```
d := 2*a
```

```
x := d / 2
```

```
xxx := x == 2
```

```
!b := a == 1 AND c
```

```
!q := b == false
```

Слика 2 - Пример правила (`b` и `q` су излазна правила)

Касније у евалуацији вредности израза, све вредности третирају као реалне вредности, где у случају логичких вредности 0 представља логичку неистину, док све остале представљају логичку истину. Такође, при евалуацији релационих и логичких израза, резултат је 0 или 1, уколико се ради о логичкој неистини, односно логичкој истини, респективно. Ова особина система омогућује

реализовање једноставних условних израза. Наиме, моделовање Хевисајдове јединичне степ функције могуће је извршили следећом спецификацијом правила.

```
// Heaviside unit step function.
```

```
ls := x < 0
```

```
gr := x > 0
```

```
eq := x == 0
```

```
!y := ls*0 + eq*0.5 + gr*1
```

*Слика 3 - Реализација једноставног условног израза - Хевисајдова функција*

Након учитавања и парсирања спецификације, у систему ће постојати стабло спецификације (које ће касније бити коришћено при евалуацији стања пацијента), табела свих симбола у спецификацији са својим вредностима, као и списак свих излазних симбола.

## Евалуатор израза

Евалуатор израза користи табелу симбола и стабло спецификације, који представљају резултат парсирања. Од евалуатора могу да се затраже сви улазни параметри који су потребни да би се извршила евалуација одређеног стања. У том случају се полази од правила које дефинише то стање и сва правила које се налазе са десне стране оператора доделе се сматрају зависним стањима, чија евалуација мора да се изврши пре него што се одреди вредност траженог стања. Даље се итеративно пролази кроз сва остала правила и траже се дефиниције зависних стања и оне се поново обрађују на описан начин. На крају се добије скуп свих стања и параметара од којих зависи тражено стање. Да би се одредили само улазни параметри који су потребни да би се израчунале сва зависна стања, од претходно поменутог скупа се направи разлика у односу на скуп свих стања која се налазе са леве стране оператора доделе (стања која се рачунају у систему) и тиме се добије скуп свих улазних параметара. Од евалуатора израза такође може да се затражи списак свих излазних стања, где он једноставно врати списак излазних симбола који је добио од парсера.

На крају, од евалуатора израза може да се затражи да изврши евалуацију на основу вредности улазних параметара. Овај процес почиње тако што евалуатор прво направи копију табеле симбола. Након тога се вредности параметара унесу у табелу симбола. Затим се покрене евалуација стабла симбола његовим обиласком са дна ка врху. Због овог начина обиласка и начина писања спецификације правила, прво се обилазе поједини делови израза, у редоследу израчунавања израза, па тек онда сама додела вредности. Зато се практично стабло спецификације обилази у истом поретку као и при израчунавању израза у постфиксној нотацији, па се ова особина користи у процесу израчунавања израза. Направи се стек вредности на који се убацују вредности симбола како се они обилазе у стаблу. Са стека се узимају вредности када се наиђе на оператор, израчуна се нова вредност и она се поново убади на стек. На крају, када се коначно дође до оператора доделе, на стеку остаје само једна вредност, која представља вредност стања и њом се онда освежи привремена табела симбола. Уколико се примети да нема довољно података да се дати израз израчуна (улаз у табели симбола нема додељену вредност), онда се прескаче то правило. На крају

привремена табела симбола садржи све вредности правила које су могле да се закључе из достављених података.

## REST интерфејс

REST интерфејс се састоји од три наредбе, које одговарају операцијама евалуатора изрази, и то су:

- <http://localhost:8080/states> - листање стања (GET)
- <http://localhost:8080/dependencies?target=<targetName>> - листинг улазних параметара који су потребни да би се одредило стање „targetName“. Може да се стави и „all“ у којем случају се враћају сви улазни параметри. (GET)
- <http://localhost:8080/evaluate> - израчунавање вредности стања на основу параметара који су прослеђени. (POST)

## Пример рада система

Табела 2 - Пример интеракције са REST сервисом - листинг стања

URL	<a href="http://localhost:8080/states">http://localhost:8080/states</a>
Header	/
Response	<pre>{   "States": [     "autism",     "broken or cracked ribs",     "heart attack",     "kidney failure",     "kidney stone",     "flue",     "cognitive deterioration",     "possible alzheimers"   ],   "Description": "List of all the states." }</pre>

Табела 3 - Пример интеракције са REST сервисом - листа улазних података потребних за евалуацију стања "autism"

URL	<a href="http://localhost:8080/dependencies?target=autism">http://localhost:8080/dependencies?target=autism</a>
Header	/
Response	<pre>{   "Target": "autism",   "Dependencies": [     "Abnormal Body Posturing",     "Abnormal Facial Expressions",     "Avoidance of Eye Contact",     "Poor Eye Contact",     "Delay in Learning to Speak",     "Intense Focus on One Topic",     "Lack of Empathy",     "Lack of Understanding Social Cues",     "Repetitive Movements",     "Social Withdrawal"   ],   "Description": "List of all the dependencies for the 'autism' state." }</pre>

Табела 4 - Пример интеракције са REST сервисом - евалуација стања "autism" када нису достављени сви подаци

URL	<a href="http://localhost:8080/evaluate">http://localhost:8080/evaluate</a>
Header	<pre>{   "Target": "autism",   "Data": {     "Abnormal Body Posturing": 1.0,     "Abnormal Facial Expressions": 0.0,     "Avoidance of Eye Contact": 1.0,     "Poor Eye Contact": 1.0,     "Delay in Learning to Speak": 0.0,     "Intense Focus on One Topic": 0.0,     "Lack of Empathy": 0.0,     "Lack of Understanding Social Cues": 1.0,     "Repetitive Movements": 1.0   } }</pre>
Response	<pre>{   "Requested": "autism",   "Results": [],   "Missing data": [     {       "Target": "autism",       "Dependencies": [         "Social Withdrawal"       ],       "Description": "The data points that were not provided in the request, but are needed in the evaluation of autism"     }   ],   "Description": "The evaluation results for the 'autism' state." }</pre>

Табела 5 - Пример интеракције са REST сервисом - евалуација стања "autism" када су достављени сви подаци

URL	<a href="http://localhost:8080/evaluate">http://localhost:8080/evaluate</a>
Header	<pre>{   "Target": "autism",   "Data": {     "Abnormal Body Posturing": 1.0,     "Abnormal Facial Expressions": 0.0,     "Avoidance of Eye Contact": 1.0,     "Poor Eye Contact": 1.0,     "Delay in Learning to Speak": 0.0,     "Intense Focus on One Topic": 0.0,     "Lack of Empathy": 0.0,     "Lack of Understanding Social Cues": 1.0,     "Repetitive Movements": 1.0,     "Social Withdrawal": 1.0   } }</pre>
Response	<pre>{   "Requested": "autism",   "Results": [     {       "Target": "autism",       "Result": 1.0,       "Description": "Result of the evaluation of the 'autism' state."     }   ],   "Missing data": [],   "Description": "The evaluation results for the 'autism' state." }</pre>

Табела 6 - Пример интеракције са REST сервисом - евалуација свих стања када су достављени неки подаци

URL	<a href="http://localhost:8080/evaluate">http://localhost:8080/evaluate</a>
Header	<pre>{   "Target": "all",   "Data": {     "Abnormal Body Posturing": 1.0,     "Abnormal Facial Expressions": 0.0,     "Avoidance of Eye Contact": 1.0,     "Poor Eye Contact": 1.0,     "Delay in Learning to Speak": 0.0,     "Intense Focus on One Topic": 0.0,     "Lack of Empathy": 0.0,     "Lack of Understanding Social Cues": 1.0,     "Repetitive Movements": 1.0,     "Social Withdrawal": 1.0,     "back pain": 1.0,     "burning urination": 1.0,     "blood in urine": 1.0,     "Fever": 0.0   } }</pre>



## Response

```
{
  "Requested": "all",
  "Results": [
    {
      "Target": "autism",
      "Result": 1.0,
      "Description": "Result of the evaluation of the 'autism' state."
    },
    {
      "Target": "kidney_stone",
      "Result": 1.0,
      "Description": "Result of the evaluation of the 'kidney_stone' state."
    }
  ],
  "Missing data": [
    {
      "Target": "broken_or_cracked_ribs",
      "Dependencies": [
        "abdominal injury",
        "chest pain"
      ],
      "Description": "The data points that were not provided in the request, but are needed in the evaluation of broken_or_cracked_ribs"
    },
    {
      "Target": "heart_attack",
      "Dependencies": [
        "Shortness of breath",
        "Cold sweat",
        "Fatigue",
        "chest pain",
        "shoulder pain"
      ],
      "Description": "The data points that were not provided in the request, but are needed in the evaluation of heart_attack"
    },
    {
      "Target": "kidney_failure",
      "Dependencies": [
        "reuced amount of urine",
        "swelling of legs",
        "swelling of feet",
        "Farigue",
        "chest pain"
      ],
      "Description": "The data points that were not provided in the request, but are needed in the evaluation of kidney_failure"
    }
  ],
  "Description": "Evaluation results for all possible states (based on the provided data).",
}
```

## Закључак

Реализовани систем одговара спецификацијама пројекта и могао би да се користи у процесу дијагностиковања пацијента и његовог лечења. Осим ове намене, систем би могао да се користи и у другим ситуацијама када је потребно извршити прераду појединачних група података на основу унапред дефинисаног сета правила. Тривијалан пример једног таквог израчунавања (израчунавања Хевисајдове функције) дат је у овом извештају.

Оно шта недостаје оваквом систему јесте могућност накнадног убацивања спецификација правила. Наиме, верзија система која је описана у овом извештају читава спецификације правила при покретању и не постоји наредба у REST интерфејсу за додавање нових спецификација. Оваква наредба значајно би побољшала квалитет сервиса, пошто би се омогућило корисницима да сами специфицирају своја правила. Још једна мана јесте *statelessness* система. Систем сада не чува никакве податке које му корисник пошаље, нити разликује кориснике. Ово се јасно види при евалуацији израза кроз REST интерфејс – ако неки подаци недостају, није довољно да их корисник пошаље у следећем захтеву, већ је потребно да поново пошаље и све претходне податке. Ово није битна мада, али би препознавање корисника уз неки вид аутентикације пружио бољи осећај током интеракције са системом.

Додатне идеје за развој набројане су у наставку.

- Даље ширење граматике језика спецификације додатним операторима.
- Ширење граматике језика спецификације уграђеним функцијама, као што су `max`, `min`, `sum`, итд.
- Могућност обрада података који се не односе само на једног пацијента. Тренутно за овакву неку обраду би било потребно послати посебан захтев за евалуацију за сваког пацијента, али би било погодније уколико би могао да се пошаље скуп података за неколико пацијената (рецимо као једна `*.csv` датотека) и да се као одговор добију дијагнозе свих пацијената.