MIF14-BDD: Project
Implementing a Top-down Query Evaluation Engine for Datalog

PÉREZ MARTÍNEZ Eduardo
DERBAL Younes

Juin 10, 2021

Índice

1. Introduction

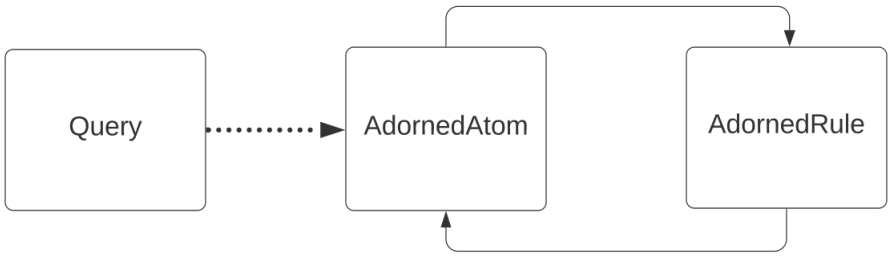
This document is a complement to our poor implementation code of the project *'Implementing a Top-down Query Evaluation Engine for Datalog'* that allows us to clarify some aspects of the practice.
The practice is composed of two parts: implementing the engine and analyzing its performance.

Due to the fact that we were not able to finish the project in the established time, our version of the practice only contains a plain introduction of the development of the engine. Because of this, concerning the second part, we only could develop the sample inputs set and check them with *DES*.

2. Implementing the engine

First of all, before start implementing the engine, we needed to understand how it have to work. After revising a couple of times the explanation of the practice, we had a superficial idea of it works.

As the name of the algorithm says, we have to work with queries to implement its recursion and calculating the information and passing it to the other rules. To get this information we have to develop the class *AdornedAtom* that allows us to add information to an atom and then combining them into an *AdorndedRule*.



The queries will communicate with them in order to calculate the result and transmitting into the others.

3. Analyzing the performance

In order to check the functioning of an hypothetical ended version of the engine we designed a small database with some queries.
The database is inspired in the popular video game League of Legends, specifically in its competitive side, where we take three of the European teams and its roosters to generate some date to play with.

We represented the players with their in game identifiers and put some extra information as their role/-position in the game and their country of origin, the teams with their names and the region, and we added a relation *play* to represent in a specific year in what team a player competed.

We developed this database and checked the queries with *DES* and then translated into the syntax of the engine that unfortunately we couldn't test.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                         League of Legends Database
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                FACTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  player(pID, role, nationality)                      key: pID
%%  team(tID, region)                                      key: tID
%%  play(pID, tID, year)                                   key: pID, tID
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

:-type(player(pID:string,role: string,nationality:string)).

player('Caps','mid','Denmark').
player('Mikyx','support','Slovenia').
player('Jankos','jungle','Poland').
player('Wunder','top','Denmark').
player('Rekkles','adc','Sweden').
player('P1noy', 'adc', 'Denmark').
player('Perkz','adc','Croatia').
player('Upset','adc','Germany').
player('Bwipo','jungle','Belgium').
player('Nisqy','mid','Belgium').
player('Hylissang','support','Bulgaria').
player('Adam','top','France').
player('Carzzy','adc','Czech').
player('Humanoid','mid','Czech').
player('Armut','top','Turkey').
player('Elyoya','jungle','Spain').
player('Kaiser','support','Germany').

:-type(team(tID:string,region: string)).

team('G2Esports','EUW').
team('Fnatic','EUW').
team('MADLions','EUW').

:-type(play(pID:string,tID:string,year: int)).

play('Caps','G2Esports',2021).
play('Mikyx','G2Esports',2021).
play('Jankos','G2Esports',2021).
play('Wunder','G2Esports',2021).
play('Rekkles','G2Esports',2021).
play('P1noy','G2Esports',2021).
play('Caps','G2Esports',2020).
play('Mikyx','G2Esports',2020).
play('Jankos','G2Esports',2020).
play('Wunder','G2Esports',2020).
play('Perkz','G2Esports',2020).
play('P1noy','G2Esports',2020).
play('Rekkles','Fnatic',2020).
play('Upset','Fnatic',2021).
play('Bwipo','Fnatic',2021).
play('Nisqy','Fnatic',2021).
play('Hylissang','Fnatic',2021).
play('Adam','Fnatic',2021).
play('Carzzy','MADLions',2021).
play('Humanoid','MADLions',2021).
play('Armut','MADLions',2021).
play('Elyoya','MADLions',2021).
```

```
play('Kaiser','MADLions',2021).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                  RULES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 1. All the names of the players with 'jungle' role.
%% player(N,'jungle',_) -> jungleplayer(N)
queryone(N) :- player(N,'jungle',_).

% 2. All the names of the players of the MAD Lions team.
%% player(N,_,_) AND play(N,'MADLions',_) -> madlplayer(N)
querytwo(N) :- player(N,_,_), play(N,'MADLions',_).

% 3. All the names of the players of G2 Esports in the past year.
%% player(N,_,_) AND play(N,'G2Esports',2020) -> gtwopastteam(N)
querythree(N) :- player(N,_,_), play(N,'G2Esports',2020).

% 4. All the names of the players who played 'adc' for G2 Esports.
%% player(N,'adc',_) AND play(N,'G2Esports',_) -> adcsgtwo(N)
queryfour(N) :- player(N,'adc',_), play(N,'G2Esports',_).

% 5. All the names of the players who played in more than one team.
%% player(N,_,_) AND play(N,T,_) AND play(N,L,_) AND T!=L -> manyplayedteams(N)
queryfive(N) :- player(N,_,_), play(N,T,_), play(N,L,_),T\=L.

% 6. All the nationalities of the Fnatic players.
%% player(P,_,N) AND play(P,'Fnatic',_) -> fncnationallity(N)
querysix(N) :- player(P,_,N), play(P,'Fnatic',_).

% 7. All the names of the teams with french players.
%% play(P,T,_) AND player(P,_,'France') -> teamswithfrench(T)
queryseven(T) :- play(P,T,_), player(P,_,'France').

% 8. All the nationalities that are repeated.
%% player(A,_,N) AND player(B,_,N) AND A!=B -> nationrepeated(N)
queryeight(N) :- player(A,_,N), player(B,_,N), A\=B.

% 9. All the teams without german players
%% player(P,_,'Germany') AND play(P,T,_) -> teamofgermanplayer(T)
%% team(T,_) AND NOT teamofgermanplayer(T) -> teamnogerman(T)
teamofgermanplayer(T) :- player(P,_,'Germany'), play(P,T,_).
querynine(T) :- team(T,_), not teamofgermanplayer(T).

% 10. All the partners of the german players
%% player(P,_,'Germany') -> germanplayer(P)
%% player(P,_,_) AND germanplayer(P) AND player(Q,_,_) AND NOT germanplayer(Q) AND P!=Q AND play(P,T,Y) AND play(Q,T,Y) ->germanp
germanplayer(P) :- player(P,_,'Germany').
queryten(Q) :- player(P,_,_), germanplayer(P), player(Q,_,_), not germanplayer(Q), P\=Q, play(P,T,Y), play(Q,T,Y).
}
```