Proposal for a sports calendar calculated using CLINGO

Department of Computer Science: University of Turin

Autori: Alessandro Saracco, Leonardo Magliolo , Mattia Marra

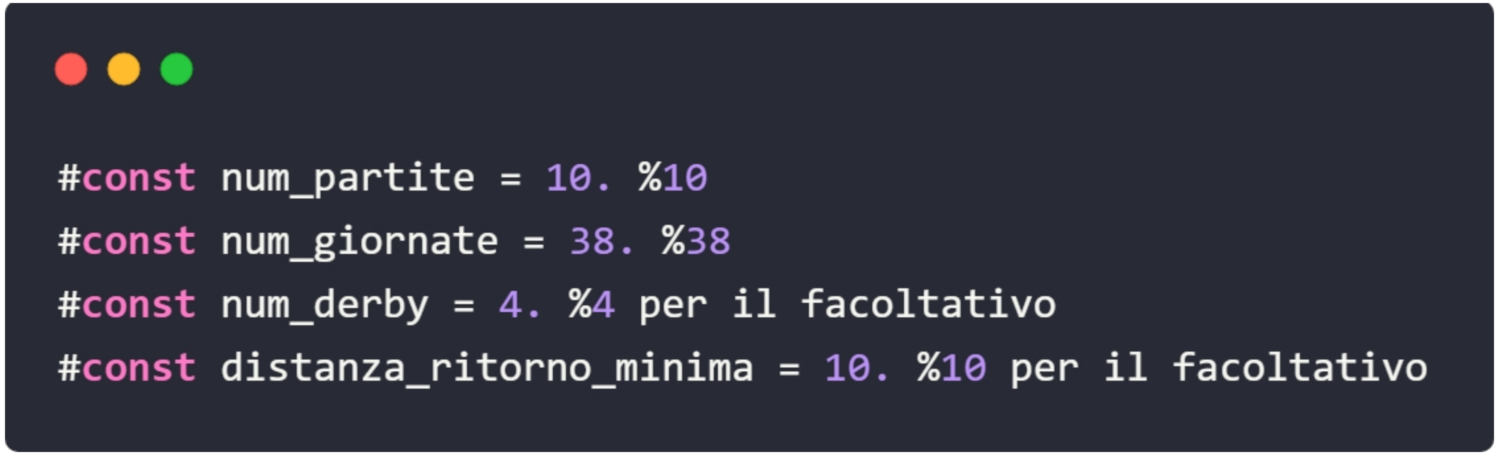
Authors: Alessandro Saracco, Leonardo Magliolo , Mattia Marra

Artificial intelligence and laboratory

A.Y. 2021/2022

Parameterization

To facilitate debugging of the program, constants have been added to make the treated problem parameterizable.



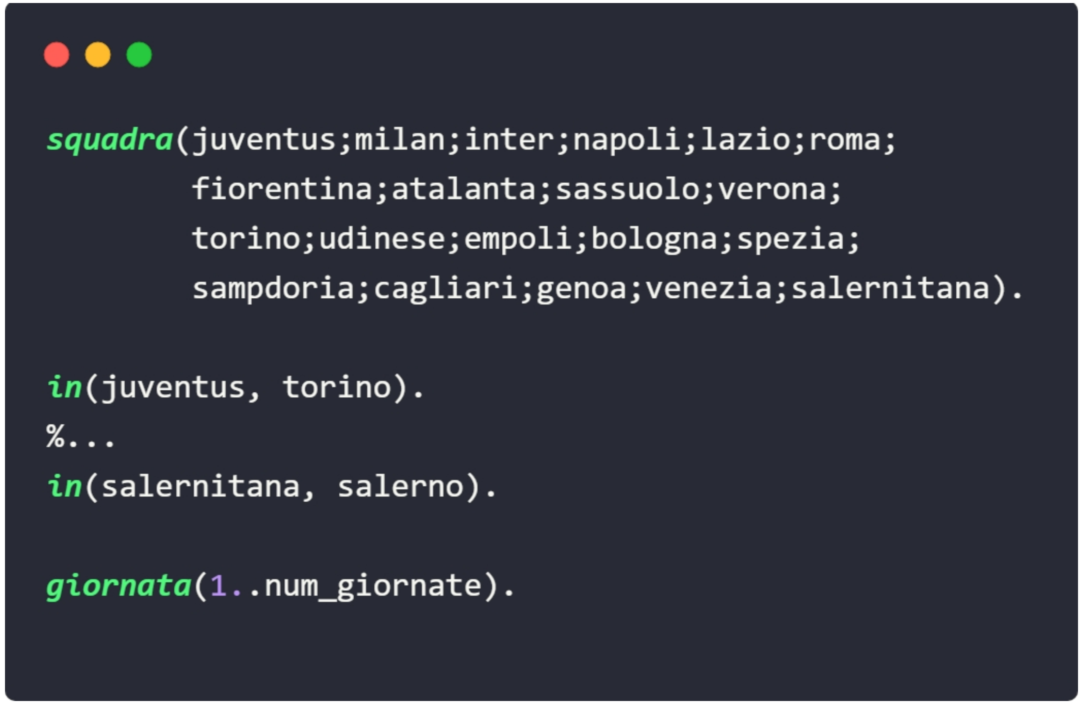
```
#const num_partite = 10. %10
#const num_giornate = 38. %38
#const num_derby = 4. %4 per il facoltativo
#const distanza_ritorno_minima = 10. %10 per il facoltativo
```

Representation of the championship

It was decided to use, for the generation of the sports calendar, the Italian soccer championship of the year 2021/2022.

Aggregates were used to define teams and days to make the descriptive syntax more compact.

The home city for each team was also indicated by means of Of the binary predicate **in(team, city)**.



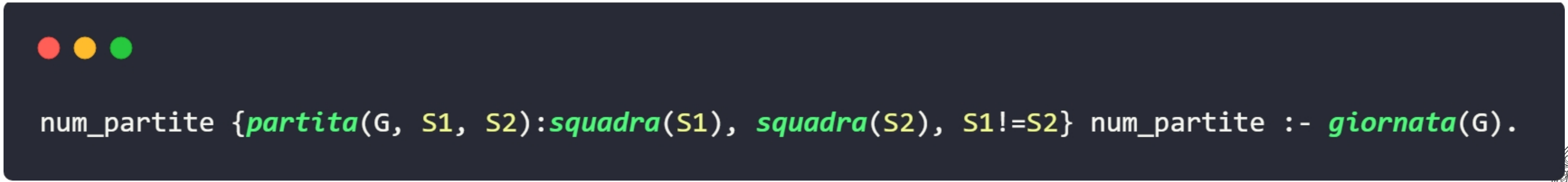
```
squadra(juventus;milan;inter;napoli;lazio;roma;
        fiorentina;atalanta;sassuolo;verona;
        torino;udinese;empoli;bologna;spezia;
        sampdoria;cagliari;genoa;venezia;salernitana).

in(juventus, torino).
%...
in(salernitana, salerno).

giornata(1..num_giornate).
```

(constraint 1)

Representation of matches

The concept of match was modeled by means of the ternary predicate **match(Day, Team1, Team2)**.
By convention, it was chosen to indicate the team playing at home within the second parameter (**Team1**).

An aggregate was used to generate the matches, so that it creates exactly ***num_parties*** for each day of the championship.
The condition was also met such that there cannot be a match having a team facing itself.



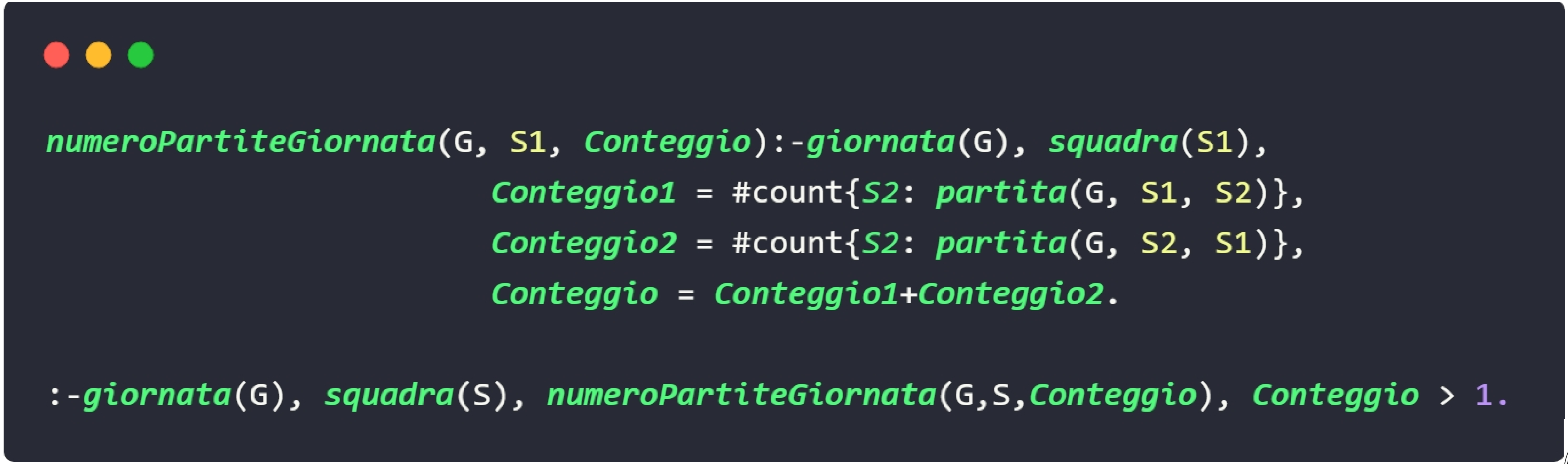
```
num_partite {partita(G, S1, S2):squadra(S1), squadra(S2), S1!=S2} num_partite :- giornata(G).
```

(constraint 2)

At most one game played on the same day

The predicate **numberGamesDay(Day, Team, Count)** was introduced to calculate how many games a given team played in a day.

This predicate was used to implement the constraint that prevents a team from playing multiple times on the same day.

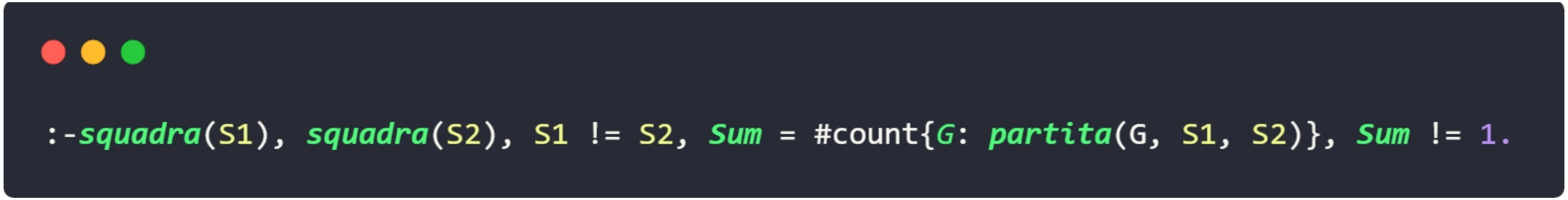


```
numeroPartiteGiornata(G, S1, Conteggio):-giornata(G), squadra(S1),
                                Conteggio1 = #count{S2: partita(G, S1, S2)},
                                Conteggio2 = #count{S2: partita(G, S2, S1)},
                                Conteggio = Conteggio1+Conteggio2.

:-giornata(G), squadra(S), numeroPartiteGiornata(G,S,Conteggio), Conteggio > 1.
```

(constraint 3)

Back-to-back games

A constraint was introduced so that each team faces all other teams twice, once at home and once
time away from home, that is, once in one's hometown and once in the
Of the other team.



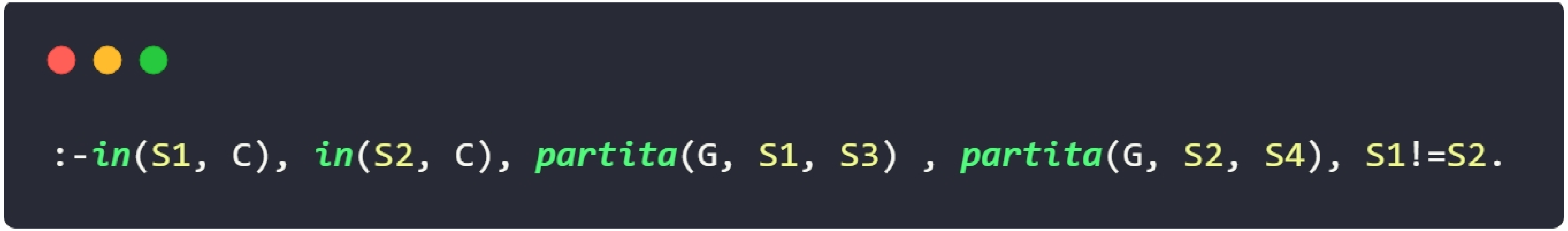
```
:-squadra(S1), squadra(S2), S1 != S2, Sum = #count{G: partita(G, S1, S2)}, Sum != 1.
```

Given the generic pair of **teams (Team1, Team2)**, it is possible to instantiate **S1**
as **Team1 and S2** as **Team2** and vice versa,
This imposes the existence of only one game played home and away for each pair of teams (taking advantage of the match declaration convention described above).

(constraint 4)

most one home game per day

A constraint was introduced so that two teams from the same city share the same playing facility, so they cannot both play at home on the same day.

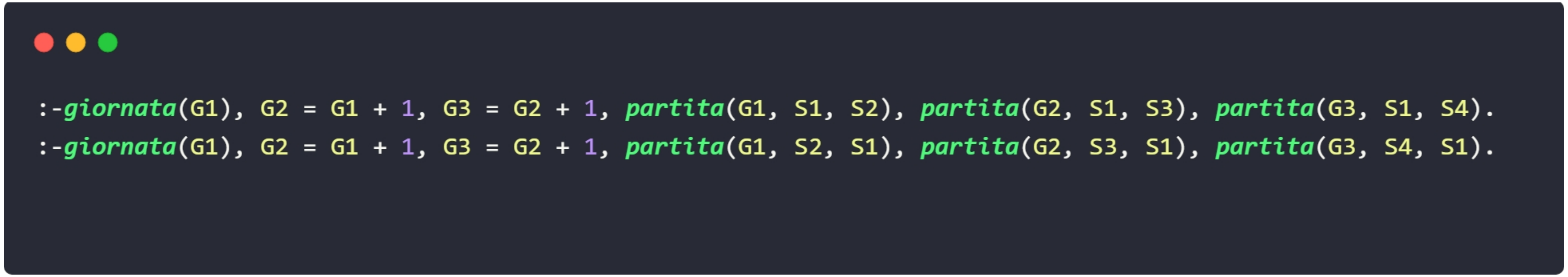


```
:-in(S1, C), in(S2, C), partita(G, S1, S3) , partita(G, S2, S4), S1!=S2.
```

(constraint 5)

No more than two consecutive home games

A constraint was introduced so that each team never plays more than two consecutive home or away games.



```
:-giornata(G1), G2 = G1 + 1, G3 = G2 + 1, partita(G1, S1, S2), partita(G2, S1, S3), partita(G3, S1, S4).
:-giornata(G1), G2 = G1 + 1, G3 = G2 + 1, partita(G1, S2, S1), partita(G2, S3, S1), partita(G3, S4, S1).
```

It was decided to divide the constraint into two independent constraints, one for home games and one for away games.

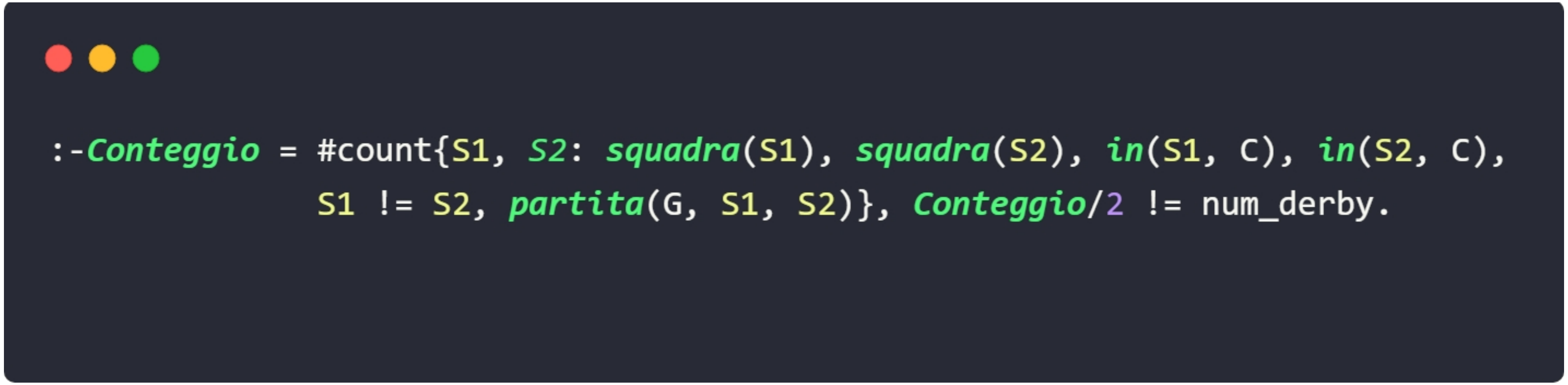
The constraints exploit the ownership of matches such that the home team is declared first and then the away team,

therefore in the first constraint the team fixed **S1** is indicated inside the second parameter Match, and in the second constraint **S1** is indicated in the third parameter Match.

(constraint 6)

Number of derbies

A constraint was introduced to introduce the presence of ***num_derby*** derbies, that is, pairs of teams referring to the same city.



```
:-Conteggio = #count{S1, S2: squadra(S1), squadra(S2), in(S1, C), in(S2, C),
              S1 != S2, partita(G, S1, S2)}, Conteggio/2 != num_derby.
```

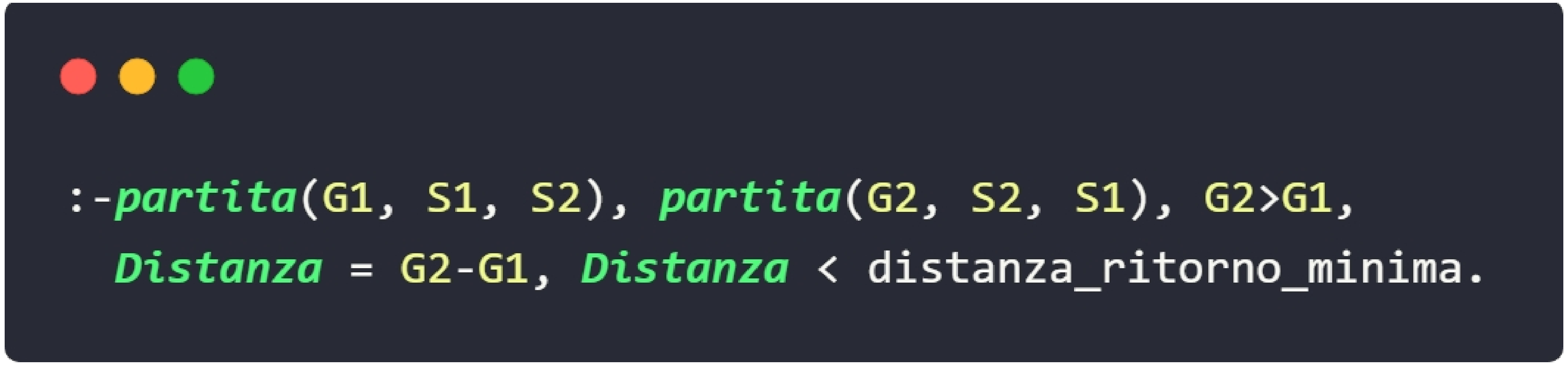
If **Team1** and **Team2** belong to the same city then **match(Day, Team1, Team2)** and **match(Day, Team2, Team1)** represent the same derby; therefore, the match count within the constraint was divided by two avoid counting the same derby twice.

(constraint 7)

Minimum round trip distance

A constraint was introduced so that the distance between a pair round-trip races is at least ***distance_return_minimum*** days.

Assuming the minimum distance is 10, then if **TeamA** vs **TeamB** is scheduled for Day 12, the return **TeamB** vs **TeamA** will be scheduled no earlier than Day 22.



```
:-partita(G1, S1, S2), partita(G2, S2, S1), G2>G1,
  Distanza = G2-G1, Distanza < distanza_ritorno_minima.
```

(constraint 8)

Performance Analysis

Hardware specifications:

- CPU: AMD Ryzen 7 5800X
- RAM: Corsair VENGEANCELPX16GB DDR4
- SSD: Samsung MZ-V8V1T0 980

All performance tests were performed using the parameters
Of execution:

- --time-limit=0, to set an arbitrary computation time limit
- --parallel-mode=16, to use all 16 threads whose architecture features

Performance Analysis (2)

Although the computation time with respect to the program (formulated in relation to the first delivery with the optional points) exceeds eight hours before finding a solution, the constraints indicated in the slides as 4, 6, 8 were selectively omitted in order to derive suboptimal solutions in less time:

- Individually, the {4}, {6}, {8} constraint sets were executed within timings of less than 20 seconds.
- Even by modifying the parameters given in delivery by a certain amount, a solution could not be obtained within acceptable time frames for the {4,6} constraint set.
- Keeping the program parameters unchanged as delivered, a solution was found for the {6,8} constraint set with timings of less than 40 seconds.
- Reducing the minimum distance between first and return games from 10 to 4 was found to the {4,8} constraint set with timings of less than 2 minutes.

Ultimately, the constraint that proved most problematic in order to calculate an acceptable timetable delivery was 4.