



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Campus São Carlos

DEPARTAMENTO DE COMPUTAÇÃO

Relatório do Segundo Trabalho

Inteligência Artificial

Prof. Dra. Heloísa Camargo de Arruda

Frankson Teotonho de Sousa - 619540

Pedro Henrique Dumont Mauad - 619736

São Carlos - SP

2017

Etapas 1 - Sentenças Lógicas

Uma base de conhecimento foi construída, usando sentenças da lógica de primeira ordem, para identificar problemas com um time de futebol.

As sentenças são:

- 1) Ruim (Preparo_fisico) Problema(Preparador_fisico)
- 2) Constantes (Atritos) Ruim (Situação_psicológica) Problema(Equipe_tecnica)
- 3) Bom (Preparo_fisico) Ruim (Situação_de_gols) Problema(time)
- 4) $\forall x$ Jogador(x) Discute_tecnico(x) Constantes (Atritos)
- 5) $\forall x$ jogador(x) Suspenso(x) Ruim (Situação_psicologica)
- 6) $\forall x,y$ gols_sofridos(x) gols_feitos(y) maior(x,y) Ruim (Situação_de_gols)
- 7) $\forall x$ cartão_vermelho(x) Suspenso(x)

Uma possível situação específica foi definida com as seguintes sentenças:

- 8) Bom (Preparo_fisico)
- 9) Jogador(David)
- 10) Cartão_vermelho(David)
- 11) $\exists x$ (Jogador(x) Discute_tecnico(x))
- 12) Gols_sofridos(1)
- 13) Gols_feitos(2)

A transformação das sentenças da lógica para o formato de cláusula definida de primeira ordem, eliminando os quantificadores universal e existencial foi realizada, conforme instruções do trabalho:

- 1) Ruim(Preparo_fisico) -> Problema(Preparador_fisico)
- 2) Constantes(Atritos) ^ Ruim(Situacao_psicologica) -> Problema(Equipe_tecnica)
- 3) Bom(Preparo_fisico) ^ Ruim(Situacao_de_gols) -> Problema(Time)
- 4) Jogador(x) ^ Discute_tecnico(x) -> Constantes (Atritos)
- 5) Jogador(x) ^ Suspenso(x) -> Ruim (Situacao_psicologica)
- 6) gols_sofridos(x) ^ gols_feitos(y) ^ maior(x,y) -> Ruim (Situacao_de_gols)
- 7) cartão_vermelho(x) -> Suspenso(x)

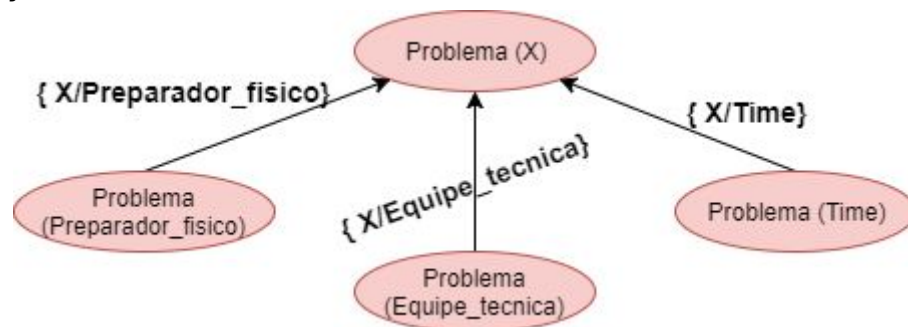
- 8) Bom(Preparo_fisico)
- 9) Jogador(David)
- 10) Cartão_vermelho(David)
- 11.1) Jogador(J1)
- 11.2) Discute_tecnico(J1)
- 12) Gols_sofridos(1)
- 13) Gols_feitos(2)

Etapa 2- Algoritmo de Encadeamento para trás.

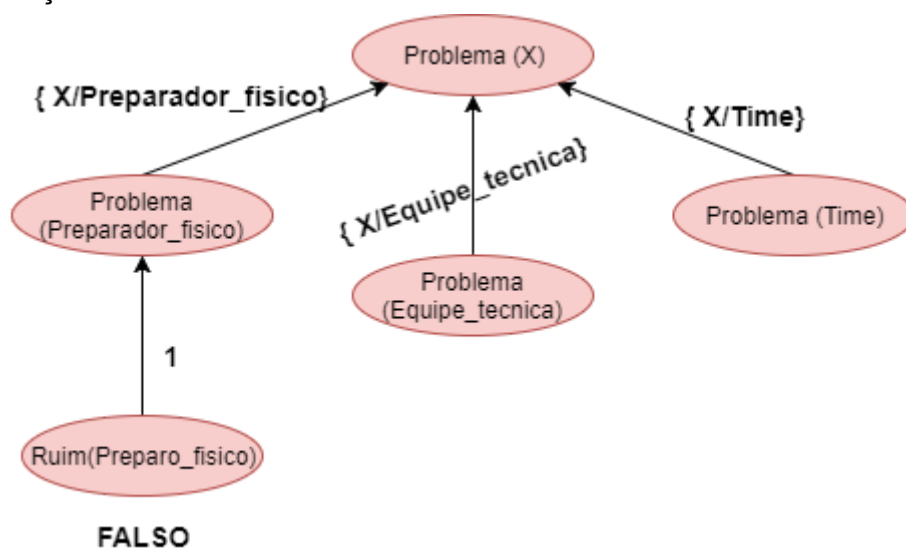
A aplicação do algoritmo de raciocínio com encadeamento para trás com busca em profundidade no conjunto de sentenças foi feita por meio de sucessivas iterações, conforme visto em sala de aula. A consulta (objetivo) deve ser definida por Problema(x) e o processo de raciocínio deve parar quando encontrar a primeira resposta. Devem ser construídos o grafo de execução e a tabela de execução.

Grafo de execução:

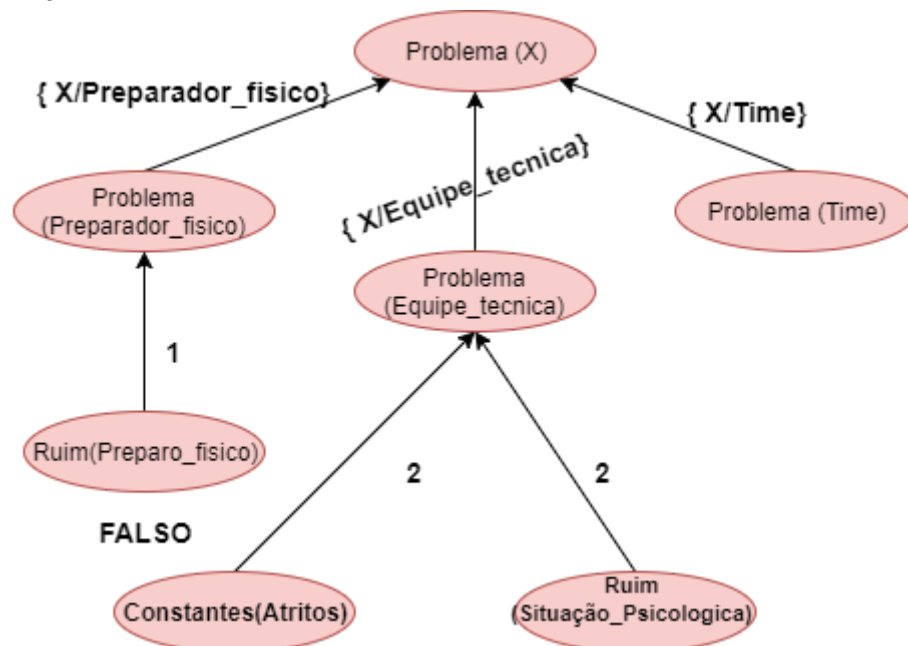
1ª Iteração:



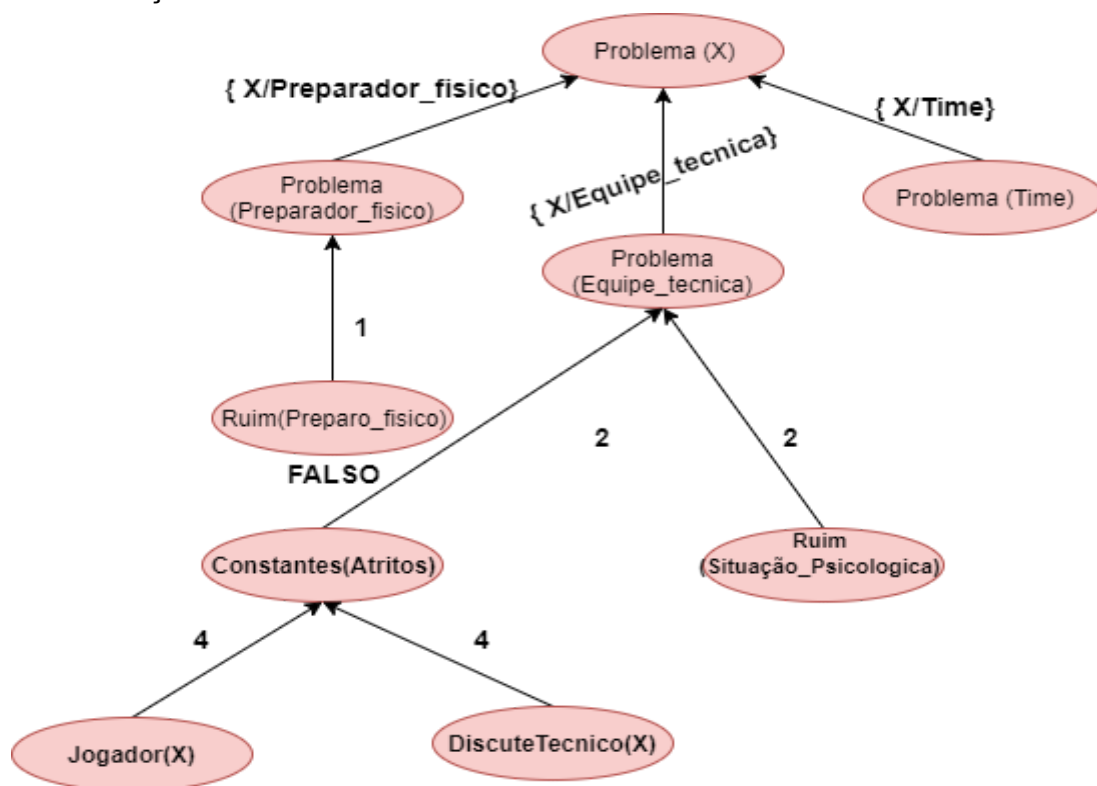
2ª Iteração:



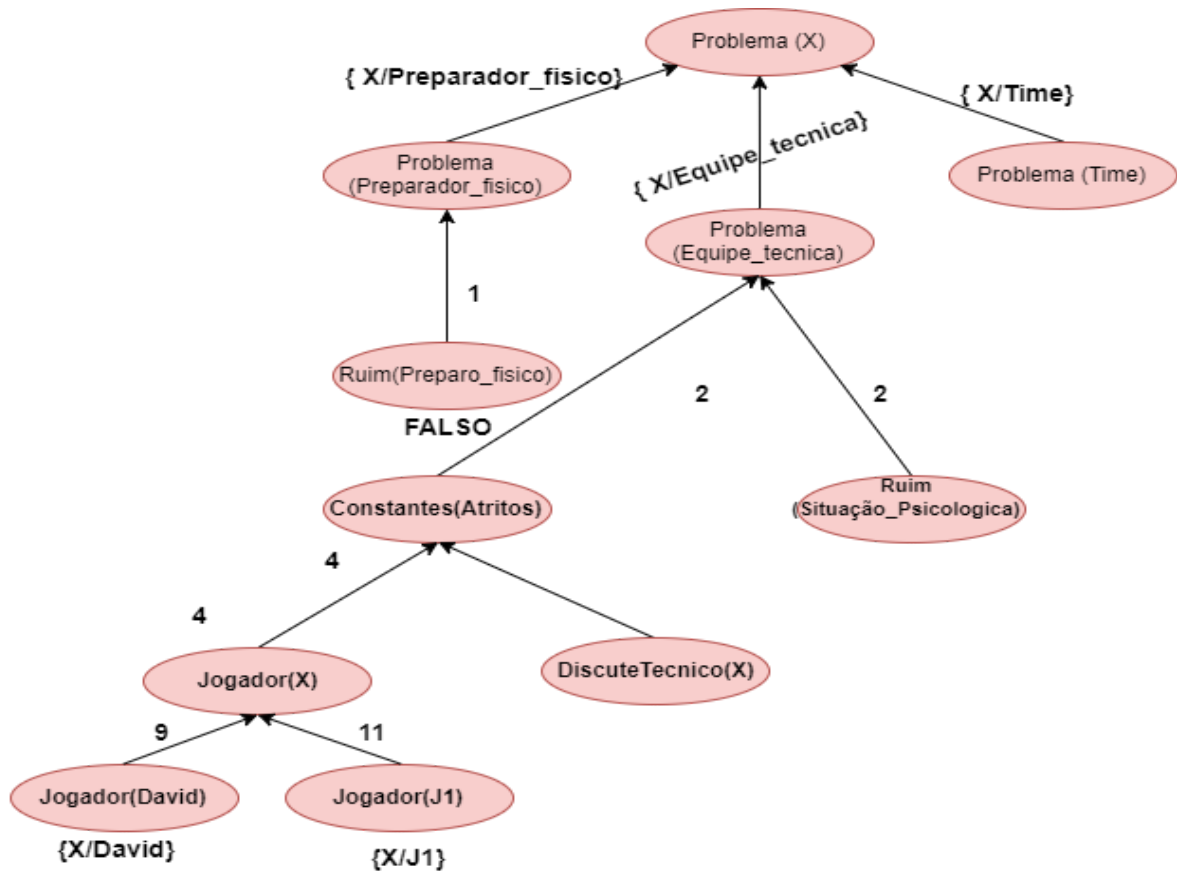
3ª Iteração:



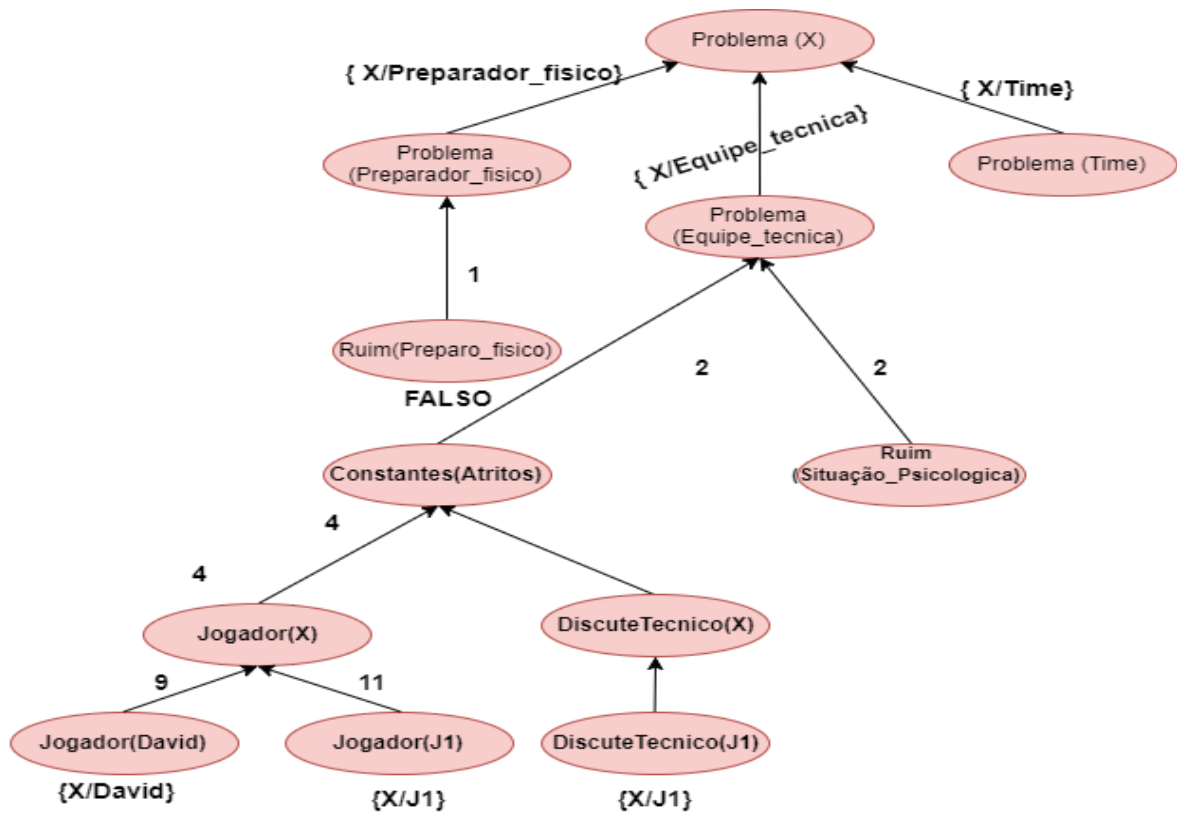
4ª Iteração:



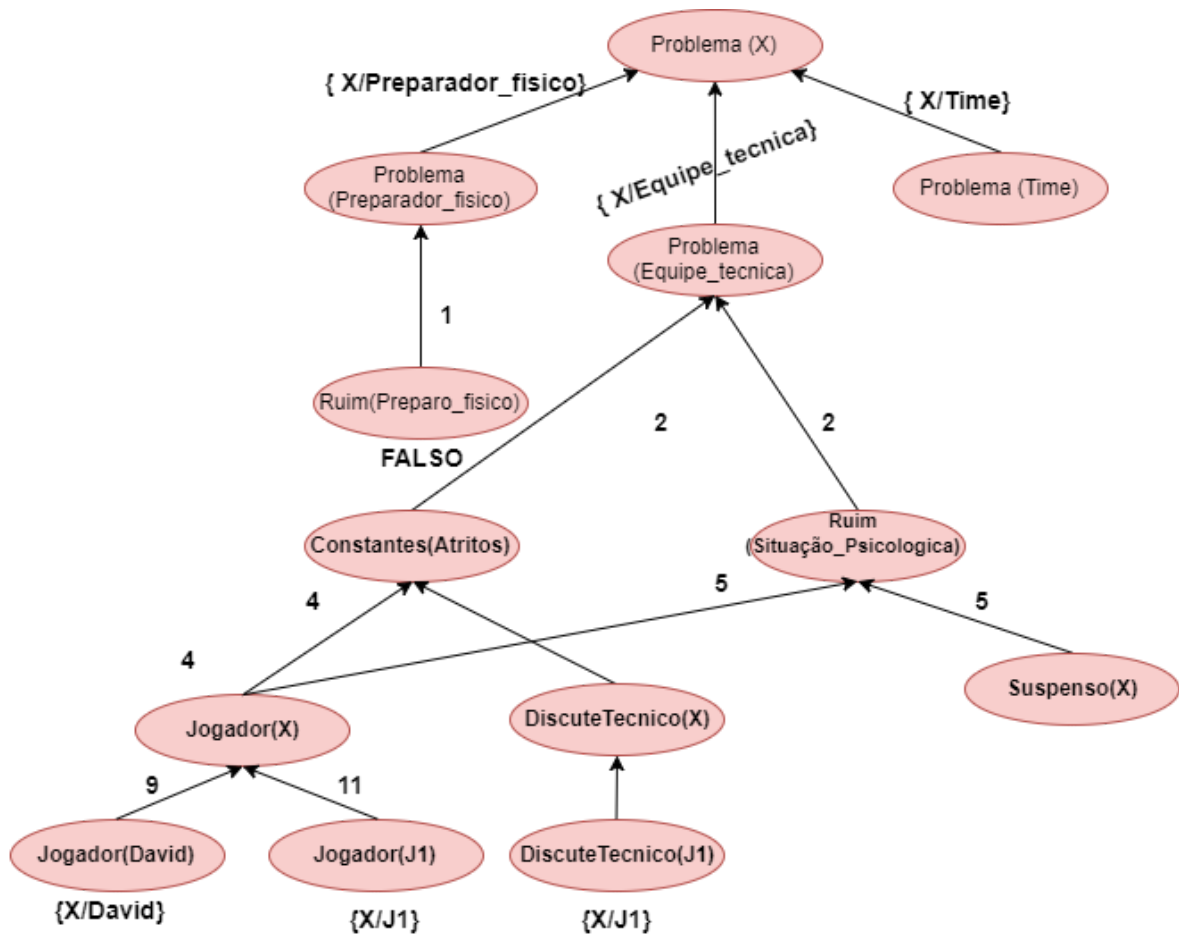
5ª Iteração:



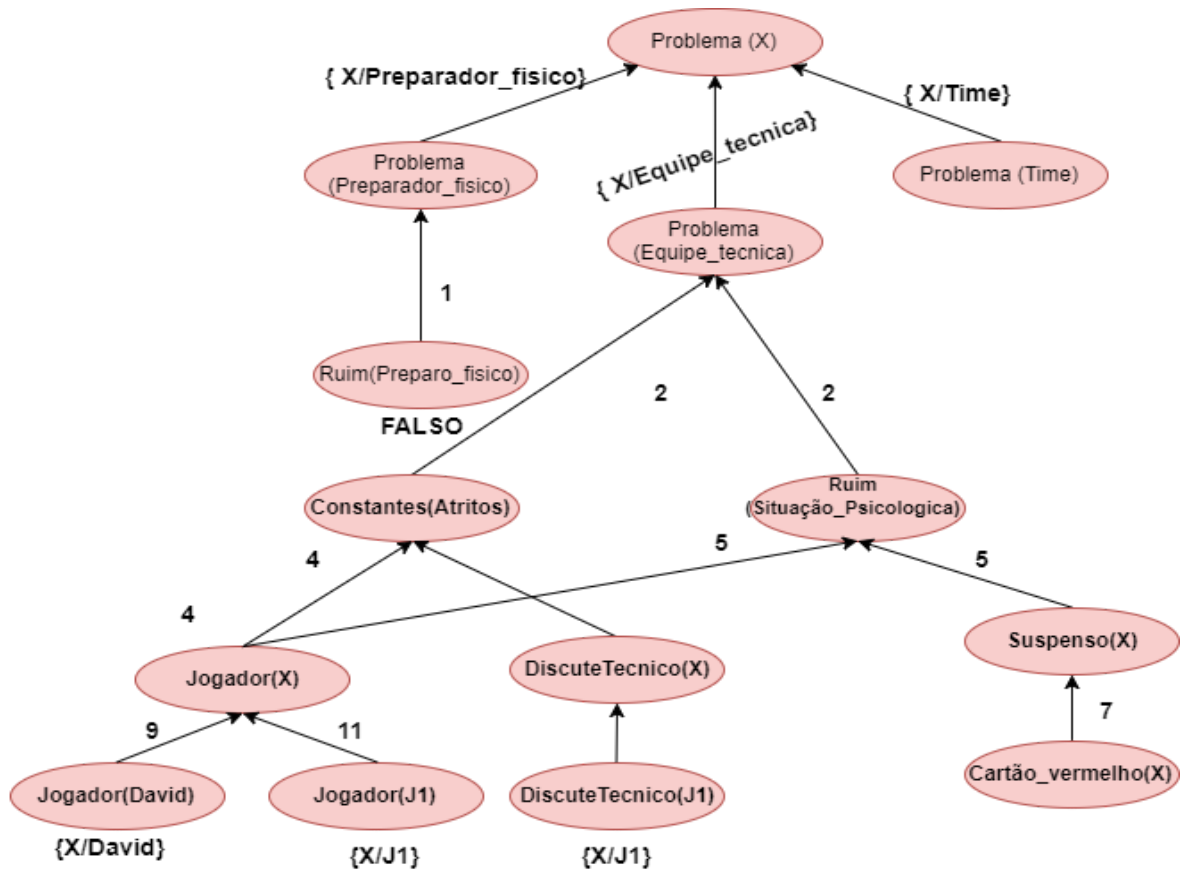
6ª Iteração:



7ª Iteração:



8ª Iteração



9ª Iteração:

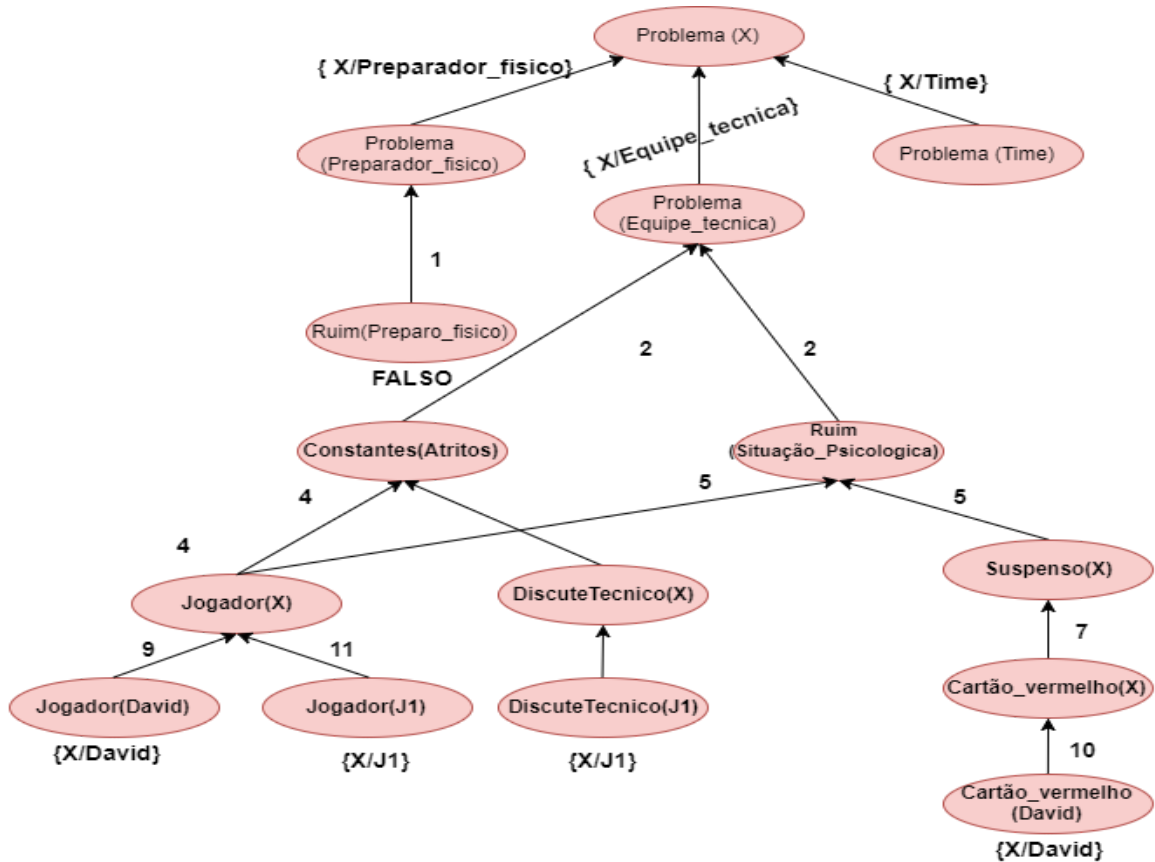


Tabela de Execução:

Tabela 1 - Tabela com os passos do algoritmo

Iteração	Memória de Trabalho	Conjunto de Conflito	Disparo
1	Problema(X)	1- {X/Preparador_fisico} 2- {X/Equipe_tecnica} 3- {X/Time}	1
2	Ruim(Preparador_fisico)	-	Retrocesso
Retrocesso até Iteração 1			
1	Problema(X)	1- {X/Preparador_fisico} 2- {X/Equipe_tecnica} 3- {X/Time}	2
2	Constantes(Atritos) Ruim(Situação_psicologica)	2- {X/Atritos} 2- {X/Situação_psicologica}	4
3	Jogador(X) Discute_tecnico(X) Ruim(Situação_psicologica)	4- {} 9- {X/David}	9
4	Jogador(David) (V) Discute_tecnico(David) (F) Ruim(Situação_psicologica)	-	Retrocesso
Retrocesso até Iteração 3			
3	Jogador(X) Discute_tecnico(X) Ruim(Situação_psicologica)	4- {} 9- {X/J1}	11
4	Jogador(J1) (V) Discute_tecnico(J1) (V) Ruim(Situação_psicologica)	5- {X/Situação_psicologica}	5
5	Jogador(J1) (V) Discute_tecnico(J1) (V) Jogador(X) Suspenso(X)	5- {} 9- {David}	9
6	Jogador(J1) (V) Discute_tecnico(J1) (V) Jogador(David) (V) Suspenso(X)	7- {}	7
7	Jogador(J1) (V) Discute_tecnico(J1) (V) Jogador(David) (V) Cartão_vermelho(X)	10- {X/David}	10

8	Jogador(J1) (V) Discute_tecnico(J1) (V) Jogador(David) (V) Cartão_vermelho(David) (V)	-	Sucesso
---	--	---	---------

Fonte: Autores

Etapa 3 - Implementação Prolog e Testes

A execução do programa Prolog com três situações iniciais diferentes.

A implementação em Prolog da base de conhecimento que contém as sentenças definidas.

Imagem 1 - Resultado da execução do script seguindo os dados convencionais

```
?- problema(X).
X = equipe_tecnica ;
false.
```

Fonte: Autores

Imagem 2 - Script contendo a base de dados convencional e as regras

```
bom(preparo_fisico).
jogador(david).
cartao_vermelho(david).
jogador(j1).
discute_tecnico(j1).
gols_sofridos(1).
gols_feitos(2).

maior(X,Y) :- X>Y.

problema(preparo_fisico) :- ruim(preparo_fisico).
problema(equipe_tecnica) :- constantes(atritos) , ruim(situacao_psicologica).
problema(time) :- bom(preparo_fisico), ruim(situacao_de_gols).
constantes(atritos) :- jogador(X), discute_tecnico(X).
ruim(situacao_psicologica) :- jogador(X), suspenso(X).
ruim(situacao_de_gols) :- gols_sofridos(X), gols_feitos(Y), maior(X,Y).
suspenso(X) :- cartao_vermelho(X).

discontiguous(jogador/1).|
```

Fonte: Autores

A execução do programa Prolog com três situações iniciais diferentes.

CASO 1:

Imagem 3 - Resultado da execução do script seguindo os dados para o caso 1

```
?- problema(X).
X = preparo_fisico ;
X = equipe_tecnica ;
false.
```

Fonte: Autores

Imagem 4 - Script contendo a base de dados do caso 1 e as regras

```
%% Problema é o preparo fisico e a equipe tecnica

ruim(preparo_fisico).
bom(holder).
jogador(ronaldinho).
cartao_vermelho(ronaldinho).
jogador(j1).
discute_tecnico(j1).
gols_sofridos(7).
gols_feitos(1).

maior(X,Y) :- X>Y.

problema(preparo_fisico) :- ruim(preparo_fisico).
problema(equipe_tecnica) :- constantes(atritos) , ruim(situacao_psicologica).
problema(time) :- bom(preparo_fisico), ruim(situacao_de_gols).
constantes(atritos) :- jogador(X), discute_tecnico(X).
ruim(situacao_psicologica) :- jogador(X), suspenso(X).
ruim(situacao_de_gols) :- gols_sofridos(X), gols_feitos(Y), maior(X,Y).
suspenso(X) :- cartao_vermelho(X).

discontiguous(jogador/1).
discontiguous(ruim/1).
```

Fonte: Autores

CASO 2

Imagem 5 - Resultado da execução do script seguindo os dados para o caso 2

```
?- problema(X).
X = equipe_tecnica ;
X = time.
```

Fonte: Autores

Imagem 6 - Script contendo a base de dados do caso 2 e as regras

```
%% Problema é a equipe tecnica e o time

bom(preparo_fisico).
jogador(pele).
cartao_vermelho(pele).
jogador(j1).
discute_tecnico(j1).
gols_sofridos(5).
gols_feitos(3).

maior(X,Y) :- X>Y.

problema(preparo_fisico) :- ruim(preparo_fisico).
problema(equipe_tecnica) :- constantes(atritos) , ruim(situacao_psicologica).
problema(time) :- bom(preparo_fisico), ruim(situacao_de_gols).
constantes(atritos) :- jogador(X), discute_tecnico(X).
ruim(situacao_psicologica) :- jogador(X), suspenso(X).
ruim(situacao_de_gols) :- gols_sofridos(X), gols_feitos(Y), maior(X,Y).
suspenso(X) :- cartao_vermelho(X).

discontiguous(jogador/1).
```

Fonte: Autores

CASO 3

Imagem 7 - Resultado da execução do script seguindo os dados para o caso 3

```
?- problema(X).  
X = time.
```

Fonte: Autores

Imagem 8 - Script contendo a base de dados do caso 2 e as regras

```
%% Problema é somente o time  
  
bom(preparo_fisico).  
jogador(robinho).  
cartao_vermelho(holder).  
jogador(j1).  
discute_tecnico(j1).  
gols_sofridos(8).  
gols_feitos(2).  
  
maior(X,Y) :- X>Y.  
  
problema(preparo_fisico) :- ruim(preparo_fisico).  
problema(equipe_tecnica) :- constantes(atritos) , ruim(situacao_psicologica).  
problema(time) :- bom(preparo_fisico), ruim(situacao_de_gols).  
constantes(atritos) :- jogador(X), discute_tecnico(X).  
ruim(situacao_psicologica) :- jogador(X), suspenso(X).  
ruim(situacao_de_gols) :- gols_sofridos(X), gols_feitos(Y), maior(X,Y).  
suspenso(X) :- cartao_vermelho(X).  
  
discontiguous(jogador/1).
```

Fonte: Autores