

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

---

**Detekce vzorů chování pro pohyb  
robotů ve hře fotbal robotů**

**Behaviour Patterns for Robot  
Movement in Robot Soccer**

2012

Lukáš Palatinus

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student: **Lukáš Palatinus**  
Studijní program: B2647 Informační a komunikační technologie  
Studijní obor: 2612R025 Informatika a výpočetní technika  
Téma: **Detekce vzorů chování pro pohyb robota ve hře fotbal robotů**  
**Behaviour Patterns for Robot Movement in Robot Soccer**

### Zásady pro vypracování:

V bakalářské práci se student zaměří na analýzu informací získaných z analýzy obrazu ve hře fotbalu robotů. Přesněji se jedná o detekci vzorů chování pohybu robotů, použitelných například pro odvození soupeřovi strategie nebo také pro predikci pohybu soupeřových robotů.

### Cíle práce:

1. Prostudovat možnosti a metody detekce vzorů chování použitelných ve hře fotbalu robotů.
2. Implementace zvolené metody a její zakomponování do knihovny pro fotbal robotů.
3. Vytvoření sady experimentů pro otestování této funkčnosti.

### Seznam doporučené odborné literatury:

A. D. Lattner, A. Miene, U. Visser and O. Herzog: Sequential Pattern Mining for Situation and Behavior Prediction in Simulated Robotic Soccer, Springer-Verlag, 2006  
F. Ramos, H. Ayanegui: Discovering Tactical Behavior Patterns Supported by Topological Structures in Soccer-Agent Domains, AAMAS, 2008  
H. Liu, H. Zha, K. Chen and P. Wang: A New Real-time Collision Prediction Model for Soccer Robots, 2011  
A. Bezek, M. Gams, I. Bratko: Multi-agent strategic modeling in a robotic soccer domain, AAMAS, 2006

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Václav Svatoň**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Havířově dne 15. dubna 2012

..........

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Havířově dne 15. dubna 2012

..........

V první řadě bych rád vřele poděkoval mému vedoucímu bakalářské práce, panu Ing. Václavu Svatoňovi za jeho trpělivost, pomoc a věcné připomínky při řešení komplikací, které vznikaly v průběhu řešení této bakalářské práce.

Dále bych rád poděkoval rodině, blízkým přátelům a všem ostatním, za jejich podporu.

## **Abstrakt**

Hlavním cílem této bakalářské práce bylo seznámit se s danou problematikou detekce chování robotů a zvolenou metodiku zaimplementovat. Implementace byla vytvořena v programovacím jazyce C#, ve kterém je potřeba daný problém řešit. Při hře fotbalu robotů, jako při lidském fotbalu, je velmi důležité přizpůsobovat hru dle stylu hry soupeře. Proto je cílem této práce nalézt nejlepší způsob jak detekovat a predikovat soupeřovy strategie a pohyby pokud možno v reálném čase za běhu hry. Díky těmto strategiím lze adaptovat naše vlastní a posunout tým blíže k vítězství.

**Klíčová slova:** C#, fotbal robotů, detekce chování, predikce vzorů, detekce soupeřových strategií, simulátor fotbalu robotů

## **Abstract**

The aim of this thesis is to introduce the topic of detection behavior of robots and implement the chosen methodology. The implementation was developed in C# programming language in which it is necessary to solve the problem. In the game, soccer robots, as in human football, it is very important to adapt the game by the style of the opponent's game. Therefore, the aim of this work is to find the best way to detect and predict the opponent's strategy and moves in real time during the game. Thanks these strategies can adapt our own strategies and will move the team closer to victory.

**Keywords:** C#, robot soccer, behaviour patterns, prediction of behavior, detection opponent's strategy, robot soccer simulator

## **Seznam použitých zkratk a symbolů**

MASDA	– Multiagentní strategické modelování
VYG	– Vygenerovaná strategie
ORG	– Originální strategie
OST	– Ostatní strategie

---

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Fotbal robotů</b>	<b>6</b>
2.1	Technický popis . . . . .	6
2.2	Kategorie fotbalu robotů . . . . .	7
<b>3</b>	<b>Predikce pohybů robotů ve hře</b>	<b>9</b>
3.1	Možnosti metodiky řešení . . . . .	9
3.2	Strategie . . . . .	12
3.3	Hra simulátoru . . . . .	13
<b>4</b>	<b>Sestavení strategií</b>	<b>16</b>
<b>5</b>	<b>Implementace kódu</b>	<b>18</b>
5.1	Externí verze . . . . .	18
5.2	Interní verze . . . . .	20
<b>6</b>	<b>Experimenty a analýza</b>	<b>23</b>
6.1	Použité programy . . . . .	23
6.2	Výsledky experimentů . . . . .	25
6.3	Experimenty a analýzy simulátoru . . . . .	28
<b>7</b>	<b>Závěr</b>	<b>29</b>
<b>8</b>	<b>Reference</b>	<b>30</b>

## Seznam tabulek

1	Výsledky všech zápasů . . . . .	26
2	Tabulka s průměrem podobností a celkovým skórem . . . . .	27



---

## Seznam obrázků

1	Hrací plocha [8] . . . . .	6
2	Různé typy robotů pro hru fotbal [8] . . . . .	7
3	Robot pro hru MiroSot [8] . . . . .	7
4	Freemanova kodifikace cest [2] . . . . .	10
5	Část taxonomie roli konceptu [4] . . . . .	12
6	Ukázka jednoho pravidla strategie . . . . .	12
7	Ukázka grafického zobrazení simulátoru . . . . .	14
8	Možný rozdíl mezi záznamy . . . . .	19
9	Ukázka průběhu hry upraveného simulátoru . . . . .	24
10	Ukázka pro dosažení maximální hodnoty . . . . .	25

## Seznam výpisů zdrojového kódu

1	Výpis z logu . . . . .	15
2	Volání metody isSimillar() . . . . .	20
3	Ukázka ukládání souřadnic pro třídu DetectionBehavior.cs . . . . .	21
4	Přidání nových záznamů do starých . . . . .	22

## 1 Úvod

V dnešní době si svět bez robotiky někteří z nás jen stěží dokážou představit. Jsou sestavováni, aby pomáhali a usnadňovali lidstvu život, ale taky aby podpořili zábavu a pokořili další úskalí, například ve sportu. V těchto oblastech se kloubí mnoho vědeckých oblastí jako je robotika, mikroelektronika, řízení, umělá inteligence, zpracování obrazu a komunikace.

Tým katedry informatiky na Vysoké škole báňské – Technické univerzitě v Ostravě, se již snaží vyvinout vlastní simulátor pro hru fotbalu robotů, za účelem simulace hry pro testování strategií a taktik, kdy nejsou fyzičtí roboti prozatím k dispozici. Toto konstruování programu je rozděleno na několik dílčích úkolů. Ať už jde o pohyb z místa „a“ na místo „b“, otáčení robotů kolem své osy, nebo predikce chování soupeřových strategií. Jednotlivé dílčí části jsou pak implementovány do celkového programu, který bude použit pro hru samotnou.

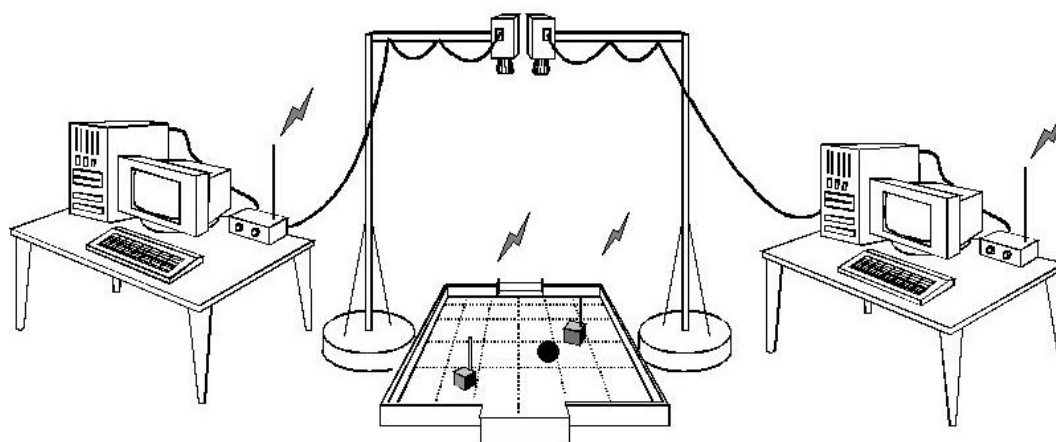
Vytvořením a zaimplementováním vybrané metody pro detekci soupeřových pohybů do 2D simulátoru fotbalu robotů, má za účel lépe reagovat na soupeřův útok či obranu během hry, a tak dosáhnout ve všech odehraných zápasech lepších výsledků.

Celá bakalářská práce je rozdělena do několika částí. V první řadě je potřebné se řádně dozvědět něco o fotbale robotů, abychom si dokázali představit, jak taková hra vypadá a probíhá. Hlavní částí bakalářské práce je seznámit se s problematikou predikce pohybů soupeřových robotů a správně je pochopit, aby mohla být vyřešena. V další části je zapotřebí se seznámit s již vytvořeným simulátorem fotbalu, který je již z části funkční a na jeho základě, by mohlo dojít ke zkonstruování řešení daného problému, kterým se tato práce zabývá. Mezi dalšími tématy jsou pak návrhy řešení a realizace kódu. Na tuto část pak bude navázáno s různými ukázkami a vysvětleními dvou verzí programu. V neposlední řadě pak proběhnou pokusy a experimenty, kde bude porovnán originál proti vygenerované strategii.

## 2 Fotbal robotů

### 2.1 Technický popis

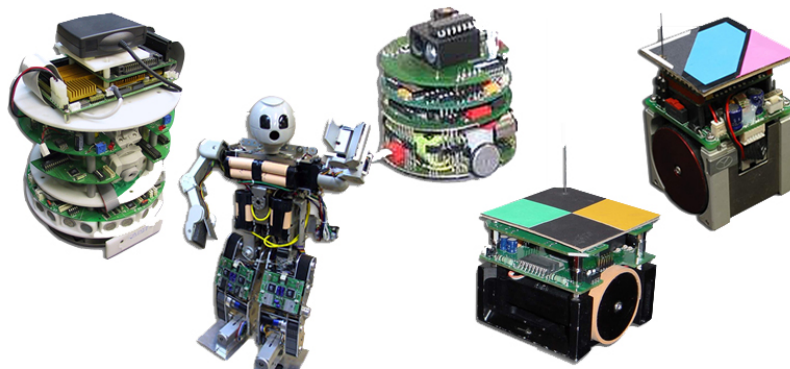
Robotický fotbal si lze představit jako klasický fotbal, se kterým je obyčejný člověk každodenně v kontaktu. Je zde pouze pár rozdílů a to, že místo lidských fotbalistů zde figurují robotičtí fotbalisté. Každý z těchto hráčů se pohybuje na dvou nezávisle poháněných kolech. Jednotlivé zápasy v robotickém fotbale probíhají mezi dvěma týmy, kde počet robotů na hřišti je úměrná typu robotického fotbalu. Každý tým má k dispozici vlastní počítač a barevnou kameru, která snímá aktuální pozice všech robotů a míče. Celá hra probíhá bez jakéhokoliv zásahu člověka, až na výjimky, kdy se rozhodčí (člověk) rozhodne v určité situaci pro faul, a nebo nastane nějaká patová situace (např. kdy se roboti navzájem blokují v rohu hřiště, kde se momentálně nachází míč a z této situace se nemohou sami dostat v určitém časovém intervalu).



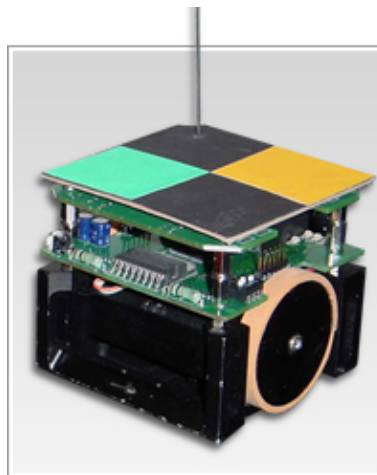
Obrázek 1: Hrací plocha [8]

## 2.2 Kategorie fotbalu robotů

Již existuje mnoho kategorií, které se liší jak rozměry hracího hřiště, tak i samotnými roboty. Pro každou takovou kategorii existují i odlišná pravidla.



Obrázek 2: Různé typy robotů pro hru fotbal [8]



Obrázek 3: Robot pro hru MiroSot [8]

### 2.2.1 MiroSot

Tým Vysoké školy báňské se snaží přiblížit simulátorem hry typu MiroSot [8], který bude v budoucnu ovládat fotbalové roboty (obrázek č. 4) dané pro tuto kategorii. Nejvíce se bude blížit kategorii hry MiroSot. V tomto typu se tým skládá z pěti robotických hráčů, z nichž jeden může být brankář, dále se skládá z manažera a trenéra (člověk), kteří můžou být pouze v jevišti. Velikost každého robota musí být menší než 7,5cm x 7,5cm x 7,5cm, kde výška antény se do rozměrů robota nepočítá.

Parametry hry:

- **Robot:** max. 7,5cm x 7,5cm x 7,5cm
- **Míč:** oranžový golfový míček
- **Hrací plocha:** pro střední ligy: 220cm x 180cm, pro velké ligy: 400cm x 280 cm

### 3 Predikce pohybů robotů ve hře

Všechny týmy jakéhokoliv sportu světa mají předpřipravené a nazkoušené strategie, které jim pomáhají v útočných situacích zvýšit šance, dát svým soupeřům gól nebo při obraných situacích, kdy se snaží soupeřům v útoku zabránit. Každé družstvo má připraveno několik těchto taktik, které využijí ve správnou situaci a na základě určitých aspektů. V našem případě, ve hře fotbalů robotů, je hlavním aspektem aktuální pozice robotů i míče.

Cílem této práce je nalézt způsob, jak nejrychleji a nejpřesněji určit soupeřův aktuální pohyb a strategii. V případě správné predikce pohybu či strategie zajistí týmu velmi významnou výhodu, která pravděpodobně zajistí vítězství v zápase.

#### 3.1 Možnosti metodiky řešení

Světové roboticko-fotbalové týmy řeší tuto problematiku na základě mnoholetých zkušeností a s pomocí mnoha vědeckých odvětví.

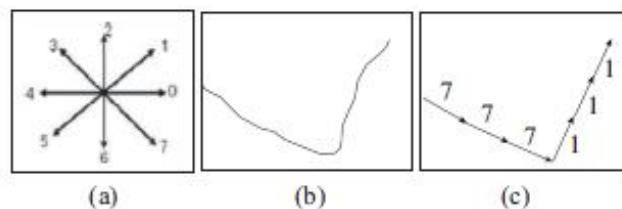
##### 3.1.1 Freemanova kodifikace

Družstvo z Mexika řeší tuto problematiku na základě takzvaných Freemanových kodifikací [2], která se vztahuje na práci Vissera a jeho kolegů [6], kteří využívají neuronových sítí pro predikování pohybů. Jednotlivé pozice hráče jsou pak vkládány do neuronové sítě a snaží se je zařadit do předem vymezených útvarů. Hlavním rozdílem přístupu pana Vissera, pana Ramose a jejich spolupracovníků je v tom, že Visserova metodika nepředstavuje několikanásobné vztahy mezi hráči což znamená, že Ramosova metoda bere v potaz různé kombinatoriky těchto vztahů mezi hráči.

Objevování taktiky soupeře pomocí Freemanovy kodifikace je rozdělen do šesti následujících kroků:

1. krok: **Načtení vstupních dat** – načtení všech pozic hráčů i míčku ze souboru (logu).
2. krok: **Eliminace podobných cest** – jsou vyhledány veškeré podobné cesty míče, které jsou pak odstraněny, až na jeden exemplář. Dále jsou rozděleny na dvě části, vždy začínající od středu hřiště směrem k brance.

3. krok: **První Freemanová kodifikace** – cesty míčku jsou kódovány do sekvence čísel použitím Freemanova pravidla, kdy každý směr míčku má specifickou hodnotu (obrázek č. 4 (a)). Dotyčná cesta je pak na základě těchto směrů zakódována do pole čísel. V případě obrázku č. 4 (c) je kód 7-7-7-1-1-1.
4. krok: **Druhá Freemanova kodifikace sekvence** – každé pole čísel, které jsme získali z kroku 3, bude převedeno do více abstraktního kódu. Necht' existují segmenty A, B, . . . ., H, kdy každý má stejnou číselnou orientaci jako v případě na obrázku č. 4. Segment A představuje číselnou hodnotu 0, segment B reprezentuje hodnotu 1, atd. V našem případě, když je pole hodnot 7-7-7-1-1-1 je nový kód HB.
5. krok: **Zjištění všeobecných vzorů** – v tomto kroku se zjišťuje nejvíce používané pod-sequence. Ty jsou zjištěny pomocí metody zobecnění stromu, který nalezne obecný vzorec pro taktický pohyb.
6. krok: **Asociace hráčů a zón** - topologická struktura používaná pro nalezení formací je velmi dobrá pro stanovení taktických her a využití za běhu hry.



Obrázek 4: Freemanova kodifikace cest [2]

### 3.1.2 Multiagentní strategické modelování

Univerzita v Lublani publikovala v roce 2006 novou metodu pro detekování pohybů soupeře, jménem multiagentní strategické modelování [4] zkratkou MASDA. Metoda je schopna popsat soupeřovi strategie graficky i symbolicky.



---

Celkový algoritmus MASDA může být reprezentován o třech krocích, kdy každý krok má za úkol další drobnější úkoly.

**1. krok: Předzpracování dat**

- detekce akcí ze surových dat (z atributů či souboru)
- generace akčních sekvencí (symbolické sekvence)
- úvod doménových znalostí (hierarchická koncepce)

**2. krok: Grafický popis vytváření**

- krok, který pomáhá při vytvoření grafu
- proces abstrakce
- výběr strategie

**3. krok: Symbolický popis určení**

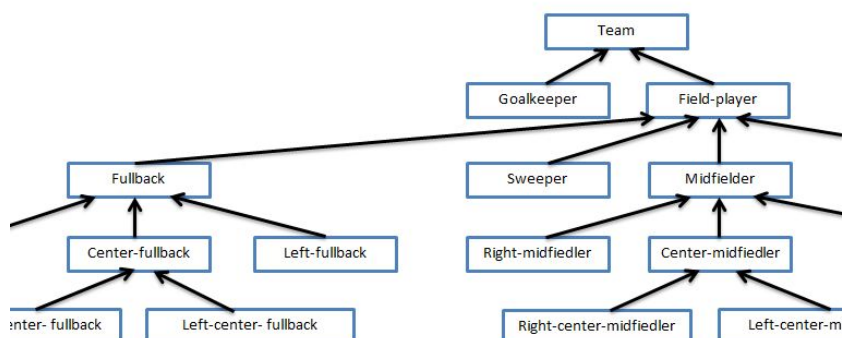
- popis vygenerované akce
- vytváření strategických dat
- pravidla přerušení

Výstup každého kroku je vstup pro každý následující postup.

První dva úkoly prvního kroku MASDA zjistí akce z multiagentních dat a vygeneruje odpovídající akční sekvenci. Třetí úkol prvního kroku zajišťuje za pomoci taxonomických koncepcí vytvoření hierarchické struktury akcí a jeho součástí (viz obrázek č. 5).

Druhý krok zajišťuje a popisuje proces výstavby grafické reprezentace akce z pozorovaného sekvenčního útoku, který vznikl v předzpracování dat (krok 1).

V posledním (třetím) kroku je proces vytváření symbolického popisu z grafického znázornění, které jsou získány ze strategických činností.



Obrázek 5: Část taxonomie roli konceptu [4]

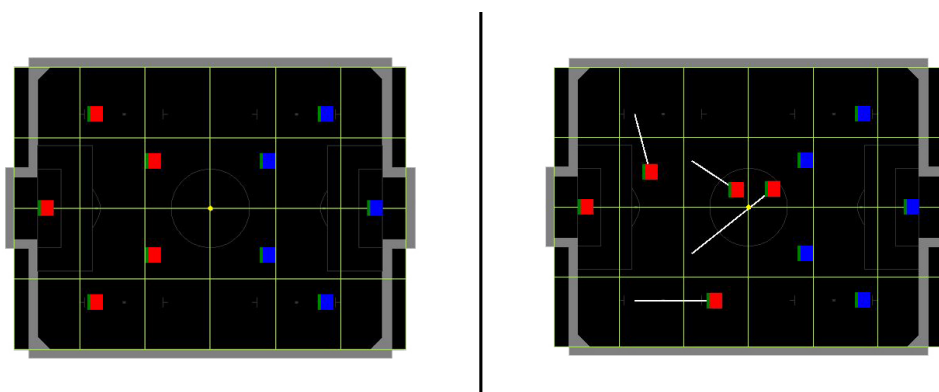
### 3.1.3 Skrytý Markovův model

Jiný tým z Pittsburské univerzity využívá Skrytý Markovův model [5], který predikuje pohyb na základě využití matematických konečných automatů, složitých vzorců a algoritmů, které vytváří automaty za běhu programu.

## 3.2 Strategie

Je potřeba porozumět tomu, co přesně tento pojem znamená a jak se vyznačuje v simulátoru.

Strategií rozumíme soubor, který je předem vytvořen a obsahuje herní situace na hřišti.



Obrázek 6: Ukázka jednoho pravidla strategie

Na obrázku č. 6 můžete vidět dva týmy (červený vs. modrý), kdy červená barva představuje náš tým. Celé hrací pole je znázorněno jako matice nebo dvourozměrné pole, podle kterého jsou tak i ohodnocovány. Například červený brankář se nachází na gridu s hodnotou 2,2. Tyto gridové souřadnice jsou více informativní pro určení strategie nežli souřadnice reálné, protože robot získává každých 100 milisekund nově vypočtené souřadnice pro jeho pohyb, a tudíž by bylo těchto informací příliš pro zpracování, kdy většina z těchto údajů by nebyla podstatná.

Toto je pouze ukázka jedné příkladné situace, kdy obě části obrázku představují startovací nebo také rozehrávací stav hry, kdy všichni hráči jsou na startovní pozici a míč je v samotném středu kruhu. Pravá půlka obrázku určuje, kterým směrem by se jednotliví hráči měli vydat po zahájení. Tomu se říká strategie hry, která je zapsána v souboru.

Soubor strategií můžeme rozdělit do dvou částí:

#### 1. Část

Je to takzvaná hlavička souboru, která obsahuje detailní informace, jako například autora, typ použitého algoritmu či velikost hracího pole a další.

#### 2. Část

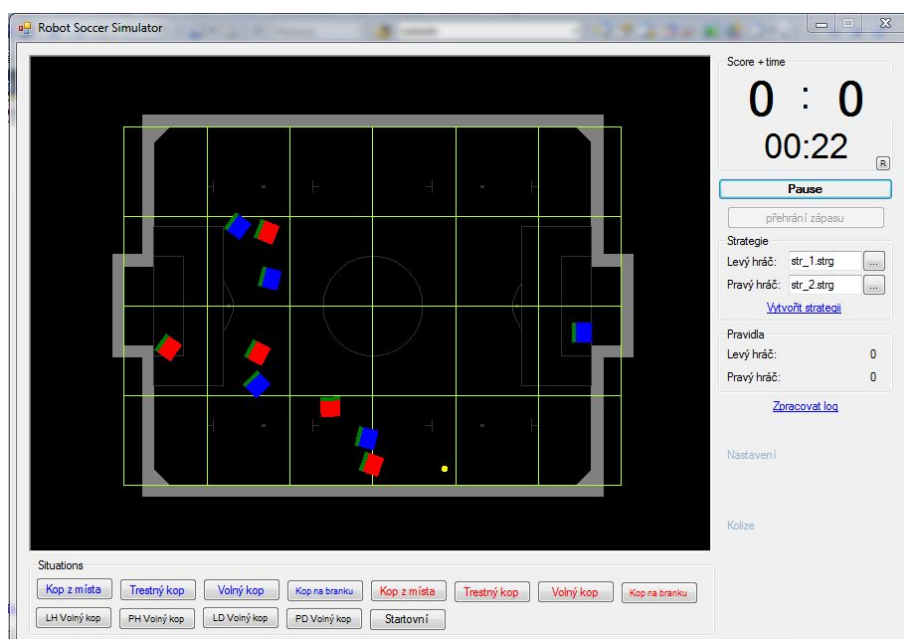
Tato část obsahuje již jednotlivá gridová pravidla, která reprezentují konkrétní situace ve hře a kam se mají robotičtí hráči přesunout.

**Pozn.** Všechny záznamy v souboru obsahují pouze čtyři souřadnice robotů i přesto, že každý tým má k dispozici celkem pět hráčů. Pátý robot je brankář, který neopouští brankoviště a tudíž, nejsou potřeba informace, kde se nachází, nebo kam se přemístí, protože ve většině případů zůstává na stejné pozici a tím neaplikuje žádnou speciální strategii.

### 3.3 Hra simulátoru

Vytvořený simulátor hry zpracovává strategie pro levý a pravý tým nezávisle na sobě. Ty jsou však od sebe separovány a každý pracuje zvlášť, tudíž nemají přístup k informacím druhého týmu. Tak to funguje i ve skutečném fotbalu robotů, kdy

každý tým má svůj vlastní počítač, kde běží program pro vlastní ovládání svých robotů. Po zahájení hry jsou všechny informace, příkazy a výpočty provedeny v herních tících, kdy jeden herní tik trvá přibližně 100 milisekund. V každém tomto herním tiku se skrze analýzu obrazu z kamery načtou a uloží v programu aktuální souřadnice míčku a pozice všech robotů do speciálních proměnných. Na základě toho, že se nyní jedná jen o simulátor reálných robotů, tak se nevyužívají kamery pro snímání obrazu celé hrací plochy. Jednotlivé souřadnice jsou uloženy v proměnných, které znamenají aktuální pozici jednotlivých objektů na hracím poli a ty jsou vysílány do grafického zobrazení hry. Později, kdy se bude realizovat a implementovat program pro fyzický fotbal robotů, bude vytvořena třída algoritmů, která bude zpracovávat analýzu obrazu, a jejich výsledky budou ukládány do proměnných, se kterými bude tato práce při implementaci pracovat.



Obrázek 7: Ukázka grafického zobrazení simulátoru

Veškeré hodnoty pozic jsou vepsány do souboru s koncovkou \*.log. Log obsahuje v první řadě použitá pravidla ze strategií z jednotlivých tiků. Také dále gridové i reálné souřadnice míčku i všech robotů. Na konci záznamu nalezneme i aktuální skóre hry

---

a čas, který zbývá do konce zápasu. Během dvou minut zápasu log obsahuje přibližně 1200 řádků záznamů.

Vytvořený program bude zpracovávat v každém tiku všechny záznamy z logu a zpětně se bude pokoušet tyto strategie sestavit a přidávat je do svých vlastních strategií. V případě, že by soupeř použil stejnou nebo do určité míry podobnou strategii, tým již bude o tomto pohybu vědět a roboti se tak budou moci přesunout na správné místo a jejich útok, případně obranu zmařit.

---

```
leftRule rightRule; ball . real . x; ball . real . y; ball . grid . x; ball . grid . y; lr0 . real . x; lr0 . real . y ;...
8 1 ;0;0;4;3;-36,67;-30;3;2;-36,67;30;3;3;-73,33;-60;2;1;-73,33;60;2;4;-105;0;1;3;36,67;...
8 1 ;0;0;4;3;-34,72;-28,4;3;2;-34,39;28,92;3;3;-71,28;-58,54;2;1;-70,9;59,33;2;4;-105;...
8 1 ;0;0;4;3;-31,93;-26,12;3;2;-31,14;27,38;3;3;-68,35;-56,44;2;1;-67,43;58,37;2;4;-105;...
8 1 ;0;0;4;3;-29,14;-23,84;3;2;-27,88;25,84;3;3;-65,42;-54,35;2;1;-63,96;57,41;2;4;-105;...
8 1 ;0;0;4;3;-26,36;-21,57;3;2;-24,63;24,3;3;3;-62,49;-52,26;2;1;-60,49;56,46;2;4;-105;...
```

---

Výpis 1: Výpis z logu

## 4 Sestavení strategií

Budou vytvořeny dvě verze programu, kdy jedna z nich bude vytvářena na vývoj celkového programu pro sestavování soupeřových strategií a pro pozdější experimenty. Druhá verze (externí), která bude zaimplementována do celkového simulátoru hry fotbalu robotů.

I přesto, že jsou k dispozici různé druhy řešení metod pro generaci soupeřových pohybů a taktik, bude mít tato práce vlastní způsob predikování strategií, která je z části podobná Freemanové kodifikaci. Ta na rozdíl od řešení této bakalářské práce spoléhá na sekvenci čísel určené z tras míčku. Bude vytvořen nový způsob generování pravidel a to proto, že vytvořené soubory, obsahující pravidla pro pohyb robotů v daných situacích, jsou předem dané. Pravděpodobnost, že se tento předefinovaný pohyb naskytne minimálně podruhé za celý průběh hry, je velice vysoká. Z tohoto důvodu bude program zpracovávat a vyhodnocovat jednotlivé tahy na základě předešlých pozicích hráčů i míčku na hřišti.

Celý projekt je rozpracován do několika dílčích částí:

### 1. část: Načtení dat

Pro obě verze programu bude rozdílný přístup a načtení souřadnic pozic robotů. Pro první verzi, která bude využívána pro následující vývoj celého programu a pro následné experimenty a pokusy, budou získávána data z logu. Pro druhou verzi (interní) budou hodnoty přiděleny z jiné třídy.

### 2. část: Sestavení hrubé verze strategií

Na základě herních tiků jsou sestaveny pohyby, které v minulých tazích proběhly. Pak nastane kontrola a následná eliminace duplikátů a jejich následný zápis do souboru.

### 3. část: Kontrola podobnosti

Využití části kódu simulátoru proběhne kontrola podobnosti všech strategií tak, aby nedošlo ke zbytečnému opětovnému zapsání pravidla, které je z části podobné jinému a tudíž by bylo zbytečné tuto informaci uchovávat.

#### 4. část: **Přivlastnění pravidel**

Pomocí upraveného kódu podobnosti pravidel, budou všechny pravidla porovnána se stávajícími a v případě, že dané pravidlo je nové, bude přivlastněno do strategií týmu, který je bude moci dřív či později využít.

#### 5. část: **Použití určitého pravidla**

V okamžiku, kdy se jedna z dřívějších situací bude opakovat, může nově vygenerované a přivlastněné pravidlo pomoci robotům zasáhnout na správné místo a tím překazit soupeřovu šanci.

Všechny tyto části jsou vyhodnoceny během celého jednoho tiků hry a v co nejkratším čase tak, aby nezpomalovali průběh celého simulátoru.

## 5 Implementace kódu

Jak již bylo řečeno, budou existovat nyní dvě verze programu, kdy obě budou plnit stejnou funkci a budou použity v rozličných případech.

### 5.1 Externí verze

Tato verze byla vyvíjena jako samostatná část programu nezávisle na simulátoru v jazyce C#.

Pro tuto verzi bylo potřeba zajistit souřadnicové hodnoty pro jejich další zpracování. To je možné jen ze souboru, který je generován za běhu grafického simulátoru hry (log). Pro jejich získání byla použita metoda parsování. Všechny následné hodnoty jsou poté uloženy do patřičně pojmenovaných proměnných.

Pro splnění druhého kroku vytvoření hrubé verze seznamu všech strategií byla vytvořena řada metod. Jedna z funkcí je `createString()`, která se stará o vytvoření textového řetězce pravidla strategie a mohla být posléze uložena do souboru. Z důvodů, že herní tik trvá přibližně 100 milisekund se velice často stává, že se někteří roboti, za tak krátký čas, nestihnou přesunout z jednoho gridu matice do druhého. To řeší metoda `nextMove()`, která se stará o to, aby našla rozdílné gridové souřadnice od těch, kde se hráči momentálně nacházejí. A taktéž je zde `checkDuplicity()`, která zajišťuje kontrolu duplicity záznamu o tom, že nový záznam nebyl dřív vytvořen. V takovém případě, by byl tento údaj vynechán. Všechny nalezené údaje jsou posléze uloženy v daném formátu a uloženy do souboru „`enemyStrategy.strg`“.

Vygenerovaný soubor obsahuje hrubý seznam pravidel, která byla vytvořena z možných dvou tisíc herních tiků. Avšak některá pravidla se liší od toho dalšího, která nastává v dalším možném kole. To znamená, že se liší jen o jeden grid, jak představuje obrázek č. 8.



.Rule 1 o Name	.Rule 1 o Name
.Mine 3,3 3,3 3,3 3,3	.Mine 4,4 3,3 3,3 3,3
.Oppnt 3,3 3,3 3,3 3,3	.Oppnt 3,3 3,3 3,3 3,3
.Ball 3,3	.Ball 3,3
.Move 3,3 3,3 3,3 3,3	.Move 3,3 3,3 3,3 3,3

Obrázek 8: Možný rozdíl mezi záznamy

Při řešení tohoto problému v konečném stádiu externí programu byla využita část naprogramovaného kódu simulátoru, který je nezbytný pro poslední sérii metod pro redukci duplicity vygenerovaných strategií. Jde o metodu porovnání podobnosti matic.

Tato metoda načte všechny předchozí uložené pravidla ze souboru „enemyStrategy.strg“, uloží si je jako novou instanci třídy GridStrategy. Při načtení každého nového záznamu ze souboru je spuštěna metoda isSimillar(rules, r), která vrací hodnotu „true“ nebo „false“ a to na základě toho, jak moc jsou si dané matice v rámci mezí podobné. Parametr „rules“ je list typu GridRule a obsahuje veškerý seznam pravidel, které prošly kontrolou a parametr „r“ je následující záznam ze souboru. Metoda isSimillar() se vyhodnotí jako „true“ ve chvíli, kdy záznam načtený z enemyStrategy.strg, je vyhodnocen na základě podobnosti pravidel a je menší než 3.

Pro konkrétní příklad může být použit opětovně obrázek č. 8, kde existují pouze tyto dva záznamy. Záznam nalevo, jakožto první načtené pravidlo, bude automaticky uloženo do listu „rules“, protože je jako pravidlo první a nemá se s kým porovnat. Následně bude načteno druhé pravidlo (pravá strana obrázku). Projde celou metodou isSimillar(), která se vyhodnotí jako „true“ a nebude do seznamu přidáno. Je to z důvodu, že podobnost obou těchto pravidel (matic) byla vyhodnocena jako hodnota „1“ a to znamená, že druhé pravidlo je do patřičné míry podobné prvnímu.

---

```
rules = new List<GridRule>();
try
{
    while (!reader.EndOfStream)
    {
        reader.ReadLine();
        GridRule r = new GridRule(reader.ReadLine(), reader.ReadLine(), reader.ReadLine(), reader.
            ReadLine(), reader.ReadLine());
        if ( isSimillar (rules, r) == false)
        {
            rules.Add((mirror ? r.Mirror((int)size.X) : r);
        }
    }
}
catch (Exception) { }
```

---

#### Výpis 2: Volání metody isSimillar()

Takto budou zkontrolována všechna pravidla a ta, která jsou od sebe odlišná a jsou zaznamenána v listu, jsou posléze uložena do souboru „enemyStrategy2.strg“ a to z důvodu pro pozdější experimenty, pokusy a analýzy obou verzí, které nám řeknou jestli tato práce je v programu užitečná či nikoliv.

Protože se jedná jen o externí verzi programu, která nijak nespolupracuje s grafickým simulátorem hry a slouží jen pro vygenerování souboru strategií, nejsou navazující stanovené kroky 4 a 5 v této verzi důležité ani řešeny.

## 5.2 Interní verze

Interní verze programu je zaimplementována do celkového simulátoru hry, tak aby predikovala za běhu hry strategii soupeře a pomáhala týmu tak lépe reagovat v reálném čase. Metoda pro detekci chování, predikcí pohybů robotů a strategií je vložena do simulátoru jako nová třída jménem DetectionBehavior.cs.

### 5.2.1 Změny

Hned první změnou je načítání informací o pozicích, se kterými tato práce nadále pracuje. Toto načítání údajů již neprobíhá ze souboru s koncovkou \*.log, ale přímo z vytvořené třídy Storage, která obsahuje veškeré reálné souřadnice všech hráčů a míče. Přestože bakalářská práce pracuje na základě gridových souřadnic, je potřeba převést reálné souřadnice pozic robotů i míčku do gridových souřadnic. K tomuto účelu je zavolána metoda RealToGrid() s jejíž pomocí se převedou reálné souřadnice na gridové a ty jsou uloženy z třídy Simulator.cs do patřičných listů naší třídy Detectionbehavior.cs.

---

```
db.BallGridList .Add((int)RealToGrid(storage.Ball.Position, SimulatorGameSetting).X);
db.BallGridList .Add((int)RealToGrid(storage.Ball.Position, SimulatorGameSetting).Y);

db.LeftRob0List.Add((int)RealToGrid(storage.MyRobots[0].Position, SimulatorGameSetting).X);
db.LeftRob0List.Add((int)RealToGrid(storage.MyRobots[0].Position, SimulatorGameSetting).Y);
```

---

Výpis 3: Ukázka ukládání souřadnic pro třídu DetectionBehavior.cs

Následně probíhá metoda sestavení strategií standartním způsobem podle externí verze. Do doby než je volán konstruktor třídy GridStrategy, kde byl přidán nový parametr detectionMethod, který říká, zdali konstruktor byl vytvořen třídou pro predikci soupeřových pohybů či jakoukoliv jinou třídou.

V neposlední řadě pak zbývá výsledné vytvořené soupeřovy strategie přidat k vlastním strategiím týmu. Toho se docílí opětovným zavoláním metody isSimilar() z třídy GStrategy, která porovná současné strategie využívané týmem s novými, kdy v případě nalezení nové strategie se přidá k stávajícím strategiím týmu, v opačném případě se záznam přeskočí a pokračuje s dalšími v seznamu.

---

```
db.CreateStrg();

if (number > 3)
{
    foreach (GridRule item in db.gs.Rules)
    {
        if (leftPlayerStrategy .GStrategy.isSimillar ( leftPlayerStrategy .GStrategy.Rules, item) ==
            false)
        {
            leftPlayerStrategy .GStrategy.Rules.Add((false) ? item.Mirror((int)new Vector2D(6, 4).X) :
                item);
        }
    }
}
```

---

#### Výpis 4: Přidání nových záznamů do starých

Náš tým je flexibilnější a učí se soupeřovým strategiím za pochodu a přidává je k vlastním. Proto zbývá vyčkat na danou situaci či jí podobnou, která se dříve stala a poté se zavolá nové pravidlo, které bude schopné zasáhnout na správné místo hrací plochy tak, aby soupeřovu týmu překazil útočný nebo obraný plán.

## 6 Experimenty a analýza

Pro otestování, zdali tato práce bude mít přínos pro celkový simulátor hry či nikoliv, bude provedena řada experimentů jak originálních částí, tak i jejich vygenerovaných strategií. Poté provedeme analýzu výsledků těchto experimentů, abychom zjistili, zdali jsou si navzájem podobné, a jak si vedou v zápasech proti stejným soupeřům ve hře oproti originálu strategií a jsou-li použitelné pro náš simulátor pro hru fotbal robotů.

Pro provedení experimentů bylo vytvořeno pět zkušebních souborů strategií. Jde o soubory:

1. Strategie\_1.strg
2. Strategie\_2.strg
3. Strategie\_3.strg
4. Strategie\_4.strg
5. Strategie\_5.strg

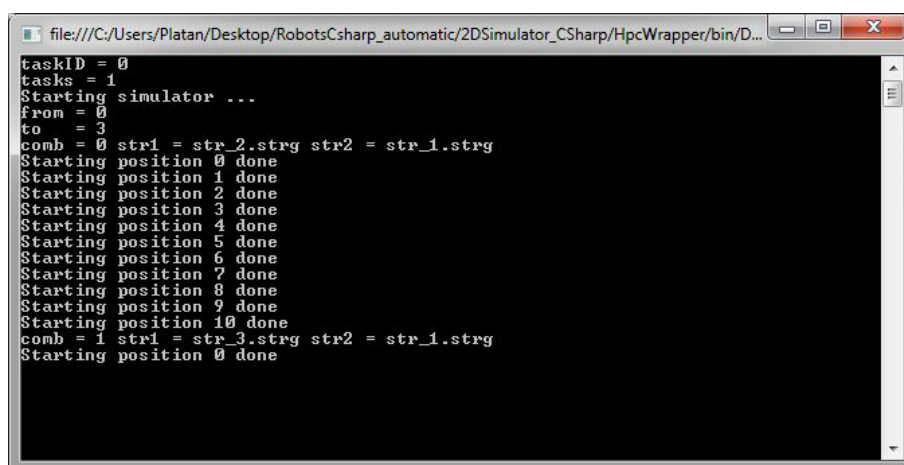
Každá z uvedených strategií obsahuje celkem 15 pravidel a navzájem se od sebe ve všech těchto bodech liší, s výjimkou hlaviček.

### 6.1 Použité programy

Pro provedení experimentů budeme využívat řadu programů, které nám pomůžou při těchto pokusech a pak i při následné analýze.

V první řadě se bude používat speciálně upravená verze simulátoru hry, kde není zobrazen grafický průběh hry. Typický čas zápasu jsou dvě minuty, avšak díky úpravě simulátoru je zápas dvojnásobně zrychlen. V adresáři se nacházejí dvě složky „Strategies1“ a „Strategies2“. V první složce se nachází pokaždé čtyři soubory strategií pro levý tým a ve složce „Strategies2“ se nachází právě testovaný soubor strategií. Po spuštění programu je mezi těmito odlišnými strategiemi odehráno celkem jedenáct

zápasů (viz obrázek č. 9), kdy v každé z těchto zápasů mají roboti odlišnou startovní pozici. Tím chceme zjistit, která pozice je nejlepší pro startovní pozici našich robotů, aby dosáhli lepších výsledků ve hře. Během simulace hry v čase dvou minut je pro každý zápas vygenerován speciální soubor (\*.log), který obsahuje veškeré pozice v jednotlivých tících hry. Každý soubor bude obsahovat celkem 1201 řádků (záznamů), kdy první řádek je hlavička souboru. Tyto soubory budou následně zpracovány parsovací metodou pro následné vygenerování strategií.



```
file:///C:/Users/Platan/Desktop/RobotsCsharp_automatic/2DSimulator_CSharp/HpcWrapper/bin/D...
taskID = 0
tasks = 1
Starting simulator ...
from = 0
to = 3
comb = 0 str1 = str_2.strg str2 = str_1.strg
Starting position 0 done
Starting position 1 done
Starting position 2 done
Starting position 3 done
Starting position 4 done
Starting position 5 done
Starting position 6 done
Starting position 7 done
Starting position 8 done
Starting position 9 done
Starting position 10 done
comb = 1 str1 = str_3.strg str2 = str_1.strg
Starting position 0 done
```

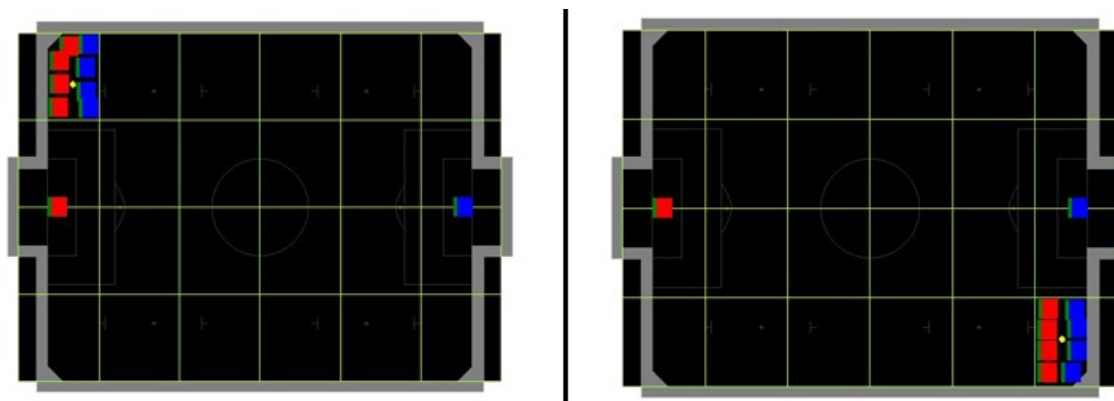
Obrázek 9: Ukázka průběhu hry upraveného simulátoru

Během experimentů bude použita další verze programu, a to externí verze sestavení soupeřovy strategie a pohybu. Za pomoci vygenerovaných logů během zápasů a této verze programu budou zpětně sestaveny strategie, které byly za běhu hry soupeřem použity.

V konečné fázi byl vytvořen program pro porovnání podobnosti matic, kdy byla využita část kódu simulátoru hry. Budou se zde porovnávat vygenerované části programu s originálem, kdy nám vyjde hodnota říkající, jak moc jsou si tyto dva soubory podobné. Díky projektu pro výpočet podobnosti matic budou moci být zjištěny hodnoty, které dále řeknou, jak moc se daný vygenerovaný seznam pravidel ve strategiích liší od originálu. Jednotlivé hodnoty jsou pro každé pravidlo sečteny dohromady a vyděleno počtem (v našem případě patnácti), tak abychom zjistili

průměrnou podobnost všech pravidel. Minimální hodnota podobnosti je „0“, která značí úplnou shodu obou matic ve všech bodech a maximální hodnota je „45“, která říká, že pravidla jsou úplně jiná a pokaždé na opačné straně hřiště.

**Pozn.** Maximální hodnota podobnosti nastane jen ve specifických situacích, kterou představuje obrázek č. 10. V prvním pravidle (levá část obrázku) jsou všichni roboti i s míčkem v levém horním rohu, oproti tomu porovnávané druhé pravidlo zobrazuje výskyt robotů na úplně opačné straně hřiště. Tyto rozdíly byly programem vypočteny na hodnotu podobnosti „45“.



Obrázek 10: Ukázka pro dosažení maximální hodnoty

## 6.2 Výsledky experimentů

Veškeré strategie byly postaveny v jednotlivých zápasech proti zbývajícím čtyřem strategiím. Každý soubor ve svém vlastním experimentu prošel na každé z jedenácti startovních pozic 4 zápasy, tedy celkem 44 zápasů. Veškeré výsledky zápasů byly zaznamenány do tabulky č. 2. Druhý sloupec označuje průměrný počet programem vytvořených pravidel na základě logů všech zápasů. Třetí sloupec oznamuje průměrnou podobnost ze všech čtyřiačtyřiceti her. Poslední tři sloupce obsahují celkové skóre jednotlivých zápasů.

Vyg strategie	Průměr vyg pravidel	Průměr podobnosti vyg vůči org	org:ost	vyg:ost	vyg:org
Strategie_1.strg	35	7,04	24:14	24:26	2:6
Strategie_2.strg	38	5,13	15:25	31:19	11:5
Strategie_3.strg	33	5,22	17:16	25:16	11:5
Strategie_4.strg	27	4,10	27:22	22:20	6:7
Strategie_5.strg	41	4,58	21:20	32:12	5:5

Tabulka 1: Výsledky všech zápasů

**Pozn.** V tabulce pro zkrácenou verzi budou znázorněny názvy strategií zkratkami: Vygenerovaná strategie z logu, které byly zpětně sestaveny programem a bude v tabulce zkrácen na „Vyg“. Originální strategie bude zapisována zkratkou „Org“. A poslední zkratkou jsou ostatní strategie vyjma právě testované „Ost“.

V celkovém souhrnu experimentů budou vítězství ohodnocena dvěma body, jako v opravdovém fotbalu. Za jednotlivé kategorie, kterými jsou zápasy proti ostatním pokusným pravidlům a také zápasům originální verze proti vygenerované verzi. Za vyrovnané skóre je pak oběma stranám připočítán jeden bod. V celkovém součtu bodů činí skóre 12:8 ve prospěch této práce, která měla za úkol predikovat soupeřovy strategie a pohyby v reálném čase za běhu hry, které pak ukládá do svých vlastních strategií. Proto je tato práce přínosem a vyplatí se jí využít v simulátoru hry.

Za pomocí analýzy veškerých experimentů byly dosaženy důležité a podstatné informace, které by měly dokázat pomoci robotům ve hře a tím jim usnadnit vítězství.

První z těchto zjištěných informací je, která ze startovních pozic se průměrně nejvíce podobá originálním strategickým souborům, které byly pro tento experiment využity.



Pozice	Podobnost	Skóre
0	5,09	13:32
1	4,95	24:25
2	5,35	<b>23:16</b>
3	4,83	19:19
4	5,04	11:19
5	5,34	22:22
6	5,88	12:13
7	5,23	24:23
8	5,70	22:15
9	<b>4,71</b>	19:27
10	5,26	5:19

Tabulka 2: Tabulka s průměrem podobností a celkovým skórem

Největší pravděpodobnost pro určení největší podobnosti vytvoření kopie soupeřových strategií máme (dle prostředního sloupce) na pozici číslo devět, kdy je shoda 4,71. Pro zjištění co nejautentičtější strategie vůči originálu je nejlepší využít startovní pozici číslo 9, která zajistí informace, které budou ve prospěch týmu a pomohou tak snáze k vítězství.

Z těchto údajů mohou být vydedukovány informace, že nejlepšího výsledku lze dosáhnout na startovní pozici s číslem 2, kdy roboti našeho potencionálního týmu dokázali skórovat třiadvacetkrát a dostat vstřelených pouze šestnáct gólů.

**Pozn.** Některé výsledky mají na svědomí patové situace během hry, kdy se míč dostane do rohu, kde se robotičtí hráči nemohou sami z této situace dostat a není zde ani žádný rozhodčí, který by hru pozastavil a nechal hru opětovně začít z patřičné pozice.

### 6.3 Experimenty a analýzy simulátoru

Nyní budou provedeny řady pokusů s grafickou verzí simulátoru, která obsahuje již implementovanou třídu „DetectionBehavior.cs“. Proběhne řada zápasů, kdy levý (červený) tým má funkci detekce pohybů a strategií soupeře k dispozici.

Bylo odehráno celkem 15 zápasů mezi jednotlivými strategiemi, které byly použity pro dřívější experimenty. Nastane stejné bodování jako při bodování v předchozích experimentech, které vyhodnotí výsledek, jak si na tom levý tým s novou třídou stojí. Vítězství týmu bude bodováno opětovně dvěma body, kdy při prohře nula body a při remíze obdrží oba týmy po jednom bodu.

Za běhu dvou minutové hry bylo levému týmu přidáno průměrně čtyřicet až padesát nových vygenerovaných pravidel, které program zpětně vytvořil za běhu programu. Levý (červený) tým získal z dříve uvedených patnácti her celkem 18 výherních bodů, oproti tomu pravý (modrý) tým obdržel bodů jen 11. Z těchto výsledků se dá usoudit, že třída „DetectionBehavior.cs“ je doopravdy užitečná a funkční pro predikci pohybů soupeřových robotů a strategií.

## 7 Závěr

Za průběhu vykonávání bakalářské práce jsem se detailně seznámil s různými metodami pro predikci soupeřových strategií a pohybu. Bylo důležité se přiblížit fungování již vytvořeného simulátoru hry fotbalu robotů, kdy bylo nepostradatelné nalézt nejlepší způsob jakou metodu zvolit a jak ji poté zimplementovat. Pro dosažení těchto cílů bylo důležité se zaměřit na programovací jazyk C#, ve kterém je program napsán a implementován.

Veškeré tyto nabitě vědomosti jsem využil pro rozšíření simulátoru univerzity pro rozšíření třídy jménem „DetectionBehavior.cs“, která řeší určování pohybů soupeře. V jiných třídách pak byl zdrojový kód buď upraven anebo byly přidány nové kódy, které jsou nezbytné pro fungování výše uvedené třídy. Tato vytvořená třída umožňuje našemu týmu fotbalových robotů zpětně sestavovat soupeřovy pohyby, které již vykonal a po zkontrolování podobnosti nově vygenerovaného pravidla s pravidly našeho týmu, budou tyto nové poznatky přidány k vlastním pravidlům. Tyto nové informace by mohly být využity v patřičné situaci, kdy se protivník opět pokusí o znovupoužití této taktiky. V tom okamžiku budou naši roboti o této hrozbě vědět a budou schopni jí čelit. Viz podrobný popis kapitola 5.1 pro externí verzi, která obsahuje podrobný popis celé struktury metodiky a kapitola 5.2, kde se nalézají informace přizpůsobení první verze pro simulátor.

Během provádění experimentů a pokusů jsem zjistil, že ne pokaždé se podařilo vygenerovat podobnou kopii originálních strategií nebo která by byla lepší než samotný originál. Avšak ve většině případů si vedla vytvořená pravidla mnohem lépe než ta prvotní a dokázala je tak přehrát a častěji skórovat či zabránit útoku soupeře. V průběhu testování jsem zjistil i další zajímavé informace a to ohledně výběru startovních pozic. Rozdílné reálné souřadnice pro rozehrávací situaci ve hře zajišťují i rozdílné výsledky. Stejně tak i jsou částečně odlišné průměrné podobnosti, kdy různé startovní pozice zaznamenávají i různé údaje. Proto je zapotřebí, aby se tato metoda pak dále vyvíjela a aby výsledky byly posléze co nejlepší.

## 8 Reference

- [1] A. D. Lattner, A. Miene, U. Visser and O. Herzog, *Sequential Pattern Mining for Situation and Behavior Prediction in Simulated Robotic Soccer*, Springer-Verlag, Germany: Deutsche Forschungsgemeinschaft, 1994.
- [2] F. Ramos, H. Ayanegui, *Discovering Tactical Behavior Patterns Supported by Topological Structures in Soccer-Agent Domains*, Mexico: AAMAS, 2008.
- [3] H. Liu, H. Zha, K. Chen and P. Wang, *A New Real-time Collision Prediction Model for Soccer Robots*, China, 2011.
- [4] A. Bezek, M. Gams, I. Bratko, *Multi-agent strategic modeling in a robotic soccer domain*, Slovenia, 2011.
- [5] Kwun Han and Manuela Veloso, *Automated Robot Behavior Recognition*, Pittsburgh, 2000.
- [6] Ubbo Visser, Christian Drücker, Sebastian Hübner, Esko Schmidt and Hans-Georg Weland, *Recognizing Formations in Opponent Teams*, Germany, 2001.
- [7] Ing. Jakub Hrabec, *Robotický fotbal* [online], Dostupné z: <http://www.robohemia.cz/Robosoccer.html.cz>.
- [8] Federation of International Robot-soccer Association, *Robo Soccer* [online], Dostupné z: <http://www.fira.net>.
- [9] Microsoft, *Visual C#* [online], Dostupné z: <http://http://msdn.microsoft.com>.