

საქართველოს ტექნიკური უნივერსიტეტი

ხელნაწერის უფლებით

გიორგი არჩვაძე

სწავლების და ცოდნის მოპოვების მეთოდების დამუშავება და
კვლევა მრავალკომპონენტურ მეხსიერებიან სისტემებისთვის

წარმოდგენილია დოქტორის აკადემიური

ხარისხის მოსაპოვებლად

აკტორეფერატი

სადოქტორო პროგრამა “ინფორმატიკა” შიფრი 0401

საქართველოს ტექნიკური უნივერსიტეტი

თბილისი, საქართველო

2015 წელი

სამუშაო შესრულებულია საქართველოს ტექნიკური უნივერსიტეტში

ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

ეკონომიური ინფორმატიკის დეპარტამენტი

ხელმძღვანელი: პროფ. ზურაბ ბოსიკაშვილი

რეცენზენტები: -----

დაცვა შედგება ----- წლის "-----" -----, ----- საათზე

საქართველოს ტექნიკური უნივერსიტეტის -----

----- ფაკულტეტის სადისერტაციო საბჭოს
კოლეგიის

სხდომაზე, კორპუსი -----, აუდიტორია -----

მისამართი: 0175, თბილისი, კოსტავას 77.

დისერტაციის გაცნობა შეიძლება სტუ-ს ბიბლიოთეკაში,

ხოლო ავტორეფერატისა - ფაკულტეტის ვებგვერდზე

სადისერტაციო საბჭოს მდივანი პროფ. თინათინ კაიშაური

ბოლო ხუთი წელია შეინიშნება მზარდი ინტერესი ხელოვნური ინტელექტის ამოცანების მიმართ, კერძოდ მრავალ აგენტიანი სისტემებით ისეთი ამოცანების ამოხსნაზე, როგორებიცაა სხვადასხვა სახის ობიექტების სახეთა გამოცნობა, კოლექტიური გადაწყვეტილებების მიღება, არასრული ინფორმაციის შემთხვევაში დაგეგმვის ამოცანების ამოხსნა და სხვა. ეს განპირობებულია იმით, რომ მათ ამოსახსნელად საჭიროა დიდი გამოთვლითი რესურსები, ცოდნის მოპოვების და დაგროვების, ცოდნის განზოგადოების და დასკვნების კეთების ახალი მექანიზმების, მეთოდების და ალგორითმების შემუშავება.

კვლევის საგანი: დისერტაციის კვლევის საგანს წარმოადგენს კოლექტიური გადაწყვეტილების მიღების და დასკვნების კეთების მექანიზმები, მეთოდები და ალგორითმები სხვადასხვა სახის ობიექტების (სახეთა) გამოცნობასთან მიმართებებში.

სამეცნიერო ბაზისი: სადისერტაციო ნაშრომში კოლექტიური გადაწყვეტილების, დასკვნების კეთების და გამოცნობის პროცესებში გამოიყენება ხელოვნური ინტელექტის ისეთი მიღწევები, როგორებიცაა, ევრისტიკული ძებნა, ხელოვნური ნეირონული ქსელები, მანქანური სწავლება, არგუმენტაციის თეორია, სახეთა გამოცნობის თეორია და პრაქტიკა. მნიშვნელოვანია დისერტანტის სამეცნიერო ხელმძღვანელის მიერ შემუშავებული ცოდნაზე დაფუძნებული სისტემების არქიტექტურის წარმოდგენის პრიციპების გამოყენება.

კვლევის მიზანი: სადისერტაციო ნაშრომის ძირითად მიზანს წარმოადგენს შეიქმნას კოლექტიური დასკვნების კეთების (გადაწყვეტილების მიღების) ფრეიმვორკი(ბაზისი) და შემუშავდეს შესაბამისი განაწილებული სისტემის აგების პრინციპები ხელოვნური ინტელექტის კომპლექსური ამოცანების ამოსახსნელად.

ამ მიზნების მისაღწევად დისერტაციაში განხილულია შემდეგი ამოცანები:

- გამოცნობის ობიექტების (სახეების) წარმოდგენისთვის მოდელის შერჩევის ამოცანა.
- მრავალ აგენტური დასკვნების კეთების მექანიზმების შემუშავება
- სისტემის განაწილებული არქიტექტურის შემუშავება
- ობიექტების და ობიექტების კომპონენტების ეტალონებთან პარალელური შეთავსების და გამოცნობის მექანიზმების და მეთოდების შემუშავება

მიზნის მღწევის უპირველეს ამოცანას წარმოადგენს გამოცნობის ობიექტების (სახეების) წარმოდგენისთვის მოდელის შერჩევის ამოცანა. მოდელმა უნდა უზრუნველყოს ობიექტების მახასიათებლების და მიმალტებების ისეთი წარმოდგენა , რომ შემდგომ ეფექტურად განხორციელდეს მათი ამოცნობა. რადგანაც ობიექტების შესახებ ინფორმაცია შეიძლება იყვეს არასრული , ამიტომ უნდა აირჩეს დასკვნების კითხვის ისეთი მექანიზმი , რომელიც არ მოითხოვს სრულ ცოდნას და ამასთან შეიძლება მისი დაპარალელება.

მეცნიერული სიახლე: სადისერტაციო ნაშრომის ძირითად მეცნიერულ სიახლეს წარმოადგენს:

- მრავალ აგენტური გამოცნობის სისტემებისთვის შავი დაფის არქიტექტურის განვრცობა.
- გამოცნობის ობიექტების სტრუქტურულ-გრაფული წარმოდგენის შემუშავება, რომელიც საშუალებას იძლევა გამოცნობის პროცესი წარიმართოს , როგორც „ზევიდან ქვევით“ ასევე „ ქვევიდან ზევით“, რაც უზრუნველყოფს გამოცნობის ქვეამოცანების დაპარალელებას.
- არგუმენტაციის თეორიის ბაზაზე შემუშავებული იქნა დასკვნების კეთების ალგორითმი ობიექტების და მის შემადგენელი კომპონენტების ეტალონებთან შეთავსებისა და გამოცნობისთვის, რაც

ასევე უზრუნველყოფს გამოცნობას ობიექტების შესახებ ინფორმაციის არასრულობის შემთხვევაში.

დისერტაციის პრაქტიკული მნიშვნელობა: მიღებული შედეგები შეიძლება განზოგადდეს ადამიანის საქმიანობის სხვადასხვა სფეროზე, სადაც კოლექტიური გადაწყვეტილებები და დასკვნების კეთებაა საჭირო. შემუშავებული ალგორითმები შეიძლება გამოყენებული იყოს ადამიანის სახის გამოცნობაში, მოძრავი ობიექტების მიდევნებაში, სწავლების სისტემებში, ცოდნის წინააღმდეგობებზე ანალიზსა და ცნებების სახვადას სემანტიკის შემთხვევაში ერთ აზრზე დაყვანის ამოცანებში, აგენტების კომუნიკაციაში და სხვა.

დისერტაციის მოცულობა და სტრუქტურა: დისერტაცია შედგება 120(ასოცი) გვერდისგან და მოიცავს შემდეგ საკითხებს : ინტელექტუალური აგენტის თეორია, ობიექტების(სახეთა) ამოცნობა, ხელოვნური ნეირონული ქსელი, მრავალ აგენტიანი ამოცნობის არქიტექტურა, აგენტების მოდელი, არგუმენტაცია , სემანტიკური შეთავსების მოდელი, ევრისტიკული ძებნა , შავი დაფის არქიტექტურა და აგენტების მიერ სიმბოლოების ამოცნობის მაგალითს.

თავი 1 : ინტელექტუალური აგენტების თეორია

სიტყვა „აგენტი“ ლათინური წარმოშობისაა და ნიშნავს მოქმედებას. ხელოვნურ ინტელექტში გამოყენებული ტერმინი - აგენტი, შინაარსობრივად შეესაბამება ლათინურ ენაში არსებული - „აგენტის“ მნიშვნელობას. აქედან გამომდინარე, ხელოვნური ინტელექტის თეორიაში ტერმი „აგენტის“ ფორმალიზებული განსაზღვრისათვის გვაქვს:

აგენტი ეწოდება ნებისმიერ ობიექტს ან სისტემას რომელიც მოქმედებს. როგორც წესი, ხელოვნური ინტელექტის თეორიაში ან პრაქტიკაში ასეთ ობიექტს ან სისტემას შეიძლება წარმოადგენდეს კომპიუტერული პროგრამა, რომელსაც ჩვეულებრივ პროგრამებისგან განსხვავებისთვის ხასითდება მთელი რიგი თვისებებით:

1. შეუძლია ავტონომიურად ფუნქციონირება;
2. შეუძლია იმ გარემოს აღქმა რომელშიც მოქმედებს;
3. საკმაოდ დიდი ხნის განმავლობაში არსებობა;
4. აქვს ადაპტაციის უნარი;
5. შეუძლია სხვების მიერ დასახლუ მიზნების მიღწევა

აგენტის მოქმედება შეიძლება შეფასდეს მისი ფუნქციონირების შედეგების მიხედვით, კერძოდ იმ საშვალეებით, რომლებსაც აგენტი იყენებს დასახული მიზნის მისაღწევად. ცხადია, სხვადასხვა აგენტი განსხვავებულად მოქმედებს ერთიდაიგივე მიზნის მისაღწევად, კერძოდ, ზოგიერთი აგენტი მოქმედებს ცუდად, ზოგიერთი დამაკმაყოფილებლად, სხვები - კარგად. აქედან გამომდინარეობს რაციონალური აგენტის ცნების განსაზღვრის აუცილებლობა.

თავი 2 : ობიექტების(სახეთა) ამოცნობა

ობიექტების (სახეთა) ამოცნობა არის მანქანური სწავლების ცალკე განშტოება, რომელიც ყურადღებას ამახვილებს ნიმუშების და მონაცემებში მსგავსებების ამოცნობაში, თუმცა რიგ შემთხვევებში ის განიხილება მანქანური სწავლების თითქმის სრულ სინონიმად. ხშირ შემთხვევებში ობიექტების (სახეთა) ამოცნობის სისტემების გავარჯიშება ხდება მარკირებული „სავარჯიშო“ მონაცემებით (კონტროლირებადი სწავლება), თუმცა როცა მარკირებული მონაცემები არ მოიძიება სხვა ალგორითმების

გამოყენება არის შესაძლებელი აქამდე უცნობი ნიმუშების ამოსაცნობად (უკონტროლო სწავლება).[2, 4]

ტერმინების ნიმუშების ამოცნობის, მანქანური სწავლების, მონაცემის მოპოვების და ცოდნის აღმოსაჩენი მონაცემთა ბაზების (KDD) განსხვავება არის ძალიან რთული, რადგან მათ ფუნქციებში არის ძალიან დიდი თანაკვეთები. მანქანური სწავლება არის ხშირად ნახმარი ტერმინი კონტროლირებადი სწავლების მეთოდებისთვის, რომლის დასაბამი მოდის ხელოვნური ინტელექტისგან, როცა KDD-ს და მონაცემის მოპოვების უფრო ყურადღებას ამახვილებენ არაკონტროლირებად სწავლების მეთოდებზე და უფრო ხშირად გამოიყენება ბიზნესში. ობიექტების (სახეთა) ამოცნობის დასაბამი მოდის ინჟინერიიდან და ტერმინი ძალიან პოპულარულია კომპიუტერული ხედვის კონტექსტში: ამ მომენტისთვის წამყვან კომპიუტერული მხედველობის კონფერენციას ქვია „კონფერენცია კომპიუტერულ მხედველობაში და ობიექტების (სახეთა) ამოცნობაში“. ნიმუშების (ობიექტების) ამოცნობაში უფრო მეტი ყურადღება ეთმობა ნიმუშის ფორმალიზებას, ახსნას და ვიზუალიზაციას, რომლისგან განსხვავებით მანქანური სწავლება ტრადიციულად მეტ ყურადღებას უთმობს ამოცნობის პროცენტულობის გაზრდას. თუმცა, ყველა აღნიშნული დომენის მოცულობა გაიზარდა თავიანთ საწყის მიზნებთან მიმართებაში და არსებულ მომენტში მოიცავენ ერთამენთის ელემენტებს და იდეებს.

ობიექტების ამოცნობა

ადამიანების და ცხოველების ქმედებების მნიშვნელოვანი ნაწილი არის მათი შესაძლებლობა რათა ამოიცნონ ობიექტები, ცხოველები და ადამიანები, რომლებიც მნიშვნელოვანია მათი გადარჩენისთვის. ადამიანებს შეუძლიათ ამოიცნონ უამრავი სხვა ადამიანი ანბანის ასოები, ნაცნობი შენობები და ა.შ. ცხოველებს სჭირდებათ ტერიტორიების, შესაბამისი

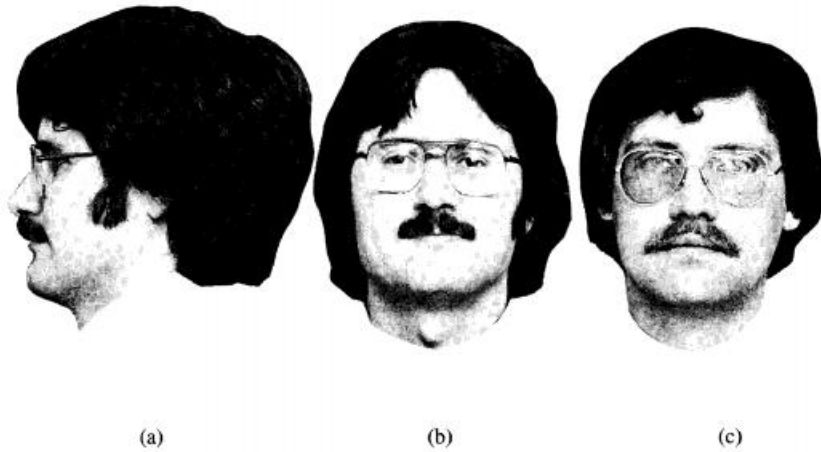
მსხვერპლის, პოტენციური მტაცელის ამოცნობა და მათან შესაბამისად მოქცევა.

თუ ჩვენ დავუშვებთ, რომ ადამიანისთვის ან ცხოველისთვის ხელმისაწვდომი ინფორმაცია არის ორ განზომილებიანი სურათი რეტინაზე, მაშინ უმაღლესი წარმოიქმნება პრობლემა მათი ვიზუალური ამოცნობის ახსნასთან დაკავშირებით. მაგალითისთვის ავიღოთ ადამიანის უნარი რომ ამიცნოს ანბანის ასო: პრობლემა მდგომარეობს იმაში რომ არსებობს უსასრულო რაოდენობა რეტინის სურათებისა, რომლებიც შეიძლება შეესაბამებოდეს კონკრეტულ ასოს. ეს დამოკიდებულია როგორ არის ასო დაწერილი, რა ზომის არის დაწერილი ასო, რომელი კუთხიდან ვუყურებთ და ა.შ სურ (4.1.) ამისადა მიუხედავად, ჩვენ ყველა ვხვდებით რომ ყველა ნიმუში შეესაბამება ასო A-ს.



სურ: 2.1 სურათზე ნაჩვენებია ასო A-ს სხვადასხვა ფორმები.

ასევე შეიძლება განვიხილოთ მეგობრის სახის ამოცნობის პრობლემა. სახის სურათი რეტინაზე დამოკიდებული იქნება განათების პირობებზე, მანძილზე, კუთხეზე, სახის გამომტყველებაზე და ა.შ. ამ შემთხვევაშიც ყველა ეს სურათი კლასიფიცირდება როგორც ერთი, იმისადა მიუხედავად რომ ისინი ერთმანეთისგან საკმაოდ განსხვავებულები შეიძლება იყოს (სურათი 4.2)



სურთი 2.2 სურათზე ნაჩვენებია ერთი და იგივე პიროვნება სხვადასხვა რაკურსით

ზემოთ აღწერილი ორივე მაგალითი არის მასტიმულირებელი ტოლფასობა პრობლემა. თუ სტიმული, რომელიც აკონტროლებს ქმედებას არის შუქის ან რეტინაზე სურათის ნიმუში, მაშინ არებობს უსასრული რაოდენობა სურათებისა, რომლებსაც გააჩნიათ ტოლფასი ეფექტი, მაგრამ ხვდებიან სურათების განსხვავებულ სიმრავლეში. ობიექტის ამოცნობის უამარავი გავლენიანი კვლევა უშვებს რომ ყველა სურათს რომელიც შეესაბამება კონკრეტულ ნივთს, იქნება ეს ანბანის ასო თუ სახე აქვთ რაღაც საერთო. პრობლემას წარმოადგენს იმის გარკვევა თუ რა არის ეს საერთო და როგორ უნდა მოხდეს მისი დადგენა. სწორედ ამ პრობლემაზე ვისაუბრებთ შემდეგ თავში.

ობიექტის გრაფად წარმოდგენა და ამოცნობის სხვადასხვა მიდგომები

მოცემულია გამოსაცნობი ობიექტის აღწერა უნდა მოიძებნოს გამოსაცნობ X ობიექტთან ყველაზე ახლო მდგომი პათერნი და შესაბამის კლასის ინდექსი:

$$i^* = \min_{i,j} \text{Dist}(X, P_{ij})$$

$$i,j$$

$$1 \leq i \leq n$$

$$1 \leq j \leq k$$

მანძილის გამოსათვლელად თითოეული პათერნი წარმოდგება მრავალშრიანი გრაფის სახით. თითოეულ შრეში თავსდება გრაფი რომელიც შედგენილია ასოცირების მიმართებით შედგენილი კლასების წვეროებისგან და თვისებების მნიშვნელობების წვეროებისგან. ხოლო ყოველ მომდევნო შრეში თავსდება მიმდინარე შრის კლასის წვეროების შემადგენელი კლასების წვეროებისგან, მათი თვისებათა მნიშვნელობების და მიმართებით დაკავშირებული კლასებისგან. ანალოგიურად ხდება გამოსაცნობი წვეროების წარმოდგენა.

გამოცნობის პროცესი ამ შემთხვევაში წარმოადგენს ერთი ერთი დონის შრის ობიექტის და პათერნის გრაფების შეთავსების ამოცანას, რომელიც ექვივალენტურია ორი გრაფის მაქსიმალური საერთო ქვეგრაფის მოძებნის ამოცანისა.

რთული ობიექტების გამოცნობისთვის არის შემდომი მიდგომები :

1. კლასიკური ქვევიდან -ზევით, როცა განისაზღვრება ჯერ ქვედა დონეზე ობიექტის შემადგენელი ნაწილების მათ პატერნებთან სიახლოვე და შემდგომ მშობელი ნაწილების სიახლოვე და ასე შემდეგ სანამ არ მივალწევთ საწყის ობიექტს .

$$M(x, Y) = \min_{t_{ij} \in \cup_{c_i \in Y} T_{c_i}} \left(\text{Dist} (g(x), g(t_{ij})) + \sum_{x_i \in S(x)} M(x_i, \sigma(c_i)) \right)$$

სადაც

$T_{c_i} = \{t_{i1}, t_{i2}, \dots, t_{ik}\}$ - არის ნიმუში c_i - კლასის,

$Dist(g(\), g(\))$ ფასდება როგორც ორი გრაფის შედარება და ეს ამოცანა არის NP-complexity

$\sigma(c_i) = \{c_{i1}, c_{i2}, \dots, c_{im}\}$ – კლასების სია c_i ,

$g(x)$ - გრაფი რომელიც შედგება თვისებებით და კავშირებით x
- ის

რეკურსიის დაწყების პირობა : $M(o, C)$,

სადაც $C = \{c_1, c_2, \dots, c_m\}$ - არის სია ნასწავლი კლასების

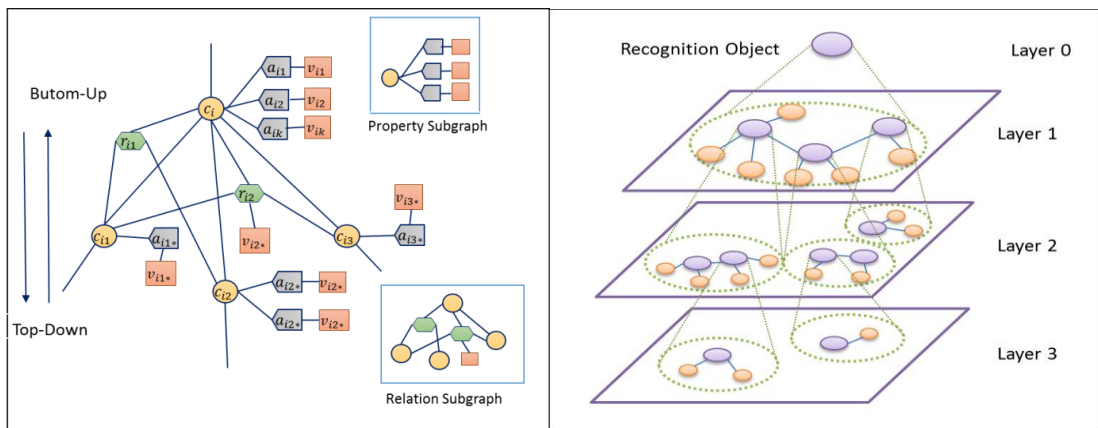
რეკურსიის დასრულების პირობა : $\sigma(c_i) = \emptyset$

$opt \{M(o, C)\} = M(o, \{c_i^*\})$ – ოპტიმალური მნიშვნელობა

რთული ობიექტის და განუზღვრელობების შემთხვევაში გადასაჭრელი ამოცანა ძალიან ბევრ გამოთვლით რესურსებს თხოულობს და პრაქტიკულად უმეტეს შემთხვევაში შეუძლებელია მისი გამოყენება.

2. მეორე მიდგომაა ზევიდან-ქვევით, როცა ზედა დონეზე გამოითქმება ჰიპოთეზა და შემდგომ ეს ჰიპოთეზა მოწმდება ობიექტის შემადგენელი ნაწილებზე, შემდგომ მათ შემადგენელ ნაწილებზე და ასე შემდეგ სულ მცირე ნაწილებამდე. ეს მეთოდი მოითხოვს აპრიორულ ცოდნას საწყისი ობიექტის შესახებ, რათა სწორედ შედგეს ჰიპოთეზა, მაგრამ რთული ობიექტებისთვის ამ აპრიორული ინფორმაციის მოპოვება როგორც წესი რთულია
3. კომბინირებული მიდგომა, როცა გამოიყენება ერთდროულად ორივე ზევით მოყვანილი მეთოდი. ეს მეთოდი ხასიათდება შემდეგი თვისებებით:
 - ქვევიდან-ზევით გამოცნობის მეთოდი უზრუნველყოფს მაღალ პარარელიზმს, ამიტომ შესაძლებელია მრავალ აგენტური მოდელის გამოყენება

- ზევიდან-ქვევით გამოცნობის მეთოდი მოითხოვს ცნებების სემანტიკასთან მუშაობას და დასკვნების მექანიზმის გამოყენებას. ამიტომ აგენტებს უნდა ჰქონდეთ ონთოლოგიაზე -დაფუძნებული არქიტექტურა, დასკვნების კეთების და სემანტიკური სიახლოვის განსაზღვრის მექანიზმებით



სურ. 2.3 ამოცნობის არქიტექტურა

თავი 3 : ხელოვნური ნეირონული ქსელი

რა არის ნეირონული ქსელი?

არსებობს ამოცანები რომლის ფორმულირება ვერ ხერხდება განსაზღვრული ალგორითმით. პრობლემები რომლებიც დამოკიდებულია მრავალ დახვეწილ ფაქტორზე მაგალითად უძრავი ქონების შესყიდვის ფასი რომელსაც ჩვენი ტვინი დაახლოებით გამოთვლის. გარე კომპიუტერულ ალგორითმს არ შეუძლია ამის გაკეთება. აქედან გამომდინარე საინტერესოა როგორ შევქმნათ ალგორითმი რომელიც მოახდენს მსგავსი პრობლემების გადაჭრას. [8]

ზუსტად - ჩვენ ვსწავლობთ და ეს შესაძლებლობა კომპიუტერს არ გააჩნია. ადამიანებს აქვთ ტვინი რომელსაც შეუძლია ისწავლოს.

კომპიუტერებს კი აქვთ გადამამუშავებელი ერთეულები და მეხსიერება. ჩვენ ვაძლევთ კომპიუტერს დავალებას შასრულოს დავალებები მაგალითად როგორცაა მათემატიკური კალკულაციები რასაც იგი ძაან სწრაფ დროში ახერხებს მაგრამ იგი არ არის ადაპტური.

თუ ჩვენ შევადარებთ კომპიუტერს და ადამიანის ტვინს, ჩვენ შევძლებთ ავლნიშნოთ რომ კომპიუტერი უფრო ძლიერია ვიდრე ჩვენი ტვინი. იგი მოიცავს 10^9 ტრანზისტორს რის გადართვასაც ახერხებს 10^{-9} წამში. ადამიანის ტვინი შედგება 10^{11} ნეირონებისგან და მათი გადართვის დრო დაახლოებით 10^{-3} წამია.

უდიდესი ნაწილი ტვინის მუშაობს მუდმივად მაშინ როდესაც უდიდესი ნაწილი კომპიუტერის არის პასიურ მდგომარეობაში. ამდენად ტვინი არის პარალელური და აქედან გამომდინარე იგი ასრულებს თავისი შესაძლებლობების მაქსიმუმს.

	გონება	კომპიუტერი
დამუშავების ერთეული	$\approx 10^{11}$	$\approx 10^9$
დამუშავების ერთეულის ტიპი	ნეირონები	ტრანსისტორები
მონაცემთა დამუშავების ტიპი	პარალელური	სერიალური
მონაცემების შენახვა	ასოციაციური	მისამართებით
გადართვის დრო	$\approx 10^{-3}$	$\approx 10^{-9}$
სავარაუდო გადართვის ოპერაციები	$\approx 10^{13}$	$\approx 10^{18}$

რეალური ოპერაციები	გადართვის	$\approx 10^{12}$	$\approx 10^{10}$
-----------------------	-----------	-------------------	-------------------

დამატებით კომპიუტერი არის სტატიკური - ხოლო ტვინი რომელიც შედგება ბიოლოგიური ნეირონული ქსელისგან შეუძლია ამოიციოს სიცოცხლის პროცესი აქედან გამომდინარე მას შეუძლია ისწავლოს შეცდომებზე და კიდევ ბევრი რამ. აღნიშნულით ვცდილობ გამოვხატო რამდენად შეგვიძია გამოვიყენოთ ტვინის მახასიათებლები კომპიუტერულ სისტემებში.

ასე რომ სწავლება ხელოვნური ნეირონული ქსელის არის მოტივირებული მსგავსებით თუ როგორ მუშაობს ბიოლოგიური სისტემა და მისი შედარება მთელ სისტემასთან - იგი შედგება ძალიან მარტივი მაგრამ მრავალრიცხოვანი ნერვული უჯრედებისგან რომლებიც მუშაობენ მასიურად და ამავდროულად პარალელურად (რაც ჩემის აზრით ერთერთი ძალიან მნიშვნელოვანი ასპექტია) და აქვთ შესაძლებლობა ისწავლონ. ერთერთი შედეგი სწავლების პროცედურისა არის შესაძლებლობა ნეირონული ქსელის რომ მოახდინოს განზოგადება და ასოცირება მონაცემების, მას შემდეგ რაც წარმატებული ტრენინგი ნეირონული ქსელის შეძლებს იპოვოს გონივრული გადაწყვეტა მსგავსი პრობლემების იგივე კლასში რომელშიც იგი არ იყო ზუსტად მომზადებული.

თავი 4 : მრავალ აგენტისანი ამოცნობის არქიტექტურა

სოციალური სამყაროს გაფართოვება ავტონომიური კომპიუტერული სისტემების დახმარებით ყოველთვის წარმოადგენდა საოცრებას, თუ საშიშს არა, პერსპექტიულს. თუმცა ეს უკვე ხდება შესაძლებელი და საჭირო

ხელოვნური ინტელექტის განვითარების კუთხით. წარსულ წლებში ხელოვნური ინტელექტი მნიშვნელოვნად განვითარდა, გახდა მძლავრი და კომპლექსური. სხვა დანარჩენ წარმატებებთან ერთად, შეგვიძლია მაგალითად მოვიყვანოთ თვითმართვადი ავტომობილი, რომელმაც ამერიკის შეერთებული შტატების ირგვლივ 95% ზე მეტი მანძილი ადამიანის ჩარევის გარეშე გაიარა ALVINN სისტემის საშუალებით. ამ და სხვა გამოწვევების საფუძველზე AI მკვლევარებმა მოიპოვეს უფლება ჩაეტარებინათ გამოკვლევები მრავალ ავტონომიურ სისტემებზე რომლებიც ურთიერთობენ ერთმანეთთან.[14] თუ არსებობს ერთი თვითმართვადი ავტომობილი, აუცილებლად იარსებებს მეტი, რომლებიც ივლიან ინდივიდუალურად და ჩვენთვის საჭიროა ვიცოდეთ თუ რას მოიმოქმედებენ ეს ავტომობილები სხვა ავტომობილებთან ესტაკადაზე შეხვედრის შემთხვევაში.

მრავალ აგენტის სისტემები ხელოვნური ინტელექტის ქვედარგია, რომლის მიზანია კომპლექსური სისტემების უზრუნველყოფა მრავალი აგენტებისა და ინდივიდუალური აგენტების კოორდინაციის მექანიზმების საშუალებით. რადგანაც ხელოვნურ ინტელექტში არ არსებობს აგენტის ზოგადი განსაზღვრება, შეგვიძლია ის წარმოვიდგინოთ როგორც რობოტი, მიზნებით, ქცევებით და გარემოს შესაბამისი ცოდნით. აგენტის (რობოტის) შესაძლებლობა მოახდინოს კოორდინაცია ამა თუ იმ სიტუაციაში, ახალი დარგია, რაც ნიშნავს ცოდნის სწრაფად გაღრმავებას წინამავალი დავალების შესრულების საფუძველზე განაწილებულ ხელოვნურ ინტელექტში (Distributed Artificial Intelligence DAI).[15]

DAI ხელოვნური ინტელექტის ქვედარგია ბოლო ორი ათწლეულის მანძილზე. ტრადიციულად ის დაყოფილია ორ ქვე-დისციპლინად: განაწილებული პრობლემის აღმოფხვრა (Distributed Problem Solving DPS) და მრავალაგენტის სისტემები (Multi Agent Systems MAS). მთავარი თემები DPS ში არის ინფორმაციის მართვის საკითხები, როგორცაა დავალების დეკომპოზიცია და გადაწყვეტის სინთეზი. მაგალითად, იძულებითი

კმაყოფილების პოზიციის დეკომპოზიცია ხშირად შესაძლებელია მრავალ სხვა ქვეპრობლემებად, რომელთა გადაწყვეტა შესაძლებელია სხვადასხვა პროცესორზე, შემდეგ კი ამ გადაწყვეტილებების სინთეზირება ერთ გადაწყვეტილებაში.

MAS საშუალებას აძლევს იძულებითი კმაყოფილების პრობლემას იყოს მონაწილე სხვადასხვა პრობლემების გადამჭრელ აგენტებთან ერთად, რომელთაც გააჩნიათ თავიანთი ინტერესები და მიზნები.

სტატია ორგანიზებულია მრავალი ზოგადი მრავალაგენტიანი სცენარებით. თითოეული სცენარისთვის აღწერილი საკითხი შეიცავს ტექნიკის ნიმუშს, რომელიც გამოიყენება მის გადასაწყვეტად. აღწერილი მიდგომა არ არის ძალზედ დამღლევი და ის გვაჩვენებს თუ როგორი შეიძლება იყოს მრავალაგენტიანი სისტემა და როგორ გამოვიყენოთ ის კომპლექსური სისტემების ასაგებად.

MAS ის კომპლექსურობიდან გამომდინარე, დიდ ინტერესს იწვევს Machine Learning ის მიდგომები ამ სირთულესთან გამკლავებაში. როცა არსებობს რამდენიმე სხვადასხვა სისტემა რომლებსაც შეუძლიათ ერთი და იგივე ან დაახლოებით ერთნაირი MAS მიდგომების ილუსტრირება, შეიძლება ითქვას რომ ეს სისტემები მიკერძოებულია Machine Learning - თან. მსგავსი გამოკვლევები არ არის ფოკუსირებული მხოლოდ რობოტულ სისტემებზე. თუმცა გვჯერა რომ მთავარი გამოკვლევები არა-რობოტულ MAS ში შეესაბამება რობოტულ MAS-ს.

თუმცა ბევრი შესაძლო გზაა დავყოთ MAS. კვლევა ორგანიზებულია ორ ძირითად განზომილებაში: აგენტის ჰეტეროგენურობასა და აგენტებს შორის კომუნიკაციების რაოდენობაში. აღწერთ მარტივ მულტიაგენტურ სცენარს, ჰომოგენური არაკომუნიკაბელური აგენტებით, შესაძლო მრავალაგენტიანი სისტემებით და მაღალი ჰეტეროგენული კომუნიკაბელური აგენტებით.[16]

სტატია ორგანიზებულია შემდეგნაირად: 2 სექცია MAS ის დარგს, მნიშვნელოვან მძლავრ პუნქტებს და წარმოადგენს ტაქსონომიას. სტატიის

ძირითადი ნაწილი, სექცია 3-7 აღწერს მრავალაგენტოვანი სისტემების სხვადასხვა სცენარებს ამ დარგში.

მრავალაგენტოვანი სისტემები

ორი აშკარა კითხვა ნებისმიერი ტიპის ტექნოლოგიების შესახებ არის:

- რა უპირატესობებს გვთავაზობს ის სხვა ალტერნატივებთან შედარებით ?
- რა სიტუაციებში არის ის სასარგებლო ?

უგუნური იქნებოდა მტკიცება, რომ MAS - ის გამოყენება იქნებოდა ყველაზე ოპტიმალური ნებისმიერი კომპლექსური სისტემის ასაგებად. როგორც ნებისმიერი საჭირო მიდგომა, არის სიტუაციები, როცა ის განსაკუთრებით შესაფერისია, ან პირიქით. ამ სექციის მიზანია ხაზი გავავლოთ MAS ის საჭიროებასა და უფრო მეტიც, აუცილებლობას ტიპურ სიტუაციებში.

ზოგ გაერთიანებას (domain) ჭირდება MAS. უფრო დეტალურად, თუ ამ გაერთიანებაში არიან სხვადასხვა ორგანიზაციები, სხვადასხვა ტიპისა და მიზნების მქონე ხალხი, საჭიროა მრავალაგენტოვანი სისტემა მათი ურთიერთობისათვის. იმ შემთხვევაშიც კი, თუ თითოეულ ორგანიზაციას სურს საკუთარი შინაგანი უშიშროების მოდელირება ერთი საერთო სისტემით, ორგანიზაციები არ მისცემენ ამ სისტემის შექმნის უფლებას მხოლოდ ერთ ადამიანს. სხვადასხვა ორგანიზაციებს ჭირდებათ საკუთარი სისტემა, რომელიც ყველაზე უკეთ აკმაყოფილებს მათ მოთხოვნებს. [13]

მაგალითისათვის წარმოვიდგინოთ კომპანია X, რომელიც აწარმოებს საბურავებს, მაგრამ ხელშეკრულებით თანამშრომლობს თხილის პროდუქციაზე კომპანია Y თან. ამ პროდუქციის პროცესის ერთი სისტემაში მოსაყვანად, უნდა მოხდეს X-სა და Y-ის შიდა საქმეების მოდელირება. თუმცა არცერთ კომპანიას არ აწყობს საკუთარი ინფორმაციის ან კონტროლის სხვა კომპანიისათვის გასაჯაროება ან გადაცემა. ორი

კომპანიის შემთხვევაში შესაძლოა მიიღწიოს შეთანხმებას, მაგრამ მრავალი კომპანიის შემთხვევაში MAS მნიშვნელოვანია.

სხვა მაგალითი შეიძლება იყოს საავადმყოფოს განრიგის შედგენა. ეს გარემო მოითხოვს სხვადასხვა აგენტებს რომ წარმოადგინოს სხვადასხვა ადამიანების ინტერესები საავადმყოფოში (სხვადასხვა გამოცდილებისა და პროფესიის ექიმები, რომლებსაც გააჩნიათ განსხვავებული განრიგი, ექთნები, ასევე ოპერატორები და ა.შ). რადგანაც ამ ადამიანებს გააჩნიათ სხვადასხვა განრიგი, სხვადასხვა კრიტერიუმების გათვალისწინებით, მათი წარმოდგენა საჭიროა სხვადასხვა აგენტებით.

იმ გარემოებებშიც, სადაც არ გამოიყენება დისტრიბუციული სისტემები, ზოგჯერ საჭიროა MAS. მრავალი აგენტის არსებობამ შესაძლებელია ოპტიმიზაცია გაუკეთოს სისტემის წარმადობას, პარალელური გამოთვლის მეთოდის წარმოებით. MAS ის პარალელიზმი შესაძლოა დაეხმაროს დროით შეზღუდული ან სივრცით შეზღუდული მოთხოვნების დაკმაყოფილებას.

გარდა იმისა, რომ დავალებების სხვადასხვა აგენტებზე განაწილებით მიიღწევა პარალელიზმი, სიმძლავრე არის MAS ის უპირატესობა, რომელსაც ყავს ზედმეტი აგენტები.

თუ კონტროლი და შესაძლებლობები თანაბრადაა გადანაწილებული სხვადასხვა აგენტებზე, სისტემის მუშაობა ბევრად სტაბილური და ეფექტურია. თუ ერთი აგენტი (პროცესორი) მართავს ყველაფერს, ერთი უმნიშვნელო შეცდომა ხშირ შემთხვევაში განაპირობებს სისტემის ავარიულად დასრულებას. თუმცა არ არის აუცილებელი მრავალაგენტიანი სისტემა განაწილებული იყოს სხვადასხვა პროცესორებზე, სისტემის ავარიული მარცხის თავის ასარიდებლად აგენტები უნდა იყვნენ გადანაწილებული სხვადასხვა მანქანებზე.

მრავალაგენტიანი სისტემების კიდევ ერთი უპირატესობაა მასშტაბურობა (scalability). ისინი არსებითად მოდულურები არიან და კარგ მოდელირებულ სისტემაში ადვილია ახალი აგენტების დამატება რათა

სისტემა გაფართოვდეს ახალი შესაძლებლობებით. ასევე ეს უპირატესობა მოიცავს არსებული მოდულების შესაძლებლობებისა და კონტროლის ცვლილების სიმარტივესა და მოქნილობას.

პროგრამისტის თვალსაზრისით, მოდულურობას მივყავართ მარტივ პროგრამირებასთან. ერთი რთული ამოცანის ცენტრალიზებულ აგენტზე დავალების ნაცვლად, მას შეუძლია განსაზღვროს ქვე-ამოცანები და დაუწესოს კონტროლი სხვადასხვა აგენტებს ამ დავალებების შესასრულებლად.

მრავალაგენტიანი სისტემა, ინტელექტთან ერთად ასევე შეიძლება გამოყენებულ იქნას სოციალური და ცხოვრებისეული მეცნიერების ფუნდამენტური პრობლემის გადასაჭრელად. ინტელექტი ღრმად დაკავშირებულია ურთიერთქმედებასთან. ითვლება რომ ყველაზე საუკეთესო გზის დასაწყისი ინტელექტუალური მანქანების შესაქმნელად არის სოციალური მანქანების შექმნა. ეს თეორია ეფუძნება სოციალურ-ბიოლოგიურ თეორიას რომელიც მიიჩნევს რომ ხელოვნური ინტელექტი სწორედ სოციალური ურთიერთობის საფუძველზე შეიქმნა.

გარდა იმისა, რომ MAS თავსებადია ზოგადად ყველა ყველა ზემოთ ხსენებულ პრინციპთან, არსებობს რამდენიმე არგუმენტი მულტი-რობოტულ სისტემებში. დავალებებში, რომლებიც ითხოვენ რობოტების განლაგებას კონკრეტულ ადგილებში, უპირატეობა აქვთ რობოტების ჯგუფს ერთ რობოტთან შედარებით, რადგან შესაძლებელია მათი გადანაწილება სხვადასხვა გეოგრაფიულ ადგილებში.

საბოლოო ჯამში, როგორც ვიმსჯელებთ, მრავალ-რობოტიანი სისტემა, რომელიც შედგება შედარებით იაფფასიანი, შეზღუდული შესაძლებლობების, მაგრამ კონკრეტული დავალების შესრულების უნარის მქონე რობოტებისაგან, მეტი უპირატესობით სარგებლობს, ვიდრე ერთ-რობოტიანი სისტემა, მძლავრი და ყოვლისშემძლე რობოტით.

ქვემოთ ჩამოთვლილია მიზეზები, თუ რატომ უნდა გამოვიყენოთ MAS :

მრავალაგენტოანი სისტემების გამოყენების მიზეზები

- რაღაც გარემოებები ითხოვენ მას
- პარალელიზმი
- სიმპლავრე
- მასშტაბურობა
- მარტივი პროგრამირება
- ინტელექტუალურობის შესასწავლად
- გეოგრაფიული გადანაწილება
- ფასის ეფექტურობა
-

ერთ აგენტოანი და მრავალაგენტოანი სისტემა

მრავალაგენტოანი სისტემების შესწავლამდე, უნდა გავითვალისწინოთ მათი სხვა ნათელი ალტერნატივა: ცენტრალიზებული ერთაგენტოანი სისტემა. ცენტრალიზებულ სისტემა შედგება ერთი აგენტისაგან, რომელიც ყველა გადაწყვეტილებას თავად იღებს, ხოლო სხვა დანარჩენები მოიაზრება როგორც მეორადი.

შესაძლებელია ერთაგენტოანი სისტემები შედგებოდეს რამდენიმე ობიექტისაგან - სხვადასხვა მოქმედი, ცალკეული კომპონენტებისაგან. ერთ-აგენტოანი სისტემაში თითოეული ობიექტი იღებს დავალებებს და მოქმედებების ინსტრუქციებს ერთი ცენტრალური პროცესიდან. ცენტრალიზებული პროცესი ახდენს ყველა ამ ობიექტების მოდელირებას. ამ სექციაში შევადარებთ ერთ-აგენტოანი და მრავალ-აგენტოანი სისტემებს.

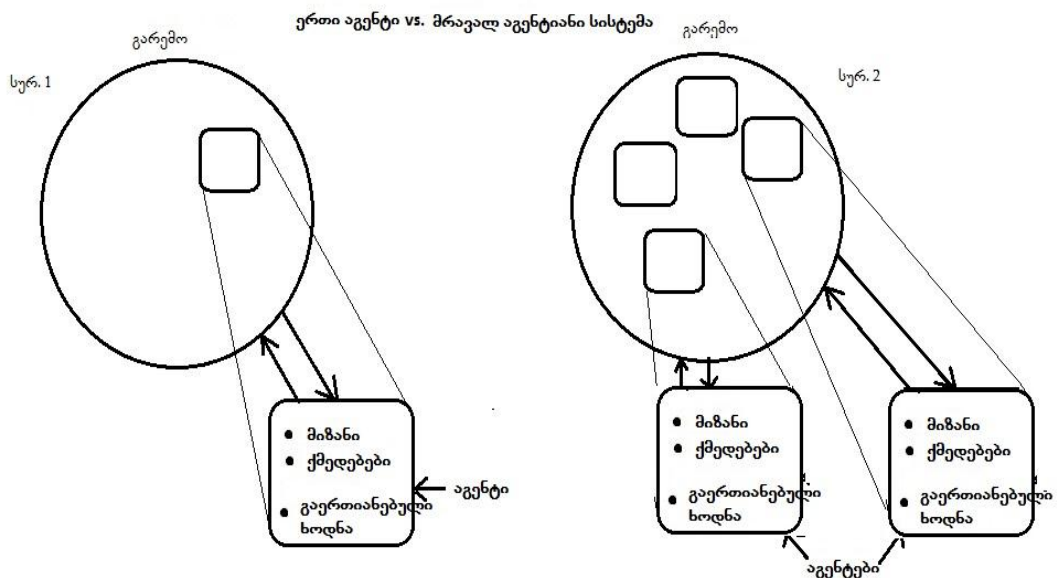
ერთ-აგენტოანი სისტემა

ზოგადად, აგენტი, ერთ-აგენტოანი სისტემაში ახდენს საკუთარი თავის, გარემოსა და ქმედებების მოდელირებას. რა თქმა უნდა აგენტი თავადვე არის გარემოს ნაწილი, მაგრამ დამატებითი გარემოს შესაბამისი კომპონენტებით. ისინი არიან დამოუკიდებელი ობიექტები, საკუთარი

მიზნებით, ქცევებით და ცოდნით. ერთ-აგენტიან სისტემაში აგენტი არ სცნობს სხვა ასეთ ობიექტებს. სხვა დანარჩენები არ მოიაზრება როგორც მიზნებისა ქცევებისა და ცოდნის მქონე.

მრავალაგენტიანი სისტემები

განსხვავება მულტიაგენტურ და ერთაგენტიან სისტემებს შორის არის ის, რომ მულტიაგენტურ სისტემაში არსებობს მრავალი აგენტი, რომლებიც ახდენენ ერთმანეთის მიზნებისა და ქცევების მოდელირებას. ზოგად მულტიაგენტურ სცენარში, შეიძლება შევხვდეთ პირდაპირ ურთიერთქმედებას (კომუნიკაციას) აგენტებს შორის. მიუხედავად იმისა რომ შესაძლებელია ამ ურთიერთქმედებების წარმოდგენა როგორც გარემოს შესაბამისი სტიმულები, ჩვენ ამ კომუნიკაციას განვაზოგადებთ გარემოსაგან. თითოეული აგენტისათვის, მრავალაგენტიანი სისტემები ერთაგენტიანიდან იმით განსხვავდება, რომ გარემოს დინამიკაზე ზემოქმედების მოხდენა სხვა აგენტებსაც შეუძლიათ.



სურ.4.1 ერთ აგენტიანი და მრავალ აგენტიანი სისტემა

სურ.4.1: ზოგადი, ერთ-აგენტიანი სტრუქტურა. აგენტი ახდენს საკუთარი თავის, გარემოსა და მათი ურთიერთქმედების მოდელირებას. სხვა

აგენტების არსებობის შემთხვევაში, ისინი მოისაზრებიან როგორც გარემოს ნაწილები.

სურ.2 ახდენს სცენარის ილუსტრირებას, სადაც თითოეული აგენტი არის როგორც გარემოს ნაწილი, ასევე მოდელირებული სხვა ობიექტების მიერ. აქ შესაძლებელია იყოს აგენტების სხვადასხვა რაოდენობა, სხვადასხვა ხარისხისა და ჰეტეროგენური მახასიათებლების, კომუნიკაბელური / არაკომუნიკაბელური აგენტები.

სურ.2: სრულიად ზოგადი მრავალაგენტიანი სცენარი. აგენტები ახდენენ ერთმანეთის მიზნების, ქცევებისა და ცოდნის მოდელირებას. ასევე, როგორც ნახაზზე, ისრებით არის წარმოდგენილი, მათ შეუძლიათ პირდაპირ ერთმანეთთან კომუნიკაცია.

თავი 5 : არგუმენტაცია

არგუმენტებით სწავლება ერთ-ერთი ძირითადი მეთოდია ხელუვნურ ინტელექტში, არგუმენტების ლოგიკაზე დაფუძნებული მოდელი უმთავრესია კომფლიქტების მოგვარების, მიზნის მიღწევის და გადაწყვეტილების მიღებისთვის. არსებითი დიალექტიკური ხასიათი ამ მოდელისა გადმოგვცეს დამფუძნებლებმა დიალოგზე დაფუძნებული არგუმენტების ფორმალიზაციისთვის, მაგალითად როცა აგენტი ცდილობს დააარწმუნოს რწმენაში სხვა პერსონა ის უკვე დარწმუნებულია ჭეშმარიტებაში ან როცა აგენტი მიზანმიმართულია რაღაცის გაკეთებისკენ ის აწარმოებს მოლაპარაკებებს, ბოლოდროინდელი კვლევითი პროექტი ავითარებს ზოგად მოდელს არგუმენტაციისას, რომელსაც დაფუძნებულია დასკვნასა და გადაწყვეტილებაზე[3,12]

კლასიკური არგუმენტაცია

არგუმენტაციის ფრეიმვორკი არის წყვილი $AF = (AR, attacks)$, სადაც AR არის სია არგუმენტების ხოლო attacks არის ორობითი კავშირი AR-თან. შეტევა (A, B) ნიშნავს რომ არგუმენტი A უტევს არგუმენტ B. სია

არგუმენტების S უტევს არგუმენტ B თუ არგუმენტ B -ს უტევს ერთერთი არგუმენტი S სიიდან. ტავარი შეკითხვა არის მისაღებია თუ არა არგუმენტი A , $A \in AR$, ამ შემთხვევაში გონირული პასუხი არის , რომ არგუმენტი უნდა იყოს მიღებული მხოლოდ იმ შემთხვევაში, ყოველი თავდასხმა მასზე არის მისაღები არგუმენტი. (Dung, 1995) [11]

არგუმენტაცია თეორიის მნიშვნელოვანი სფეროში ხელოვნური ინტელექტი. მრავალ აგენტთან გარემოში სადაც აგენტები ცდილობენ მიაღწიონ საკუთარ მიზანს , თანამშრომლობა მათ შორის არ არის გარანტირებული. რომ მივალწიოთ შეთანხმებას და მოლაპარაკებას თვით მოტივირებულ აგენტებს შორის ჩვენ უნდა გამოვიყენოთ არგუმენტაციის ფრამევორკი. მიზანი არგუმენტაციის ფრამევორკის არის გადაწყვიტოს კონფლიქტები აგენტებს გადაწყვეტილებებს შორის. არგუმენტაციის ფრამევორკს დიდი გავლენა აქვს მრავალ აგენტთან გარემოში.

გაფართოებული არგუმენტაცია

ჩვენ შეგვიძლია გავაფართოვოთ არგუმენტაციის ფრამევორკი და მას დავარქვათ მნიშვნელობებზე დამოკიდებული არგუმენტაციის ფრამევორკი(VAF). VAF საშუალებას გვაძლევს განვსაზღვროთ მისაღებია თუ არა ესა თუ ის არგუმენტი რომელიც წარმოდგენილია სხვადასხვა აუდიენციის აგენტების მიერ. ჩვენს აგენტებს გააჩნიათ სხვადასხვა მიდგომები რათა მოახდინონ თავიანთი შედეგების(არგუმენტების) გაცვლა.

VAF შეუძლია გამოყოს თავდასხმები თუ რომელია მათ შორის წარმატებული, რომელი არგუმენტებით მოხდა სხვა არგუმენტებზე თავდასხმა. ის საშუალებას აძლევს სხვადასხვა აუდიენციის სხვადასხვა აგენტების ინტერესების გამომჟღავნებას.

მნიშვნელობებზე დამოკიდებული არგუმენტაცია (VAF) შეგვიძლია წარმოვადგინოთ როგორც $VAF = (AR, attacks, V, val, P)$, სადაც $(AR, attacks)$ არის არგუმენტაციის ფრამევორკი, V არის არა ცარიელი სია მნიშვნელობების ხოლო val არის ფუნქცია. არგუმენტი შეგვიძლია ჩავთვალოთ მიღებულად თუ $A \in AR, val(A) \in V$.

თავი 6 : სემანტიკური შეთავსების მოდელი

შეთავსები მოდელის ასაგებად დავუშავთ, რომ ობიექტი ან ცნება წარმოდგება სასრული რაოდენობის ფენების სახით. ნულოვან-საწყის ფენაში მოთავსებულია თვითონ ობიექტი (ცნება) თავისი ატრიბუტებით (თვისებებით და მიმართებებით), ხოლო ყოველ შემდეგ ფენაში თავსდება წინა ფენის ობიექტის შემადგენელი ნაწილები-ქვეობიექტები (ცნების შემადგენელი ქვეცნებები) თავიანთი ატრიბუტებით.

შემოვიტანოთ შეთავსების ფუნქცია

$\varphi: g(O) \times g(T) \rightarrow R$, რომელიც განსაზღვრავს გამოსაცნობი ობიექტებისა და ცნებების პატერნების სიახლოვეს . აქ $g: X \rightarrow \{G_1, G_2, \dots\}$ არის ოპერატორი რომელიც გამოსაცნობ ობიექტს ან კლასის პატერნს ასახავს მათი აღმწერი გრაფების სიმრავლეში. განვსაზღვროთ φ ფუნქციის მნიშვნელობების მიღების პირობები თვისებების და მიმალტებების შემთხვევაში:

1. თვისებების შეთავსება :

$$\varphi(g^p(o), g^p(t)) \geq 0 \text{ თუ}$$

$f_p(g(t)) \subseteq f_p(g(o)) \& \forall p \{ p \in f_p(g(t)) \& (f_v(p_o) \subseteq f_v(p_t)) \}$ ობიექტის თვისებების სიმრავლე სრულად მოიცავს პატერნის შესაბამის სიმრავლეებს და პატერნის თითოეული თვისების მნიშვნელობებისთვის სრულდება კონკრეტული პრედიკატებით განსაზღვრული პირობა ობიექტის შესაბამის მნიშვნელობებთან მიმართებაში.

მაგ: თუ პატერნისთვის განსაზღვრულია თვისება „სიგრძე < 10“ და ობიექტს გააჩნია ეს თვისება მაშინ შეთავსება , განხორციელდება თუ თვისების მნიშვნელობა ნაკლები ინება 10-ზე , მაგ. 5.

ამ შემთხვევაში გვაქს სრული განსაზღვრულობა (*certainty*).

2. მიმართებების შეთავსება :

$$\varphi(g^r(o), g^r(t)) \geq 0 \text{ თუ}$$

$$f_r(g(t)) \subseteq f_r(g(o)) \& \forall r \{ if \exists p((r, p) \in g(t)) \& (f_v(p_o) \subseteq f_v(p_t)) \} \& \text{Matches Certainty } (\sigma(t), \sigma(o)) ,$$

სადაც σ -ოპერატორია, რომელიც ნებისმიერ ობიექტს უკვშირებს მისი ქვეობიექტების სიმრავლეს. ობიექტების მიმართებები უთავსდებიან პათერნის მიმართებებს თუ პათერნის ქვეობიექტების სიმრავლესრული განსაზღვრულობით უთავსდება ობიექტის ქვეობიექტების სიმრავლეს, პათერნის მიმართებების სიმრავლეს სრულად ფარავს ობიექტების მიმართებების სიმრავლე და პათერნის ნებისმიერი მიმართების თვისების მნიშვნელობის დაკმაყოფილება ხდება თუ მას თვისება გააჩნია.

3. $\varphi(g(o), g(t)) < 0$ სხვა შემთხვევაში, როცა პატერნი ნაწილობრივ უთავსდება ობიექტს ან საერთოდ არ უთავსდება და ამ შემთხვევაში ვამბობთ რომ გვაქს განუზღვრელობა (*uncertainty*).

ვამბობთ რომ o ობიექტი უფრო ახლოა t_i პატერნთან ვიდრე t_j -თან, თუ $|\varphi(o, t_i)| < |\varphi(o, t_j)|$.

შემოვიტანოთ ოპერატორი g_l^p, g_l^r , რომელიც შესაბამისად ობიექტის l -ფენის თვისებების და მიმართებების ქვეგრაფებს აღწერს. მაშინ, ობიექტის პატერნთან სიახლოვე შეიძლება გამოითავალოს შემდეგი ფორმულით:

$$\varphi(g(o), g(t_i)) = \sum_{l=1}^k [\alpha_l^p * \varphi(g_l^p(o), g_l^p(t_i)) + \alpha_l^r * \varphi(g_l^r(o), g_l^r(t_i))]$$

სადაც α_l^p, α_l^r - წონითი კოეფიციენტებია.

რადგანაც $g_l^p(.)$ ქვეგრაფი ბუჩქს წარმოადგენს, ხოლო $g_l^r(.)$ - ორდონიან ხეს (იხილეთ ნახ. ?), ამიტომ მათი შეთავსების ოპერატორების აგება შედარებით გამარტივებულია.

თავი 7 : ევრისტიკული ძებნა

კომპიუტერულ მეცნიერებაში, ხელოვნურ ინტელექტში და მათემატიკურ ოპტიმიზაციაში ევრისტიკული მიდგომა არის შექმნილი რომ მოხდეს პრობლემის გადაწყვეტა უფრო სწრაფად როდესაც სტანდარტული მეთოდები არის უფრო ნელი, ან ვიპოვოთ ამოცანის მიახლოებული გადაწყვეტაროდესაც ცლასიკური მეთოდი ვერ ახერხებს ზუსტი ამონახსნის პოვნას.

- ხელოვნურ ინტელექტში ევრისტიკულ გადაწყვეტას იყენებენ ორ სიტუაციაში
 1. ამოცანას არ აქვს ზუსტი გადაწყვეტა. ამოცანა პირობაში არის ორაზროვნება ან ხელმისაწვდომი მონაცემების პრობლემა.
 2. ამოცანას აქვს ზუსტი გადაწყვეტა მაგრამ გამოთვლის დრო რაც ამოცანას სჭირდება არის მიუღებელი.
- ევრისტიკულ მიდგომა არის პრობლემის ამოხსნაში შემდგომი ნაბიჯის გააზრებული (ინფორმაციაზე დაფუძნებული) გამოცნობა.
- ევრისტიკული მიდგომა არის დაფუძნებული გამოცდილებაზე და ინტუიციაზე. სხვა სიტყვებით იგი მომავლის დაფუძნებულია ცოდნაზე და მიმდინარე ვითარებაზე.
- ევრისტიკული მიგომა ზოგჯერ იწვევს არაეფექტურ გადაწყვეტას და ვერ პოულობს გამოსავალს სიტუაციიდან.
-

A* ძებნა

რა არის A* ძებნა?

კომპიუტერულ მეცნიერებაში A* არის კომპიუტერული ალგორითმი რომელიც ფართოდ გამოიყენება გზის საპოვნად და გრაფის შემოვლისთვის. ამ ძებნის მეთოდის მიზანია ევრისტიკული ფუნქციის გამოყენებით შეამციროს ძებნის(მიზნის მიღწევის) ღირებულება. ხარბი ძებნა მნიშვნელოვნად ამცირებს ძებნის დროს მაგრამ იგი არ არის ოპტიმალური და და სრული. [22][23]

შედარებისთვის ერთიანი ძებნის ღირებულება ამცირებს გზის ღირებულებას $g(n)$,

მაგრამ ერთიანი ძებნა არის ოპტიმალური და სრული და ასევე შეიძლება იყოს ძან არაეფექტური.

$$f(n) = g(n) + h(n)$$

ჩვენ შეგვიძლია განვახორციელოდ A* ძებნა შემდეგნაირად :

სტანდარტული A* ალგორითმის პსევდო კოდი გამოიყურება შემდეგნაირად [33]:

```
function A*(start,goal)
  closedset := the empty set //სია კვანძებისა რომლებიც უკვე შეასებულია
  openset := {start} //სია კვანძებისა რომლებიც ჯერ არ შეფასებულა , საწყის
  ეტაპზე არის საწყისი კვანძი
  came_from := the empty map //კვანძი საიდანაც მოვედით.

  g_score[start] := 0 //ღირებულება საწყისი მდებარეობისა.
  f_score[start] := g_score[start] + heuristic_cost_estimate(start, goal)
  while openset is not empty
  current := the node in openset having the lowest f_score[] value
    if current = goal
      return reconstruct_path(came_from, goal)
    remove current from openset
    add current to closedset
    for each neighbor in neighbor_nodes(current)
      if neighbor in closedset
        continue
      tentative_g_score := g_score[current] + dist_between(current,neighbor)
      if neighbor not in openset or tentative_g_score < g_score[neighbor]
        came_from[neighbor] := current
        g_score[neighbor] := tentative_g_score
        f_score[neighbor] := g_score[neighbor] +
        heuristic_cost_estimate(neighbor, goal)
      if neighbor not in openset
```

```

        add neighbor to openset
    return failure
function reconstruct_path(came_from,current)
    total_path := [current]
    while current in came_from:
        current := came_from[current]
        total_path.append(current)
    return total_path

```

სადაც $g(n)$ არის გზის ღირებულება საწყისი კვანძიდან საბოლოო კვანძამდე ხოლო $h(n)$ არის სავარაუდო იაფი ღირებულება მიზნამდე.

თავი 8 : შავი დაფის არქიტექტურა

ავტომატური სამუშაო პროცესი თამაშობს მნიშვნელოვანი როლის თანამედროვე ბიზნესში. ერთი ვარიანტი სამუშაო პროცესის განვითარების არის მიღებული იქნას სამუშაოს მართვის სისტემა. თუმცა, ასეთი სისტემები ხშირად მოუქნელია და მისი გაფართოება და ინტეგრაცია არსებულ სისტემასთან საკმაოდ რთულია პროცესია. ამიტომაც არსებულის სისტემის ცვლება საყოველთაოდ არჩეული ალტერნატივაა. ამ სიტუაციაში, ჩვენ გთავაზობთ რომ მეტაფორა და მითითებითი არქიტექტურა იქნებოდა გამოსადეგი. ჩვენი მუშაობა აჩვენებს, თუ როგორ შეიძლება შავი დაფის არქიტექტურა გამოყენებულ იქნას ასეთი სამუშაო არქიტექტურისთვის.[17]

შავი დაფის არქიტექტურა არის რთული ამოცანების გადაჭრის სტრატეგია სხვადასხვა ცოდნის წყაროების გამოყენებით. შავი დაფა არის გლობალური წვდომადი მონაცემთა ბაზა რომელიც გამოიყენება შუალედურად, ამოცანის გადაჭრის ნაწილობრივ შედეგებზე. შავი დაფის არქიტექტურა შეგვიძლია წარმოვიდგინოთ როგორც ჯგუფი სპეციალისტების რომლებიც სხედან შავი დაფის წინ. ისინი მუშაობენ ერთობლივად იმისთვის რათა მოახდინონ ამოცანის გადაჭრა და ამისთვის

ისინი იყენებენ შავ დაფას როგორც სამუშაო ადგილს ამოცანის გადაწყვეტისთვის.

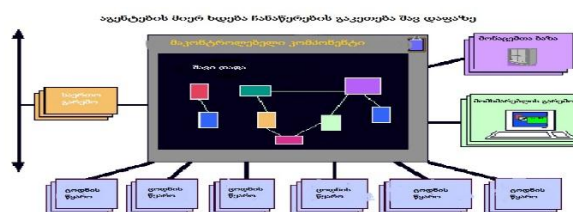
პრობლემის გადაჭრა იწყება მისივე გამოცხადებით და ხდება თავდაპირველი მონაცემების დაწერა შავ დაფაზე. სპეციალისტები უყურებენ შავ დაფას და ეძებენ შესაძლებლობას რომ წვლილი შეიტანონ ამოცანის განვითარებაში. როდესაც სპეციალისტი იპოვის შესაძლებლობას ის აკეთებს ჩანაწერს შავ დაფაზე და იმედოვნებს რომ ისინი გამოიყენებენ მის წვლილს რათა იპოვონ ამოცანის გადაჭრის გზა. ეს პროცესი გრძელდება მანამ სანამ არ მოხდება ამოცანის გადაწყვეტა.

შავი დაფის არქიტექტურა შედგება სამი კომპონენტისგან

1. ცოდნის წყაროები
2. შავი დაფა
3. მაკონტროლებელი კომპონენტი

ცოდნის წყაროები არიან დამოუკიდებელი მოდულები რომლებიც შეიცავენ საჭირო ცოდნას ამოცანის გადაწყვეტისთვის. მათ არ სჭირდებათ იცოდნენ არსებობა სხვა ცოდნის წყაროების, მაგრამ მათ უნდა იცოდნენ თუ რა მდგომარეობაშია ამოცანის გადაჭრის პროცესი და ინფორმაცია რომელიც წარმოდგენილია შავ დაფაზე.[21]

ცოდნის წყაროები შესაძლებელია წარმოდგენილი იყოს სხვადასხვა სახეობის ცოდნით, ისინი შეიძლება მოიცავდნენ წესებზე დაფუძნებულ სისტემას , შემთხვევებზე დაფუძნებულ სისტემას, გენეტიკურ ალგორითმებს. სურათზე მოყვანილია შავი დაფის მაგალითი.



სურ. 8.1 აგენტების მუშაობა შავ დაფაზე

შავი დაფა გამოიყენება როგორც გლობალური მონაცემთა ბაზა რომელიც აზიარებს სხვადასხვა ინფორმაციას როგორც შემომავალ ინფორმაციას ასევე ამოცანის ნაწილობრივ გადაწყვეტას ალტერნატივას და საბოლოო გადაწყვეტას. შავი დაფის აპლიკაციებს როგორც წესი აქვთ რთული სტრუქტურა მრავალი დონე ანალიზის და აბსტრაქცია.

მაკონტროლებელი კომპონენტი იღებს გადაწყვეტილებას ამოცანის მუშაობის პროცესში, თუ რომელი ცოდნის წყაროს გამოყენება მოხდეს რათა ნაპოვნი იქნას ამოცანის ოპტიმალური ამოხსნის გზა.

შავი დაფის არქიტექტურის უპირატესობები

შავი დაფის არქიტექტურით ამოცანის გადაჭრას გააჩნია შესაბამისი უპირატესობები რაც გვეხმარება ამოცანის ამონახსნის პოვნაში

1. მაკონტროლებელი სისტემის მიერ ხდება ცოდნის წყაროების ინტეგრაცია სისტემაში
2. მოდულირება - თუთოეული ცოდნის წყარო არის დამოუკიდებელი კომპონენტი რაც ქმნის ამოცანის შემდგომ განცითარებას და შენარჩუნებას უფრო ადვილს.
3. მოქნილობა - შავი დაფის არქიტექტურა საშუალებას აძლევს შავი დაფის აპლიკაციებს მოერგონ ცვლილებებს უფრო მოქნილას ვიდრე ტრადიციული პროგრამული აპლიკაციები.
4. გაფართოება - ახალი ცოდნის წყაროები შეიძლება იქნან შემუშავებული და გამოყენებული სისტემაში ისე რომ არ მოხდეს ცვილება არსებული სიტემის და არ მოხდეს მისი სპეციფიკაციის განსაზღვრა არსებულ ცოდნის წყაროებთან.
5. ეფექტურობა და ხარისხი- მრავალ ცოდნის წყაროს აქვს საშუალება შეასრულოს შესაბამისი ფუნქციონალი, ხოლო მაკონტროლებელ

კომპონენტ შეუძლია არჩევანი გააკეთოს ცოდნის წყაროებს შორის რომელიც უფრო სასარგებლო იქნება ამოცანის ამოხსნისთვის რათა მოხდეს საუკეთესო გადაწყვეტილების მიღება. ეს ყველაფერი კი აისახება როგორც ამოცანის გადაჭრის ეფექტურობაზე ასევე მისის გადაწყვეტის ხარისხზე.

6. თანამშრომლობა - შავი დაფის არქიტექტურაში თანამშრომლობა მთავარი მომენტია , ცოდნის წყაროებს შეუძლიათ ასახონ შავ დაფაზე ამოცანის ნაილობრივი გადაწყვეტა იმის იმედად რო სხვა ცოდნის წყაროები მოახდენენ ამოცანის სრულ გადაჭრას.
7. ცოდნის წყაროები და შავი დაფა ერთმანეთისგან დამოუკიდებელი კომპონენტებია და შესაძლებელია მათი გამოყენება ცალ ცალკე სხვა განსხვავებული ამოცანის გადასაჭრელად.

დასკვნა

თემაში განხილული იყო ობიექტების(სახეთა) გამოცნობის მრავალ აგენტური მიდგომები. განმარტებულია თუ რა მიდგომები არსებობს მრავალ აგენტურ სისტემებში. განმარტებულია მისი მარტივი მექანიზმები და ასევე უფრო კომპლექსური მიდგომები, აღწერილია თუ როგორ შეიძლება ობიექტების წარმოდგენა და როგორ უნდა მოხდეს მათი გარჩევა. თემაში განხილულია თუ როგორ არის შესაძლებელი ობიექტების სტრუქტურულად დალაგება. აღწერილია უპირატესობები მრავალ აგენტური მიდგომების კლასიკურ მეთოდებთან შედარებით, მკაფიოდ არის გამოხატული თუ როგორ უნდა მოხდეს აგენტების ურთიერთობა რათა მათი კომუნიკაცია იყოს წარმატებული. თემაში განხილულია თუ როგორ ახდენენ აგენტები სხვა აგენტების აზრის გათვალისწინებას ან როგორ უარყოფენ სხვა აგენტის აზრს. ახსნილი თუ როგორ უნდა მოხდეს არგუმენტაციის გამოყენება მრავალ აგენტურ სისტემებში, განხილულია რა სახის არგუმენტაცია არსებობს. აგენტების მთავარი მიზანი თუ როგორ

დაასაბუთონ თავიანთი აზრი არგუმენტების გამოყენებით მკაფიოდაა ხაზგასმული.

ასევე თემაში განხილულია ხელოვნური ნეირონული ქსელი აღწერილია თუ რა არის ადამიანური და ხელოვნური ნეირონული ქსელი. აღწერილია თუ როგორ ახდენს ადამიანი ობიექტების ამოცნობას და როგორ უნდა მოხდეს ამ ყველაფრის გამოყენება ხელოვნურ ინტელექტში. მოყვანილია მაგალითები თუ როგორ უნდა მოახდინონ აგენტებმა ობიექტების გამოცნობა ლოგიკაზე დაყრდნობით. ახსნილია თუ რა არის ნეირონი და როგორ შეიძლება მოხდეს ამ ყველაფრის გამოყენებით ნიმუშების ამოცნობა.

თემაში განხილულია თუ როგორ უნდა მოხდეს ობიექტების შეთავსება მისი თვისებებისა და სტრუქტურების მიხედვით და ასევე განხილულია თუ როგორ უნდა მოხდეს შესაბამისი არგუმენტების გენერირება. განხილულია ევრისტიკული მიდგომები და მოყვანილია შესაბამისი მაგალითები. აღწერილია თუ რაში გვჭირდება ევრისტიკული ფუნქციის გამოყენება და რა უპირატესობები აქვს ევრისტიკულ მიდგომებს.

დისერტაციის თემასთან დაკავშირებით გამოქვეყნებული პუბლიკაციები

1. მრავალ აგენტის სისტემები სახეთა ამოცნობისთვის - ზურაბ ბოსიკაშვილი, გიორგი არჩვაძე - საქართველოს საინჟინრო სიახლენი N2(vol.74), 2015, თბილისი 2015 გვ.17-22
2. სახეთა გამოცნობისთვის მრავალ აგენტებზე დაფუძნებული არქიტექტურა - ზურაბ ბოსიკაშვილი, გიორგი არჩვაძე - შრომები მართვის ავტომატიზირებული სისტემები N2(18) თბილისი 2014 გვ.11-16

3. მათემატიკური მოდელი და არგუმენტაციის თეორია სახეთა გამოცნობისთვის მულტიაგენტებზე დაფუძნებული არქიტექტურით - ზურაბ ბოსიკაშვილი, გიორგი არჩვაძე - შრომები მართვის ავტომატიზირებული სისტემები N1(19) თბილისი 2015 გვ. 187-196

გამოყენებული ლიტერატურა

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 2003
2. Zurab Bosikashvili, “ Extansion of the Architecture of Software Systems with Artificial Intelligence Elements“ Modern Computer Applications in Science and Education, Editor Constantin Buzatu, 75-84, ISBN:978-960-474-363-6, 2014
3. Trevor J.M. Bench-Capon, Paul E. Dunne. (2005). Argumentation and Dialogue in Artificial Intelligence. Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, U.K.
4. Alexander J. Faaborg. (2002). Using Neural Networks to Create an Adaptive Character Recognition System. Cornell University, Ithaca NY
5. Yoav Shoham, Leyton-Brown K. (2009). Multiagent systems, Algorithmic, Game-Theoretic and Logical Foundations. Cambridge University Press , Cambridge, England
6. Hunt J. (2002). Blackboard Architectures. JayDee Technology Ltd 27.
7. http://en.wikipedia.org/wiki/Blackboard_system, Wikipedia, the free encyclopedia.
8. Shahrin Azuan Nazeer, Nazaruddin O., Khairol F., Jumari and Marzuki Khalid. (2007). Face detecting using Artificial Neural Networks Approach. First Asia International Conference on Modelling & Simulation. Seri Kembangan
9. L. Amgoud. A formal framework for handling conflicting desires. In Proc. 7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'2003), pages 552–563, Aalborg, Denmark, 2003.

10. L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *International Journal of Automated Reasoning*, Volume 29 (2):125–169, 2002.
11. L. Amgoud and C. Cayrol. A reasoning model based on the production of accept-able arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197– 215, 2002.
12. L. Amgoud, Y. Dimopolous, and P. Moraitis. A unified and general framework for argumentation-based negotiation. In *Proc. 6th International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'2007)*, Honolulu, Hawai'i, 2007.
13. L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-00)*, pages 31–38, 2000.
14. https://en.wikipedia.org/wiki/Argumentation_framework , Wikipedia, the free encyclopedia.
15. Wei-Min Shen, Jafar Adibi, Rogelio Adobbati, Bonghan Cho, Ali Erdem, Hadi Moradi, Behnam Salemi, and Sheila Tejada. Building integrated mobile robots for soccer competition. In *Proceedings of the International Conference on Robotics and Automation*, 1998.
16. Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.
17. Peter Stone, Manuela Veloso, and Patrick Riley. The CMUnited-98 champion simulator team. In Minoru Asada and Hiroaki Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag, Berlin, 1999.
18. Hunt J., “Blackboard Architectures”, *JayDee Technology Ltd* 27, 2002.
19. Cavazza M., Mead S.J., Strachan A.I., Whittaker A., “A Blackboard System for Interpreting Agent Messages”, *Proceedings GameOn 2000: International Conference on Intelligent Games & Simulation*, UK, 2001.

20. van Liere R., Harkes J., de Leeuw W., “A Distributed Blackboard Architecture For Interactive Data Visualization”, *Proceedings of the conference on Visualization '98*, 1998, pp. 225 – 231.
21. Metzner C., Cortez L., Chacin D., “Using A Blackboard Architecture In A Web Application”, *The Journal of Issues in Informing Science and Information Technology*, Volume 2, 2005, pp. 743-756.
22. http://en.wikipedia.org/wiki/Blackboard_system , Wikipedia, the free encyclopedia.
23. [https://en.wikipedia.org/wiki/Heuristic_\(computer_science\)](https://en.wikipedia.org/wiki/Heuristic_(computer_science))
24. Chenoweth, S., and Davis, H. (1992). New Approaches for Understanding the Asymptotic Complexity of A* Tree Searching. *Annals of Mathematics and AI* 5
25. Davis, H., Pollack, R., and Sudkamp, T. (1984). Toward a Better Understanding of Bidirectional Search. *Proceedings AAAI*,
26. Field, R., Mohyeldin-Said, K., and Pohl, I. (1984). An Investigation of Dynamic Weighting in Heuristic Search, *Proc. 6th ECAI*,
27. Gaschnig, J. (1979). Performance Measurement and Analysis of Certain Search Algorithms. Ph.D. Dissertation CMU CS 79-124.
28. Korf, R.,E., and Zhang, W. (2000). Divide and Conquer Frontier Search Applied to Optimal Sequence Alignment. *AAAI-00*, pp. 910–916.
29. Pearl, J. (1983). Knowledge versus search: A quantitative analysis using A*. *Artificial Intelligence*:20, 1–13.
30. Pohl, I. (1970a). First Results on the Effect of Error in Heuristic Search. In B. Meltzer and D. Michie (Eds.), *Machine Intelligence* 5, pp. 219–236. Edinburgh University Press.
31. Pohl, I. (1973). The Avoidance of (Relative) Catastrophe, Heuristic Competence, Genuine Dynamic Weighting and Computational Issues in Heuristic Problem Solving. *IJCAII* 3, pp. 20–23.
32. Slagle, J. R. (1963). A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus. *Journal of the ACM* 10, 507–520.
33. https://en.wikipedia.org/wiki/A*_search_algorithm

Abstract

What is intelligent agent, how we can describe environment and how we can recognize pattern using many intelligent agent. In this paper are described recognition mechanisms as simple as well complex. There is described how we can compare objects and how divide it as different structures.

When agents communicate they do not necessarily use the same vocabulary or ontology. For them to interact successfully they must find correspondences between the terms used in their ontologies. This paper describes our work constructing a formal framework for reaching agents' consensus on the terminology they use to communicate. Each agent can decide according to its interests whether to accept or refuse the candidate correspondence. We are trying to construct pattern recognition model and how to use different ways to recognize patterns, in this paper we are constructing agent model which consist of different sub model. There are shown different approaches to pattern recognition and different types of multiagent systems. We are trying to explain what the pattern recognition is and how the agents can interact successfully. Defines a very simple description of coordination and negotiation schemes through multi agent systems.

This paper describes Artificial Neural Network. There is briefly described how human brain operates as a multiprocessor. We want to show in this paper advantages and disadvantages of artificial neural network and compare it to human neural network. There is detail explanation how human can recognize pattern and how we can implement it computer system.

This paper presents heuristics approaches. Explains what is heuristic approach and what type problems can we solve using heuristic approach. This paper also generates heuristic function and shows problem solving using heuristic approach.

BlackBoard architecture is one of the famous approaches to solve artificial intelligence tasks. In this paper is described what is blackboard architecture and its

components are briefly described. We are going to show how we can use blackboard architecture in multi agent systems and describe advantages and disadvantages.