

საქართველოს ტექნიკური უნივერსიტეტი

გიორგი არჩვაძე

სწავლების და ცოდნის მოპოვების მეთოდების დამუშავება და კვლევა
მრავალკომპონენტურ მეხსიერებთან სისტემებისთვის

წარმოდგენილია დოქტორის აკადემიური
ხარისხის მოსაპოვებლად

სადოქტორო პროგრამა “ინფორმატიკა” შიფრი 0401

საქართველოს ტექნიკური უნივერსიტეტი
თბილისი, საქართველო
ივლისი, 2015 წელი

საავტორო უფლება © 2015, გიორგი არჩვაძე
თბილისი
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

საქართველოს ტექნიკური უნივერსიტეტი
ინფორმატიკის და მართვის სისტემების ფაკულტეტი

ჩვენ, ქვემოთ ხელისმომწერი ვადასტურებთ, რომ გავეცანით გიორგი არჩვამის მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: „სწავლების და ცოდნის მოპოვების მეთოდების დამუშავება და კვლევა მრავალკომპონენტურ მეხსიერებიან სისტემებისთვის” და ვაძლევთ რეკომენდაციას საქართველოს ტექნიკური უნივერსიტეტის ინფორმატიკისა და მართვის სისტემების ფაკულტეტის სადისერტაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

თარიღი: __/__/__

ხელმძღვანელი: პროფ. ზურაბ ბოსიკაშვილი

რეცენზენტი:

რეცენზენტი:

საქართველოს ტექნიკური უნივერსიტეტი

2015 წელი

ავტორი: გიორი არჩვაძე

დასახელება: „სწავლების და ცოდნის მოპოვების მეთოდების
დამუშავება და კვლევა მრავალკომპონენტურ მეხსიერებიან
სისტემებისთვის”

ფაკულტეტი: ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

ხარისხი: დოქტორი

სხდომა ჩატარდა:

ინდივიდუალური პიროვნებების ან ინსტიტუტების მიერ
ზემომოყვანილი დასახელების ნაშრომის გაცნობის მიზნით მოთხოვნის
შემთხვევაში მისი არაკომერციული მიზნებით კოპირებისა და გავრცელების
უფლება მინიჭებული აქვს საქართველოს ტექნიკურ უნივერსიტეტს.

ავტორის ხელმოწერა

ავტორი ინარჩუნებს დანარჩენ საგამომცემლო უფლებებს და არც
მთლიანი ნაშრომის და არც მისი ცალკეული კომპონენტების გადაბეჭდვა ან
სხვა რაიმე მეთოდით რეპროდუქცია დაუშვებელია ავტორის წერილობითი
ნებართვის გარეშე.

ავტორი ირწმუნება, რომ ნაშრომში გამოყენებული საავტორო
უფლებებით დაცული მასალებზე მიღებულია შესაბამისი ნებართვა (გარდა
ის მცირე ზომის ციტატებისა, რომლებიც მოითხოვენ მხოლოდ სპეციფიურ
მიმართებას ლიტერატურის ციტირებაში, როგორც ეს მიღებულია
სამეცნიერო ნაშრომების შესრულებისას) და ყველა მათგანზე იღებს
პასუხისმგებლობას.

რეზიუმე

რა არის ინტელექტუალური აგენტი როგორ უნდა მოხდეს გარემოს აღწერა და როგორაა შესაძლებელი ბევრი ინტელექტუალური აგენტის საშუალებით ობიექტების ამოცნობა. ნაშრომში განხილულია ამოცნობის მექანიზმები როგორც მარტივი ასევე კომპლექსური. აღწერილია თუ როგორ უნდა მოხდეს ობიექტების დამთხვევა და მათი სტრუქტურებად დაყოფა.

როდესაც აგენტები ურთიერთობენ ისინი არასდროს იყენებენ ერთიდაიგივე ლექსიკონს და ონთოლოგიას. იმისთვის რომ მათი ურთიერთობა იყოს წარმატებული მათ უნდა იპოვონ კორესპონდენცია ტერმინებს შორის რომლებსაც ისინი იყენებენ თავიანთ ონთოლოგიებში. დისერტაციაში აღწერილია თუ როგორ შეიძლება მოხდეს შეთანხმების მიღწევა აგენტებს შორის იმ ტერმინოლოგიით რომლებსაც ისინი მოიხმარენ საურთიერთობოდ. თითოეულ აგენტს შეუძლია თავისი ინტერესიდან გამომდინარე დაეთანხმოს ან უარყოს სხვა აგენტის კორესპონდენცია. ჩვენ ვცდილობთ შევქმნათ ობიექტების(სახეთა) ამომცნობი მოდელი და გამოვიყენოთ სხვადასხვა გზები ამოცნობისთვის. დისერტაციაში ჩვენ ვაწყობთ აგენტების მოდელს რომელიც შედგება განსხვავებული ქვე მოდელებისგან. ასევე ნაჩვენებია სხვადასხვა მიდგომები ობიექტების(სახეთა) გამოცნობისთვის. ჩვენ ვცდილობთ ავხსნათ თუ რა არის ობიექტების(სახეთა) გამოცნობა და როგორ შეძლებენ აგენტები წარმატებულ კომუნიკაციას. ასევე აღწერილია კორდინაციისა და მოლაპარაკების სქემა მრავალ აგენტთან სისტემებისთვის.

დისერტაციაში აღწერილია თუ რა არის ხელოვნური ნეირონული ქსელი, ასევე დეტალურადაა განხილული როგორ მუშაობს ადამიანის ტვინი. ჩვენ გვინდა გაჩვენოთ უპირატესობები ხელოვნური ნეირონული ქსელის და შევადაროთ იგი ადამიანურ ნეირონულ ქსელს. დეტალურადაა განხილული თუ როგორ ახერხებს ადამიანი ობიექტების ამოცნობას და როგორ არის შესაძლებელი იგივე გააკეთოს კომპიუტერმა.

ნაშრომში წარმოდგენილია ევრისტიკული მიდგომები. ახსნილია რა არის ევრიტიული მიდგომა და რა ტიპის ამოცანების გადაჭრა არის შესაძლებელი. ასევე განხილულია ევრისტიკული ფუნქციის გენერირება. ნაჩვენებია ევრისტიკული მიდგომით ამოცანის გადაჭრის გზები.

შავი დაფის არქიტექტურა ერთერთი ცნობილი მიდგომაა ხელოვნური ინტელექტის ამოცანების გადაჭრისათვის. ახსნილია თუ რას წარმოადგენს შავი დაფის არქიტექტურა და რა კომპონენტებისგან შედგება იგი. აღწერილია თუ როგორ შეიძლება მრავალ აგენტთან სისტემაში მისი გამოყენება და რა უპირატესობები ექნება ესეთ მიდგომას.

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 advanatges and disadvantages.

შინაარსი

შესავალი	12
1. ლიტერატურის მიმოხილვა	16
2. შედეგები და მათი განსჯა.....	18
ინტელექტუალური აგენტების თეორია	18
2.1. ინტელექტუალური აგენტების ცნების განსაზღვრა.....	18
2.2. რაციონალური აგენტის ცნების განსაზღვრა.....	22
2.3. გარემოს აღწერა.....	25
ობიექტების(სახეთა) ამოცნობა	28
3.1. ობიექტების ამოცნობა	30
3.2. ამოცნობის მარტივი მექანიზმები	32
3.3. ამოცნობის უფრო კომპლექსური პროცესი.....	35
3.4. შაბლონური დამთხვევა	35
3.5. სტრუქტურული აღმწერები	40
3.6. ობიექტის გრაფად წარმოდგენა და ამოცნობის სხვადასხვა მიდგომები	45
ხელოვნური ნეირონული ქსელი.....	49
4.1. ხელოვნური ნეირონული ქსელი.....	51
4.2. ადამიანური და ხელოვნური ნეირონული ქსელი - მსგავსებების გამოძიება	52
4.3. ადამიანების ნეირონული ქსელიდან ხელოვნურ ნეირონულ ქსელამდე ...	53
4.4. მარტივი ნეირონი	54
4.5. გასროლის წესები	54
4.6. ნიმუშის ამოცნობა - მაგალითი	56
მრავალ აგენტიანი ამოცნობის არქიტექტურა	60
5.1. მრავალაგენტიანი სისტემები.....	62
5.2. ერთ აგენტიანი და მრავალაგენტიანი სისტემა.....	65
5.3. ერთ-აგენტიანი სისტემა.....	66
5.4. მრავალაგენტიანი სისტემები.....	66
აგენტების მოდელი.....	69
6.1. ონთოლოგიური მოდელი.....	69
6.2. გადაწყვეტილებების მიღების მოდელი.....	71
6.3. სემანტიკა	71

არგუმენტაცია	73
7.1. კლასიკური არგუმენტაცია.....	77
7.2. მნიშვნელობებზე დამოკიდებული არგუმენტაცია.....	78
სემანტიკური შეთავსების მოდელი.....	80
8.1. თვისობრივი არგუმენტების გენერირება	81
8.2. მიმართებითი არგუმენტების გენერირება	82
8.3. სტრუქტურული არგუმენტების გენერირება.....	83
ევრისტიკული ძებნა	85
9.1. განხორციელება	87
9.2. მაგალითი.....	88
9.3. ხარბი ძებნა	89
9.4. A* ძებნა.....	91
9.5. 8 - თავსატეხი	94
9.6. ევრისტიკული ფუნქციის ეფექტურობა.....	96
შავი დაფის არქიტექტურა.....	97
10.1. შავი დაფის არქიტექტურის უპირატესობები.....	99
10.2. შავი დაფის აპლიკაცია	100
10.3. შავი დაფის სისტემა აგენტების მესიგების ინტერპრეტაციისთვის..	101
10.4. მობილური რობოტი დერეფნის ნავიგაციისთვის	102
აგენტების მიერ სიმბოლოების ამოცნობა.....	104
დასკვნა	106
გამოყენებული ლიტერატურა.....	108

ნახაზების ნუსხა

სურ. 1 აგენტების მიერ გარემოს აღქმა და გარემოდან ზემოქმედება.....	31
სურ: 2 სურათზე ნაჩვენებია ასო A-ს სხვადასხვა ფორმები.....	42
სურ 3 სურათზე ნაჩვენებია ერთი და იგივე პიროვნება სხვადასხვა	42
სურ 4 თევზი უტევდა ყველა ობიექტს, რომელსაც ქონდათ წითელი მუცელი...	44
სურ 5 სანამ დაიწყეთ შაბლონის დამთხვევა ნიმუშს შესაძლოა შეუცვალონ ორიენტაცია, დაატრიან ვერტიკალურად ან გაზარდონ ზომაში.....	48
სურ 6 სურათზე ნაჩვენებია თუ როგორ შეიძლება ასო A-ს შაბლონი დაემთხვეს ასო R-ს შაბლონს და პირიქით.....	49
სურ 7 რით ხდება Q-ს და O-ს ერმანეთისგან განსხვავება? ეს არც წრის ფორმა არაა, არც Q-ს ხაზი, რადგან ის შეიძლება რამოდენიმე წერტილში იყოს განთავსებული.....	49
სურ 8 შინაარსი მიხედვით ადამიანის ასოების აღქმის უნარი.....	52
სურ 9 ასო T-ს სტრუქტურული აღწერა.....	54
სურ 10 ასო T.....	55
სურ 11 სათამაშოების ნიმუში რომელთა ცნობაც შეეძლო ვინსტონის პროგრამას.....	56
სურ 12 ტრენინგის მიმდევრობა.....	57
სურ 13 ამოცნობის არქიტექტურა.....	61
სურ 14 ნეირონის კომპონენტები და სინაფსისი.....	66
სურ 15 ხელოვნური ნეირონული მოდელი.....	67
სურ 16 მარტივი ნეირონი.....	68
სურ 17 ობიექტის ამოცნობის მოდელი.....	70
სურ 18 ერთ აგენტიანი და მრავალ აგენტიანი სისტემა.....	81
სურ 19 აგენტების მოდელი და ონთოლოგია.....	85
სურ 20 დანგის არგუმენტაციის ფრეიმვორკის მაგალითი.....	92

რ 21 მნიშვნელობებზე დამოკიდებული არგუმენტაციის მაგალითი.....	94
სურ 22 ევრისტისტიკული ძეგლის მაგალითი.....	105
სურ 23 აგენტების მუშაობა შავ დაფაზე.....	115
სურ 24 მრავალ აგენტიანი სისტემა შავი დაფის გამოყენებით.....	120
სურ 25 სხვადასხვა დონის და ცოდნის მქონე აგენტები.....	122

შესავალი

მრავალ აგენტური სისტემები არგუმენტაციის გამოყენებით მთავარი კომპონენტია განაწილებულ სისტემებში და ადამიანის ინტელექტში. აგენტების საურთიერთობოდ როდესაც ისინი მუშაობენ საერთო პრობლემაზე საჭიროა გამოვიყენოთ არგუმენტაცია რათა მოხდეს მათივე წამოყენებული მოსაზრებების დასაბუთება. ობიექტების ამოცნობისთვის როდესაც ბევრი აგენტი ცდილობს მიიღოს გადაწყვეტილება წარმატების მისაღწევად იყენებენ არგუმენტაციას. ასეთ შემთხვევებშიც ნებისმიერ აგენტს საშუალება აქვს დაეთანხმოს ან უარყოს სხვა აგენტის მოსაზრება. როდესაც აგენტებს აქვთ კომუნიკაცია ისინი არასდროს არ იყენებენ ერთიდაიგივე ლექსიკონს, რათა იურთიერთობონ წარმატებულად მათ უნდა გამოიყენონ არგუმენტები რათა იპოვონ კორესპონდენცია ტერმინებს შორის რომლებსაც ისინი იყენებენ[1].

წარმატებული კომუნიკაცია მთავარი პრობლემაა ისეთ არქიტექტურაში სადაც ბევრი აგენტი მუშაობს ერთი გადაწყვეტილების მიღებისთვის. აგენტებს აქვთ საშუალება ერთმანეთს გაუზიარონ ინფორმაცია, უარყონ და დაეთანხმონ სხვა აგენტის აზრს შესაბამისი არგუმენტის გამოყენებით. მათი მთავარი მიზანი მხოლოდ ის არის, რომ საბოლოოდ დავიდნენ ერთ დასკვნამდე რომელიც გამომდინარეობს შესაბამისი გამოცდილებიდან და ცოდნიდან. დისერტაციაში მოყვანილია მაგალითები თუ როგორ ახერხებენ აგენტები ობიექტის ამოცნობას. ჩვენი მთავარი მიზანია აღვწეროთ რა არის არგუმენტაცია და როგორ გამოვიყენოთ იგი მრავალ აგენტურ სისტემებში ობიექტების ამოცნობისთვის, თუ როგორ ვაურთიერთქმედოდ აგენტები რო მათი გადაწყვეტილება იყოს უშეცდომო.

ობიექტების(სახეთა) ამოცნობა არის მეცნიერება თუ როგორ შეიძლება მანქანები დააკვირდნენ გარემოს, გამოყონ ობიექტები(სახეები)

მათი ინტერესის და გამოცდილების მიხედვით, გახადონ აშკარა გადაწყვეტილების მიღება ობიექტების(სახეთა) კატეგორიების მიხედვით. საუკეთესო ობიექტების(სახეთა) ამომცნობები არიან ადამიანები, მაგრამ ჩვენ არ გვესმის თუ როგორ ახერხებენ ადამიანები ამის გაკეთებას[3].

როსი ხაზს უსვამს ნობელის ლაურეატის ჰერბერტ სიმონის მუშაობას რაც გულისხმობს რო ობიექტების ამოცნობა არის ყველაზე კრიტიკული ამოცანა ადამიანურ გადაწყვეტილებებში რომლებსაც ისინი ღებულობენ: „ნიმუშების შესაბამისობა თქვენს განკარგულებაშია, მაგრამ მთავარი გადაწყვეტილების მიღებაა.“ ეს არის იმედის მომცემი ახალი ამბავი ხელოვნური ინტელექტის მომხრეებისთვის, მას შემდეგ რაც კომპიუტერი შეძლებს ისწავლოს თუ როგორ მოახდინოს ობიექტების (სახეთა) ამოცნობა. მართლაც რომ წარმოვიდგინოთ წარმატებული კომპიუტერული პროგრამა რომელიც დაეხმარება ბანკებს შეაფასოს საკრედიტო მომხმარებლები, დაეხმაროს ექიმებს პაციანტის დიაგნოზის დადგენაში, დაეხმარება მფრინავებს დასვან თვითმფრინავი და ეს ყველაფერი დამოკიდებული იქნება სხვადასხვა ობიექტების (სახეთა) ამოცნობაზე. ჩვენ დიდი ყურადღება უნდა გავამახვილოთ სწავლებაზე ობიექტების ამოცნობისთვის. შვენი მიზანია ავხსნათ რა არის ობიექტების(სახეთა) ამოცნობა ხელოვნური ნეირონული ქსელის გამოყენებით, რა არის საუკეთესო მივილოთ გადაწყვეტილება ავტომატურად[2].

მრავალ აგენტთან სისტემებში ობიექტების(სახეთა) ამოცნობისთვის როგორც ავლნიშნეთ ჩვენ ვიყენებთ არგუმენტაციას, არგუმენტაცია არის მნიშვნელოვანი ასპექტი ადამიანების ინტელექტში, გვამღევეს საშუალებას თუ როგორ მოახდინონ ადამიანებმა ინფორმაციის ანალიზი , გამოყონ დადებითი და უარყოფითი მხარეები როდესაც ცდილობენ მიიღონ გადაწყვეტილება. არგუმენტაციის ფორმალიზაცია კომპიუტერულ გარემოში იწვევს ზრდად ინტერესს ხელოვნური ინტელექტის მიმართულებით კვლევებში. აგენტებმა როგორც ადამიანები უნდა მოხადნინო ანალიზი თავიანთი ინფორმაციის და დაადგინონ უარყოფითი და დადებითი მხარეები ამა თუ იმ მოსაზრების.

მიმდინარე ნაშრომი წარმოადგენს ხელოვნური ინტელექტში ობიექტების(სახეთა) ამოცნობის მრავალ აგენტის მიდგომის ანალიზს არგუმენტაციის გამოყენებით. იგი მოიცავ შემდეგ თავებს

თავი 2 : განხილულია ხელოვნური ინტელექტის ისტორია, ახსნილია მისი გამოყენება სხვადასხვა სფეროებში და მოყვანილია შესაბამისი მაგალითები.

თავი 3 : აღწერილია თუ რა არის ინტელექტუალური აგენტი, რა ფუნქცია აკისრია მას და რა სამუშაოს უნდა ასრულებდეს იგი. ასევე აღწერილია თუ რა არის რაციონალური აგენტი, მკაფიოდ არის ახსნილი თუ როგორ უნდა მოხდეს აგენტებისათვის გარემოს განსაზღვრა.

თავი 4 : ამ თავში განხილულია ობიექტების(სახეთა) ამოცნობის მაგალითები, აღწერილია მათი ამოცნობის როგორც მარტივი ასევე კომპლექსური მექანიზმები, განხილულია ობიექტების დამთხვევის სხვადასხვა შემთხვევები, ასევე ობიექტების წარმოდგენის სხვადასხვა ვარიანტები და შესაბამისი მექანიზმები. თავში მოყვანილია ობიექტების ამოცნობისთვის სხვადასხვა მაგალითები.

თავი 5 : განხილულია თუ რა არის ხელოვნური ნეირონული ქსელი და მოყვანილია შედარებები ხელოვნური და ადამიანურ ნეირონულ ქსელებს შორის. ახსნილია თუ როგორ ხდება გადასვლა ადამიანური ნეირონული ქსელიდან ხელოვნურ ნეირონულ ქსელამდე. განხილულია რა არის მარტივი ნეირონი და მოყვანილია შესაბამისი მაგალითები.

თავი 6 : ამ თავში განხილულია მრავალ აგენტის ამომცნობი არქიტექტურა მოყვანილია მაგალითები ერთ აგენტის და მრავალ აგენტის სისტემებზე. შედარებულია მრავალ აგენტის სისტემა ერთ აგენტის სისტემასთან და აღწერილია უპირატესობები.

თავი 7 : თავში განხილულია ორი მოდელი ონთოლოგიური და გადაწყვეტილების მიღების მოდელები. დეტალურად არის აღწერილი თუ რას წარმოადგენს თითოეული მათგანი.

თავი 8 : თავში განხილულია არგუმენტაციის ფრეიმვორკი. ახსნილია თუ როგორ უნდა მოხდეს არგუმენტების გენერირება, შედარებულია კლასიკური და მნიშვნელობებზე დამოკიდებული არგუმენტაციები.

თავი 9 : აღწერილია ობიექტების(სახეთა) შეთავსებების მოდელები. ახსნილია როგორ უნდა მოხდეს თვისობრივი არგუმენტების გენერირება , ასევე განხილულია მიმართებითი არგუმენტების გენერირება და შესაბამისად სტრუქტურული არგუმენტების გენერირება.

თავი 10 : ამ თავში განხილულია ევრისტიკული ალგორითმები. აღწერილია თუ რა არის ევრისტიკული ძებნა და მოყვანილია მისი რამოდენიმე მაგალითი და შესაბამისი განხორციელება.

თავი 11 : თავში განხილულია შავი დაფის არქიტექტურა, ახსნილია თუ რას წარმოადგენს ამოცანის გადაჭრის ესეთი მიდგომა და როგორ შეიძლება გამოვიყენოთ იგი მრავალ აგენტთან სისტემებში. ახსნილია თუ რა არის შავი დაფის აპლიკაცია და მოყვანილია შესაბამისი მაგალითები.

1. ლიტერატურის მიმოხილვა

დისერტაციაში დასმული პრობლემები მომდინარეობს ხელოვნურ ინტელექტში დასმული ცნობილი პრობლემებიდან. კერძოდ კოლექტიური გადაწყვეტილების მიღების, დასკვნების კეთების, სახეთა გამოცნობის და სხვა კომპლექსური ამოცანების ამოხსნის განაწილებული სისტემების აგების ამოცანებიდან. ამ ამოცანებში საერთო ის არის, რომ რადგანაც ისინი კომპლექსური ამოცანებია მათი გადაწყვეტისთვის საჭიროა დიდი გამოთვლითი რესურსები. თანამედროვე მიდგომა ამ ამოცანების ამოსახსნელად გულისხმობს მრავალ აგენტური არქიტექტურის სისტემების გამოყენებას.

სხვადასხვა დისერტაციებში თუ ნაშრომებში ამოცანის გადაწყვეტის მრავალი გზა არის აღწერილი, თუ როგორ უნდა მოხდეს ამოცანის სტრუქტურულად დაყოფა და შესაბამისი მაგალითები ასევე დასკვნების კეთების სხვადასხვა მიდგომები, სწავლის და ცოდნის მოპოვების საშუალებები.

მაგალითად, სტატიაში [4] „სიმბოლოების ამოცნობა ხელოვნური ნეირონული ქსელების გამოყენებით“ სიმბოლოების წარმოდგენა ხდება მატრიცის სახით და მსგავსი სიმბოლოები გაერთიანებულია ერთ ჯგუფში. სწავლის მექანიზმისთვის გამოიყენება სიმბოლოების შაბლონური დამთხვევა. ასევე სტატიაში [8] „გამოვიყენეთ ნეირონული ქსელი რათა შევქმნათ ადაპტირებული სიმბოლოების ამოცნობის სისტემა“ სიმბოლოების ამოცნობისთვის აქაც მათი წარმოდგენა ხდება მატრიცის სახით.

სტატიაში „არგუმენტაცია და დიალოგი ხელოვნურ ინტელექტში“ აღწერილია არგუმენტების სტრუქტურები და დიალოგი, რომელიც ხორციელდება არგუმენტაციის გამოყენებით. ხდება არგუმენტების მოდელირება და სხვადასხვა ჯგუფებად დაყოფა, იმისდამიხედვით თუ რა ტიპის არგუმენტია. დიალოგს გააჩნია შესაბამისი ტიპი და მას მიეკუთვნება შესაბამისი არგუმენტების სია.

მრავალ დისერტაციაში და ნაშრომში ობიექტების ამოცნობისთვის გამოყენებულია შავი დაფის არქიტექტურა, რომელიც თავისთავად მოიცავს ცოდნის მოპოვებას, სწავლებას და დასკვნების კეთებას.

ხელონურ ინტელექტში არსებობს ამოცანები რომლებიც ძალიან დიდი რესურს მოითხოვს შედეგის მისაღებად, რასაც რეალურ დროში თანამედროვე კომპიუტერები ვერ ახერხებენ. საკითხი რთულდება როცა განზომილებასთან ერთად იზრდება ცოდნის ან ინფორმაციის განუზღვრელობის ხარისხი. დისერტაციაში არის მცდელობა ამ პრობლემების გადაჭრის მრავალაგენტური არქიტექტურის და კოლექტიური დაკვნების კეთების მექანიზმებით.

2. შედეგები და მათი განსჯა

ინტელექტუალური აგენტების თეორია

2.1. ინტელექტუალური აგენტების ცნების განსაზღვრა

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 განსხვავებისთ 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 საშუალებებით, 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.

რაციონალური აგენტი ეწოდება ისეთ აგენტს, რომლის მოქმედებით მიიღწევა შესაძლო წარმატებული (საუკეთესო) შედეგი ან, თუ გარემო განუსაზღვრელია (არადეტერმინირებული ანუ სტოქასტიკური), მაშინ მიიღწევა საუკეთესო მოსალოდნელი შედეგი.

რაციონალური აგენტის შექმნისათვის დიდი მნიშვნელობა აქვს ადმიანის მოქმედების მოდელირებას, რაც - ხელოვნური ინტელექტის სფეროში არსებული ერთერთი პრობლემაა. გარდა ამისა, ადამიანთან მიმართებაში კარგადაა გარკვეული და განზოგადებული რაციონალური მოქმედების სახე. უნდა აღინიშნოს, რომ გამოიყენება იდეალური რაციონალურობის კონცეფციაც, რომელსაც აქვს არა პრაქტიკული, არამედ უფრო თეორიული დატვირთვა რადგან იდეალურობის ზუსტი განსაზღვრა ძალიან რთულია, ხოლო იდეალურად რაციონალური აგენტის აგებადა ფორმირება პრაქტიკულად შუძლებელია, რაც თეორიულადაცაა დატკიცებული. იდეალური აგენტის ცენაზ გამოსადგეია რაციონალური აგენტის შესაფასებლად, როგორც ათვლის წერტილი მოქმედებათა სივრცეში.

აგენტის თეორიაში არსებობს „შეზღუდული რაციონალურობის“ ცნება

შეზღუდული რაციონალურობის აგენტი ეწოდება აგენტს, რომელიც გარემოს ან სხვა მიზეზით გამოწვეული შეზღუდვებისას ახლორციელებს მხოლოდ მისაღებ ქმედებას. უნდა აღინიშნოს რო მისაღები ქმედებები ყოველთვის არ არის რაციონალური.

რთული გარემოს არსებობის შემთხვევაში, იდეალური რაციონალურობის მიღწევა, როდესაც ყოველთვის მიიღება სწორი გადაწყვეტილებები და მათი შესაბამისი მოქმედებები, პრაქტიკულად განუხორციელებელია.

მაგალითისათვის განვიხილოთ ჩვენი საინტერესო ობიექტი აგენტი, პროგრამული მოდულის სახით, რომელიც ერთდროულად წარმოადგენს ერთერთ პროგრამულ უზრუნველყოფას კომპიუტერისათვის. აგენტი

პროგრამული უზრუნველყოფის სახლით გარემოდან იღებს სენსორულ ინფორმაციას, ანუ სიგნალებს, კლავიატურაზე თითების დაჭერით, ფაილების წაკითხვით, კომპიუტერის გარე მოწყობილობებით, როგორცაა ოპტიკური სისტემა-ობიექტივი, სკანერი, სმენითი-მიკროფონი, შეხებითი - ტაქტილური გადაამწოდები და ა.შ. გარემოს ზემოქმედება გამოიხატება მიღებული გადაწყვეტილებებისა და მონაცემების გამოტანით ეკრანზე, ახალი ფაილების შექმნით და მაში ინფორმაციის ჩაწერით. მხედველობაში უნდა მივიღოთ რომ ნებისმიერ აგენტს შეუძლია თავისი მოქმედებებით გამოწვეული შედეგების აღქმა.

ტერმინი - „აღქმა“, აგენტისათვის წარმოადგენს დროსი კონკრეტულ მომენტში მიღებულ ინფორმაციას, ხოლო ტერმინი - „აღქმის თანამიმდევრობითი აქტები“ აღნიშნავს მთელ ინფორმაციას, რაც აღქმულია აგენტის მიერ მისი არსებობის განმავლობაში მოცემულ მომენტამდე.

გარემოს ზემოქმედებით გამოწვეულ აღქმების კონკრეტულ თანამიმდევრობას, აგენტის ფუნქცია ეწოდება.

თუ ხერხდება აგენტის მოქმედებათა აღწერა ყველა აღქმის თანამიმდევრობით აქტებისათვის, მაგალითად ფუნქციების, გრაფიკული, ცხრილების ან მატრიცების სახით მაშინ გვაქვს აგენტის „გარე აღწერა“.

აგენტის „შიდა აღწერა“ ეწოდება კონკრეტული აგენტის მოქმედებათა აღწერას გარე ზემოქმედებაზე აგენტის ფუნქციის გამოყენებით. ასეთ აღწერას „აგენტის პროგრამა ეწოდება“.

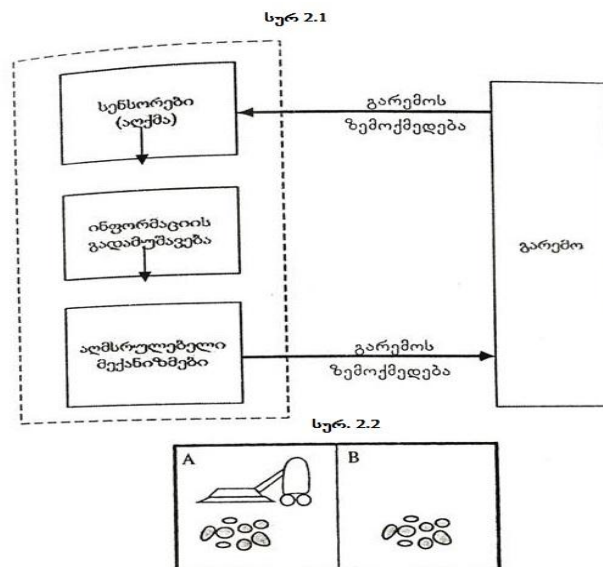
აგენტის ფუნქცია წარმოადგენს ფორმალიზებულ აბსტრაქტულ - მათემატიკურ აღწერას, ხოლო აგენტის პროგრამა- აგენტის ფუნქციის კონკრეტულ რეალიზაციას, რომელიც დამოკიდებულია აგენტის აგებულების თავისებურებებზე, ანუ აგენტის არქიტექტურაზე.

აგენტის გარემოსთან ურთიელთქმედების ზოგადი სქემა მოცემულია 3.1 სურათზე სადაც წყვეტილი ხაზებითაა მოცემული აგენტის ზოგადი აგებულება- სტრუქტურა; გარემოს მოქმედება აგენტზე მოცემულია სიგნალთა ერთობლიობით „გარემოს ზემოქმედება“, რაც აგენტის მიერ

აღიქმება სენსორების საშუალებით; მოცემულ ინფორმაციას გადაამუშავებს აგენტი რის შედეგადაც იღებს გადაწყვეტილებას.

მიღებული გადაწყვეტილება მიეწოდება აგენტის აღმსრულებელ მექანიზმებს რომლებიც მიღებული გადაწყვეტილების შესაბამისად ახდენენ გარემოზე ზემოქმედებას.

ზემოთ აღწერილი ტერმინებისა და სიტუაციების დასაზუსტებლად განვიხილოთ უმარტივესი აგენტის მოქმედების მაგალითი, კერძოდ აგენტი , რომლის დანიშნულებაა ორი კვადრატისაგან (ოთხასისაგან) შედგენილი სივრცის (იატაკის) დასუფთავება, ანუ ამ შემთხვევაში, აგენტი წარმოადგენს მტვერსასრუტს სურათზე 1 წარმოდგენილი ა და ბ კვადრატებით.



სურ. 1 (აგენტების მიერ გარემოს აღქმა და გარემოდან ზემოქმედება)

მტვერსასრუტი რომელიც მოძრაობს ავტონომიურად მოთავსებულია ა ოთახში. თუ მტვერსასრუტი ინტელექტუალურია, მაშინ იგი თვითონ აღგენს თავის მდებარეობას და მტვერის ან ანაგვის არსებობას მოცემულ ოთახში. აგენტს შეუძლია შემდეგი მოქმედებების განხორციელება გადაადგილება მეზობელ ოთახში, ნაგვის შეწოვა და უმოქმედობა. აგენტის ფუნქცია მარტივია: თუ ოთახში არის ნაგავი, შისრრუტოს; წინააღმდეგ შემთხვევაში გადავიდეს მეორე ოთახში. ამ მოქმედებიებიდან გამომდინარე, ცხადი ხდება , რომ დასახული მიზნის მიღწევა , ანუ ოთახების

დასუფთავება აგენტს სხვა ხერხებითაც შეუძლია. ამთან დაკავშირებით, ჩნდება კითხვა : აგენტის მოქმედება ერთობლიობებიდან რომელია სწორი? ამ კითხვაზე პაუზის გაცემით გავარკვევთ, თუ როგორია აგენტი: ცუდი თუ კარგი; ინტელექტუალური თუ არაინტელექტუალური.

2.2. რაციონალური აგენტის ცნების განსაღვრა

რაციონალური აგენტები ეწოდება ისეთ აგენტებს, რომლებსაც შეუძლიათ სწორი მოქმედებების განხორციელება. რას ნიშნავს სწორი მოქმედებები? სწორი მოქმედებების შესაფასებლად გამოიყენება აგენტის ფუნქციონირების შედეგების ამსახველი მახასიათებლები. კერძოდ სწორ მოქმედებებად ჩაითვლება ისეთი ქმედებები, რომლებიც უზრუნველყოფს აგენტის საუკეთესო ფუნქციონირებას მოცემულ გარემოში. აქედან გამომდინარე საუკეთესო ფუნქციონირების გარდა, აგენტის რაციონალურობის განსაზღვრისათვის საჭიროა გარემოს, სენსორული შემსრულებელი მექანიზმების აღწერა რაც წარმოადგენს სრულ ინფორმაციას აგენტის შესახებ., ანუ იმ პრობლემური გარემოს აღწერას, რომელშიც მოქმედებს აგენტი.

აგენტის მოქმედების ერთერთი მნიშვნელოვანი მახასიათებელია წარმოებლურობა. მას შემდეგ რაც აგენტი იწყებს გარემოში მოქმედებას, ანუ ზემოქმედებს გარემოზე, შესაბამისად იცვლება გარემო აგენტის ყოველი თანამიმდევრულად განხორციელებული ქმედების შემდეგ. რაც ნიშნავს რომ იცვლება გარემოს მდგომარეობები დროის გარკვეულ მომენტებში. თუ მოცემულ მომენტში გარემოს მდგომარეობები სასურველია მაშინ აგენტის ფუნქციონირება შეგვიძლია შევაფასოთ, როგორც კარგი.

უნდა აღინიშნოს , რომ ერთი მახასიათებლით შეუძლებელია აგენტის შეფასება. მაგალითისათვის განვიხილოთ ზემოთ აღწერილი ქმედება აგენტი მტვერსასრუტის სახით. ასეთი აგენტისთვის მწარმოებლურობის

მახასიათებელი იქნება დროის ერთეულში მოწმენდილი იატაკის ფართობი ანუ აღებული ნაგავი. აგენტის მოქმედების რაციონალურობიდან გამომდინარე, მხოლოდ წარმოებლობის კრიტერიუმის გამოყენებით, აგენტს აწყობს თუ შეუძლია დასუფთავებული იატაკიდან შეგროვებული ნაგავი ისევ დაყაროს იატაკზე და განმეორებით ისევ აიღოს იატაკიდან და ასე რომ არ მოხდეს აუცილებელია აგენტს ქონდეს დასუფთავებისათვის სხვა კრიტერიუმებიც, მაგალითად დაჯილდოვება ყოველი დასუფთავებული ფართობის ერთეულის შესაბამისად. აქ შეიძლება შემოვიღოთ ჯარიმაც დახარჯული ელექტროენერჯისა და გამოწვეული ხმაურის გამო.

აღწერილი მაგალითიდან გამომდინარე, შევადგინოთ იმ თვისებების ნუსხა რაც უნდა ქონდეს რაციონალურ აგენტს:

1. მიზნობრივი ფუნქცია, რომელიც უზრუნველყოფს აგენტსი წარმატებულ ფუნქციონირებას;
2. აგენტსი ცოდნა იმ გარემოს შესახებ, რომელშიც მოუწევს მოქმედება; (ეს პუნქტი უნდა შესრულდეს აგენტის მოქმედების დაწყებამდე);
3. კონკრეტული აგენტი შესაძლო მოქმედებების ჩამონათვალი რომელშიც მას შეუძლია შეასრულოს;
4. აღქმის თანამიმდევრობათა სიმრავლის აღწერა (დამახსოვრება) მოცემულ მომენტამდე.

ამ მოთხოვნების საფუძველზე შეიძლება ჩამოვაცალიბოთ რაციონალური აგენტის ზუსტი განსაზღვრა.

რაციონალური ეწოდება აგენტს, რომლის ნებისმიერი აღქმის თანამიმდევრობისათვის ირჩევს იმ შემუშავების მოქმედებას, რაც მაქსიმალურს გახდის მიზნობრივი ფუნქციის მნიშვნელობას. მიზნობრივი ფუნქცია ფრომირდება აღქმის თანამიმდევრობითი აქტისა და ყველა აპრიორულად არსებული ინფორმაციის მხედველობაში მიღებით. აგენტი მტვერსასრუტის შემთხვევაში იმის გასარკვევად, არის თუ არა ეს აგენტი

ზემოთ მოყვანილი მახასიათებლების(კრიტერიუმების) მიხედვით უნდა ავლწეროთ აგენტის ფუნქციონირების პირობები, კერძოდ:

1. აგენტი უნდა დაჯილდოვდეს წინასწარ შედგენილი ქულების რაოდენობით, მაგალითად, დასუფთავებული ფართობის, მგზავროის კომფორტულობის ან სხვა მახასიათებლების მიხედვით. ამასთან , დაჯილდოების მექანიზმი უცვლელი უნდა იყოს აგენტის არსებობის ტელი, საკმაოდ ხანგრძლივი პერიოდის განმავლობაში;
2. აპრიორულად (წინასწარ) უნდა იყოს ცნობილი აგენტის მოქმედების არეალი (გარემო), მაგალითად, დასუფთავებული ოთახების რაოდენობა და მათი განლაგება , მგზავროის პირობები და სხვა. ამ დროს , აგენტმა თვითონ უნდა გაარკვიოს სად იმყოფება და განსაზღვროს შესაძლო გადაადგილების მიმართულებები;
3. აღქმის აქტებმა ანუ სენსორებიდან მიღებულმა ინფორმაციამ უნდა უზრუნველყოს მტვერსასრუტის ადგილმდებარეობის განსაზღვრა და ასევე , არის თუ არა მოცემული ოთახის დასუფთავებული.

თუ აგენტი მტვერსასრუტი ყველა ზემოთ ჩამოთვლი პირობას აკმაყოფილებს მაშინ ის რაციონალურია.

აგენტის რაციონალურობა არ ნიშნავს მისი მოქმედებების სრულყოფას. რაციონალურობა ნიშნავს მიზნობრივი ფუნქციის პრაქტიკულად შესაძლო მაქსიმალური მნიშვნელობების მიღწევას; სრულყოფა ნიშნავს აგენტის მიერ თეორიულად გაანგარიშებული მაქსიმუმის მიღწევას. უნდა აღინიშნოს, რომ ტერმინი „სრულყოფილი აგენტი“ გამოიყენება რაციონალური აგენტის მოქმედების შესაფასებლად. დამტკიცებულია, რომ სრულყოფილი აგენტის აგება შექმნა შეუძლებელია რეალური გარემოსათვის, ამიტომ ასეთ აგენტს, თეორიულის გარდა პრაქტიკული გამოყენება არა აქვს.

მეორე პუნქტის ეფექტურად შესრულებისათვის, რაციონალური აგენტი უნდა იყოს სწავლადი, რაც გულისხმობს წინასწარი (აპრიორული) ინფორმაციის შეფასების და გადამუშავების უნარს იმ მიზნით, რომ შესძლოს გარემოს საკუთარი აღწერის შექმნა და შესაბამისად, ამ გარემოში

უკეთესად მოქმედება. განვიხილოთ აგენტი მტვერსასრუტის მოქმედებების მაგალითზე სწავლადობისა და დამახსოვრების უნარების აუცილებლობა. რაციონალურ აგენტს თავისი ფუნქციის ეფექტურად შესასრულებლად უნდა ქონდეს წინასწარი, შემქმნელის მიერ მოპოვებული ინფორმაცია გარემოს შესახებ და ასევე, სწავლადობის უნარი. მაგალითად, თუ დასუფთავებული ოთახების რაოდენობა ბევრია და ზოგიერთი ოთახი ისვრება უფრო ხშირად, მაშინ აგენტი მტვერსასრუტი გამოცდილების შედეგად დააგროვებს ინფორმაციას ოთახების დანაგვიანების თაობაზე, მაგალითად, ზოგიერთი ოთახი ხშირად გამოიყენება, სხვები კი ნაკლებად, მიიღებს შესაბამის გადაწყვეტილებას რომ მეტად გამოიყენებადი ოთახი შეამოწმოს და დაასუფთაოს უფრო ხშირად, ხოლო ნაკლებად გამოიყენებადი ოთახები იშვიათად.

2.3. გარემოს აღწერა

რაციონალური აგენტის გარემოს აღსაწერად გამოიყენება ტერმინი - „პრობლემური გარემო“, რომელიც წარმოადგენს მიზნობრივ ფუნქციის უშუალო გარემოს, სენსორული სისტემების (გადამწოდების) და შმსრულებელი მექანიზმების ერთობლიობას.

რაციონალური აგენტის დაპროექტების პირველ ეტაპზე პრობლემური გარემოს მონაცემები (მახასიათებლები) რაც შეიძლება სრულად უნდა გაირკვეს და განისაზღვროს. ამ ინფორმაციაზე დამოკიდებულია რაციონალური აგენტის შესაძლებლობა და ეფექტურობა. განვიხილოთ აგენტის - ტაქსის მძღოლოს დაპროექტების მაგალითი. უნდა აღინიშნოს, რომ ასეთი ექსპერიმენტული სისტემა აშშ-ში შექმნილია სამარშუტო ტაქსის სახით, რომელიც 2000 კმ-ის გავლისას 98% იმართება ავტომატურად ხოლო 2%-ის განმავლობაში საჭიროა მძღოლის ჩარევა.

ავღწეროთ ავტომატური მძღოლის პრობლემური გარემო შინაარსობრივად:

აგენტ- ტაქსის მძღოლი;

მიზნობრივი ფუნქციის ატრიბუტები: უსაფრთხო, სწრაფი და კომფორტული მგზავრობა მოძრაობის წესების დაცვით, მოგების მაქსიმიზაცია; საწვავის ხარჯის მინიმიზაცია.

გარემოს ატრიბუტები: გზები და მათი ხარისხი; სხვა სატრანსპორტო საშუალებები; ფეხით მოსიარულეები; კლიენტები; სხვადასხვა დაბრკოლებები (ცხოველები, სარემონტო საშუალებები და ა.შ)

აღმასრულებელი მექანიზმები: საჭე, აქსელერატორი, მუხრუჭები, სასიგნალო სისტემები- ოპტიკური(მაშუქები) და სმენითი (კლაკსონი), დისპლეი;

სენსორული მექანიზმები : ვიდეო კამერები; მანძილის საზომი (ულტრაბგერითი), სპიდომეტრი, ადგილმდებარეონისა და ნავიგაციის გლობალური სისტემა, აქსელომეტრი, და ძრავის გადამწოდები.

მიზნობრივი ფუნქციის ფორმალიზება ერთერთი ურთულეი ამოცანაა იმის გამო, რომ გარემოს მახასიათებლები როგორც რაოდენობრივი, ისე თვისებრივია. მაგალითად მგზავრობის უსაფრთხოება და კომფორტულობა თვისებრივი მახასიათებლები, ხოლო მოგებისა და საწვავის ხარჯების პარამეტრები, რაოდენობრივ პარამეტრებს მიეკუთვნება კლიენტების სურვილებიც : მგზავრობის დროის ან ფასის მინიმიზაცია, რომელიც შედის მგზავრობის კომფორტულობის მახასიათებელში.

გარემო წარმოადგენს ობიექტების ერთობლივას რომლებთანაც აქვთ შეხება ტაქსის მძღოლს. გზების ხარისხი: სწორი გზა, სერპანტინი, გზა ორმოებით ან ბორცვებით, მოყინული, თოვლიანი საფარით; მოძრაობის სისტემა : მარჯვენა ან მარცხენა.

აღმასრულებელი მექანიზმები - იგივე ობიექტები და სისტემები, რაც აქვს მძღოლს ადამიანს, საჭე და მუხრუჭები, კლაქსონები და მაშუქები, სენსორული სისტემები - ავტომობილების თანამედროვე მოდულებში შეიძლება არსებობდეს(დაკვეთით) მონიტორი, რომელზედაც გამოიტანება ინფორმაცია ავტომობილის ადგილმდებარეობის შესახებ, აგრეთვე სხვა მახასიათებლები, ჰაერის წნევა საბურავებში, გამაგრებული სითხისა და ზეთის დონეების, მუხრუჭების ხუნდების ცვეთის ხარისხის და ა.შ

მონაცემები. სხვა ავტომობილებამდე ან ობიექტამდე მანძილის საზომები;
ნავიგაციის გლობალური სისტემა, შესაძლოა კლავიატურის ან მიკროფონის
არსებობა მგზავრებთან კომუნიკაციისათვის.

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).

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 ერთმანეთის ელემენტებს და იდეებს.

მანქანურ სწავლებაში, ობიექტების (სახეთა) ამოცნობა არის მოცემული შესატანი მნიშვნელობისთვის შესაბამისი მარკის მინიჭება. სტატისტიკურ ანალიზში სწორედ ამ მინზისთვის 1936 წელს მოხდა დისკრიმინანტის ანალიზის შემოღება. ობიექტების (სახეთა) ამოცნობის მაგალითია კლასიფიკაცია, რომელიც ცდილობს ყოველი შესატანი მნიშვნელობისთვის მიაწოდოს კონკრეტული კლასის ტიპი (მაგალითად, განსაზღვროს მოცემული ელექტრონული წერილი არის „სპამი“ თუ არა). თუმცა, ობიექტების (სახეთა) ამოცნობა არის უფრო ზოგადი პრობლემა, რომელიც მოიცავს სხვა ტიპის შედეგებსაც (output). სხვა მაგალითია რეგრესი, რომელიც ყოველ შენატანს ანიჭებს რეალურ მნიშვნელობის შედეგს. მიმდინარეობითი მარკირება გულისხმობს სხვადასხვა კლასის მინიჭებას მნიშვნელობების რიგის ყოველი ელემენტისთვის (მაგალითად: საუბრის დაკავშირების ნაწილი, რომელიც საუბრის ყოველ ნაწილს ანიჭებს თითოეულ სიტყვას) და მათ პარსირებას, რომელითაც შემომავალ წინადადებას ანიჭებს პარსირებულ ხეს, რითაც აღწერს წინადადების სინტაქსურ სტრუქტურას. ობიექტების (სახეთა) ამოცნობის ალგორითმი ზოგადად მიმართულია ყველა შეკითხვისთვის მისაღები პასუხის დაბრუნებაზე და „ყველაზე სავარაუდოს“ მეთოდით შენატანების სტატისტიკური სხვაობების გათვალისწინებაზე მათი ერთმანეთთან დაკავშირებისას. ეს არის ნიმუშების დაკავშირების ალგორითმის საპირისპირო, რომლებიც შენატანში ეძებენ ზუსტ (სრულ) დამთხვევას, უკვე არსებული ნიმუშების საფუძველზე. ნიმუშების დაკავშირების ალგორითმის ხშირად გამოყენებადი მაგალითია რეგულარული გამოსახულების ალგორითმი, რომელიც ეძებს კონკრეტული ტიპის ნიმუშებს ტექსტურ მონაცემებში და რომელიც გათვალისწინებულია უმარავი ტექსტური ედიტორის ძიების ფუნქციაში. ნიმუშის ამოცნობისგან განსხვავებით, ნიმუშის დაკავშირების ალგორითმი ზოგადად არ განიხილება მანქანური სწავლების ტიპად, თუმცა ნიმუშების დაკავშირების ალგორითმებმა ხანდახან შეიძლება ნიმუშების ამოცნობის ალგორითმის იდენტური ხარისხის შედეგი მოგვცენ.

3.1. ობიექტების ამოცნობა

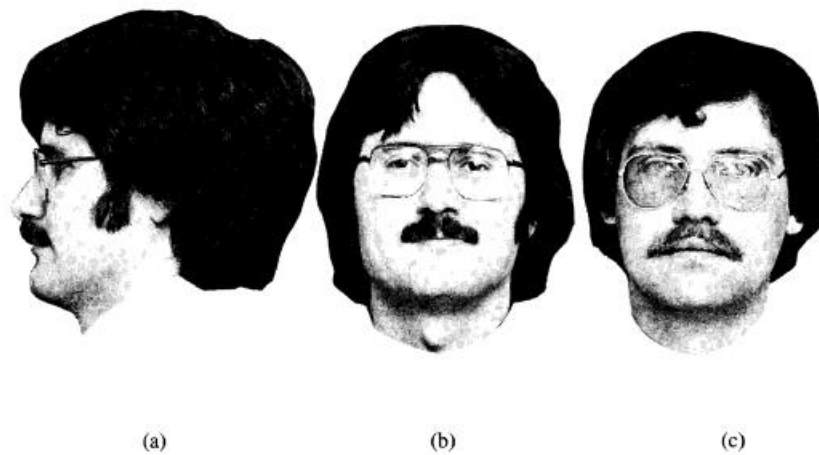
ადამიანების და ცხოველების ქმედებების მნიშვნელოვანი ნაწილი არის მათი შესაძლებლობა რათა ამოიცნონ ობიექტები, ცხოველები და ადამიანები, რომლებიც მნიშვნელოვანია მათი გადარჩენისთვის. ადამიანებს შეუძლიათ ამოიცნონ უამრავი სხვა ადამიანი ანბანის ასოები, ნაცნობი შენობები და ა.შ. ცხოველებს სჭირდებათ ტერიტორიების,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

თუ ჩვენ დავუშვებთ, რომ ადამიანისთვის ან ცხოველისთვის ხელმისაწვდომი ინფორმაცია არის ორ განზომილებიანი სურათი რეტინაზე, მაშინ უმალ წარმოიქმნება პრობლემა მათი ვიზუალური ამოცნობის ახსნასთან დაკავშირებით. მაგალითისთვის ავიღოთ ადამიანის უნარი რომ ამოცნოს 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 (2) ამისადა მიუხედავად, ჩვენ ყველა ვხვდებით რომ ყველა ნიმუში შეესაბამება ასო A-ს.



სურ: 2 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, იმისდა მიუხედავად რომ ისინი ერთმანეთისგან საკმაოდ განსხვავებულები შეიძლება იყოს (სურათი 3)



სურ 3. სურათზე ნაჩვენებია ერთი და იგივე პიროვნება სხვადასხვა რაკურსით

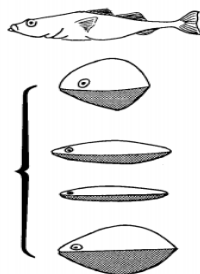
ზემოთ აღწერილი ორივე მაგალითი არის მასტიმულირებელი ტოლფასობა პრობლემა. თუ სტიმული, რომელიც აკონტროლებს ქმედებას არის შუქის ან რეტინაზე სურათის ნიმუში, მაშინ არებობს უსასრული რაოდენობა სურათებისა, რომლებსაც გააჩნიათ ტოლფასი ეფექტი, მაგრამ ხვდებიან სურათების განსხვავებულ სიმრავლეში. ობიექტის ამოცნობის უამრავი 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.

3.2. მოცნობის მარტივი მექანიზმები

ძალიან ბევრი ცხოველი, უფრო კონკრეტულად კი მწერები და თევზები, მასტიმულირებელი ტოლფასობის პრობლემას წყვეტენ იმ რაღაც მარტივის დადგენით, რომლის სურათსაც აქვს რაღაც საერთო კონკრეტული ობიექტთან. ამის კარგი მაგალითია თევზის ერთ-ერთი ჯიში, რომელთა მამრები აშენებენ ბუდეებს და მის დასაცავად ემუქრებიან ჯიშის სხვა მამრებს. შესაბამისად, მათ აქვს უნარი ამოიცნონ სხვა მეტოქე მამრები და განასხვავონ ისინი სხვა თევზებისგან და ობიექტებისგან, რომლებიც მის ბუდეს ჩაუვლიან. მეტოქე მამრების რეტინალური სურათები იქნება განსხვავებული მათი მანძილის და კუთხის მიხედვით და ამ სურათების მიხედვით მათ კლასიფიცირებას ჭირდება შესაბამისი განმსაზღვრელი პირობა.

როგორც 1951 წელს ტინბერგენმა განსაზღვრა, ამას თევზები ახერხებენ საკმაოდ მარტივი მექანიზმის გამოყენებით. დაკვირვებების შედეგად ტინბერგენმა დააფიქსირა რომ აღნიშნული ჯიშის მამრი თევზები უტევდნენ მათკენ მიმავალ ყველა ობიექტს, რომლებსაც ქონდათ წითელი მუდელი (სურათი 4).

ობიექტის ან ცხოველის მახასიათებელს (მაგალითად : წითელი მუცელი), რომელიც სხვა ცხოველისგან იწვევს რეაქციას ეძახიან გასაღებს ან სტიმულის ნიშანს და ის ძალიან ამარტივებს ამოცნობის პრობლემას.



სურ 4. თევზი უტევდა ყველა ობიექტს, რომელსაც ქონდათ წითელი მუცელი

სანამ ობიექტები და თევზები წითელი მუცლით არ არის ბევრ თევზების გარემოცვაში, ამ ჯიშის თევზებს შეუძლიათ ამ გასაღების გამოყენება მეთოქეების ამოსაცნობად და არაფერში არ ჭირდება ინფორმაცია სხვა თევზების დეტალურ აგებულებაზე და ფერებზე.

თევზების მიერ მეტოქე მამრების დადგენა არ ხდება რეტინალური სურათში მხოლოდ კონკრეტული ზომის წითელი წერტილის დადგენით . როგორც ტინბერგენმა დაადგინა ობიექტს, რომელსაც ქონდა წითელი წერტილი ზურგზე თევზები უტევდნენ უფრო იშვიათად ვიდრე მას რომელსაც იდენტური ზომის წითელი წერტილი ქონდა მუცელზე. ამასთან ერთად, მოდელებს რომლებსაც თავი ქონდათ მიმართული დაბლა (აგრესიული თევზის დგომა) უტევდნენ უფრო ხშირად, ვიდრე ჰორიზონტალურ მდგომარეობაში მყოფ მოდელებს. ამისდა მიუხედავად, უნიკალური თვისების ქონა საშუალებას იძლევა უფრო მარტივად და ეფექტურად მოხდეს ამოცნობა.

კიდევ ბევრი სხვა მაგალითი არსებობს ისეთი გასაღებებისა, რომლებიც ძალიან მნიშვნელოვანია ცხოველებში მათი ჯიშის (სახეობის) სხვა წევრების ამოსაცნობად და ჩვენ გაჩვენებთ ტინბერგენის მიერ გამოკვლეულ კიდევ 2 მაგალითს. პირველი მაგალითი აჩვენებს თუ როგორ ამოიცნობენ პეპლები თავისი ტიპის მდედრებს. ტინბერგენმა დაადგინა რომ მამრები მოდელების ფერების განურჩევლად მიიწევდნენ იმ მოდელებისკენ, რომლებიც მაღლა მიფრინავდნენ. მასტიმულირებელი გასაღები ამ შემთხვევაში აღმოჩნდა მოდელების მოძრაობის ნიმუში: მამრები მიფრინავდნენ მოდელისკენ იმ შემთხვევაში თუ ის მოძრაობდა მაღლა-დაბლა (არა სწორ-ხაზოვნად) ანუ იმეორებდა პეპლის მოძრაობებს. იმისდან მიუხედავად რომ მამრ პეპლებს დრო ეკარგებათ სხვა მამრი პეპლების დევნაში, ამ ამოცნობის მარტივი მექანიზმით მათ თავისი ჯიში არ ეშლებათ სხვა მწერებში.

ბოლო მაგალითი გვიჩვენებს თუ როგორ იცნობენ ბუდეში მყოფი შაშვის ბარტყები მათ მშობლებს. როცა მშობლებს ბუდეში საკვები მიაქვთ, ბარტყები მათ უბრუნდებიან და ალებენ პირს საკვების მისაღებად. როგორც ტინბერგენმა დააფიქსირა ბარტყები ანალოგიურად იქცეოდნენ ნებისმიერ ზომის და ფორმის მუქი სილუეტის მიმართ, რომელიც მათ ბუდესთან ახლოს ჩაივლიდა. სავარაუდოდ ამოცნობის ეს მარტივი მექანიზმი არის საკმარისად ეფექტური, რადგან იმის ალბათობა რომ აღნიშნული მასტიმულირებელი გასაღების თვისება ექნება მშობელი შაშვის გარდა სხვასაც არის ძალიან მცირე.

მასტიმულირებელი გასაღები ასევე მნიშვნელოვანია მსხვერპლის ამოცნობისას. ბაყაყები იკვებებიან მათთან ახლოს ჩაფრენილი მწერებით, რომლებსაც იჭერენ თვიანი წებოვანი ენით. ევარტმა 1974 წელს დაადგინა რომ ისინი მწერებს იცნობენ საკმაოდ მარტივი პირობით. ისინი იწვედნენ პატარა ოთხკუთხა ფორმის კარდონის ობიექტებზე, იმისდა მიუხედავად ეს ობიექტი მოძრაობდა თუ იყო უძრავ მდგომარეობაში. იმისდა მიუხედავად რომ ბაყაყები არჩევენ მსხვერპლს მათი ზომის და მოძრაობის სიჩქარის მიხედვით, შედეგებმა ცხადად აჩვენა რომ არჩევსას მათ არ შეუძლიათ მწერების გამოსახულების სხვა დეტალების გათვალისწინება.

შესაბამისად რიგ ცხოველებისთვის საჭირო ობიექტების ამოცნობის პრობლემა დადის კონკრეტული სპეციფიური მახასიათებლის (მასტიმულირებელი გასაღების) დადგენის პრობლემამდე. ეს მახასიათებელი შესაძლოა იყოს საკმაოდ მარტივი - მაგალითად თევზებში წითელი მუცლის დაფიქსირება. ამისდა მიუხედავად, აღნიშნული ტიპის დადგენის მექანიზმების წარმატება დამოკიდებულია მათ გარშემო არსებულ გარემოზე: როცა მეცნიერებმა მათ გარემოში გაუშვეს სათამაშო თევზები, ქალაქის პეპლები და კარდონის მწერები, ცხოველების მიერ დაშვებული ამოცნობის ვარაუდები უკვე დაირღვა.

სხვა ცხოველებს, განსაკუთრებით პრიმატებს უფრო მეტი თავისუფლება აქვთ თავიანთ დასკვნებში და ქმედებებში და შეუძლიათ ამოცნობა განახორციელონ უფრო კომპლექსური პირობების გამოყენებით. როგორც ამას შემდგომში ვნახავთ, ესეთ შემთხვევებში, ისევე როგორც ადამიანებში მასტიმულირებელი ტოლფასობის მიღების პრობლემა ხდება უფრო კომპლექსური.

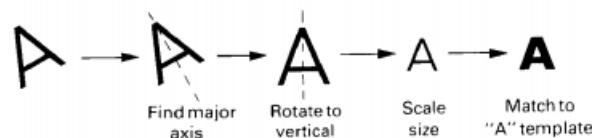
3.3. ამოცნობის უფრო კომპლექსური პროცესი

ჩვენ შეიძლება ვიკამათოთ რომ ადამიანებში ზოგიერთი ქმედება არის მასტიმულირებელი გასაღების კონტროლის ქვეშ. თუმცა კონკრეტულ სიტუაციებში ადამიანის ამ ქმედებების ჩამოყალიბება ხდება სწავლების პროცესის მეშვეობით. ჩვილი ბავშვი სწავლობს თავისი მშობლების სახის გარჩევას განათების, კუთხის და სახის გამომეტყველების და მიუხედავად დედა მისი შვილისთვის მაინც დარჩება დედიკოდ მას შემდეგ რაც ის დაიხვევს თმებს და მამა დარჩება მამიკოდ იმისდა მიუხედავად თუ მას ბოლო რამოდენიმე დღის განმავლობაში არ გაუპარსავს წვერი. მოგვიანებით ბავშვი ისწავლის უცხოებისგან მასწავლებლების და მეგობრების გარჩევას, ხოლო განათლების რთული პროცესის გავლის შემდეგ მას ექნება საშუალება ერთმანეთისგან განასხვაოს დამწერლობის სხვადასხვა ასოები. რა სახის შიდა აღქმა აძლევს საშუალებას რომ განასმოახდინოს ამ კომპლექსური კონფიგურაციების განსხვავება და რა სახის პროცესები მუშაობენ რეტინალურ სურათზე რათა მისცენ წვდომა მათ შიდა აღქმაზე? ეს არის შეკითხვები, რომლებზეც პასუხის გაცემას ცდილობს ადამიანების ნიმუშების და ობიექტების ამოცნობის სწავლებები.

3.4. შაბლონური დამთხვევა

ყველაზე მარტივი ახსნა თუ როგორ ხდება ასო-ციფრული ელემენტების ამოცნობა არის შაბლონური დამთხვევის მეშვეობით. ყოველი

ასოსთვის ან ციფრისთვის იარსებებს შაბლონი, რომელიც ინახება გრძელვადიან მეხსიერებაში. შემომავალი ნიმუში შედარდება შენახულ შაბლონების სიმრავლესთან და თუ იარსებებს საკმარისი დამთხვევა მაშინ ნიმუშს მიენიჭება შესაბამისი შაბლონის შესაბამისი კლასი. ესეთ სტრუქტურაში, ნიმუშის ზომაში ან კუთხეში მცირედი ცვლილება მოგვარდება სტანდარტიზაციის და ნორმალიზაციის პროცესების მეშვეობით. მაგალთად: შესაძლოა მოხდეს ყველა ნიმუშის ატრიალება ისე რომ მათი აბსიცა გახდეს ვერტიკალური. დამატებით შესაძლოა გახდეს საჭირო სურათის წინასწარი დამუშავება / გასუფთავება, რადგან როგორც ადამიანებს ისევე ცხოველებს შეუძლიათ ნიმუშის დაზიანებული ან გაურკვეველი ნაწილების ამოცნობა / შევსება (სურათი 5).



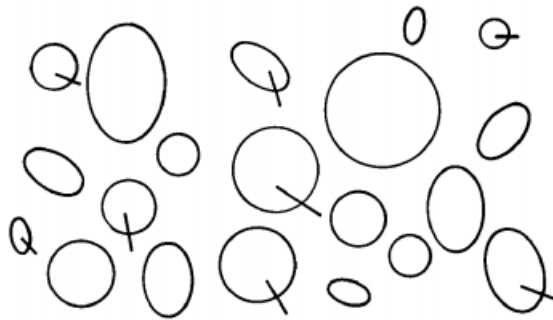
სურ 5. სანამ დაიწყება შაბლონის დამთხვევა ნიმუშს შესაძლოა შეუცვალონ ორიენტაცია, დაატრიალნ ვერტიკალურად ან გაზარდონ ზომაში.

ამ ტიპის შაბლონური დამთხვევის სქემა იმ შემთხვევაში იმუშავებდა თუ ნორმალიზაციის პროცედურა იქნებოდა საკმარისი შედეგების ნიმუშების ცალსახად გარდასაქმნელად. სამწუხაროდ აღნიშნულის მიღწევა არის თითქმის შეუძლებელი, ასო-რიცხვითი ელემენტების მარტივ შემთხვევაშიც კი. ასო A-ს შაბლონი შესაძლოა უფრო მეტად დაემთხვეს ასო R-ის შაბლონს ვიდრე თავს საკუთარ შაბლონს და პირიქით (იხილეთ სურათი 6).



სურ 6. სურათზე ნაჩვენებია თუ როგორ შეიძლება ასო A-ს შაბლონი დაემთხვეს ასო R-ს შაბლონს და პირიქით.

ის ხაზი რომელიც ასო Q-ს ანსხვავებს ასო O-სგან შესაძლოა განთავსებული იყოს რამოდენიმე წერტილში (იხილეთ სურათი 7).



სურ 7. რით ხდება Q-ს და O-ს ერმანეთისგან განსხვავება? ეს არც წრის ფორმა არაა, არც Q-ს ხაზი, რადგან ის შეიძლება რამოდენიმე წერტილში იყოს განთავსებული.

შესაბამისად არ შეიძლება იმის თქმა რომ ბავშვები, ასოებს და რიცხვებს შეისწავლიან სწორედ ამ სექმის გამოყენებით.

თუმცა შაბლონური დამთხვევის სქემა მუშაობს ძალიან წარმატებულად თუ ამოსაცნობი სიმბოლოები არის სტატიკურად (ხისტად) გაწერილი. შედეგად კომპიუტერები, რომლებსაც ევალებათ ჩეკის ბოლოში დაწერილი ანგარიშის ნომრების წაკითხვა მარტივად ახორციელებენ ამ ოპერაციას მათში ჩაწერილი შაბლონებთან შედარებით. ამ მაგალითში რამე სახის გაურკვევლობის თვიდან ასაცილებლად სიმბოლოებს მუდამ აქვთ

ერთი ფორმა და არიან ერთმანეთისგან მაქსიმალურად განსხვავებულები. სიმბოლოები, რომლებსაც ადამიანი ცნობს არ არიან ესე ერთმანეთისგან განსხვავებულები. თუ ჩვენ დავუფიქრდებით თუ როგორ ახერხებს ადამიანი A-ს და R-ს ან Q-ს და O-ს ერთმანეთისგან გარჩევას დავინახავთ რომ უნდა არსებობდეს გარკვეული კრიტიკული თვისებები, რომლითაც ამ სიმბოლოებს ერთმანეთისგან განვასხვავებდით. ხაზი რომელიც კვთავს ასო Q-ს არის ძალიან მნიშვნელოვანი O-სგან განმასხვავებელი ელემენტი. ამავდროულად ამ ელემენტების განსხვავებისას წრის ფორმას უფრო ნაკლები დატვირთვა აქვს. შესაძლოა მოდელი, რომლითაც ხდება თვისებების კომბინაციების დადგენა უფრო წარმატებული იყოს ვიდრე შაბლონებზე დაფუძნებული მოდელის.

თვისებების ანალიზის მოდელი იყო ძალიან პოპულარული ფსიქოლოგიასა და კომპიუტერულ მეცნიერებებში 1960-იანი წლების განმავლობაში. ამ დროს შეიქმნა სავარაუდოდ ფსიქოლოგიისთვის ყველაზე წარმატებული სელფრიჯის პანდემონიუმის სისტემა, რომელიც თვიდან წარმოადგენდა კომპიუტერულ პროგრამას, რომელსაც შეეძლოა მორზე სიგნალების ამოცნობა, მაგრამ შემდგომში პოპულარობა ჰპოვა როგორც ასორიცხვითი სიმბოლოების ამომცნობმა მოდელმა, რომელიც შექმნეს ნეისერმა (1967) და ლინდეიმ და ნრომანმა (1972). პანდემონიუმის სისტემის ილუსტრაცია ნაჩვენებია 7 სურათზე.

სისტემა შედგენა სხვადასხვა რაოდენობის და სხვადასხვა კლასისი „დემონებისგან“. ჩვენი მიზნებისთვის ყველაზე მნიშვნელოვანი არიან თვისების დემონი და შემეცნებითი დემონი. თვისების დემონი არჩევითან პახუობს როცა მას მიეოდება კონკრეტული ლოკალური კონფიგურაცია (მარჯვენა კუთხე, ვერტიკალური ხაზი...). ხოლო შემეცნებითი დემონი, წარმოადგენს კონკრეტულ ასოებს და თვისებების დემონთან ემბეს თვისებების კონკრეტულ კომბინაციას. შედეგად შემეცნებითი დემონი, რომელიც წარმოადგენს ასო H-ს შესაძლოა ემბდეს ორ ვერტიკალურ და ერთ ჰორიზონტალურ ხაზს და დამაბით ითოხვდეს ოთხი მართი კუთხის არსებობას. რაც უფრო მეტი საჭირო თვისებების მოძიება მოხდება, უფრო

ხმამალა შეძლებს შემეცნებითი დემონი იმის ყვირილს, რომ ის წარმოადგენს დამთხვევის უმაღლეს ალბათობას. ხოლო არჩევანის გამკეთებელი დემონი ირჩევს ისეთ შემეცნებით დემონს, რომელიც ყველაზე ხმა მალა ყვირის. შედეგად, ამ სისტემაში თითოეული სიმბოლო წარმოდგენილია როგორც კრიტიკული თვისებების მქონე სიმრავლე და ყოველი სურათის პროცესირება ხდება იერარქიული სტრუქტურით აბსტრაქციის სხვადასხვა დონეებზე. ამ მოდელში ბარლოუ და სხვები გულისხმობდნენ რომ ტვინის მარტივი უჯრედები ასრულებდნენ პანდომინუმის სისტემის თვისების დემონების მოვალეობას და აწვდიდნენ ინფორმაციას სხვა უჯრედებს, რომლებიც შესაბამისად მოექცეოდნენ გადაწოდებულ აბსტრაქტულ პარამეტრებს. ესეთი ჰიპოთეტურ უჯრედებს ეძახდნენ „ბებია უჯრედებს“ ან „ყვითელი ვოლცვაგენის დეტექტორებს“, რათა ყურადღება გაემხავილებინათ იმ აბსტრაქტულ მასტიმულირებელ გარემოებებზე, რომლებზეც ისინი მოქმედებდნენ.

პანდომინუმის სისტემას შეუძლია სხვადასხვა თვისებას მიანიჭოს სხვადასხვა რანგი, იმისდა მიხედვით თუ რამდენად კარგად ჯდება ეს თვისებები კონკრეტულ ნიმუშებში. ამ სისტემის გარდა არსებობენ კიდევ სხვა სისტემები, რომლებიც ზემოთ აღწერილი სისტემის ანალოგიურად სწავლობენ რანგის ცვლილებას სტიმულსა და სტიმულზე განხორციელებული რეაქციის მიხედვით. პანდომინუმის ტიპის სისტემას ასევე შეუძლია გარკვეული შინაარსიდან შესაბამისი დასკვნების გამოტანა. მაგალითისთვის სურათ 2.7-ზე ნაჩვენებია მაგალითი, თუ როგორ მოქმედებს შინაარსი ადამიანის ასოების აღქმის უნარზე.

TAE CAT

სურ: 8 (შინაარსი ადამიანის ასოების აღქმის უნარი)

იმისდა მიხედვით თუ რომელი ასოების გარემოცვაშია მოქცეული, ერთი და იგივე ფორმა შესაძლოა იქნას დანახული როგორც ასო H ასევე ასო A-ად. პანდომინუმის სისტემაში, ამ ფაქტებზე დაყრდნობით ჩვენ შეგვიძლია

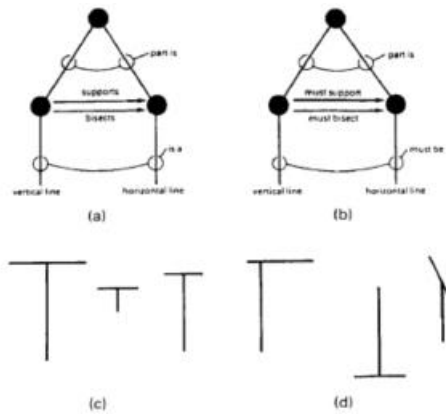
ზედა - დონის დემონებს უფლება დავრთოთ რომ გამოაცოცხლონ ის ქვედა დონის დემონები, რომლებიც შეესაბამებიან შესაბამის ნიმუშს რათა მათ უფრო ხმამაღლა დაუძახონ გადაწყვეტილების მიმღებ დემონს.

თუმცა როგორც ადამიანის ნიმუშის და ობიექტების ამოცნობის ზოგადი მოდელი პანდომონიუმის სისტემა არ არის დამაკმაყოფილებელი. ძირითადად იმიტომ რომ ის დაფუძნებულია ნიმუშებში არსებული თვისებების სიმრავლის აღწერებზე, რომლებიც თავის მხრივ არიან მინი-შაბლონების ანალოგიები. ერთ-ერთი საბაზი რის გამოც პანდომონიუმი იყო პოპულარული იყო მისი თავსებდაობა ნეუროფსიქოლოგიასთან. თუმცა ჩვენ უკვე ვნახეთ რომ ცალი უჯრედი ვერ ჩაითვლება „თვისების დამდგენად“. ამას მართალია არ აქვს მნიშვნელობა წმინდა ფსიქოლოგიური და კომპიუტერული მეცნიერებების თვალთაზრისით, მაგრამ არსებობს სხვა პრობლემები. თვისებების აღმწერ სიას არ შეუძლია სრულად გადმოსცეს დანახული დეტალების სრული სტრუქტურა. შესაბამისად, პანდომონიუმის სისტემას შესაძლოა სიმბოლოები შეემალოს უფრო იშვიათად გამოყენებად სიმბოლოებთან. მაგალითად : E და Σ ანუ ისეთი ტიპის შეცდომები, რომლებიც ზოგადად ადამიანებს არ შეეშლებოდათ. ამასთან ერთად, პანდომონიუმის სისტემა არ ითვალისწინებს ერთი ნიმუშის სხვადასხვა მაგალითის ინფორმაციას ანუ გადაწყვეტილების მიმღები დემონის შედეგი მუდამ იქნება ერთი იმისდა მიუხედავად ასო A-ს რომელი ვერსია იყო ნაჩვენები. საჭიროა არსებობდეს გზა, რომლითაც ჩვენ შევძლებდით განგვესხვავებინა სხვადასხვა ნიმუშების სხვაობა და ამასთან ერთად შევძლებდით ერთიდაიგივე ტიპის ნიმუშების დაჯგუფებას.

3.5. სტრუქტურული აღმწერები

ზოგადი და მოხერხებული ფორმატი ადამიანის ნიმუშის და ობიექტების ამოსაცნობად მოცემულია სტრუქტურული აღმწერი ენების მეშვეობით. სტრუქტურული აღმწერები არ ეხმარებიან თეორიას რომელიც

ადგენს თუ როგორ ხდება ცნობა, ისინი უბრალოდ გვაძლევენ გამოხატვის სწორ ტიპს, რომელზეც შესაძლებელია აღნიშნული თეორიის აგება. სტრუქტურული აღმწერები გვაძლევენ თეორიების სიმრავლეს კონკრეტული კონფიგურაციის შესახებ. ესეთი თეორიები აღწერენ კონფიგურაციის კომპონენტის ბუნებას და წვრილმანდებიან ამ ნაწილების სტრუქტურული განლაგებებით. შესაბამისად ასო T-ს სტრუქტურული აღწერა, შესაძლოა გამოიყურებოდეს შემდეგნაირად (სურ 9):

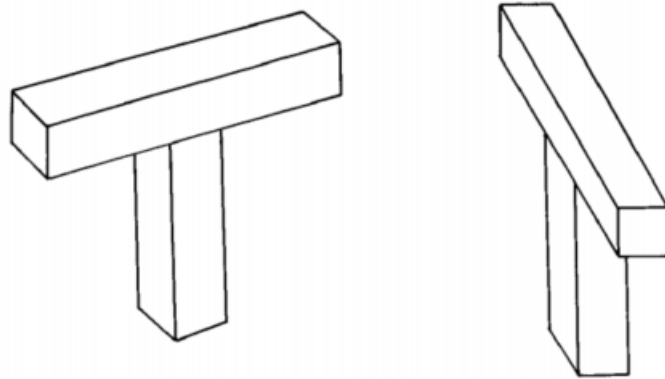


სურ: 9 ასო T-ს სტრუქტურული აღწერა.

სტრუქტურული აღწერის ენის გამოყენებით შესაძლებელია კონკრეტული კონცეფციის და კატეგორიის მოდელის აგება, რომელიც შეძლებს ნებისმიერი შემომავალი ობიექტისთვის შესაბამისი კლასის დადგენას. ესეთი მოდელები ავლენენ სტრუქტურის აუცილებელ თვისებებს, მაგრამ შესაძლოა ნაკლებ ყურადღებას უთმობდეს სხვა დეტალებს. შესაბამისად ასო T-ს მოდელი შესაძლოა გამოიყურებოდეს როგორც სურათი 4.8 b ნახაზი, ხოლო სურათი 4.8 c-ზე ნაჩვენებია ის მოდელები, რომლებიც შესაძლოა კატეგორიზირდეს როგორც ასო T, ხოლო იგივე ნახატის c-ში ნაჩვენებია მოდელები, რომლებიც არ გამოდგება ასო T-ს აღწერად.

სტრუქტურული აღმწერების მორგება უფრო მარტივია ობიექტების ამოცნობაზე ვიდრე შაბლონების ან თვისებების გამოსახულებებზე. ობიექტის სურათი შესაძლოა აღწერილი იყოს სხვადასხვა დონის

აბსტრაქციის მქონე სტრუქტურული აღმწერების მიხედვით. თითოეული ეს დონე წარმოადგენს აღწერის შესაძლო „დომენებს“. აღნიშნული მაგალითის განსახილველად ავიღოთ სურათი 10.



სურ: 10. ასო T

ეს სურათი შესაძლოა აღწერილი იქნას სხვადასხვა უნიკალური დომენის მიხედვით, რომელთა ზოგადი კატეგორიზირება შესაძლებელია მოხდეს როგორც „2 განზომილებიანი“ ან „3 განზომილებიანი“. 2 განზომილებიანი აღმწერი გადმოსცემს მიცემულ სურათს და ამ სურათის აღწერა შესაძლოა მოხდეს ძალიან აბსტრაქტული და ზოგადი ტერმინებით. ის შესაძლოა აღიწეროს როგორც სხვადასხვა სიკაშკაშის მქონე წერტილების ერთობლიობა, ან როგორც ხაზების ერთობლიობა და ა.შ. მნიშვნელობა არ აქვს თუ როგორი და რა დონის „2 განზომილებიანი“ აღწერა იქნება მოცემული ზემოთ ნაჩვენები ფიგურების, ამ აღწერების მიხედვით ზემოთ ნაჩვენები ორი ფიგურის გამოხატულება იქნება ერთმანეთისგან ძალიან განსხვავებული. ეს მხოლოდ „3 განზომილებიანი“ აღწერის მოდელშია შესაძლებელი რომ ამ ორ ფიგურას შორის მოხდეს ტოლფარდობის დაფიქსირება, რადგან ამ ტიპის აღმწერები გადმოსცემენ ობიექტის წყობის და აგებულების აღწერებს. შესაბამისად, ზემოთ ნაჩვენები 2 ნახაზი

ერთამენთის ანალოგიურია მხოლოდ ობიექტების აღწერის დონეზე, რომელიც არ არის დამოკიდებული დაკვირვების წერტილზე.

შედეგად ერთი და იგივე ობიექტის სხვადასხვა ვიზუალს სურათების დომეინში ექნება განსხვავებული სტრუქტურული აღწერა, მაგრამ იქნებიან ანალოგიურები ობიექტების დომეინში. ჩვენი პრობლემა ეხლა მდგომარეობს იმაში რომ გავარკვიოთ როგორ უნდა მოხდეს სტრუქტურული აღწერების აგება 3D დონეზე, მათი შედარება და სასურველი შედეგის მიღება.

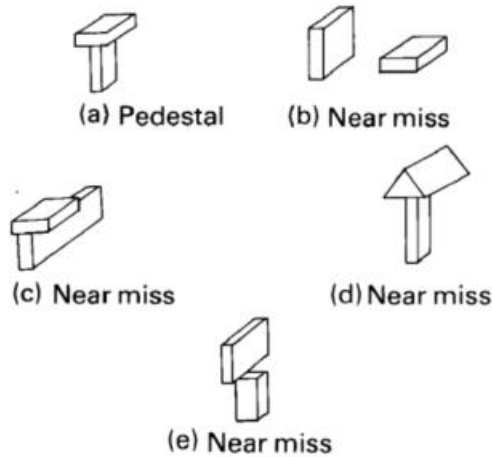
ვინსტონმა 1975 წელს, წარმოადგინა ადრეული ილუსტრაცია იმისა თუ როგორ არის შესაძლებელი სტრუქტურული აღმწერების გამოყენება ობიექტის ამოცნობისას. აღნიშნულით მან გვაჩვენა თუ როგორ შეიძლება მაგალითებზე სწავლების გზით მოხდეს ობიექტებზე საერთო წარმოდგენის ჩამოყალიბება. მან შექმნა პროგრამა რომელსაც შეძლო ესწავლა მარტივი სათამაშოების სტრუქტურის ცნობა (სურ:11).



სურ 11: სათამაშოების ნიმუში რომელთა ცნობაც შეეძლო ვინსტონის პროგრამას.

კომპიუტერულ პროგრამას გადაეწოდებოდა თითოეული ფორმის სათამაშოს ნიმუში და ასევე ამ ფორმასთან მიახლოებული ნიმუშები. სწავლების პროცესი იყო შემდგომი:

თავდაპირველად მას გადაეწოდებოდა შესასწავლი სათამაშოს ასლი ნიმუში (დავუშვათ რომ ეს შესასწავლი ობიექტი იყო სურ:4.10-ზე ნაჩვენები მეორე ფიგურა, პედესტალი). მას პროგრამა მიაჩვენებდა შესაბამის სტრუქტურულ აღწერას (სურ: 12 ნახაზი a).



სურ: 12 ტრენინგის მიმდევრობა

შემდეგში პროგრამას გადაეწოდებოდა პედესტალი ფორმასთან მიახლოებული ფორმის მქონე ფიგურები, რომლებიც არ წარმოადგენენ პედესტალს (სურ: 12 ნახაზი b-e). მნიშვნელოვანი ამ გადაცემულ სათამაშოებში იყო ის, რომ ყველა ფიგურა შედგებოდა ორი ნაწილისგან: აგურისგან და დაფისგან. შესაბამისად სისტემას პედესტალის ფორმის მქონე ობიექტის და არა-პედესტალის ფორმის მქონე ობიექტების შედარებით, უკვე შეეძლო აეგო პედესტალის მოდელი, რომელშიც ნაჩვენები იქნებოდა თუ რა პირობებს უნდა აკმაყოფილებდეს ერთმანეთთან მიმართებაში ფიგურის ნაწილები, რათა ფიგურას დაერქვას პედესტალი. ობიექტს რომ დაერქვას პედესტალი სავალდებულოა რომ „მისი ერთი ნაწილი მყარდებოდეს მისი მეორე ნაწილისგან“, „ჰორიზონტალურ მდგომარეობაში მყოფი დაფა უნდა იდგეს ვერტიკალურ მდგომარეობაში მყოფ აგურზე“...

ჩვენ სპეციალურად გაჩვენეთ სწავლების მაგალითი პედესტალის ნიმუშზე, რადგან მას გააჩნია ასო T-სთან მიახლოებული 3 განზომილებიანი ფორმა, შესაბამისად ასო T-ს 3 განზომილებიანი სტრუქტურული აღწერა იქნება პედესტალის აღწერასთან ძალიან მიახლოებული მხოლოდ ერთი სხვაობით. პედესტალის შემთხვევაში მისი შემდგენი ნაწილებიც თავის მხრივ წარმოადგენენ 3 განზომილებიან ობიექტს, ხოლო ასო T-ს შემთხვევაში მისი შემდგენები არიან ჰორიზონტალური და ვერტიკალური ხაზები.

დავუბრუნდეთ ვინსტონის პროგრამას. პედესტალის ამოცნობის იდენტური პროცესის გამოყენება შესაძლებელია მოხდეს სხვა ფიგურების, მაგალითად სურათზე ნაჩვენები სახლის ან არკის სათამაშოს ობიექტების დასადგენად.

შედეგად ვხედავთ რომ ვინსტონის ობიექტი ოპერირებს ობიექტების დომეინში. მას შეუძლია მიიღოს აგურის ან დაფის პროექცია და მათ მიანიჭოს შესაბამისი კატეგორია. თუმცა თავად აგურის და დაფის სტრუქტურული აღწერა უნდა იყოს განსაზღვრული პროგრამის სხვა დონეზე. სწორედ ეს დაბალი დონის პროგრამები, რომლებიც ამუშავებენ აღწერებს და გადასცემენ ინფორმაციას შემსწავლელ პროგრამას წარმოადგენენ ვინსტონის სისტემის პრობლემას. ჩვენ გვჭირდება უკეტესი გზა, რომლითაც ვინსტონის ტიპის პროცედურებისთვის შევძლებდით საკმარის რაოდენობის სტრუქტურული აღწერების მომარაგებას, რომელიც არ იქნებოდა შეზღუდული ხელოვნური სამყაროს ფარგლებში.

ამისთვის ჩვენ უნდა დავუბრუნდეთ ობიექტების ამოცნობის ფუნდამენტალურ პრობლემას. გავიხსენოთ რომ ობიექტების პროექცია რეტინაზე დამოკიდებულია დაკვირვების წერტილზე. შესაბამისად თუ ჩვენ დაეყრდნობოდით დამკვირვებლის კუთხიდან აგებულ ობიექტის აღწერაზე, ჩვენ დაგვჭირდებოდა ბევრი სხვადასხვა დაკვირვების წერტილების შესაბამისი აღწერების შენახვა.

3.6. 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 მივაღწევთ 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,

$\text{Dist}(g(\quad), g(\quad))$ 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

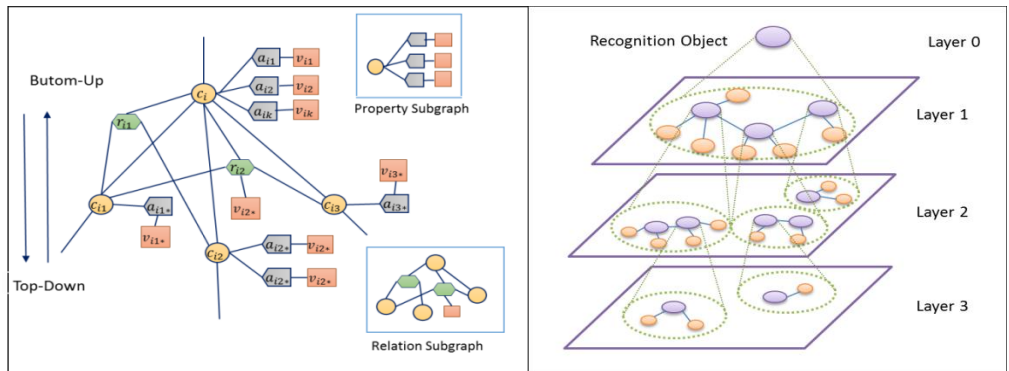
რეკურსიის დასრულების პირობა : $\sigma(c_i) = \emptyset$

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

-დაფუნქციონირებადი არქიტექტურა, დასკვნების კეთების და სემანტიკური სიახლოვის განსაზღვრის მექანიზმებით



სურ. 13 ამოცნობის არქიტექტურა

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.

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 როგორიცაა მათემატიკური კალკულაციები რასაც იგი ძან 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 გადართვის ოპერაციები	$\approx 10^{12}$	$\approx 10^{10}$

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 ამავედროულად 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.

4.1. ხელოვნური ნეირონული ქსელი

ხელოვნური ნეირონული ქსელი არის ინფორმაციის დამუშავების პარადიგმა, რომლის შტაგონების სისტემა ბიოლოგიური ნერვიული სისტემაა, როგორცაა ტვინი და ინფორმაციის დამუშავება. ძირითადი ელემენტი ამ პარადიგმის არის სტრუქტურა ინფორმაციის პროცესინგის. იგი შედგება დიდი რაოდენობით ერთმანეთთან დაკავშირებული გადამამუშავებელი ელემენტებისგან (ნეირონებისგან) რომლებიც მუშაობენ პრობლემის გადაწყვეტისთვის. ხელოვნური ნეირონული ქსელი მსგავსად ადამიანებისა სწავლობს მაგალითებზე. ხელოვნური ნეირონული ქსელი კონფიგურირდება სპეციფიური აპლიკაციისთვის როგორცაა ობიექტების(სახეთა) ამოცნობა ან მონაცემთა კლასიფიკაცია და ეს ყველაფერი ხდება სწავლის პროცესის განმავლობაში. სწავლის პროცესი ბიოლოგიურ სისტემაში მოიცავს მოწესრიგებულ კავშირებს რომლებიც არსებობენ ნეირონებს შორის. ეს მსგავსად არის ხელოვნურ ნეირონულ ქსელში.

ნეირონული ქსელის განვითარება მოხდა ბოლო დროს. თუმცა ეს სფერი კომპიუტერის არსებობამდე ჩამოყალიბდა. პირველი ხელოვნური ნეირონული ქსელი იქნა ნახსენები და ჩამოყალიბებული 1943 წელს ნეუროფიზიკოსის ვაერნ მუკულოჩის და ლოგიკოსის ვალტერ პისტის მიერ. მაგრამ ტექნოლოგიების იმ დროინდელმა განუვითარებლობამ არ მისცა საშუალება მათ გაეკეთებინათ უფრო მეტი.

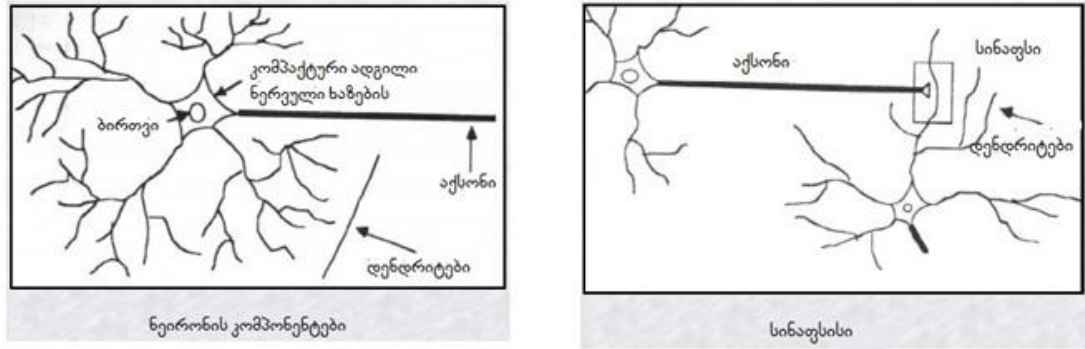
ნეირონული ქსელი, და მისი შესანიშნავი შესაძლებლობა შესაძლებელია გამოყენებული იქნას რთული და ბუნდოვანი მონაცემებისთვის, ობიექტების(სახეთა) გამოცნობისთვის რასაც ადამიანი ვერ ახერხებს. ასევე ამ მიდგომას გააჩნია სხვა უპირატესობებიც :

1. ადაპტირებული სწავლება - შესაძლებლობა თუ როგორ შევასრულოთ ამოცანები მოცემულ მონაცემებზე დაყრდნობით რომელიც მოცემული გვქვს სასწავლო პროცესის დროს , ან გამოვიყენოთ თავდაპირველი გამოცდილება.
2. თვითორგანიზება - ხელოვნურ ნეირონულ ქსელს შეუძლია თვითონ გაუკეთოს ორგანიზება ან წარმოდგენა ინფორმაციას რომელსაც იგიღებულობს სწავლის პროცესის დროს.
3. რეალურ დროში ოპერირების უნარი - ხელოვნური ნეირონული ქსელი გამოთვლები შესაძლებელია განხორციელდეს პარალელურად, შესაძლებელია შეიქმნას სპეციალური მოწყობილობა რომელიც ისარგებლებს ამ შესაძლებლობით.
4. შემწყნარებლობის მეშვეობით ვიდრე ჭარბი ინფორმაციის კოდირება იწვევს კოდის განადგურებას ქსელში, ზოგიერთმა ქსელმა შესაძლოა შეინარჩუნოს მდგომარეობა მანამ სანამ მთავარი ქსელი ა განადგურდება.

4.2. ადამიანური და ხელოვნური ნეირონული ქსელი - მსგავსებების გამოძიება

როგორ სწავლობს ადამიანის ტვინი?

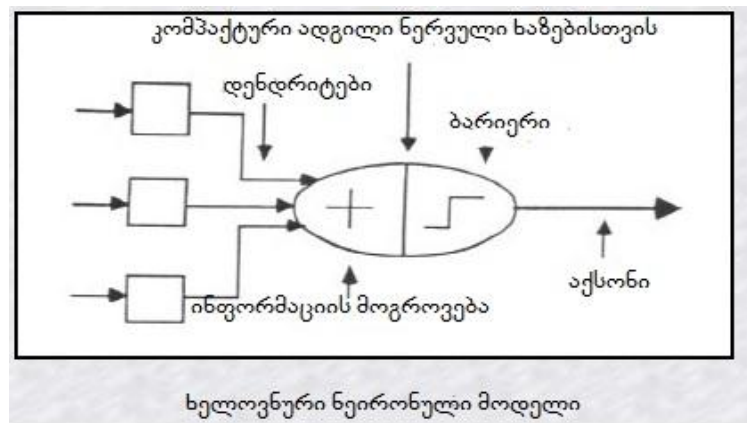
ჯერ კიდევ ბევრი რამეა უცნობი იმასთან დაკავშირებით თუ როგორ ამზადებს ტვინი ინფორმაციას რათა მოახდინოს შემდეგ მისი დამუშავება, მაგრამ მსგავსი თეორიები მრავლადაა. ადამიანის ტვინიში , ტიპიური ნეირონები აგროვებენ სიგნალებს სხვა ადგილმდებარეობიდან რომელსაც ქვია დენდრიტები. ნეირონები აგზავნიან იმპულსებს ელექტრონული საქმიანობის შესახებ , ეს სტენდი ცნობილია როგორც აქსიომა.



სურ. 14 ნეირონის კომპონენტები და სინაფსისი

4.3. ადამიანების ნეირონული ქსელიდან ხელოვნურ ნეირონულ ქსელამდე

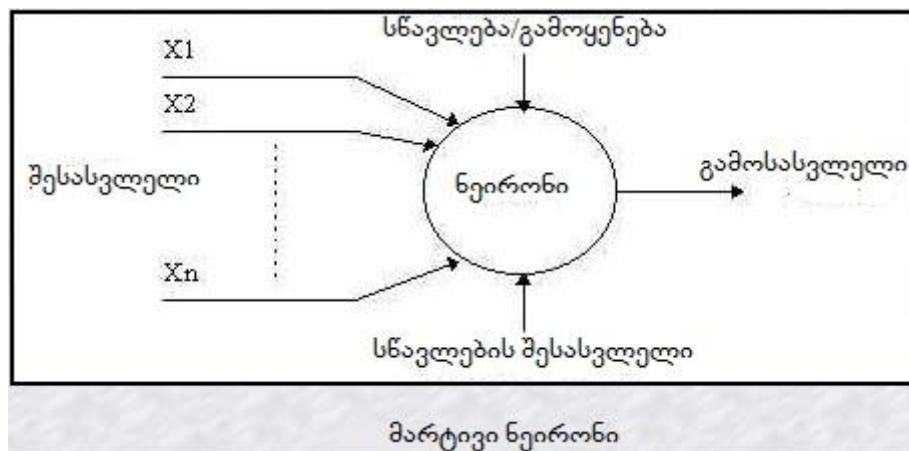
რომ ჩავატაროთ იგივე ნეირონული ქსელი ჩვენ ვცდილობთ შევამციროთ ძირითადი არსებითი ნიშნები ნეირონების და მათი კავშირები. ჩვენ ტიპიურად ვაპროგრამირებთ კომპიუტერს რათა მოვახდინოთ მსგავსი თვისებების სიმულაცია. მიუხედავად იმისა რო ჩვენი ცოდნა ნეირონებთან დაკავშირებით არასრულია და ჩვენი გამოთვლის ძალა არის შეზღუდული, ჩვენი მოდელი საჭიროებისამებრ გაზრდილია იდეალიზაციისთვის რათა მოხდეს რეალურ ნეირონულ ქსელთან მიახლოება.



სურ. 15 ხელოვნური ნეირონული მოდელი

4.4. მარტივი ნეირონი

ხელოვნური ნეირონი არის მოწყობილობა მრავალი შემავალი მონაცემით და ერთი გამომავალი პასუხით. ნეირონს აქვს ორი რეჟიმი ოპერაციის : სასწავლო რეჟიმი და გამოყენების რეჟიმი. სასწავლო რეჟიმში ნეირონი ცდილობს მოემზადოს მუშაობისთვის(ან არა), ტრენინგი დამოკიდებული კერძოდ შემავალ ობიექტებზე(ნიმუშებზე). გამოყენების რეჟიმში , როდესაც ნასწავლი ნიმუში აღმოჩნდა შემავალ მონაცემად, მასთან დაკავშირებული პასუხი დამოკიდებულია მიმდინარე პასუხთან. თუ შემავალი ობიექტი(ნიმუში) არ ეკუთვნის სიას ნასწავლი ობიექტებისა(ნიმუშებისა) მაშინ სამუშაო წესი იწყებს დადგენას დაიწყოს მუშაობა თუ არა.



სურ.16 მარტივი ნეირონი

4.5. გასროლის წესები

გასროლის წესი ნევრულ ქსელებში მნიშვნელოვანი ცნებაა და მათი მოქნილობის საწინდარია. გასროლის წესი განსაზღვრავს უნდა გაისროლოს

თუ არა ნეირონმა ნებისმიერი მოცემულობისთვის. ის ეხება ყველანაირი მოცემულობის ნიმუშს და არა მხოლოდ ისეთებს, რომლებზეც კვანძი გაწვრთნილია.

მარტივი გასროლის წესის იმპლემენტაცია შესაძლებელია ჰამინგის მანძილის ტექნიკით. წესი განისაზღვრება შემდეგნაირად:

განვიხილოთ კვანძის საწვრთნელი ნიმუშების ერთობლიობა, მათ შორის, რომლებიც იწვევენ გასროლას (1-გაწვრთნილი ნიმუშები) და რომლებიც არ ასროლინებენ (0-გაწვრთნილი ნიმუშები). შემდგომ, ყველა ისეთი ნიმუში, რომელიც არ შედის არსებულ ერთობლიობაში იწვევს კვანძზე გასროლას თუ მას უფრო მეტი საერთო ელემენტი აქვს „ყველაზე ახლოს მყოფ“ 1-გაწვრთნილ, ვიდრე 0-გაწვრთნილ ნიმუშთან. ფრეს შემთხვევაში ნიმუში რჩება განუსაზღვრელი.

მაგალითად, 3 მონაცემიანი ნეირონი გაწვრთნილია 1-ის დაბრუნებაზე როდესაც მონაცემები (X1, X2 და X3) არის 111 ან 101 და 0-ის დაბრუნებაზე, როდესაც მონაცემებია 000 ან 001. მაშინ, სანამ მას გასროლის წესს მივუყენებთ, ჭეშმარიტების ცხრილი გამოიყურება ასე:

X1:		0	0	0	0	1	1	1	1
X2:		0	0	1	1	0	0	1	1
X3:		0	1	0	1	0	1	0	1
OUT:		0	0	0/1	0/1	0/1	1	0/1	1

გასროლის წესის მიყენების მაგალითისთვის განვიხილოთ ნიმუში 010. ის განსხვავდება 000-სგან 1 ელემენტით, 001-სგან 2 ელემენტით, 101-სგან 3 ელემენტით, ხოლო 111-სგან 2 ელემენტით. შესაბამისად, „უახლოესი“ ნიმუში არის 000, რომელიც 0-გაწვრთნილ სიმრავლეს მიეკუთვნება. ანუ გასროლის წესი მოითხოვს, რომ ნეირონმა არ უნდა გაისროლოს როდესაც მოცემულობა არის 010. მეორეს მხრივ, 011 ერთიდაიგივე მანძილითაა დაშორებული ორივე ტიპის ნიმუშებისგან, ასე რომ მისი დაბრუნებული მნიშვნელობა კვლავ განუსაზღვრელია.

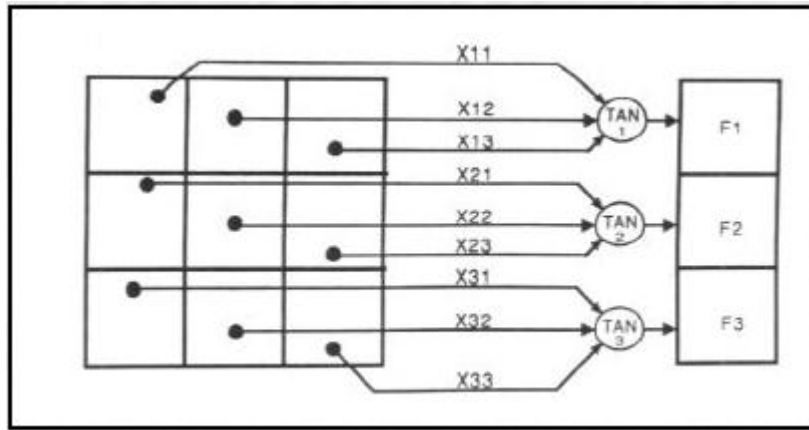
გასროლის წესის მიყენებით ვიღებთ შემდეგ ჭეშმარიტების ცხრილს:

X1:		0	0	0	0	1	1	1	1
X2:		0	0	1	1	0	0	1	1
X3:		0	1	0	1	0	1	0	1
OUT:		0	0	0	0/1	0/1	1	1	1

ამ ორ ცხრილს შორის განსხვავებას ეწოდება *ნეირონის განზოგადება*. შესაბამისად, გასროლის წესი ნეირონს აძლევს მსგავსების შეგრძნებას, რომელიც საშუალებას აძლევს მას „გონივრული“ რეაგირება მოახდინოს ისეთ ნიმუშებზე, რომლებზეც არ არის გაწვრთნილი.

4.6. ნიმუშის ამოცნობა - მაგალითი

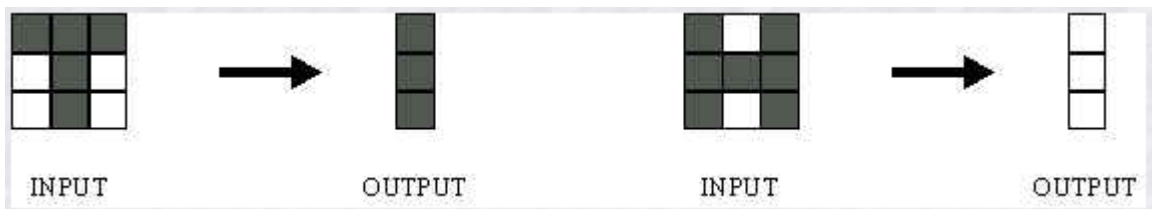
ნევრული ქსელების მნიშვნელოვანი გამოყენების სფერო არის ნიმუშის ამოცნობა. ნიმუშის ამოცნობის იმპლემენტაცია შესაძლებელია შესაბამისად გაწვრთნილი *წინ-გადაცემის* ნევრული ქსელის გამოყენებით. წვრთნის დროს ქსელი სწავლობს მოცემულობის ნიმუშების დასაბრუნებელ მნიშვნელობებთან ასოცირებას. ქსელის გამოყენებისას განისაზღვრება მოცემულობის ნიმუში და ქსელი ცდილობს დააბრუნოს ასოცირებული მნიშვნელობა. ნევრული ქსელების ძალა იბადება მაშინ, როდესაც ქსელს ეძლევა ისეთი ნიმუში რომელთანაც არ გააჩნია ასოცირებული დასაბრუნებელი მნიშვნელობა. ასეთ შემთხვევაში ქსელი აბრუნებს მნიშვნელობას, რომელიც ასოცირებულია იმ ნასწავლ ნიმუშთან, რომელიც ყველაზე ნაკლებად განსხვავდება მოცემული ნიმუშისგან.



სურ. 17 ობიექტის ამოცნობის მოდელი

მაგალითად:

პირველ ნახაზე მოცემული ქსელი გაწვრთნილია T და H ნიმუშების ამოცნობაზე. ასოცირებული მნიშვნელობები არის შესაბამისად 3-ვე შავი და 3-ვე თეთრი.



თუ ავლნიშნავთ შავ კვადრატებს 0-ით ხოლო თეთრებს 1-ით, 3 ნეირონისთვის განზოგადების შემდგომი ჭეშმარიტების ცხრილები იქნება შემდეგი:

X11:		0	0	0	0	1	1	1	1
X12:		0	0	1	1	0	0	1	1
X13:		0	1	0	1	0	1	0	1
OUT:		0	0	1	1	0	0	1	1

ზედა ნეირონი

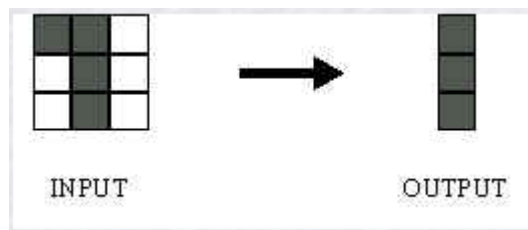
X21:		0	0	0	0	1	1	1	1
X22:		0	0	1	1	0	0	1	1
X23:		0	1	0	1	0	1	0	1
OUT:		1	0/1	1	0/1	0/1	0	0/1	0

შუათანა ნეირონი

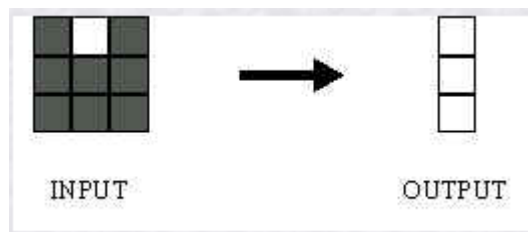
X21:		0	0	0	0	1	1	1	1
X22:		0	0	1	1	0	0	1	1
X23:		0	1	0	1	0	1	0	1
OUT:		1	0	1	1	0	0	1	0

ქვედა ნეირონი

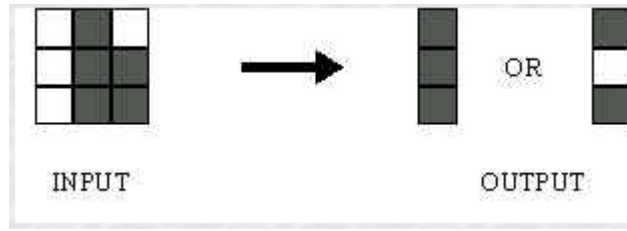
როგორც ცხრილებიდან ჩანს შესაძლებელია მივიღოთ შემდეგი ასოციაციები:



ამ შემთხვევაში ცხადია, რომ დაბრუნებული მნიშვნელობა უნდა იყოს 3-ვე შავი, რადგან მოცემული ნიმუში თითქმის იგივეა რაც „T“ ნიმუში.



აქაც ასევე ცხადია, რომ დაბრუნებული მნიშვნელობა უნდა იყოს 3-ვე თეთრი, რადგან მოცემული ნიმუში თითქმის იგივეა რაც „H“ ნიმუში.



ამ შემთხვევაში ზედა 2 სტრიქონი 2 ელემენტით განსხვავდება T ნიმუშგან და 3-თ H ნიმუშისგან და შესაბამისად აბრუნებს შავს. შუა სტრიქონი 1-ით განსხვავდება ორივე ნიმუშგან და შესაბამისად აბრუნებს შემთხვევით მნიშვნელობას. ქვედა სტრიქონი 1 ელემენტით განსხვავდება T ნიმუშისგან და 2-ით H ნიმუშისგან და შესაბამისად აბრუნებს შავს. ქსელის დაბრუნებული საბოლოო მნიშვნელობა ისევ T ნიმუშის ფორმას უფრო მიესადაგება.

მრავალ აგენტური 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 აგენტური 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 გადრეკვებას 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 მრავალაგენტიანი 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 დამლელი 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 მრავალაგენტიანი სისტემების სხვადასხვა სცენარებს ამ დარგში.

5.1. მრავალაგენტიანი 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 ტიპიურ სიტუაციებში.

ზოგ გაერთიანებას (domain) ჭირდება MAS. უფრო დეტალურად, თუ ამ გაერთიანებაში არიან სხვადასხვა ორგანიზაციები, სხვადასხვა ტიპისა და მიზნების მქონე ხალხი, საჭიროა მრავალაგენტიანი 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.

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 კოკნკრეტულ 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 :

მრავალაგენტიანი სისტემების გამოყენების მიზეზები

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

5.2. ერთ აგენტიანი და მრავალაგენტიანი სისტემა

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

შესაძლებელია ერთაგენტიანი სისტემები შედგებოდეს რამდენიმე ობიექტისაგან - სხვადასხვა მოქმედი, ცალკეული კომპონენტებისაგან. ერთ-

აგენტთან სისტემაში თითოეული ობიექტი იღებს დავალებებს და მოქმედებების ინსტრუქციებს ერთი ცენტრალური პროცესიდან. ცენტრალიზებული პროცესი ახდენს ყველა ამ ობიექტების მოდელირებას. ამ სექციაში შევადარებთ ერთ-აგენტთან და მრავალ-აგენტთან სისტემებს.

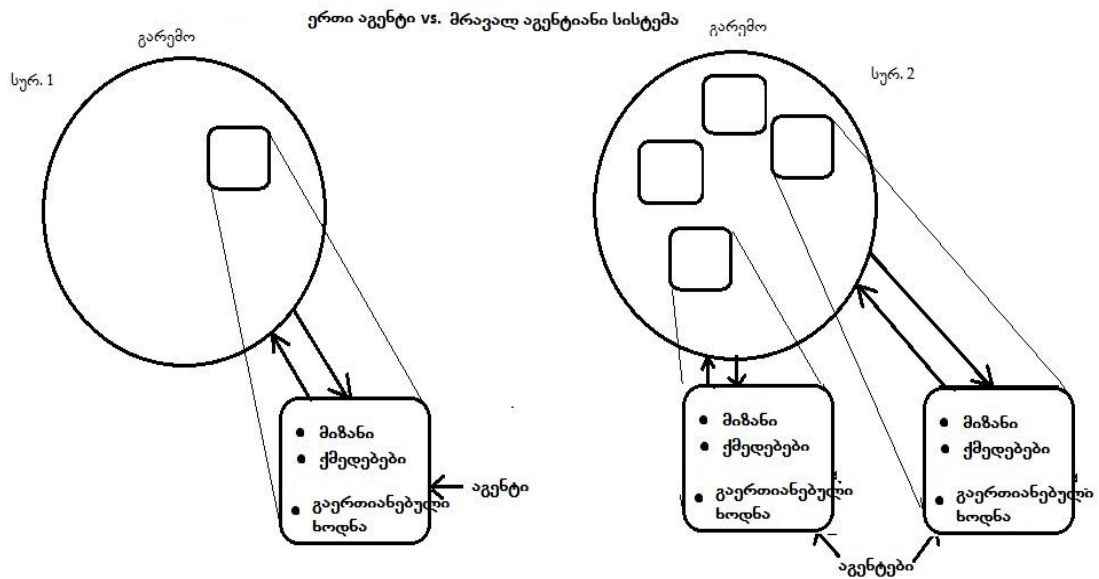
5.3. ერთ-აგენტის სისტემა

ზოგადად, აგენტი, ერთ-აგენტის 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-აგენტის სისტემაში აგენტი არ სცნობს სხვა ასეთ ობიექტებს. სხვა დანარჩენები არ მოიაზრება როგორც მიზნებისა ქცევებისა და ცოდნის მქონე.

5.4. მრავალაგენტის სისტემები

განსხვავება მულტიაგენტურ და ერთაგენტის 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, მრავალაგენტის სისტემები

ერთაგენტებიდან იმით განსხვავდება, რომ გარემოს დინამიკაზე ზემოქმედების მოხდენა სხვა აგენტებსაც შეუძლიათ.



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.1 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 მრავალაგენტოანი სცენარი. აგენტები ახდენენ ერთმანეთის მიზნების, ქცევებისა და ცოდნის მოდელირებას. ასევე, როგორც ნახაზზე, ისრებით არის წარმოდგენილი, მათ შეუძლიათ პირდაპირ ერთმანეთთან კომუნიკაცია.

აგენტების მოდელი

ბევრი კლასიკური მეთოდები არსებობენ რათა წარმოადგინონ და შეადარონ ონთოლოგიური ცოდნა ხელოვნურ ინტელექტში (აღწერების ლოგიკაში, ფრეიმებზე დაფუძნებული წარმოდგენაში, სემანტიურ ქსელში) და ეს ყველაფერი ბრინდება მოდაში თუნდა სემანტიკური ქსელის ინიციატივით. ასეთი მიდგომები უაღრესად გაურკვეველია და ბუნდოვანია აგენტის მოდელი შეიძლება ფორმალურად წარმოდგენილი იყოს შემდეგი სამეულით

$$A_M = \langle \Omega, \mathcal{R}, \mathcal{M} \rangle$$

სადაც Ω – ონთოლოგიური მოდელია, \mathcal{R} – დასკვნების კეთების მოდელია, ხოლო \mathcal{M} – სემანტიკური შეთავსების, სიახლოვის მოდელია. განვიხილოთ ცალცალკე.

6.1. ონთოლოგიური მოდელი

ონთოლოგიური მოდელი შეიძლება წარმოვადგინოთ შემდეგი ტუპლის სახით

$$\Omega_a = \langle C_a, L_a, P_a^*, R_a^*, H_a, I_a, A_o \rangle \quad [29]$$

სადაც C_a – ცნებებია, H_a – ცნებების და ატრიბუტების (თვისებების და მიმართებების) იერარქიაა, L_a – ჭდეების ლექსიკონია, I_a – ცნებების ეგზემპლარების სიმრავლეა, A_o – აქსიომების სიმრავლეა.

$$P_a^* = \langle \bar{P}, \bar{V}, Pr \rangle - \text{სადაც } \bar{P} = \{\bar{p} \mid \bar{p} \in \bar{P} \subset \mathfrak{P}(P)\},$$

P თვისებების სიმრავლეა, ხოლო $\mathfrak{P}(\cdot)$ – აღნიშნავს ყველა შესაძლო ქვესიმრავლეთა სიმრავლეს. თითოეულ c_i ცნებას (კლასს) ან მის

ეგზემპლიარს უკავშირდება \bar{p} თვისებების სიმრავლისგან შედგენილი მიმდევრობა. ამისთვის შემოვიტანოთ ასახვა

$f_p: C \rightarrow \bar{P}$, რომელიც განსაზღვრავს მოცემული კლასის თვისებების მიმდევრობას.

მაგ. $f_p(\text{მართკუთხედი}) = \{\text{სიგრძე, სიგანე}\}$

ანალოგიურად განვსაზღვროთ მიმართებების სიმრავლე.

$R_a^* = \langle \bar{R}, \bar{V}, Pr \rangle$ - სადაც $\bar{R} = \{\bar{r} \mid \bar{r} \in \bar{R} \subset \mathfrak{R}(R)\}$, R მიმართებების სიმრავლეა, ხოლო $f_r: C \rightarrow \bar{R}$, რომელიც განსაზღვრავს მოცემული კლასის მიმართებების მიმდევრობას.

მაგ. $f_r(\text{წრფე}) = \{\text{კვეთავს, ვერტიკალურია, ჰორიზონტალურია}\}$,

\bar{V} - თვისებების მნიშვნელობათა სიმრავლეა და $\bar{V} = \{\bar{v} \mid \bar{v} \in \bar{V} \subset \mathfrak{R}(V)\}$, სადაც V - ან დისკრეტულ მნიშვნელობათა სიმრავლეა ან $V \subset R^n$, R - რაციონალურ რიცხვთა სიმრავლეა.

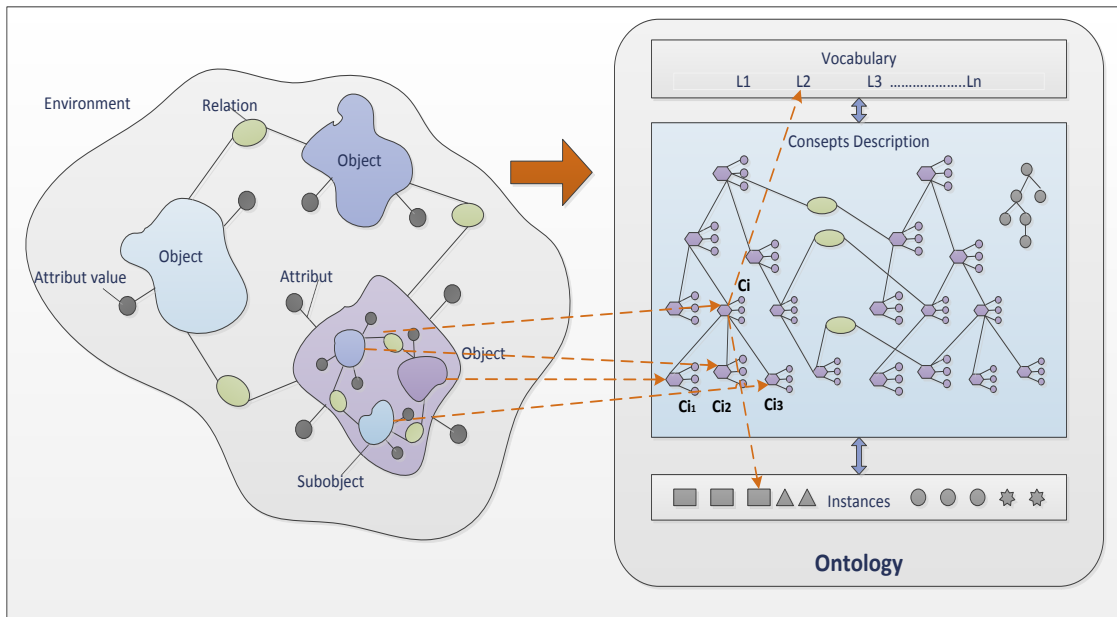
$f_v: P \rightarrow \bar{V}$, ასახვა აკონკრეტებს მოცემული თვისებისთვის კონკრეტულ მნიშვნელობას.

მაგ. $f_v(\text{სიგანე}) = 5$, $f_v(\text{სიგანე}) = [0, 1]$, $f_v(\text{ფერი}) = \{\text{შავი, თეთრი}\}$.

$Pr = \{=, \neq, <, \in, \notin, \geq, \leq, \subset, \not\subset, \subseteq, \not\subseteq, \dots\}$ - პრედიკატების სიმრავლეა, რომელიც წარმოადგენს ოპერატორს და აკონკრეტებს თვისების ან მიმართების ნიშვნელობას.

მაგ. ცნება : 'მართკუთხედი', თვისება : 'სიგანე', მნიშვნელობა : $\{< 5\}$,

ცნება : 'მართკუთხედი', თვისება : 'სიგანე', მნიშვნელობა : $\{= 5\}$.



სურ. 19 აგენტების მოდელი და ონთოლოგია

6.2. გადაწყვეტილებების მიღების მოდელი

გადაწყვეტილებების მიღების მოდელი ჩვენ შეგვიძლია წარმოვადგინოთ შემდეგი ტუპლის სახით

$$\mathcal{R} = \langle S, Tr, Arg \rangle$$

სადაც S არის სემანტიკა, Tr არის ნდობა ხოლო Arg არგუმენტაციის მოდელი.

6.3. სემანტიკა

ძირითადად სემანტიკა არის აღწერილი როგორც რეალურ სამყაროში არსებული ობიექტების ცნებების ინტერპრეტაცია და მათი კავშირების განსაზღვრა. პრედიკატულ ლოგიკაში ფორმულის ინტერპრეტაცია არის

რეკურსიულად განსაზღვრული ფორმულების კონსტრუქციის დროს, თავდაპირველად მას ენიჭება მუდმივები, ცვლადები და ფუნქციის სიმბოლოები ობიექტიდან რომელიც არსებობს რეალურ სამყაროში. ჩვენ ვართ დაინტერესებული ანდაგვარი ფორმულებით რომელიც ვალიდურია მსგავსი ინტერპრეტაციისთვის ან დაიცავს ამდაგვარ წესებს[30]. აქედან გამომდინარე სემანტიკა გამოიხატება ამგვარად

$$S = A_0 \cup A_0^*$$

სადაც A_0 არის სია აქსიომების და A_0^* მიახლოებული A_0

ნდობა

როგორც ონთოლოგია განსაზღვრავს ცოდნის წარმოდგენის მოდელს და უფრო მეტიც ცოდნა არის მიმართებული შეუთავსებლობასთან, ურთიერთობა წინადადებებსა და გამოცდილებას შორის ვერ იქნება ზუსტად წარმოდგენილი ნდობით. არაა ნდობა ამ ყველაფრის განსაზღვრად საკმარისი.

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[8]. ბევრი თეორიული და პრაქტიკული დეველოპმენტი დაფუძნებულია სემინალური არგუმენტების თეორიაზე. Dung არგუმენტაციის თეორია მიმართულია გრაფისკენ, რომელიც სედგება არგუმენტების ნაკრებისგან $args$ და ბინარული შეტევა R თან დაკავშირებით, ამ ფრეიმვორკის არგუმენტები განისაზღვრება როგორც სხვადასხვა სემანტიკა სადაც სენამტიკის არჩევანი უტოლდება სხვადასხვა ხარისხს სკეპტიციზმის. საკითხის გაფართოება განისაზღვრება „მისაღები აღრიცხვებით“ როცა არგუმენტი $X \in Args$ ამბობს რომ მისაღებია $S \in Args$ მაშინ როცა y უტევს x თითონ გამოდის დამარცხებული z არგუმენტით, მაგალითად თუ s ი არის მაქსიმალური წარმოადგენს ყველა შემავალ არგუმენტს რომელიც აკმაყოფილებს s ს მაშინ როცა s ი ამბობს რომ ის იქნება გაგრძელება რჩეული სემანტიკის. ბოლო წლების ინტენსიურ სწავლებას აქვს მრავალი მიმართულება, ის შეიძლება აიხსნას არგუმენტების ფარგლებში აბსტრაქტული ხასიათით და კოდირების სინტუიციით. ხაზგასმული და ზემოთ აღნიშნული ლოგიკა კრავს არგუმენტებს $args$ და შეტევა R ზე რჩება განუსაზღვრელი, ამგვარად საშუალება გვეძლევა ლოგიკის სხვადასხვა ფორმალიზაციისვის. თეორიის

დასკვნების შემდეგ შეიძლება განისაზღვროს პრეტენზიები არგუმენტებისა, არგუმენტები ძირითადად არის მტკიცებულება კანდიდატის დასკვნისა-არგუმენტი მოიცავს გაგებად ლოგიკას. დანკის თეორია ამიტომ შეიძლება გაგებული იქნას სემანტიკად, მონოტონურ მსჯელობად. ამ გაგებით მიზანი არაა მხოლოდ სწორი დასკვნის გაკეთება არამედ დაუპირისპირდე ოპონენტს, სად არის აპლიკაცია რომელიც გამოითვლის ლოგიკას, რაციონალობა ნიშნავს დაამკვიდრო სისწორის სტანდარტი. მართლაც ბევრი ლოგიკა პროგრამირების შეესაბამება დაგის თეორიას, დაგის სემანტიკას შეიძლება მრავალი გაგრძელება ქონდეს განსაკუთრებით არგუმენტების თეორიას, თუმცა შეიძლება აღმოვჩნდეთ პრობლემის წინაშე როცა ისმენ ერთის არგუმენტს და უნდა მიიღო გადაწყვეტილება, რომელსაც არ აქვს გაგრძელება[9]. ილუსტრაციისთვის განვიხილოთ P და Q არგუმენტები, A და B ამინდის ამინდის პროგნოზი

P1 : დღეს იქნება მშრალი ამინდი ლონდონში მაშინ როცა BBC იმ გამოაცხადაა მზე = A

Q1: დღეს იწვიმებს ლონდონში მაშინ როცა CNN მა გამოაცხადა ნალექი = B

A და B აყენებენ ურთიერთგამომრიცხავ არგუმენტებს და ეპაექრებიან ერთმანეთს (სემანტიკური პრძოლა)

დაგის თეორიის თანახმად გვაქვს ორი ვარიანტი A და B რომელთაგან არცერთი არაა გამართლებული, არსებობს პრობლემის გადაჭრის ერთი გზა, უპირატესობა მიანიჭო რომელიმე არგუმენტს, ზოგიერთი ამ მომენტში რთავს და ამართლებს დაგის თეორიას. დაგის თეორია ივრცობა უპირატესობის მინიჭების პრინციპით, ასე რომ შეტაკებისას X ის Y ზე წარმატებულია იმ შემთხვევაში თუ Y ი ვერ ჯობია X ს, დაგის ფრეიმვორკი დაფუძნებულია ღირებულებებზე და ღირებულებების მნიშვნელობაზე. ამინდის მაგალითი როლი რომელიც განვიხილეთ ზემოთ და A და B სუპირატესობა რადგან BBC უფრო სანდოა ვიდრე CNN, B ე არ იყო წარმატებული მოიერიშე ვიდრე A ამიტომ ამ ბრძოლის მაგივრად შეიძლება

დავტოვოთ მხოლოდ {A} ეს მაგალითი არის ილუსტრაცია პრობლემის გადაჭრის არგუმენტებით.

ხშირ შემთხვევაში აუცილებელია ინფორმაცია იმის სავარაუდოდ თუ ვინ იქნება გამარჯვებული, თუმცა ინფორმაცია უპირატესობის შესახებ იყოს არასწორი, შეფასება შეიძლება შეესაბამებოდეს კონტექსტს, რომელსაც აფასებს სანდო წყარო სხვადასხვა კრიტერიუმების მიხედვით, ამისათვის აუცილებელია სიტუაციის შეფასება სხვადასხვა კრიტერიუმებით, ამდენად გჭირდება კომფლიქტის უპირატესობის ინფორმაცია რომ ამტკიცო ზემოთ მოყვანილი ფაქტები. ილუსტრაციისთვის გავაგრძელოთ ზემოთ მოყვანილი მაგალითი ამინდის შესახებ :

P2: მაგრამ BBC არის უფრო სანდო ვიდრე CNN = C

Q2: მაგრამ CNN ი უფრო კომპეტეტურია ამინდის პროგნოზში ვიდრე BBC = C1

Q3: ზოგადი შედარება უფრო მნიშვნელოვანია ვიდრე ვიდრე პროგნოზი სანდოობასთან დაკავშირებული: E

C ს არგუმენტი არ ეწინააღმდეგება B ს თუმცა ის გამოხატავს არგუმენტს A ს უპირატესობას B ზე , თუმცა C გამოხატავს უპირატესობას B ს Aზე , C და C1 ეპაექრებიან ერთმანეთს რაც მათ გამოხატეს ურთიერთსაწინააღმდეგო წინადადებები, საბოლოოდ E მოითხოვს C1 უპირატესობას C ზე და მხოლოდ C1 ი ამარცხებს C ს, ამგვარად ჩვენ გვაქვს სკეპტიკური სამართლიანი არგუმენტი C1 ის რომელიც მოითხოვს B ს უპირატესობას A ზე და B არის სკეპტიკურად მართალი .

ამ თავში ჩვენ გვსურს განვაგრძოთ დაგის თეორია რომ ყურადღება მივაქციოთ არგუმენტებს, გვერდი დაყოფილია შემდეგნაირად , განვიხილავთ დაგის თეორიას და არგუმენტებს, იმ გვერდს რომელსაც ჩვენ მივყვებით აღვწერთ სწავლებას არგუმენტების ფორმალიზაციას.

დაგის არგუმენტების თეორია გრძელდება რომ ინტეგრაცია მოხდეს უპირატესობა არგუმენტებს შორის. გაფართოებული თეორია იცავს დაგის აბსტრაქტულ მიდგომას.

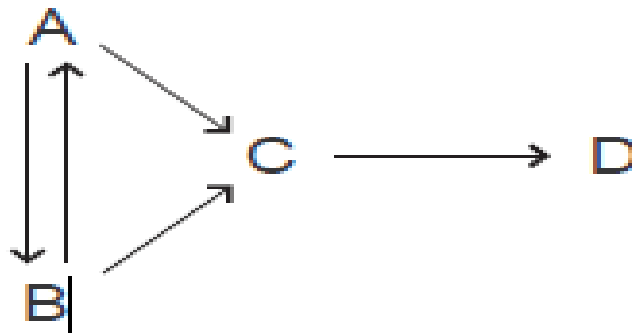
არგუმენტები რომლებიც გამოხატავენ უპირატესობას არის მარტივი კვანძი გრაფაში, უპირატესობა აპლიკაციის არის აბსტრაქტული თვისება, ახალი შეტაკების განსაზღვრა რომელიც დაკავშირებულია არგუმენტების უპირატესობასთან , შეტაკება არგუმენტებს შორის რომელიც იქნება უპირატესობის მოთხოვნის საგანი, ამგვარად არ არის სავალდებულო იცოდეთ უპირატესობის განსაზღვრა . უპირატესობის არგუმენტები გამოხატავს საწინააღმდეგო უპირატესობას ოპონენტებს შორის, შემდეგ ჩვენ განვაგრძობს დანგის სტატიებს ასე რომ არგუმენტებს და შეტაკებებს ჭირდება აღდგენა. ვრცელი სემანტიკები სემანტიკის გავრცობისთვის განისაზღვრება ზუსტად ისე როგორც როგორც დანგის თეორია, ზოგიერთი ზოგადი შედეგი რომელიც ახლავს ვრცელ სემანტიკს დანგის თეორიაში ასევე ნაჩვენებია ვრცელ ფრეიმვორკში. ჩვენი მიზანია საფუძველი ჩავუყაროთ აპლიკაციის პირდაპირ მოდიფიკაციას და გავავითაროთ დანგის არგუმენტებზე დაფუძნებული უპირატესობა. ვრცელი თეორია გვაწვდის ზოგად პარამეტრებს ვისწავლოთ არგუმენტების უპირატესობა და უფრო ზოგადად შეტაკებები, ამგვარად ზემოთ ნახსენები ფრეიმვორკი შეიძლება განვავრცოთ და მხედველობაში მივიღოთ არგუმენტები რომელიც გამომდინარეობს პრობლემის გადაწყვეტილებიდან. აქ ჩვენ განვიხილავთ არგუმენტების ფრეიმვორკის ორ ძირითად კლასს. გაფართოებული თეორია შექმნილია იმისთვის რომ გააერთიანოს ფრეიმვორკი რომელიც ავითარებს დანგის თეორიის უპირატესობის პრინციპს. ეს სექცია აღწერს როგორ შეიძლება გაფართოვდეს იერარქიულად არგუმენტებზე დამყარებული მნიშვნელობები. გაფართოებული თეორია არის შემოთავაზებული როგორც ობიექტის დონე რომელიც ალაგებს ფაქტებს პრიორიტეტების მიხედვით, ასეთი ფორმალური გავრცელება ობიექტის დონის ენის არგუმენტების კონსტრუქციისთვის წესებთან და პრიორიტეტებთან ერთად[10].

არგუმენტები კომფლიქტის შესახებ პრიორიტეტები შეიძლება ჩამოაღიბდეს სხვა ფაქტებიდან გამომდინარე, ასეთი მიდგომა აზოგადებს ფორმალიზაციას და უპირატესობები შეიძლება დამყარებული იყოს წესების პრიორიტეტებზე არამედ არგუმენტის სიმლიერეზე.

7.1. კლასიკური არგუმენტაცია

ამ თავში განვიხილავთ დანგის არგუმენტების თეორიას:

1. დანგის არგუმენტების ფრეიმვორკი არის პირობა $AF = (Args; R)$ სადაც $Args$ ი არის არგუმენტების ერთობლიობა და $R \mu Args \mathcal{E} Args$. პირველ ფიგურაში ნაჩვენებია ისრები x დან y ისკენ სადაც $(X; Y) \in R$. [12]



სურ 20 დანგის არგუმენტაციის ფრეიმვორკის მაგალითი

დანგი შემდეგ განიხილავს არგუმენტების წვდომას, ფუნქციების თვისებებს და ფრეიმვორკის დასაშვებ გაგრძელებას.

2. $AF = (Args; R)$, $S \mu Args$ და $A; B; C; \dots$ აღნიშნავს არგუმენტს $Args$ ში.
 - 1. S ი არის თავისუფალი კონფლიქტი $\delta A; B \in S$ ეს არ არის შემთხვევა როცა $(A; B) \in R$

- A ს შეუძლია პატივი სცეს S თუ $\delta B \ 2 \text{Args}$. თუ $(B;A) \ 2 \ R$ შემდეგ შესაძლებელია $C \ 2 \ (C;B)$
- ფუნქციის ტისებები AF , აღნიშნავს FAF განსაზღვრულია როგორც: - $FAF : 2\text{Args} \rightarrow 2\text{Args} \ (S) = fAjA$ წვდომადია w.r.t. Sg
- თუ S თავისუფალი კონფლიქტი, მაშინ S არის დასაშვები პირობა AF თუ ყოველი უმენტი μ წვდომადია S (i.e., $S \mu FAF(S)$)

3. $AF = (Args;R)$, S ქვეფუნქციაა $Args$ ის, და F ფუნქციაა AF . როცა :

- S ი არის ფუნქცი რომელიც მისაღებია ($S = F(S)$)
- S არის სტაბილური ფუნქცია თუ $\delta B = 2 \ S$, $\delta A \ 2 \ S$ და $(A;B) \ 2 \ R$
- S ი არის ფუნდამენტური ფუნქცია თუკი $S \mu FAF(S)$

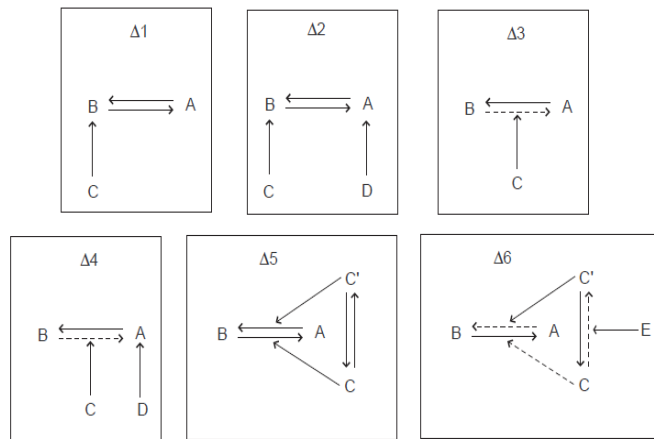
$S \in$ (სრულყოფილი, უპირატესი, სტაბილური, ფუნდამენტური) არგუმენტი გულისხმობს სემანტიკას თუკი ის ეკუთნის ფუნქციას. არგუმენტი არის სემანტიკური თუ ის ეკუთნის მხოლოდ ერთს და არა ყველას ერთად.

7.2. მნიშვნელობებზე დამოკიდებული არგუმენტაცია

ამ თავში ჩვენ გვსურს განვავრცოთ დაგის თეორია რომ ყურადღება მივაქციოთ არგუმენტებს, გვერდი დაყოფილია შემდეგნაირად, განვიხილავთ დაგის თეორიას და არგუმენტებს, იმ გვერდს რომელსაც ჩვენ მივყვებით აღვწერთ სწავლებას არგუმენტების ფორმალიზაციას.

გავიხსენოთ პირველი სექციის ამინდის პროგნოზის მაგალითი სადაც A და B სემანტიკური შეტევებია $((A;B), (B;A) \ E \ R)$ A ა უპირატესია B ზე(რასაც

ითხოვს არგუმენტი C), B ვერ ეპაექრება A ს , პარალელურად A ვერასოდეს ამარცხებს B ს , თუ ჩვენ შევაფასებთ შესაძლებლობას მაშინ A ა პირობა დასაშვებია და B არა.



სურ. 21 მნიშვნელობებზე დამოკიდებული არგუმენტაციის მაგალითი

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 მიმართებების 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 :

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

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

სადაც σ - ოპერატორია , რომელიც ნებისმიერ ობიექტს უკვშირებს მისი ქვეობიექტების სიმრავლეს. ობიექტების მიმართებები უთავსდებათ პატერნის მიმართებებს თუ პატერნის ქვეობიექტების სიმრავლესრული განსაზღვრულობით უთავსდება ობიექტის ქვეობიექტების სიმრავლეს, პატერნის მიმართებების სიმრავლეს სრულად ფარავს ობიექტების მიმართებების სიმრავლე და პატერნის 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 გვაქვს 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_i^p, g_i^r , რომელიც შესაბამისად ობიექტის l -ფენის თვისებების და მიმართებების ქვეგრაფებს აღწერს.

მაშინ , ობიექტის პატერნთან სიახლოვე შეიძლება გამოითავალოს შემდეგი ფორმულით:

$$\varphi(g(o), g(t_i)) = \sum_{i=1}^k [\alpha_i^p * \varphi(g_i^p(o), g_i^p(t_i)) + \alpha_i^r * \varphi(g_i^r(o), g_i^r(t_i))]$$

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

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

8.1. თვისობრივი არგუმენტების გენერირება

თვისობრივი არგუმენტაციის გენერაცია იწყება მაშინ , როცა ობიექტის თვისებების შეთავსების აგენტება თავიანთი საქმე დაამთავრეს.

არგუმენტაციის დასაგენერირებლად გამოიყენება $\varphi(g_i^p(o), g_i^p(t))$ ფუნქცია და დარდება მათი თვისებების სიმრავლეები. ქვევით ცხრილ 1-ში მოყვანილია c და h-ის მნიშვნელობები. თუ მოხდა პატერნთან სრული თანხვედრა მაშინ, ყველაფერი გარკვეულია და მის კონფიდენციალობის დონე არის Certainty და სხვა აგენტები ვერ შეეკამათებიან ამ არგუმენტს. მაშინ როცა ხდება ნაწილობრივი შეტავსება და შეთავსების ფუნქცია დადებითია მაშინ ცონფიდენციალობა გაურკვეველია და შესაძლებელია ამ არგუმენტაციის გაპროტესტება.

ცხრილი 1

C	+(h)
შეთავსება	თუ ყველა თვისებები ობიექტი o არის მსგავსი ობიექტი(ნიმუში) t თვისებებისა და ყველა მნიშვნელობა თვისებების ასევე შეესაბამება და $\varphi(g^p(o), g^p(t)) \geq 0$
შეუთავსებლობა	თუ ზოგიერთი თვისებები ობიექტის(ნიმუშის) o არის მსგავსი ობიექტი(ნიმუში) t გარკვეული თვისებებისა და ყველა მნიშვნელობა თვისებების არ შეესაბამება და $\varphi(g^p(o), g^p(t)) \geq 0$
C	-(h)
შეუთავსებლობა	სხვა შემთხვევაში და $\varphi(g^p(o), g^p(t)) < 0$

8.2. მიმართებითი არგუმენტების გენერირება

მიმართებითი არგუმენტაციის პროცესი იწყება მაშინ თუ თვისობრივი არგუმენტ გენერირებულია და ის განიცდის შეტევას, ამასთან ყველ გამოსაცნობი ობიექტის ყველა ქვეობიექტი. უნდა იყვეს პატერნებთან

შეთავსებული. ქვევით ცხრილ-ში მოყვანილია არგუმენტების კონფიდენციალობის და კონტრ არგუმენტების წამოყენების პირობები. აქ არგუმენტაციის გამოსათვლელად გამოიყენება $\varphi(g^r(o), g^r(t))$ ფუნქცია.

ცხრილი 2

C	+(h)
შეთავსება	თუ ყველა მიმართება ობიექტი o-სი არის მსგავსი მიმართებებისა ობიექტი(ნიმუში) t - სი და ობიექტის(ნიმუში) ყველა ქვე ობიექტების ემთხვევა ყველა ქვე ობიექტების თვისებების მნიშვნელობებს და $\varphi(g^r(o), g^r(t)) \geq 0$
შეუთავსებლობა	თუ ზოგიერთი მიმართება ობიექტი o-სი არის მსგავსი მიმართებებისა ობიექტი(ნიმუში) t - სი და ობიექტის(ნიმუში) ზოგიერთი ქვე ობიექტები ემთხვევა ზოგიერთ ქვე ობიექტების თვისებების მნიშვნელობებს ან არ ემთხვევა და $\varphi(g^r(o), g^r(t)) \geq 0$
C	-(h)
შეუთავსებლობა	სხვ შემთხვევაში და $\varphi(g^r(o), g^r(t)) < 0$

8.3. სტრუქტურული არგუმენტების გენერირება

სტრუქტურული აგენტები არგუმენტების გენერაციას განიხილავენ ცნებების ონტოლოგიური იერარქიის კონტექსტში, როც მოცემული ობიექტი შეიძლება შეუთავსდეს იერარქიის ნებისმიერ ცნებას. რა თქმა უნდა არგუმენტების გენერირება იწყება ზევიდან ქვევით და ალგორითმი უნდა ჩავიდეს რაც შეიძლება ღრმად. თუ რომელიმე დონეზე ვერ მოხერხდა

შეთავსება ტვისობრივად და მიმართებრივად და არი გაურკვეველობა მაშინ იერერქის დაბალ საფეხურებზე მდგომი ცნებები უნდა მოინიშნოს როგორც არ გამოცნობადი certainty კონფიდენციალობით და მათთვის არგუმენტები აღარ უნდა დაგენერირდეს. თუ იერერქიის რომელიმე დონეზე გვაქს განუზრვრელობის კონფიდენციალობა და კონტრ არგუმენტი არსებობს ამ დონეზე მაშინ ეს ცნებაც უნდა მოინიშნოს როგორც არა გამოცნობადი certainty კონფიდენციალობით და არგუმენტაცია უნდა გაგრძელდეს მშობელი ცნებისთვის.

ევრისტიკული ძეგნა

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[22].

- 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 გადაწყვეტასდა ვერ პოულობს გამოსავალს სიტუაციიდან.

ხელოვნური ინტელექტის ქვედარგების ევრისტიკული და ავტომატური დაგეგმვა

მჭიდროდაა დაკავშირებული, დაგეგმვის პრობლემები ხშირად საჭიროებს გაკეთდეს სტიმულაცია ამოცანის ტესტირებისთვის ძეგნის ალგორითმების გამოყენებით. კლასიკური მიდგომები ევრისტიკული ძეგნისა და დაგეგმვის წარმოადგენს განმსაზღვრელ მოდელს გადაწყვეტილების მიღებაში , რაც შედეგად გვაძლევს გარდაქმნას ამოცანა დაწყების მდგომარეობიდან

მიზანის მიღწევის მდგომარეობაში. ეფექტურობა ევრისტიკული ძიების მიდგომაში კლასიკური დაგეგმვისას ილუსტრირებულია შედეგების მიღებაზე, სადაც ოპტიმალური დამგეგმავები დაფუძნებულია A^* და Best first search ალგორითმებზე.

Best first search არის საძიებო ალგორითმი რომელიც ირჩევს გრაფში ყველაზე პერსპექტიულ კვანძს სპეციფიურ წესებზე დაყრდნობით. Judea Pearl აღწერა best first search ალგორითმი რათა შეეფასებინა კვანძის არჭერილობა „ევრისტიკული შეფასების ფუნქციით $f(n)$ “, რომელიც ზოგადად შეიძლება დამოკიდებული იყოს მიზნის აღწერაზე, მონაცემებზე რომელიც შეგროვებულია ამ კვანძამდე მისვლამდე ან გარე ცოდნიდან ამ ამოცანის შესახებ. ზოგიერთი ავტორის აზრით „best first search“ ალგორითმი ეხება ევრისტიკულ ძებნას რომელიც ცდილობს ივარაუდოს რამდენად ახლოსაა გზა დასასრულისა მიმდინარე ამოცანაში, რათა შემდეგი ნაბიჯი იყოს სწორად გადადგმული ამოცანის დასრულებისთვის. ესეთ კონკრეტულ ძებნას ეწოდება ხარბი „best first search“ და სუფთა „best first search“ [25].

ბრმა ძებნა სტანდარტულად ძალიან არაეფექტურია. ცოდნის დამატებით შესაძლებელია გაუმჯობესება მოხდეს პროცესისი და შევეცადოთ რამოდენიმე ესეთი მეთოდის გამოძიებას.

იდეა რომელიც ჰეურისტიკული ძებნის უკან დევს არის ვიპოვოთ კვანძი რომელიც იქნება ყველაზე ახლოს იმ კვანძთან რომელიც არის ჩვენი დანიშნულების ადგილი. გარანტირებული არ არის რომ ევრისტიკული ფუნქციის მიერ დაბრუნებული მნიშვნელობა იქნება ყველზე ახლოს მდებარე მნიშვნელობა რომელიც ჩვენ გვჭირდებოდა მიზნის მისაღწევად. თუ ჩვენ არ გავაკეთებთ ამას მაშინ ჩვენ შეგვიძლია მარტივად ვიმოდრაოდ კვანძებს შორის სანამ არ მივაღწევთ მიზნამდე.

რათა განვახორციელოთ ევრისტიკული ფუნქცია ჩვენ გვჭირდება გარკვეული ცოდნა გაერთიანების შესახებ. ასევე ჩვენ გვჭირდება გარკვეული ცოდნა ამოცანის შესახებ რომელიც ჩვენ უნდა გადავწყვიტოთ

რათა ვიმსჯელოთ რამდენად ახლოსაა მიმდინარე კვანძი იმ კვანძამდე რომელამდეც მიღწევა არის ჩვენი მიზანი.

თუ ჩვენ გამოვიყენებთ ბრმა ძებნას დავინახავთ ჩვენ ვიხილავთ ყველა ვარიანტებს და ვცდილობთ რომ მივიღეთ ყველაზე იაფი ღირებულებით დანიშნულების ადგილამდე. ხოლო ჰეურისტიკული მიდგომით ჩვენ ვცდილობთ ყოველ ნაბიჯზე მოვახდინოთ დადგენა თუ რამდენად ახლოს ვართ მიზანთან ყველაზე იაფი ღირებულებით.

ჩვენ შეგვიძლია ავლნიშნოთ გზის ღირებულების ფუნქცია $g(n)$ - ით, ხოლო ევრისტიკული ფუნქცია $h(n)$, სადაც $h(n)$ არის შეფასება ღირებულების საწყისი კვანძიდან იმ კვანძამდე რომელიც არის ჩვენი მიზანი.

9.1. განხორციელება

ევრისტიკული ფუნქციის განსახორციელებლად დავალაგოდ კვანძები მნიშვნელობების მიხედვით რომლებიც დაბრუნებულია ევრისტიკული ფუნქციის მიერ. ჩვენ შეგვიძლია განვახორციელოთ მსგავსი ძებნა ზოგადი ძებნის ალგორითმის გამოყენებით. ფუნქციას რომელიც ირჩევს საუკეთესო კვანძს ესწოდება best-first-search ფუნქცია.

(სინამდვილეში ჩვენ არ ვართ დარწმუნებული, რომ ჩვენ ვირჩევთ საუკეთესო კვანძს პირველად (წინააღმდეგ შემთხვევაში ეს არ იქნება ძებნა, ურალოდ ეს იქნება მარში მიმდინარე მდგომარეობიდან მიზნის მდგომარეობამდე) მაგრამ, თუ გავითვალისწინებთ ჩვენს მიმდინარე ცოდნას, ჩვენ გვჯერა რომ კვანძი რომელიც ჩვენ ავირჩიეთ არის საუკეთესო და ჩვენს ფუნქციას დავარქვათ believed-to-be-best-first-search.

ფუნქცია BEST-FIRST-SEARCH(პრობლემა, EVAL-FN) აბრუნებს გადაწყვეტილებების მიმდევრობას

შემავალი ინფორმაცია: ამოცანა რომელიც უნდა გადაწყვიტოთ

EVAL-FN: შეფასების ფუნქცია

Queueing-Fn: ფუნქცია რომელიც ალაგებს კვანძებს EVAL-FN ფუნქციის მიხედვით

ბრუნდება GENERAL-SEARCH (პრობლემა, Queueing-FN)

9.2. მაგალითი

ჩვენ შეგვიძლია შევქმნათ ევრისტიკული ფუნქცია რომელიც დაგვეხმარება ვიპოვოთ გადაწყვეტილება რუმინული მარშუტის ამოცანაში. ერთ-ერთი შესაძლო ევრისტიკული ფუნქცია არის დაფუძნებული სწორ ხაზოვან მანძილებზე ქალაქებს შორის.

ჩვენ ყოველი ქალაქში მისვლისას შეგვიძლია შევხედოთ მეზობელ ქალაქებს და ავირჩიოთ მარშუტი რომელსაც აქვს პირდაპირი კავშირი ჩვენს მიზანთან. ჩვენ ევრისტიკულ ფუნქციას ექნება შემდეგი სახე:

$h_{SLD}(n) =$ პირდაპირი კავშირი მანძილი n - სა და საბოლოო ლოკაციას შორის რა თქმა უნდა ამ ევრისტიკული ფუნქციის გამოყენებით ჩვენ ყოველთვის არ მოვხვდებით ისეთ ქალაქში რომელიც საუკეთესო იქნება ჩვენთვის. ამის ორი შესაძლო მიზეზი არის:

- 1) ქალაქში რომელშიც ჩვენ გადავწყვიტეთ წასვლა შესძლოა ამ ქალაქიდან არ არსებობდეს გზა საბოლოო დანიშნულების ადგილამდე, ამ შემთხვევაში ჩვენ მოვიწევს ჩვენი ნაბიჯების თავიდან გაკეთება მაგალითად თუ ჩვენ ვიყავით სამტრედიაში და ევრისტიკულად გადავწყვიტეთ წასვლა გრიგოლეთში, ჩვენ მოვიწევს ჩვენი ნაბიჯების თავიდან გაკეთება იმიტომ რომ არ არის სხვა გზა მოვხვდეთ ქუთაიში გრიგოლეთის გავლის გაერშე.
- 2) გზამ ჩვენს საბოლოო დანიშნულების ადგილამდე შეიძლება მიიღოს გრაგნილის სახე მიუხედავად იმისა რომ SLD არის მოკლე, ფაქტობრივი მანძილი იქნება მაინც გრძელი.

მაგრამ ზოგადი სტრატეგია გამოვიყენოთ SLD ევრისტიკული ფუნქცია გვაძლევს საშუალებას მივიღოთ ამონახსნი უფრო სწრაფად ვიდრე სხვა ამონახსნის გზები[26].

9.3. ხარბი ძებნა

ხარბი ძებნა არის განხორციელება საუკეთესო ძებნის ფილოსოფიის. იგი მუშაობს პრინციპით რაც ნიშნავს რომ ამოცანის ყოველ ეტაპზე გადაწყვეტილებისას ვიღებთ საუკეთესო ვარიანტს.

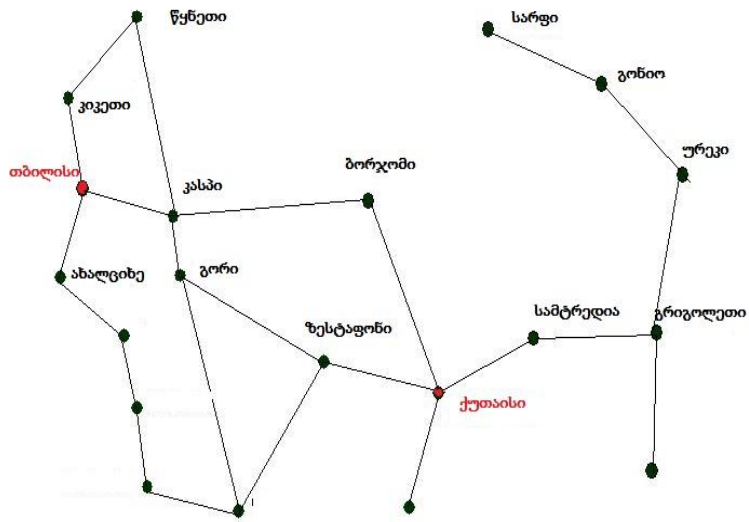
ხარბი ძებნა ცდილობს მინიმიზაცია გაუკეთოს ღირებულებას რაც გვჭირდება მიზნის მიღწევამდე. ამისთვის ის ირჩევს კვანძს რომელიც ყველაზე ახლოსაა მიზანთან. რომ მიიღოს ეს გადაწყვეტილება ის ამისთვის იყენებს ევრისტიკულ ფუნქციას h .

გვავს ევრისტიკული ფუნქცია h , ჩვენ შეგვიძლია განვახორციელოთ ხარბი ძებნა შემდეგნაირად

ფუნქცია GREEDY-SEARCH(ამოცანა) აბრუნებს ამონახსნს ან შეცდომას

აბრუნებს BEST-FIRST-SEARCH(ამოცანა, h)

მოცემული გავქვს თბილისი/ქუთაისის მარშუტის საძიებო პრობლემა და h_{SLD} ევრისტიკული ფუნქცია



სურ. 22 ევრისტიკული ძეგნის მაგალითი

ქალაქი	SLD ქუთაისამდე
თბილისი	366
ქუთაისი	0
კიკეთი	374
ახალციხე	329
კასპი	273
გორი	193
ბორჯომი	178
ზესტაფონი	98
სამტრედია	80
გრიგოლეთი	151
ურეკი	199
გონიო	226
სარფი	234
წყნეთი	380

ჩვენ ვიწყებთ თბილისიდან, სადაც გზის საფასური არის 366 ქუთაისამდე და ეს არის ერთადერთი კვანძი რომელიც ჩვენ გვაქვს საწყის ეტაპზე. ამის შემდეგ ჩვენ გვაქვს სამი ახალი კვანძი ესენია (კიკეთი, კასპი, ახალციხე), მათი გზის საფასურებია 374, 273, 329. ჩვენი ევრისტიკული მიდგომის საფუძველზე შემდეგ ვირჩევთ კასპს. ამის შემდეგ ჩნდება ოთხი ახალი ვარიანტი ესენია (თბილისი, წყნეთი, ბორჯომი, გორი) შესაბამისად გზის საფასურებია 366, 380, 178, 193. შესაბამისად ევრისტიკული მიდგომის საფუძველზე ხდება ბორჯომის არჩევა საიდანაც უკვე გვაქვს პირდაპირი გზა ქუთაისამდე რაც არის ჩვენი დანიშნულების ადგილი.

ხარბი ძეგნის ფსევდო კოდი გამოიყურება შენდეგნაირად :

```

current_distance = 0
current_stop = 0
stops = []
while current != lahore:
    next_stop = 0
    while distance(next_stop) - current_distance <= m:
        next_stop = next_stop + 1
    next_stop = next_stop - 1

    current_stop = next_stop
    current_distance = distance(current_stop)
    add next_stop to stops
return stops

```

9.4. A* ძეგნა

რა არის A* ძეგნა?

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 ძაან არაეფექტური.

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

ჩვენ შეგვიძლია განვახორციელოდ A* ძებნა შემდეგნაირად :

ფუნქცია A*-SEARCH(ამოცანა) აბრუნებს გადაწყვეტილებას ან შეცდომას აბრუნებს BEST-FIRST-SEARCH(ამოცანა $g + h$)

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.

9.5. 8 - თავსატეხი

ვინც არ იცის ამ ამოცანის შესახებ იგი შედგება მოედანზე რვა ფილისგან და აქედან ერთერთი არის ცარიელი. ჩვენი მიზანია გადავაცუროთ ფილები ისე რო მივაღწიოთ საბოლოო მიზანს. ტიპიური პრობლემა მოცემულია ქვემოთ

5	4		1	2	3
6	1	8	8		4
7	3	2	7	6	5
საწყისი მდგომარეობა			მიზანი		

ტიპიური ამონახსნი არის დაახლოებით ოც ნაბიჯში. განშტოებების ფაქტორი არის შემდეგი (ოთხი როდესაც ცარიელი ფილა განთავსებულია შუაში, ორი როდესაც ცარიელი ფილა განთავსებულია კუთხეში, და სამი როდესაც ცარიელი ფილა განთავსებულია სხვა მდებარეობაზე).

ეს ნიშნავს რომ ამომწურავი ძიება დაახლოებით 3^{20} მდგომარეობას მოიცავს ან $(3.5 * 10^9)$

თუმცა თუ მოვახდენთ შენახვას განმეორებადი მდგომარეობების ჩვენ შევძლებთ შევამციროთ იგი 362,880 ოპერაციამდე რადგან დაფაზე გვაქვს მხოლოდ 9! შესაძლო განლაგება.

თუმცა 362,880 არის საკმაოდ ბევრი მდგომარეობა ძეზისთვის ამიტომაც ჩვენ გვჭირდება კარგი ევრისტიკული ფუნქცია.

თუ ჩვენ გვინდა ვიპოვოთ ოპტიმალური გადაწყვეტა, ამისთვის გვჭირდება ევრისტიკული ფუნქცია რომელიც არასდროს არ აფასებს ნაბიჯებს ზედმეტად მიზნის მისაღწევად ანუ ჩვენ გვჭირდება მისაღები ევრისტიკული ფუნქცია[28].

ამისთვის გვავს ორი ვარიანტი

- h_1 = რაოდენობა ფილებისა რომლებიც არიან არასწორ პოზიციაზე. ზემოთ სურათში (საწყისი მდგომარეობა) გვავს 7 ფილა რომელიც არ არის განთავსებული თავის ადგილას ამიტომაც ფუნქცია დააბრუნებს 7-ს რადგან 8 ფილიდან 7 არ იმყოფება თავის ადგილას. ევრისტიკულად ჩვენ შეგვიძლია ვთქვათ ფილები რომლებიც არ არიან თავის ადგილას განლაგებული საჭიროებენ გადაადგილებას სულ მცირე ერთხელ რათა მიიღონ სწორი პოზიცია.
- h_2 = ჯამი მანძილებისა რომლებიც სჭირდებათ ფილებს რათა განლაგდნენ საბოლოო პოზიციაზე. ამ შემთხვევაში ჩვენ შეგვიძლია დავთვალოთ რამდენად შორს არის ფილა მიზნის ადგილამდე და ეს არის ჯამი ჰორიზონტალური და ვერტიკალური პოზიციების. ეს ევრისტიკული ფუნქციაც ასევე მისაღებია რადგან ყოველი ფილისთვის საჭიროა გადაადგილება იმდენჯერ სანამ არ მოხვდება მიზნის მდგომარეობაში.

მეორე ვარიანტს ექნება შემდეგი სახე

$$h_2 = 2 + 3 + 3 + 2 + 4 + 2 + 0 + 2 = 18$$

9.6. ევრისტიკული ფუნქციის ეფექტურობა

ორი 8 თავსატეხის ევრისტიკული ფუნქციისთვის ჩვენ შევქმენით 2 ვარიანტი და ვცდილობთ გავარკვიოთ რომელია მისი საუკეთესო გამოყენება. ერთი მეთოდია შევქმნათ ბევრი მაგალითი და ორივესთვის გამოვიყენოთ ორივე ევრისტიკული ფუნქცია და ვნახოთ რომელი გვაძლევს უკეთეს შედეგს.

ქვემოთ ცხრილში ნაჩვენებია დაახლოებით 100 გაშვებაზე ამოცანის სიღმეზე დამოკიდებული 2,4,6...24 A* ალგორითმის გამოყენებით h1 და h2 ევრისტიკული ფუნქციის შედარება და ასევე არა ევრისტიკული ფუნქციით შედარება.

სიღრმე	ძებნის ღირებულება		
	არა ევრისტიკული ფუნქცია	A*(h ₁)	A*(h ₂)
2	10	6	6
4	112	13	12
6	680	20	18
8	6384	39	25
10	47127	93	39
12	364404	227	73
14	3473941	539	113
16		1301	211
18		3056	363
20		7276	676
22		18094	1219
24		39135	1641

შავი დაფის არქიტექტურა

ავტომატური სამუშაო პროცესი თამაშობს მნიშვნელოვან როლს 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.

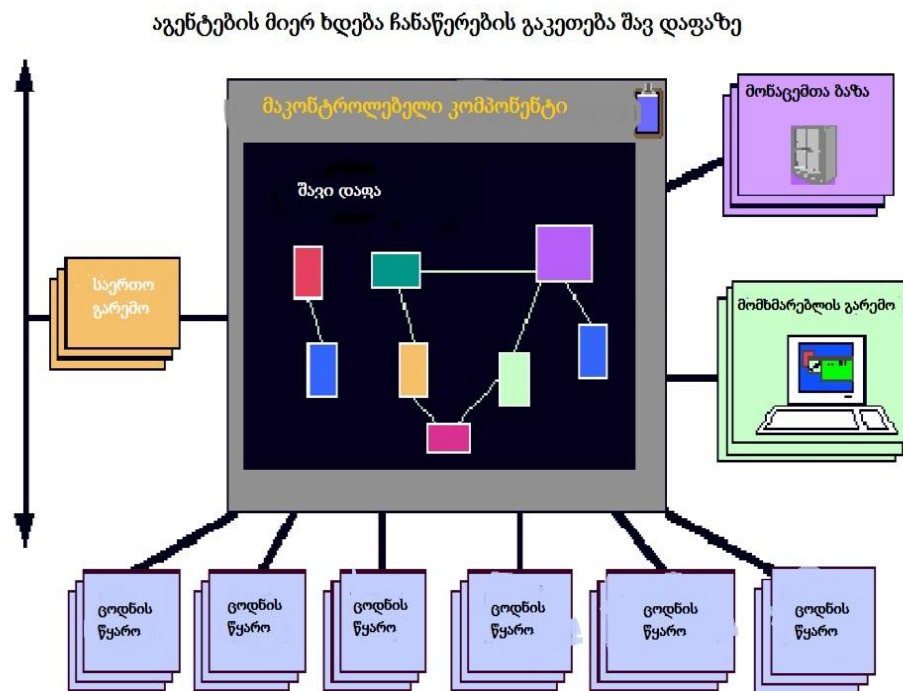
პრობლემის გადაჭრა იწყება მისივე გამოცხადებით და ხდება თავდაპირველი მონაცემების დაწერა შავ დაფაზე. სპეციალისტები უყურებენ შავ დაფას და ეძებენ შესაძლებლობას რომ წვლილი შეიტანონ ამოცანის განვითარებაში. როდესაც სპეციალისტი იპოვის შესაძლებლობას ის აკეთებს ჩანაწერს შავ დაფაზე და იმედოვნებს რომ ისინი გამოიყენებენ 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. 23 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[19].

10.1. 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 ტრადიცილი 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 მისი 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.

10.2. შავი დაფის აპლიკაცია

შავი დაფის არქიტექტურა შესაძლებელია გამოყენებული იქნას სხვადასხვა სფეროში

- სენსორული ინტერპრეტაცია;
- პროცესების კონტროლი;
- დაგეგმვა და დისპეჩირება;

- კომპიუტერული ხედვა;
- ვარიანტებზე დამოკიდებული გადაწყვეტილების მიღება;
- ცოდნაზე დაფუძნებული სიმულაცია;
- ცოდნაზე დამოკიდებული ახსნა;
- ბრძანებები და კონტროლი;
- სიმბოლების სწავლება;
- მონაცემების შერწყმა;

ძირითადად შავი დაფის არქიტექტურია გამოიყენება ისეთ სისტემებში როგორცაა მონაცემთა ინტერპრეტაცია, აპლიკაციის გარემოს ცვლილებები, მონიტორინგის პრობლემები და სსისტემები სადაც საჭიროა გადაწყვეტილებების მიღება.

10.3. შავი დაფის სისტემა აგენტების მესიჯების ინტერპრეტაციისთვის

მთავარი მიზანი რეალურ დროში სტრატეგიული თამშისა არის დამოკიდებულება ჯგუფებს შორის და არა ინდივიდუალური დამოკიდებულება, ამიტომაც საჭიროა ცენტრალიზებული სისტემის კონტროლი. ეს ამცირებს ამოცანის მთლიან სირთულეს რადგან აძლევს ინდივიდუალებს იყვნენ მარტივები, ხოლო სირთულე ექცევა ცენტრალიზებული სისტემის ფარგლებში.

აგენტის სხვა ასპექტით აგენტების კორდინაცია არის მათი კომუნიკაცია. შავი დაფის სისტემა არის მიდგომა რომელიც საშუალებას იძლევა მოხდეს აგენტების კომუნიკაცია ცენტრალიზებულად. ზუსტად შავი დაფის არქიტექტურაც წარმოადგენს ისეთ არქიტექტურას სადაც მუშაობა ხდება ცენტრალიზებულად და კომუნიკაცია სტრუქტურული მესიჯებით[19].

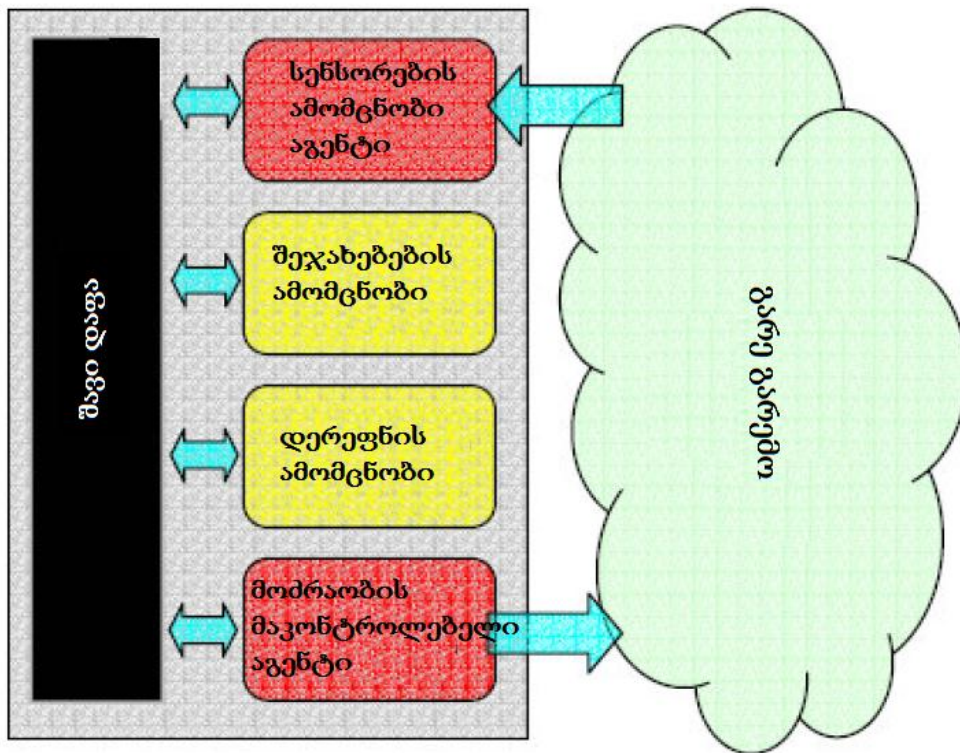
აგენტი აგენერირებს ინფორმაციას რომელიც მოიცავს ინფორმაციისმიხედვით შიდა მდგომარეობის და შემოგარენის შესახებ, ეს მესიჯი კი ისახება შავ დაფაზე. ეს მესიჯი წარმოადგენს სისტემის საწყის მდგომარეობას. ეს ყველაფერი საშუალებას აძლევს სისტემას იყოს მარტივი მოქნილი, სირთულეების გარეშე.

შავი დაფის არქიტექტურა შექმნილია იმისთვის რათა გაუკლავდეს რთული ამოცანებს, ცუდ სტრუქტურებს ეს ყველაფერი კი დაფუძნებულია საძიებო პროგრამულ ცოდნაზე და საშუალება აქვს გააერთიანოს სხვადასხვა ცოდნის წყაროები. მთავარი პრობლემა შავი დაფის არქიტექტურაში არის მოხდეს სტრატეგიის შენარჩუნება და რელურ დროში უსაფრთხოების უზრუნველყოფა.

10.4. მობილური რობოტი დერეფნის ნავიგაციისთვის

ამ მაგლითში ფოკუსირება მინდა გავაკეთო ავტონომიურ მანქანაზე, როგორც სატესტო საწოლზე რომელის მომავალი განვითარება იქნება ინტელექტუალური ეტლი იმ ადამიანებისთვის რომლებსაც სიარული არ შეუძლიათ.

ექსპერიმენტული რობოტი აღჭურვილია ოთხი აგენტისგან (სენსორების ამომცნობი აგენტი, შეჯახებების ამომცნობი აგენტი, დერეფნის ამომცნობი აგენტი და წამყვანი(მთავარი) აგენტი) და მოიცავს მაკონტროლილებად პროგრამულ უზრუნველყოფას, ალუმინის კოჭებს , პლასტიკურ კონექტორებს, ორ ნაბიჯიან ძრავს, 360 გრადუსით მბრუნავ საბურავს, ჩართვის მოდულს, ელემენტს, გადამღებ კამერას, ცხრა ინფრაწითელ სენსორს და დამატებით სხივებისა კონექტორების გამაგრებას[20].



მრავალი აგენტის სისტემა შავი დაფის გამოყენებით

სურ. 24 მრავალ აგენტის სისტემა შავი დაფის გამოყენებით

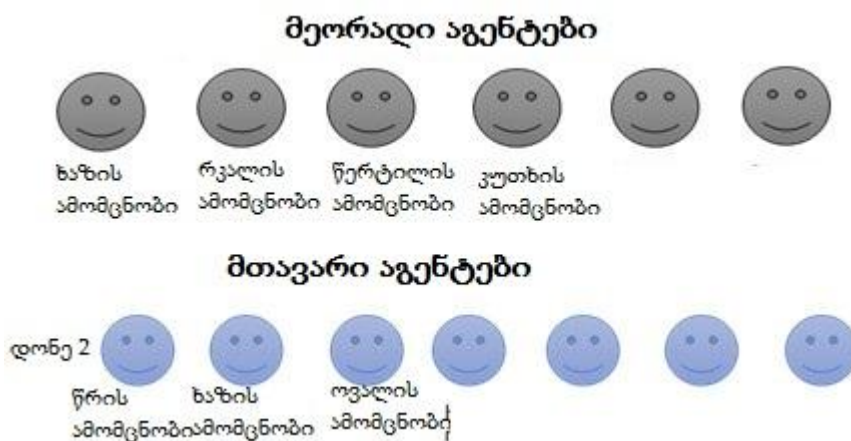
აგენტები და შავი დაფა დამოკიდებული იქნებიან მაკონტროლებელ კომპონენტზე. აგენტები თანამშრომლობენ შავ დაფასთან. შავი დაფა კი გამოიყენება ცენტრალური წყარო გაზიარებული ინფორმაციის.

მხოლოდ ორ აგენტს აქვს წვდომა გარემოსთან, ესენია სენსორის ამომცნობი აგენტი და მოძრაობის მაკონტროლებელი აგენტი.

ამ შემთხვევაში ჩვენ არ გვაქვს გლობალური მაკონტროლებელი აგენტების, აქედან გამომდინარე თითოეული დამოუკიდებლად ცდილობს წვლილი შეიტანოს სისტემის წინსვლაში. ძირითადად ოტხივე აგენტი ასრულებს სამუშაოს იღებს ინფორმაციას შავი დაფიდან ამუშავებს ამ ინფორმაციას და შემდეგ ისევ უკან აბრუნებს შავ დაფაზე დამუშავებულ ინფორმაციას.

აგენტების მიერ სიმბოლოების ამოცნობა

ამ სექციაში ჩვენ ვცდილობთ გაჩვენოთ თუ როგორ ცდილობენ აგენტები რათა მოახდინონ სიმბოლოების ამოცნობა. ჩვენ შევეცდებით დეტალურად ავხსნათ რა წყვილი შეაქვს თითოეულ აგენტს სიმბოლოების ამოცნობაში. თითოეულ აგენტს გააჩნია თავისი ცოდნა და ამაზე დაყრდნობით ისინი ცდილობენ სიმბოლოების ამოცნობას. ამ შემთხვევაში ჩვენ გვყავს მთავარი და მეორადი აგენტები, მთავარ აგენტებს გააჩნიათ უფრო მეტი ცოდნა ვიდრე მეორად აგენტებს და შესაბამისად გადაწყვეტილების მიღებაშიც უპირატესობა მათ ენიჭებათ. წარმოვიდგინოთ ჩვენ გავქვს აგენტები რომლებმაც იციან თუ როგორ უნდა ამოიცნონ წრე, წერტილი, ოთხკუთხედი, კუთხე, ოვალი, სამკუთხედი და ა.შ.



სურ. 25 სხვადასხვა დონის და ცოდნის მქონე აგენტები

სურათზე ნაჩვენებია თუ რა ცოდნა გააჩნია თითოეულ აგენტს და რის ამოცნობა შეუძლია თითოეულ მათთაგანს. ქვემოთ ავხსნათ თუ როგორ ახდენენ ისინი სიმბოლო „X“ ამოცნობას.

1. Line(x),Line(y), Cross(x,y) --> Symbol (z) {Name(z,"x")} - rank 1
2. Line(x),Line(y), Cross(x,y), Not Vertical(x) --> Symbol (z) {Name(z,"x")} - rank 2

პირველი დონის აგენტი ამბობს რომ არსებობს ორი ხზი და ისინი კვეთავენ ერთმანეთს და ეს სიმბოლო შეიძლება იყოს „X“. მეორე დონის აგენტი ამბობს რომ არსებობს ორი ხაზი და ისინი კვეთავენ ერთმანეთს და ასევე ისინი არ არიან ვერტიკალური ეს სიმბოლო არის „X“. მეორე დონის აგენტს აქვს მეტი პრიორიტეტი რომ მიიღოს გადაწყვეტილება ვიდრე პირველი დონის აგენტს

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 გამოხიზება 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.

გამოყენებული ლიტერატურა

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 2003
2. Zurab Bosikashvili, “ Extension 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. Paper 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. *IJCAI* 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