თსუ ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტის

კომპიუტერულ მეცნიერებათა დეპარტამენტი

ლექციათა კურსი

ფ ო რ მ ა ლ უ რ ი ე ნ ე ბ ი დ ა ს ა ს რ უ ლ ი

ა ვ ტ ო მ ა ტ ე ბ ი

ასოცირებული პროფესორი ტარიელ ხვედელიძე

-2013 –

ს ა რ ჩ ე ვ ი

ზოგადი ცნებები ფორმალური სისტემების შესახებ

§1. გამონათქვამები და ლოგიკური კავშირები.......... 4

§2. დე მორგანის წესები.......... 8

§3. გამონათქვამთა აღრიცხვის თეორიულ-სიმრავლური ინტერპრეტაცია.......... 9
სავარჯიშოები 9

§4. ფორმალური სისტემის ცნება.......... 12

§5. გამონათქვამთა აღრიცხვის ფორმალიზებული ვარიანტი.......... 15

§6. გამომდინარეობის მაგალითები.......... 17

§7. ფორმალური სისტემის განმარტება.......... 18

§8. ხომსკის გრამატიკა.......... 18

§9. შესაბამისობა გრამატიკებსა და ენებს შორის.......... 21

§10. არაცარიელი, სასრული და უსასრულო ენები.......... 22

§11. ენის მაგალითი, რომელიც არაა KC -ენა.......... 24

§12. ჩაკეტილობის თვისება. ოპერაციები KC - ენებზე.......... 26

§13. KC -ენების სპეციალური კლასები.......... 28

§14. ავტომატები და ენები.......... 31

§15. მაღაზიური მეხსიერების ავტომატი (МП-ავტომატი) და მისი ფუნქციონირება 32

§16. ენათა კლასი, დასაშვები МП-ავტომატში და წარმოქმნილი МП-ავტომატის მიერ.......... 34

§17. ავტომატის ფუნქციონირება ენის წარმოქმნისას.......... 35

§18. ავტომატის ფუნქციონირება ენის დაშვებისას.......... 36

§19. ავტომატური გრამატიკები.......... 37

§20. ფორმალური გრამატიკების კლასიფიკაცია.......... 39

ზოგადი ცნებები სასრული ავტომატების შესახებ

§21. დინამიური სისტემა.. 41

§22. სასრული ავტომატები 44

§23. ავტომატების ფუნქციონირების აღწერა... 46

§24. ავტომატების კლასიფიკაცია. სასრული ავტომატების მაგალითები..................57

§25. ხდომილებათა წარმოდგენა ავტომატებში.. 62

§26. ავტომატების ანალიზისა და სინთეზის ამოცანები... 68

§27. რეგულარული გამოსახულების მინიმალური ფორმა... 70

§28. რეგულარული გამოსახულებების მოცემა გრაფების ფორმით........................... 75

§29. ენები და დაპროგრამების სისტემები. გამოთვლითი ტექნიკის განვითარე-
ბის ქრონოლოგია. ზოგიერთი კომპიუტერული ფირმის მოკლე ისტორია......80

სავარჯიშოები.. 93

ფორმალური გრამატიკების თეორიას ცენტრალური შემადგენელი ადგილი უჭირავს მათემატიკურ ლინგვისტიკაში (ენათმეცნიერებაში). იგი წარმოიშვა, ერთის მხრივ, ლინგვისტიკის და, მეორეს მხრივ, გამომთვლელი მანქანების მათემატიკური უზრუნველყოფის მოთხოვნილებასთან დაკავშირებით. ფორმალურ გრამატიკებს შორის განსაკუთრებით მნიშვნელოვანია კონტექსტურად თავისუფალი გრამატიკების კლასი. იგი მეტად ხელსაყრელი აღმოჩნდა მრავალი ბუნებრივი ენის კონსტრუქციის აღსაწერად და შემდეგში ფართო გამოყენება პოვეს პროგრამირების ენების აღწერაში. თავისი სტრუქტურის სიმარტივისა და აბსტრაქტული ავტომატების თეორიის ზოგიერთ კონცეფციასთან მჭიდრო კავშირის წყალობით იგი მეტად ხელსაყრელია მათემატიკური შესწავლის თვალსაზრისითაც.

ზოგადი ცნებები ფორმალური სისტემების შესახებ

გამონათქვამები და ლოგიკური კავშირები

გამონათქვამი არის მტკიცებულება ან ისეთი წინადადება, რომლის შესახებაც შეიძლება ითქვას ჭეშმარიტია იგი თუ მცდარი. სხვა სიტყვებით რომ ვთქვათ, გამონათქვამის შესახებ ჭეშმარიტების ან მცდარობის მტკიცებულებას აზრი უნდა ჰქონდეს. რაიმე მტკიცებულების ჭეშმარიტებას ან მცდარობას ეწოდება მისი ჭეშმარიტების მნიშვნელობა.

მოვიყვანოთ ზოგიერთი წინადადება, რომელიც არაა გამონათქვამი:

ვინ ხართ? (კითხვა),

წაიკითხეთ ეს თავი შემდეგ მეცადინეობამდე! (ბრძანება),

ეს მტკიცებულება მცდარია (შინაგანი წინააღმდეგობრივი მტკიცებულება).

განვიხილოთ ორი ასეთი ფრაზა: “ორჯერ ორი ოთხია” და “ორჯერ სამი ხუთია”. ამ ორი ფრაზიდან, რომლებიც გამონათქვამებს წარმოადგენენ, პირველი მათგანი ჭეშმარიტია, ხოლო მეორე-მცდარი, რადგან პირველი მათგანის ლოგიკური მნიშვნელობაა ჭეშმარიტება, მეორის ლოგიკური მნიშვნელობა- მცდარობა. ფრაზები “ამ დღეს მოდიოდა წვიმა”, “მთელი რიცხვი არის მარტივი” არიან ცვლადიანი გამონათქვამები, რომლებიც გარემოებებზეა დამოკიდებული და შეუძლიათ მიიღონ

ჭეშმარიტი ან მცდარი მნიშვნელობა. ზოგადად ჭეშმარიტება აღინიშნება 1-იანით, ხოლო მცდარობა 0-ით.

გამონათქვამებსა და ცვლადიან გამონათქვამებს ავღნიშნავთ მცირე ლათინური p,q,r… ასოებით.

მეტყველებაში მარტივი წინადადებებიდან რთული წინადადებების მისაღებად გამოიყენება კავშირები – მეტყველების განსაკუთრებული ნაწილები, რომლებიც აერთებენ ცალკეულ გამონათქვამებს. ყველაზე ხშირად გამოიყენება "და", "ან", "არა", "თუ.. მაშინ", "მხოლოდ თუ", "და მაშინ და მხოლოდ მაშინ" კავშირები. გამონათქვამს ეწოდება მარტივი, თუ იგი არ შეიცავს არცერთ კავშირს. გამონათქვამს ეწოდება რთული, თუ იგი შეიცავს ერთ ან რამოდენიმე კავშირს.

ვთქვათ p და q აღნიშნავს შემდეგ მარტივ გამონათქვამებს:

p: "გივი ატარებს ავტომობილს",

q: "გოგის შავი თმები აქვს".

გამონათქვამი

"გივი ატარებს ავტომობილს და გოგის შავი თმები აქვს"

არის რთული გამონათქვამი და ის შედგება ორი ნაწილისაგან, რომლებიც შეერთებულია ერთმანეთთან "და" კავშირით. ეს გამონათქვამი სიმბოლურად შეიძლება შემდეგი სახით ჩავწეროთ:

p და q.

გამონათქვამი "წვიმა მოდის და ქარი ქრის" ჭეშმარიტად ითვლება მაშინ და მხოლოდ მაშინ, თუ ერთდროულად წვიმაც მოდის და ქარიც ქრის. ეს გამონათქვამი არის ორი – "წვიმა მოდის" და "ქარი ქრის" მარტივ გამონათქვამთა კონიუნქცია.

ორი p და q გამონათქვამის კონიუნქცია აღინიშნება $p \wedge q$ სიმბოლოთი ($\wedge$ სიმბოლო აღნიშნავს "და" კავშირს) და p და q-ს სხვადასხვა მნიშვნელობებისათვის აქვს შემდეგი მნიშვნელობები:

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

დიზიუნქცია $p \vee q$ ($\vee$ სიმბოლო აღნიშნავს “ან” კავშირს) ითვლება ჭეშმარიტად მაშინ და მხოლოდ მაშინ, როცა ერთ-ერთი გამონათქვამი მაინცაა ჭეშმარიტი.

p	q	$p\vee q$
0	0	0
0	1	1
1	0	1
1	1	1

ნებისმიერ q გამონათქვამს შეგვიძლია შევუსაბამოთ $\neg q$ (ან $\bar{q}$) -მისი უარყოფა, რომელიც განისაზღვრება შემდეგი ცხრილით:

q	$\neg q$
0	1
1	0

ნებისმიერ p, q გამონათქვამთა წყვილს შეგვიძლია შევუსაბამოთ გამონათქვამი $p \supset q$ (იმპლიკაცია).

p	q	$p \supset q$
0	0	1
0	1	1
1	0	0
1	1	1

$p \supset q$ გამოსახულებაში ლოგიკური ოპერატორი $\supset$ არავითარ შემთხვევაში არ უნდა ავურიოთ “$p \Rightarrow q$” მეტაიმპლიკაციაში, რომელიც აღნიშნავს, რომ q ნამდვილად გამომდინარეობს p- დან. ჩვენს შემთხვევაში p და q გამონათქვამების შინაარსი არ განიხილება და ერთადერთი, რაც მათ შესახებაა ცნობილი, არის მათი ლოგიკური მნიშვნელობა.

სავარჯიშოები:

p : ნაპოლეონი მოკვდა წმ. ელენეს კუნძულზე.

q : ვერცინჟეტორიქსი ულვაშს ატარებდა.

$p \supset q$ ღებულობს მნიშვნელობას 1 (ვერქსინჟეტორიქსი ულვაშს ატარებდა).

p : 2+2=5 .

q : 12 მარტივი რიცხვია.

$p \supset q$ ღებულობს მნიშვნელობას 1 .

p : მთვარე შედგება ცხვრის ყველისაგან .

q : 17 მარტივი რიცხვია .

$p \supset q$ ღებულობს მნიშვნელობას 1 .

p : 17 მარტივი რიცხვია .

q : 16 მარტივი რიცხვია .

$p \supset q$ ღებულობს მნიშვნელობას 0 .

თუ ჩვენ ვიცით, რომ p -ს მნიშვნელობაა 1 და $p \supset q$ -ს აქვს მნიშვნელობა 1, ჩვენ შეგვიძლია დავასკვნათ, რომ q -ს მნიშვნელობაა 1 .

ორი p და q გამონათქვამიდან შეგვიძლია შევადგინოთ $p \equiv q$ გამონათქვამი, რომელიც განისაზღვრება შემდეგი ცხრილით:

p	q	$p \equiv q$
0	0	1
0	1	0
1	0	0
1	1	1

აქ ფაქტიურად წარმოდგენილია გამონათქვამი “ p ეკვივალენტურია q –სი”, ანუ “ p - ს აქვს იგივე ლოგიკური მნიშვნელობა , რაც q – ს”.

როგორც იმპლიკაციის შემთხვევაში, ლოგიკური ეკვივალენტობის შემთხვევაშიც შესაერთებელი გამონათქვამების შინაარსი მხედველობაში არ მიიღება.

$\wedge$ და $\vee$ სიმბოლოებს ბინარულ კავშირებს უწოდებენ, რადგანაც ისინი გამონათქვამთა ორ წყვილს აერთებს. $\neg$ სიმბოლო არის უნარული კავშირი, რადგან იგი მხოლოდ ერთი გამონათქვამისათვის გამოიყენება.

ვთქვათ გვაქვს p, q ... ცვლადების სასრული ან უსასრულო რაოდენობა. მათი და ლოგიკური ოპერაციების საშუალებით შეგვიძლია შევქმნათ გამოსახულებები. მაგ: $(p \vee q) \wedge q$, $(p \supset q) \vee (q \supset p)$ და სხვ.

არსებობენ გამოსახულებები, რომლებიც ღებულობენ 1-ის ტოლ მნიშვნელობას, როგორიც არ უნდა იყოს მასში შემავალი ცვლადების მნიშვნელობები. მაგალითად: $p \vee \bar{p},\ p \supset p$. ასეთ გამონათქვამებს უწოდებენ იგიურად ჭეშმარიტ გამონათქვამებს ანუ ტავტოლოგიას.

არსებობენ გამოსახულებები, რომლებიც ღებულობენ 0-ის ტოლ მნიშვნელობას მასში შემავალი ცვლადების ნებისმიერი მნიშვნელობისათვის. მაგალითად: $p \wedge \bar{p},\ (p \vee \bar{p}) \supset (p \wedge \bar{p})$. ასეთ გამოსახულებებს უწოდებენ იგიურად მცდარ ანუ წინააღმდეგობრივ გამონათქვამებს.

არსებობენ გამოსახულებები, რომლებიც ღებულობენ ან 1 ან 0 მნიშვნელობებს მასში შემავალი ცვლადების მნიშვნელობებზე დამოკიდებულების მიხედვით. მაგალითად:$(p \supset p) \vee (p \wedge q)$. თუ ამ გამონათქვამს ავღნიშნავთ r -ით, მაშინ

p	q	r
0	0	1
0	1	1
1	0	0
1	1	1

დე მორგანის წესები

ადვილად შეგვიძლია დავრწმუნდეთ, რომ გამოსახულებები

$$\overline{\left(p \bigvee q\right)} = \bar{p} \bigwedge \bar{q}, \qquad \overline{\left(p \bigwedge q\right)} = \bar{p} \bigvee \bar{q}$$

წარმოადგენენ ტავტოლოგიურ ეკვივალენტურობას. მათ ეწოდებათ დე მორგანის წესები.

ნებისმიერი r, s, t ცვლადების ან გამოსახულებებისათვის გვაქვს:

1. $r \vee r = r;\ r \wedge r = r$: იდემპოტენტურობა.
2. $r \vee s = s \vee r;\ r \wedge s = s \wedge r$: კომუტატიურობა.
3. $r \vee (s \vee t) = (r \vee s) \vee t;\ r \wedge (s \wedge t) = (r \wedge s) \wedge t$: ასოციაციურობა.
4. $r \vee (r \wedge s) = r = r \wedge (r \vee s)$

5. $\begin{cases} r \vee (s \wedge t) = (r \vee s) \wedge (r \vee t) \\ r \wedge (s \vee t) = (r \wedge s) \vee (r \wedge t) \end{cases}$: დისტრიბუტიულობა.

6. $r \vee \bar{r} =$ ჭეშმარიტი; $r \wedge \bar{r} =$ მცდარი.

7. $\begin{cases} \overline{r \vee s} = \bar{r} \wedge \bar{s} \\ \overline{r \wedge s} = \bar{r} \vee \bar{s} \end{cases}$: გაორება (დე მორგანის წესი).

8. $\overline{(\bar{r})} = r$

გამონათქვამთა აღრიცხვის თეორიულ-სიმრავლური ინტერპრეტაცია

ავიღოთ $P, Q, \ldots$ სიმრავლეთა ოჯახი და $p, q, \ldots$ცვლადების ინტერპრეტაცია მოვახდინოთ როგორც $x \in P, x \in Q, \ldots$ სახის გამოსახულება ერთი და იმავე x-ით. მაშინ $p \wedge q$ ინტერპრეტირდება, როგორც $x \in P \cap Q$, ხოლო $p \vee q$ - როგორც $x \in P \cup Q$ და ა. შ.

გამონათქვამთა აღრიცხვა ახლა მიიღებს შემდეგ ინტერპრეტაციას: მის თეორიულ-სიმრავლურ სახეს წარმოადგენს სიმრავლეთა ალგებრა, მიღებული $P, Q, \ldots$- დან თანაკვეთის, გაერთიანების ოპერაციებისა და რაიმე უნივერსალური U სიმრავლის მიმართ დამატების აღებით, რომელიც თავის მხრივ შეიცავს ყველა $P, Q, \ldots$- ს და შეესაბამება ჭეშმარიტებას. უნივერსალური U სიმრავლის მიმართ დამატების აღება შეესაბამება უარყოფას; ცარიელი $\emptyset$ სიმრავლე, რომელიც წარმოადგენს დამატებას U - თვის, შეესაბამება სიცრუეს.

სავარჯიშოები:

1. მოცემულია გამონათქვამები: “ცხელა $\equiv c$ ”, “წვიმა მოდის $\equiv p$”, “ჰაერი მშრალია$\equiv s$’’. შემდეგი გამონათქვამები გამოსახეთ სიმბოლოებით:

ა. “წვიმა მოდის და ცხელა”.

ბ. “ცხელა, მაგრამ ჰაერი არაა მშრალი”.

გ. “ჰაერი ნესტიანია ან ცხელა” .

დ. “წვიმა არ მოდის ან ჰაერი მშრალია”.

2. დავუშვათ, რომ გამონათქვამი “ცივა და ჰაერი ნესტიანია” - ჭეშმარიტია. გამოიკვლიეთ წინამდებარე წინადადებების მნიშვნელობები.

3. ავღნიშნოთ p -თი წინადადება “თვითმფრინავი დაფრინავს მაღლა” , ხოლო q -თი წინადადება – “თვითმფრინავი კრეფს სიმაღლეს”. გადავთარგმნოთ შემდეგი გამოსახულებები: ა)$p \wedge q$ ბ)$p \wedge \bar{q}$ გ)$\bar{p} \wedge \bar{q}$ დ)$p \vee \bar{q}$ ე)$\overline{p \wedge q}$ ვ)$\overline{p \vee q}$ ზ)$\overline{\bar{p} \vee \bar{q}}$

4. ვთქვათ p აღნიშნავს “მე მაქვს ფირსაკრავი” , ხოლო q - “მე მაქვს ფირფიტები”. გადავთარგმნოთ გამოსახულება: $\overline{\bar{p} \vee \bar{q}} \wedge \bar{p}$

5. ორი ცვლადიდან ან გამოსახულებიდან გამომდინარე, შეიძლება ავაგოთ “p w q ” ფორმულა, სადაც w მოიცემა შემდეგი ცხრილით:

p	q	w
0	0	0
0	1	1
1	0	1
1	1	0

და შეესაბამება მკაცრ დიზიუნქციას: ან ერთი ან მეორე, მაგრამ არც ერთი და არც მეორე ერთად. მიეცით w - ს თეორიულ-სიმრავლური ინტერპრეტაცია. არის თუ არა ეს ოპერაცია ასოციაციური?

6. ანალოგიური კითხვა | ოპერაციისათვის (შეფერის შტრიხი ანუ უთავსებადობა), რომელიც ასე განისაზღვრება

p	q	$p\|q$
0	0	1
0	1	1
1	0	1
1	1	0

შეამოწმეთ შემდეგი ტავტოლოგიური ეკვივალენტურობა

$$\bar{p} = p|p.$$

$$p \vee q = (p \mid p) \mid (q \mid q).$$

$$p \wedge q = (p \mid q) \mid (p \mid q).$$

$$p \supset q = p \mid (q \mid q).$$

7. ქვემოთ მითითებული წინადადებებიდან იპოვეთ გამონათქვამები. მიუთითეთ მათი ჭეშმარიტი მნიშვნელობა.

ა. რომელი საათია?

ბ. რიცხვი 1 არის უმცირესი მთელი დადებითი რიცხვი.

გ. თუ $x = 3$, მაშინ $x^2 = 6$.

დ. უფრთხილდი ავტომობილს!

ე. სამხრეთ დაკოტა - სამხრეთი შტატია.

8. ვთქვათ p, q, r აღნიშნავს შემდეგ გამონათქვამებს:

p : მარსზე მოგზაურობა ძვირია.

q : მე ვიმოგზაურებ მარსზე.

r : მე მაქვს ფული.

ჩაწერეთ სიმბოლური ფორმით შემდეგი გამონათქვამები:

ა) მე არ მაქვს ფული და მე ვერ ვიმოგზაურებ მარსზე.

ბ) მე არ მაქვს ფული და მარსზე მოგზაურობა ძვირია ან მე ვიმოგზაურებ მარსზე.

გ) მე მაქვს ფული და მე გავფრინდები მარსზე.

დ) მარსზე მოგზაურობა არაა ძვირი და მე გავფრინდები მარსზე ან მარსზე მოგზაურობა ძვირია და მე ვერ გავფრინდები მარსზე .

9. შეადგინეთ ჭეშმარიტების ცხრილი თითოეული გამონათქვამისათვის 8) –ე მაგალითში.

10. ვთქვათ p, q, r აღნიშნავს შემდეგ გამონათქვამებს:

p : ეს თამაში ძალიან ძნელია.

q: მე ვთამაშობ ჭადრაკს.

r : ჭადრაკის თამაში საჭიროებს დროს.

გადათარგმნეთ შემდეგი გამოსახულებები როგორც ჩვეულებრივი გამონათქვამები:

ა) $q \wedge r$;
ბ) $\bar{p} \vee \bar{q}$;
გ) $(p \vee r) \wedge q$;
დ) $p \wedge q \wedge r$.

11 . შეადგინეთ ჭეშმარიტების ცხრილი თითოეული გამონათქვამისათვის 10) –ე მაგალითში.

12. შეადგინეთ ჭეშმარიტების ცხრილი შემდეგი გამონათქვამებისათვის :

ა) $p \wedge (q \vee \overline{r})$;
ბ) $(q \wedge \overline{r}) \vee (\bar{p} \wedge r)$;
გ) $(\overline{p \wedge r}) \vee (\bar{q} \wedge r)$;
დ) $(p \wedge r) \vee (p \wedge \bar{q})$.

ფორმალური სისტემის ცნება

ფორმალური სისტემის ცნება ისტორიულად ჩამოყალიბდა აქსიომატური თეორიის ცნების საფუძველზე. ჯერ კიდევ ევკლიდეს გეომეტრია იყო ნაწილობრივ აქსიომატიზირებული. სრულად კი აქსიომატური მიდგომა ცხადად გაფორმდა 19-ე საუკუნის ბოლოს.

აქსიომებიდან გამომდინარე, შეგვიძლია დავამტკიცოთ თეორემები. მაგალითად, თეორემა: ორ განსხვავებულ წრფეს შეიძლება ჰქონდეს ერთი საერთო წერტილი.

წერტილებისა და წრფეების შესახებ ყველა არსებული ინფორმაცია კი მოცემულია შემდეგ აქსიომებში;

წრფე არის წერტილთა სიმრავლე.

ნებისმიერ ორ წერტილზე შეიძლება გავავლოთ რაიმე წრფე.

ორ წერტილზე გაივლება მხოლოდ ერთი წრფე.

ნებისმიერ წრფეზე არსებობს არანაკლებ ორი წერტილი.

არსებობს სამი წერტილი მაინც, რომლებიც არ ეკუთვნის ერთ წრფეს.

ფორმალურ სისტემას საქმე აქვს L ენასთან, რომელიც აიგება რაიმე ალფავიტის მიხედვით აქსიომებისა და გამომდინარეობის წესების საშუალებით. რომ ვილაპარაკოთ ფორმალური სისტემის ელემენტებზე, მათ წესებსა და სხვ. , უნდა ვისარგებლოთ რომელიმე ენით. ეს ენა უნდა შეიცავდეს ობიექტების სახელებს, ურთიერთობის სახეებს და სხვ.

მაგალითი. განვიხილოთ ფორმალური სისტემა, განსაზღვრული $U = \{a, b, c\}$ ალფავიტით. რომ ვისაუბროთ ამ ფორმალური სისტემის სიტყვებზე, ჩვენ დაგვჭირდება ისეთი სახელი, როგორიცაა, მაგალითად, '$a'baca'$', რომელიც მიღებულია ალფავიტის ასოებისაგან სპეციალური ბრჭყალებით.

კონკატენაციის ოპერაციის აღსანიშნავად გამოვიყენებთ ≪·≫ სიმბოლოს და დავწერთ, მაგალითად, ასე: თუ $A = 'abaca'$ და თუ $B = 'ba'$, მაშინ $A \cdot B = 'abacaba''$. აქ “=” ნიშანი აღნიშნავს, რომ ამ ნიშნის მარცხნივ მდგომი გამოსახულება და მარჯვნივ მდგომი გამოსახულება არის ერთი და იგივე ობიექტის ორი სხვადასხვა სახელი.

ზემოთ ნათქვამი გარკვეულ წარმოდგენას გვაძლევს მეტაენის შესახებ. ზოგიერთი ავტორი ამ ნიშნით აღნიშნავს ნიშნების განსაზღვრულ სისტემას, რომლის აღსაწერად გამოიყენება მეტამეტაენა .ზოგიერთი ავტორი კი მეტაენის ქვეშ გულისხმობს ჩვეულებრივი ბუნებრივი ენისა და სპეციალური ნიშნების ერთობლიობის გაერთიანებას, რომელიც საშუალებას იძლევა აღიწეროს ფორმალური სისტემა. ჩვენ მივემხრობით მეორე ფორმულირებას.

მეტაენის სიმბოლოები, რომლებსაც შემდგომში გამოვიყენებთ, განისაზღვრებიან ცალსახად და ლოგიკურად განსხვავებული ქართული ფრაზებით. მაშასადამე, ეს სიმბოლოები თავისი არსით შემცირებას წარმოადგენენ.

მაგალითი. ვთქვათ ფორმალური სისტემის ალფავიტი არის

$$U = \{0,1,2,3,4,5,6,7,8,9, +, -\}.$$

ჩვენ გვინდა განვსაზღვროთ სიტყვათა ორი სიმრავლე. ერთი მათგანია: “მთელი რიცხვები ნიშნის გარეშე” სიმრავლე, ხოლო მეორე სიმრავლეა “მთელი რიცხვები”.

ამ სიმრავლეების შექმნის წესების ჩამოსაყალიბებლად შემოვიღოთ მეტაენური ცვლადი, რომელიც მიიღებს ნებისმიერ მნიშვნელობას {0,1,2,3,4,5,6,7,8,9} სიმრავლიდან - ESS მეტაენური ცვლადი, რომელიც ღებულობს მნიშვნელობებს “მთელი რიცხვები ნიშნის გარეშე” სიმრავლიდან და მეტაენური ცვლადი E, რომელიც თავის მნიშვნელობებს ღებულობს “მთელი რიცხვები” სიმრავლიდან.

ტეხილი “<” და “>” ფრჩხილები გამოიყენება შესასვლელი ცვლადების განსაცალკავებლად; ვერტიკალური “|” ხაზი აღნიშნავს “ან” კავშირს. “::=”სიმბოლო აღნიშნავს იმის უფლებას, რომ შეიცვალოს ვერტიკალური ხაზებით განცალკევებული მარცხნივ მდგომი გამოსახულება მარჯვნივ მდგომი რაიმე გამოსახულებით.

მოვიყვანოთ ჩვენთვის საინტერესო სიმრავლეების წარმოქმნის წესები.

წესი 1. $<ESS>::=<c> \mid <ESS><c>$

წესი 2. $<E>::=<ESS> \mid +<ESS> \mid -<ESS>$

1 წესის გამოყენებით შეგვიძლია , მაგალითად, მივიღოთ
$<ESS>::=3,$
$<ESS>::=35$

უკანასკნელზე წესი 2-ის გამოყენება მოგვცემს

$<E>::=-35$

დავუშვათ, რომ რაიმე ფორმალური სისტემის ალფავიტი შეიცავს"$\wedge$" ასოს და გვაქვს შემდეგი წესი: “თუ A და B არის θ ტიპის სიტყვები, მაშინ სიტყვა $A\cdot\wedge\cdot b$ აგრეთვე θ ტიპის სიტყვაა. გამონათქვამები “თუ ... , მაშინ .. “ . ამ კონტექსტში აბსოლუტურად ცალსახად განიხილება, მაგრამ ჩაწერის სიმოკლისათვის იგი შეიძლება შეიცვალოს მეტაენური სიმბოლოთი, მაგალითად “$\Rightarrow$”სიმბოლოთი.

საზოგადოდ, ფორმალური სისტემა იქმნება როგორც რაღაც მეორის იდეალიზირებული სახე. ამ რაღაც მეორეს შეიძლება დავარქვათ, მაგალითად, “თეორია” , რომლის მიმართებითაც ეს ფორმალურ სისტემა აზრს იძენს. მაშინ ფორმალური სისტემა წარმოადგენს რაიმე თეორიის ფორმალიზაციას, წარმოშობს რა ამ თეორიის სინტაქსს. თეორია კი წარმოადგენს ამ ფორმალური სისტემის ინტერპრეტაციას.

ფორმალური სისტემის სწორად შედგენილი სიტყვები (ფრაზები) აზრს იძენენ ფორმალიზირებული თეორიის ჩარჩოებში. პირიქით, ნებისმიერ გამონათქვამს, რომელსაც აზრი აქვს მოცემულ თეორიაში, შეესაბამება სწორად შედგენილი სიტყვების (ფრაზების) რაღაც კლასი ფორმალურ სისტემაში.

გამონათქვამთა აღრიცხვის ფორმალიზებული ვარიანტი

ვთქვათ გვაქვს აბსტრაქტული მათემატიკური ობიექტების თვლადი სიმრავლე, რომლებიც ქმნიან აბსტრაქტულ ალფავიტს. ეს ობიექტები მოიძებნებიან ურთიერთცალსახა შესაბამისობით შემდეგ გრაფიკულ სიმბოლოებთან, რომლებიც ქმნიან ალფავიტს :

$$U = \{\,], [, \neg, \supset, a_1, a_2, \ldots a_n, \ldots \}$$

აქ "{" , "}" და "," მეტაენის სიმბოლოებია; "]" და "[" ფორმალური ფრჩხილების სახელებია; "¬ " - ფორმალური უარყოფის სახელი ; "⊃ " - ფორმალური იმპლიკაციის სახელი ; a_i - ელემენტარული ობიექტების სახელები, რომლებსაც ატომები ეწოდებათ და ადგენენ თვლად სიმრავლეს.

გამოვყოთ ფორმალური ენის რაღაც ნაწილი, რომელიც შეესაბამება კონკრეტული ფორმულების რაღაც სიმრავლეს. წესები, რომლებითაც მოცემულია ფორმულების ეს სიმრავლე, საჭიროებს ფრჩხილებისა და ოპერაციების სიმბოლოების "სწორ" გამოყენებას. ეს წესებია:

F1. ყოველი ატომი არის ფორმულა.

F2. თუ X სიტყვა ფორმულაა, მაშინ $\bar{X}$ სიტყვაც ფორმულაა.

F3. თუ X და Y სიტყვები ფორმულებია, მაშინ $[X \supset Y]$ სიტყვაც ფორმულაა.

F4. არავითარი სხვა ფორმულა არ არსებობს.

აქ X და Y წარმოადგენს მეტაენის ცვლადებს.

მაგალითები: a_1 წარმოადგენს ფორმულას F1-ის ძალით. a_2არის ფორმულა F1-ის ძალით. $[a_1 \supset a_2]$ არის ფორმულა F3-ის ძალით. $\overline{[a_1 \supset a_2]}$ არის ფორმულა F2-ის ძალით; a_3 არის ფორმულა F1 -ის ძალით; $[\,\overline{[a_1 \supset a_2]} \supset a_3]$ არის ფორმულა F3-ის და F1-ის ძალით.

ყველა ფორმულების სიმრავლე შეიძლება მივიღოთ, თუ მიმდევრობით ავაგებთ უფრო დიდი სიგრძის ფორმულებს.

1 სიგრძის ფორმულები არის ატომები. რადგან ფორმულებში არაფერი არ იშლება, თითოეული ფორმულა შეიცავს ერთ ატომს მაინც.

2-ის სიგრძის ფორმულებია

$$\overline{a_1}, \overline{a_2}, \dots \overline{a_n}, \dots$$

3-ის სიგრძის ფორმულები:

$$\overline{\overline{a_1}}, \overline{\overline{a_2}}, \dots \overline{\overline{a_n}}, \dots$$

4-ის სიგრძისაა

$$\overline{\overline{\overline{a_1}}}, \overline{\overline{\overline{a_2}}}, \dots \overline{\overline{\overline{a_n}}}, \dots$$

5-ის სიგრძისაა

$$\overline{\overline{\overline{\overline{a_1}}}}, \overline{\overline{\overline{\overline{a_2}}}}, \dots \overline{\overline{\overline{\overline{a_n}}}}, \dots$$

და $[a_1 \supset a_1], [a_1 \supset a_2], \dots [a_n \supset a_p], . .$ და ა.შ.

თუ რაიმე M სიტყვა არ შეიცავს ატომებს, მაშინ M არ არის ფორმულა. თუ შეიცავს ატომებს, მაშინ საჭიროა ამ ატომებისაგან შევადგინოთ ისეთი ყველა შესაძლებელი ფორმულა, რომელთა სიგრძე არ იქნება M -ის სიგრძეზე მეტი. M სიტყვა იქნება ფორმულა მაშინ და მხოლოდ მაშინ, როცა M არის ერთ-ერთი მათგანი ამ ფორმულებიდან. იმის განსაზღვრისათვის, არის თუ არა M ფორმულა, ეს პროცესი შეიძლება "მექანიზირებული" იქნეს. ამავე დროს არ არის აუცილებელი იმის გარკვევა, თუ რამდენად "ეფექტური" ან "ხელსაყრელია" ეს პროცესი.

მაგალითები. $\overline{[[}$ არ არის ფორმულა (ამ გამოსახულებაში არ არის ატომები).

$[\overline{a_2} \supset a_2]$ არის ფორმულა, რადგანაც ეს გამოსახულება შედის ყველა იმ ფორმულათა სიმრავლეში, რომელთა სიგრძეც ≤ 6 და რომელთა შედგენაც შესაძლებელია a_2 -ით. ეს სიმრავლე შედგება შემდეგი ფორმულებისაგან: $\overline{a_2}$, $\overline{\overline{a_2}}$, $\overline{\overline{\overline{a_2}}}$, $\overline{\overline{\overline{\overline{a_2}}}}$, $\overline{\overline{\overline{\overline{\overline{a_2}}}}}$, $[a_2 \supset a_2]$, $\overline{[a_2 \supset a_2]}$, $[\overline{a_2} \supset a_2]$, $[a_2 \supset \overline{a_2}]$

განვიხილოთ ახლა სხვა ენა (ფორმალური), რომელიც შედის წინამდებარე ენაში. ამ ენას ვუწოდოთ თეორემათა ერთობლიობა.

თეორემები შეესაბამებიან იმას, რასაც არაფორმალურ ვარიანტში ეწოდებოდა ტავტოლოგიები.

ჯერ განვიხილოთ ელემენტარული თეორემები შემდეგი შეთანხმებით: თუ X , Y , Z სიტყვები ფორმულებია, მაშინ

T1. [X⊃[Y⊃X]] არის ელემენტარული თეორემა.

T2. $[[\bar{X}\supset\bar{Y}]\supset[Y\supset X]]$ არის ელემენტარული თეორემა.

T3. $[\,[X\supset[Y\supset Z]\,]\supset[[X\supset Y]\supset[X\supset Z]\,]\,]$ არის ელემენტარული თეორემა.

T4. სხვა ელემენტარული თეორემები არ არის.

თუ ელემენტარულ თეორემებს განვიხილავთ როგორც აქსიომებს, მაშინ T1-T3 გამოსახულებები, რომლებიც მეტაენას მიეკუთვნებიან, იქნებიან აქსიომათა სქემები.

ადვილი დასანახია, რომ ელემენტარული თეორემები F2 და F3-ის ძალით არიან ფორმულები. ადვილი საჩვენებელია აგრეთვე, რომ ნებისმიერი M სიტყვისათვის შეიძლება დავადგინოთ არის ეს სიტყვა ელემენტარული თეორემა თუ არა.

შევნიშნოთ, რომ T1 ასახავს იმპლიკაციის "კვაზიპარადოქსულ" ხასიათს. T2 არის კონტრპოზიციის წესი. ყურადღება მივაქციოთ, რომ როცა ჩვენ ვიყენებთ ჭეშმარიტების შინაარსობრივ ცნებას, ცხადია, რომ $\bar{\bar{P}}=P$.ამიტომ ამ შემთხვევაში სულ ერთია რას დავწერთ,$[\bar{X}\supset\bar{Y}]\supset[Y\supset X]$ an $[Y\supset X]\supset[\bar{X}\supset\bar{Y}]$.

მაგრამ საქმე ასეც არაა, როცა თავიდანვე არ ვითვალისწინებთ ჭეშმარიტების ცნებას (ე.ი. როცა მხედველობაში გვაქვს სხვა ინტერპრეტაცია). T3 განპირობებულია ტექნიკური მოხერხებულობისათვის. ელემენტარული თეორემებიდან გამომდინარე, შეიძლება ავაგოთ თეორემები გამოყვანის წესების საფუძველზე:

თუ X არის სიტყვა, ხოლო $[X\supset Y]$ თეორემა, მაშინ Y სიტყვა აგრეთვე თეორემაა.

ადვილი დასანახია, რომ თეორემები აუცილებლად წარმოადგენენ ფორმულებს. შევნიშნოთ, რომ ჩვენს მიერ შემოღებული წესი არის ცნობილი modus ponens წესი.

გამომდინარეობის მაგალითები

ვაჩვენოთ, რომ $[a_1\supset a_1]$ და ზოგადად $[X\supset X]$ არის თეორემა. ამისათვის ავაგოთ თეორემა, რომელშიც $[a_1\supset a_1]$ აღმოჩნდება Y-ის პოზიციაში წინა წესის საფუძველზე.

(1). a_1 არის ფორმულა F1 -ის ძალით.

(2). $[a_1\supset a_1]$ სიტყვა, რომელსაც ავღნიშნავთ M-ით, არის ფორმულა (1)-ისა და F3-ის ძალით.

(3). $[a_1\supset M]$ არის ფორმულა (1),(2) და F3-ის ძალით.

(4). $[a_1 \supset M]$ არის თეორემა (1) და T1 -ის ძალით.

(5). $[a_1 \supset [M \supset a_1]]$ არის თეორემა (1),(2) და T1-ის ძალით.

(6). $[a_1 \supset [M \supset a_1]] \supset [[a_1 \supset M] \supset [a_1 \supset a_1]]$ არის თეორემა (1), (2) და T3 -ის ძალით.

(7). მარჯვენა მხარე (6)-ე სიტყვისა არის თეორემა (5),(6) და გამომდინარეობის ძალით.

(8). $[a_1 \supset a_1]$ არის თეორემა (4),(7) და გამომდინარეობის ძალით.

ასევე შეიძლება ვაჩვენოთ, რომ ნებისმიერი x ფორმულისათვის$[X \supset X]$ გამოსახულება არის თეორემა.

მაშასადამე, ჩვენ ავაგეთ "სისტემა", რომელიც შედგება ორი ურთიერთ დაკავშირებული ნაწილისაგან. ამ სისტემას შეიძლება მივცეთ განსაღვრული აზრი: იგი ფორმალიზებას უკეთებს გამონათქვამთა აღრიცხვას , განხილულს გაგების ინტუიტიურ დონეზე.

ფორმალური სისტემის განმარტება

გავუკეთოთ რეზიუმე ზემოთ ნათქვამს ფორმალური სისტემის შესახებ.

ფორმალური სისტემა ეწოდება $C =< U, L, Z, R >$ ოთხეულს, სადაც U-თვლადი ალფავიტია; L-ფორმალური ენა U-ზე; Z-U სიმრავლის რაიმე ქვესიმრავლე ან აქსიომათა სიმრავლე; R-გამომდინარეობის წესების ერთობლიობა.

გამომდინარეობის R წესი არის წესი, რომელიც ყოველ $< y_1, y_2, \dots y_n >$ -ეულს შეუსაბამებს დეკარტული $U * U * \dots * U$ ნამრავლის სიმრავლიდან ერთ გარკვეულ x ფორმულას.

ფორმალური L ენა არის Z ალფავიტში შემავალი ასოებისაგან შედგენილი სიტყვათა მიმდევრობა (ჯაჭვი), გამოყოფილი რაღაც სასრული რაოდენობის წესების საშუალებით.

ხომსკის გრამატიკა

ენებსა და გრამატიკებს შორის ერთერთი მნიშვნელოვანია ხომსკის გრამატიკა და შესაბამისად მისგან წარმოქმნილი ენა, რომელსაც ხომსკის ენა ეწოდება. თვითონ ხომსკი ამ კლასის ენებს უწოდებდა "კონტექსტურად-თავისუფალ" ანუ

“უკონტექსტო” ენას. ტერმინი „კონტექსტურად-თავისუფალი” ანუ „უკონტექსტო“ აღნიშნავს იმ ფაქტს, რომ დამხმარე სიმბოლოს შეცვლა ჯაჭვით არაა დამოკიდებული კონტექსტზე (ე.ი. დამხმარე სიმბოლო ნებისმიერ დროს შეგვიძლია შევცვალოთ ჯაჭვით).

შემდეგში “კონტექსტურად-თავისუფალ” გრამატიკებსა და ენებს ავღნიშნავთ შემოკლებით, როგორც KC -გრამატიკები და KC -ენა (KC -რუს. კონტექსტურად-თავისუფალი).

განმარტება. ხომსკის გრამატიკა მოიცემა ობიექტების შემდეგი ოთხეულის მითითებით;

V_T -ტერმინარული ანუ ძირითადი სასრული ალფავიტი.

V_A -დამხმარე სასრული ალფავიტი.

$S\epsilon V_A$ აქსიომა.

სასრული სიმრავლე A→φ სახის KC -წესებისა, სადაც

$A\epsilon V_A,\ \varphi\epsilon\{V_A\cup V_t\}^*$

განმარტება. ხომსკის ენა ეწოდება ფრაზათა სიმრავლეს, წარმოქმნილს ხომსკის გრამატიკის ტერმინარული სიტყვებით.

ხომსკის გრამატიკაში “დამტკიცების” ნაცვლად მიღებულია ტერმინი “გამომდინარეობა”.

თუ გვაქვს U სიმრავლე, მაშინ U^* სიმბოლოთი ავღნიშნოთ სიმრავლე, წარმოქმნილი U- ს ელემენტების კომბინაციით. მაგალითად, თუ $A='\,baca'$ და $B='ac'$, მაშინ $AB='bacaac'$, $BA='acbaca'$. თუ $U=\{a\}$, მაშინ U^* იქნება a- ს ხარისხების სიმრავლე. თუ $U=\{0,1\}$, მაშინ U^* სიმრავლის ჯაჭვები იქნება მთელი ორნიშნა რიცხვები. ამ შემთხვევაში კონკატენაცია წარმოადგენს საკმაოდ რთულ არითმეტიკულ ოპერაციას.

მაგალითი 1. ვთქვათ გვაქვს G_M გრამატიკა

$$V_A=\{S\};\ V_T=\{a,b,c\}$$

$$\left|\begin{matrix} S\to aSa \\ S\to bSb \\ S\to c \end{matrix}\right|$$

ერთერთი გამომდინარეობა G_M -ში არის

$$S \to aSa \to aaSaa \to aabSbaa \to aabcbaa$$

ჩვენ დავწერთ $S \Rightarrow aabcbaa.$

ჩანაწერი $\varphi \Rightarrow \Psi$ აღნიშნავს, რომ არსებობს გამომდინარეობა, რომელიც იწყება φ ჯაჭვით (მიმდევრობით) და მთავრდება Ψ ჯაჭვით.

ცხადია, რომ G_M გრამატიკა წარმოშობს იმ ფრაზებს, რომლებიც აგებულია $\{xc\tilde{x}\}$ სქემით, სადაც x არის ნებისმიერი ჯაჭვი შედგენილი a და b სიმბოლოებით, ხოლო $\tilde{x}$ მისი სარკული ანასახი. ეს ასე შეიძლება ჩავწეროთ

$$L_M = L(G_M) = \{xc\tilde{x} | x\epsilon\{a,b\}^*\}$$

2. ვთქვათ გვაქვს G_0 გრამატიკა

$$V_A = \{S, A\};\ V_T = \{a, b\}$$

$$\begin{vmatrix} S \to aA \\ A \to Ab \end{vmatrix}$$

ცხადია რომ გამომდინარეობა ამ გრამატიკაში უსასრულოდ გრძელდება და არ მიგვიყვანს ტერმინარულ ჯაჭვამდე. G_0 წარმოქმნის ცარიელ ენას.

წესების კლასიფიკაცია. ნებისმიერი კონტექსტურად თავისუფალი G გრამატიკა წარმოქმნის არატერმინარული ჯაჭვების რაიმე მიმდევრობას და რაიმე ტერმინარული ჯაჭვების (შეიძლება ცარიელი) სიმრავლეს, რომლებიც ქმნიან L(G) ენას. დავარქვათ წინატერმინარული არატერმინარულ ჯაჭვს, რომლისგანაც ზუსტად ერთი წესის გამოყენებით მიიღება ტერმინარული ჯაჭვი. ცხადია, რომ წინატერმინარული ჯაჭვი შეიცავს ზუსტად ერთ არატერმინარულ სიმბოლოს.

L(G) ენის არაცარიელობის აუცილებელ პირობას წარმოადგენს G -ში ერთი ან რამოდენიმე A→τ სახის ტერმინარული წესის არსებობა, სადაც $A\epsilon V_A, \tau\epsilon V_T^*$. გრამატიკის სხვა წესებს ეწოდება არატერმინარული.

მაგალითი. G_n გრამატიკას აქვს ორი არატერმინარული წესი: $S \to aSa$ და $S \to bSb$ და ერთი $S \to c$ ტერმინარული წესი.

G_0 გრამატიკას არ აქვს არცერთი ტერმინარული წესი.

შესაბამისობა გრამატიკებსა და ენებს შორის

მაშასადამე, ყოველ KC -გრამატიკას შეესაბამება რომელიმე KC -ენა, შეიძლება ცარიელიც. ეს შესაბამისობა არ არის ურთიერთცალსახა: განსხვავებულმა გრამატიკებმა შეიძლება წარმოშვას ერთი და იგივე ენა.

მაგალითი. ვთქვათ გვაქვს G_n გრამატიკა

$$V_A = \{S, A\};\ V_T = \{a, b, c\}$$

$$\left|\begin{array}{ll} S \to aSa & (1) \\ S \to bSb & (2) \\ S \to c & (3) \\ S \to abSba & (4) \\ S \to cA & (5) \\ S \to aA & (6) \end{array}\right|$$

G_n გრამატიკა წარმოქმნის იგივე L_M ენას, როგორსაც წარმოქმნის G_m გრამატიკა. G_n გრამატიკები კი განსხვავებულია G_m-საგან შემდეგი წესებით:

(4)-ე წესის დამატებით, რომელიც უზრუნველყოფს L_M ენის ზოგიერთი ჯაჭვის უფრო სწრაფ შექმნას.

(5)-ე და (6)-ე წესების დამატებით, რომლებიც საერთოდ არ იძლევიან არცერთ ტერმინარულ ჯაჭვს.

მაშასადამე, ვთქვათ მოცემული გვაქვს რაღაც გრამატიკა. ჩვენ შეგვიძლია (მისგან წარმოქმნილი ენის შეუცვლელად) მასზე ჩავატაროთ შემდეგი გარდაქმნები:

“გავწმინდოთ” გრამატიკა იმ წესებისა და სიმბოლოების ამოყრით, რომლებიც არ მიგვიყვანენ ტერმინარულ ჯაჭვამდე.

ამა თუ იმ სახით გადავაწყოთ მისი წესები მისგან წარმოქმნილი ენის შენარჩუნებით.

შემაკავშირებული გრამატიკები. გრამატიკების განხილვისას მოსახერხებელია მას შევუსაბამოთ გრაფი, რომლის წვეროები შეუსაბამებენ არატერმინარული ანბანის $S = A_1, A_2, \dots, A_p$ სიმბოლოებს $a_1, a_2, \dots a_q$ სიმბოლოებს, რომლებიც ფიგურირებენ ტერმინალურ წესებში.

გავატაროთ რკალი A_i -დან A_j-ში (შესაბამისადa_k -ში) თუ გრამატიკაში არის $A_i \to \varphi A_i \Psi$ (შესაბამისად $A_i \to a_k$) სახის წესი.

ცხადია, რომ თუ ეს გრაფი არაა შემაკავშირებელი, მაშინ არც ერთი შემაკავშირებულობის კომპონენტი არ მოახდენს არავითარ გავლენას გრამატიკის მიერ წარმოქმნილ ენაზე. ამიტომ შესაბამისი წესები და სიმბოლოები გრამატიკიდან უნდა ამოვაგდოთ.

მაშასადამე, ყოველთვის შეგვიძლია ვიგულისხმოთ, რომ განსახილველი გრამატიკა იქნება შემაკავშირებელი, ე.ი. ისეთი, რომ მისი შესაბამისი გრაფი იქნება შემაკავშირებელი.

ცარიელი ჯაჭვის შესახებ არავითარი პრინციპული მოსაზრება არ გვიშლის ხელს გვქონდეს A→E სახის ტერმინარული წესი, სადაც E-ცარიელი ჯაჭვია. მაგრამ ტექნიკური თვალსაზრისით უფრო მოსახერხებელია ასეთი წესის ამოგდება. შეიძლება დამტკიცება, რომ ნებისმიერი KC -გრამატიკა, რომელიც წარმოქმნის L ენას და შეიცავს ცარიელ ჯაჭვს, შევცვალოთ მასთან ახლოს მყოფი KC -გრამატიკით, რომელიც წარმოქმნის $L\backslash\{E\}$ ენას.

გამომდინარეობის შეკვეცა. ვთქვათ რაიმე გრამატიკაში გვაქვს $A \Rightarrow \varphi$ გამომდინარეობა, სადაც A არატერმინალური სიმბოლოა, ხოლო φ -ნებისმიერი ჯაჭვი. ცხადია, რომ ჩვენ არ შევცვლით ენას, წარმოქმნილს ამ გრამატიკით, თუ მას დავუმატებთ A→ φ წესს. ვიტყვით, რომ ეს წესი მიღებულია გამომდინარეობის შეკვეცით.

არაცარიელი, სასრული და უსასრულო ენები

არაცარიელი ენები. ენის, რომელიც წარმოქმნილია G სახის KC -გრამატიკით, არაცარიელობის აუცილებელი და საკმარისი პირობა მდგომარეობს S⇒ τ გამომდინარეობის არსებობაში, სადაც τ -ტერმინალური ჯაჭვია.

უსასრულო ენები. რადგან ტერმინალური ანბანი სასრულია, ამიტომ L(G) ენის უსასრულობისათვის აუცილებელი პირობაა წარმოქმნილი ტერმინალური ჯაჭვების სიგრძეთა შემოუსაზღვრელობა. აქედან გამომდინარეობს, რომ წინატერმინალური ჯაჭვებიც უნდა იყოს შემოუსაზღვრელი. ასევე შემოუსაზღვრელი უნდა იყოს არატერმინალური ჯაჭვების სიგრძე.

ეს პირობა ცხადია რომ არაა საკმარისი, რადგანაც შეიძლება არსებობდეს გრამატიკები, რომლებშიც არატერმინალური ჯაჭვების სიგრძე შემოუსაზღვრელია, მაგრამ ისინი წარმოქმნიან ცარიელ ენას (მაგალითად G_0 გრამატიკა).

ენის უსასრულობის პრობლემა შეიძლება ორ ნაწილად გავხლიჩოთ:

ა). ვიპოვოთ აუცილებელი და საკმარისი პირობა გრამატიკებით წარმოქმნილი არატერმინალური ჯაჭვების უსასრულობისათვის.

ბ). ვიპოვოთ დამატებითი პირობა ტერმინალური ჯაჭვების უსასრულობისათვის.

ა) კითხვაზე პასუხი ასეთია: იმისათვის, რომ გრამატიკით წარმოდგენილი არატერმინალური ენა იყოს უსასრულო, აუცილებელი და საკმარისია, რომ რაიმე არატერმინალური A სიმბოლოსათვის არსებობდეს $A \Rightarrow \varphi A\psi, |\varphi| + |\psi| \neq 0$ სახის გამომდინარეობა. ასეთი არატერმინალური სიმბოლოს არსებობის პრობლემა ალგორითმულად გადაწყვეტადია. თუ $|\varphi|*|\psi|\neq 0$, მაშინ იტყვიან, რომ A სიმბოლო თვითჩადგმადია.

რაც შეეხება ბ) კითხვას, მისთვისაც შეიძლება მოიძებნოს კრიტერიუმი, რომელიც ამ პრობლემას გადაწყვეტს (იხ. მაგ. 7). მაშასადამე, ენის სისრულის გამოცნობისა და სიცარიელის გამოცნობის (მაგ. 6) პრობლემა გადაწყვეტადია.

თეორემა. KC - გრამატიკის თვისებები, წარმოქმნას ცარიელი ენა და წარმოქმნას სასრული ენა (შესაბამისად წარმოქმნას არაცარიელი ან უსასრულო ენა), ალგორითმულად გადაწყვეტადია.

ვინაიდან ჩვენ ამოვაგდეთ A→E სახის წესი, გამომდინარეობის ყოველ ბიჯზე KC - გრამატიკაში ჯაჭვის სიგრძე შეიძლება გაიზარდოს ან დარჩეს უცვლელი სიგრძის. იმის საპასუხოდ, ეკუთვნის თუ არა მოცემული ჯაჭვი (არა აუცილებლად ტერმინალური) ენას, წარმოქმნილს მოცემული KC -გრამატიკით , საკმარისია ავაგოთ ამ გრამატიკაში ყველა შესაძლებელი გამომდინარეობა, რომლებიც იწყება საწყისი სიმბოლოთი და მთავრდება ჯაჭვით, რომლის სიგრძეც არ აღემატება განსხვავებული ჯაჭვის სიგრძეს. ეს პროცედურა გამოიყენება ნებისმიერი ჯაჭვისათვის და ყოველთვის იძლევა "კი" ან "არა" პასუხს. საბოლოოდ ვასკვნით, რომ ნებისმიერი KC -ენა წარმოადგენს რეკურსიულ სიმრავლეს.

ენის მაგალითი, რომელიც არაა KC -ენა

შემდეგში ჩვენ დაგვჭირდება ის ფაქტი, რომ $L = \{a^n b^n c^n\}$ ენა არ არის KC -ენა.

დამტკიცება. დავუშვათ, რომ L წარმოქმნილია რაიმე KC -გრამატიკით. ვინაიდან L უსასრულოა, ამიტომ გრამატიკაში არსებობს $A \Rightarrow \varphi A\psi, |\varphi| + |\psi| \neq 0$ გამომდინარეობა. ამიტომ მასში $\forall_n$ -თვის გამომდინარეობს $\alpha\varphi^n A\psi^n b$ სახის ჯაჭვები. საბოლოოდ φ, A და ψ-მ უნდა მოგვცეს ტერმინალური ჯაჭვები. შესაბამისი გამომდინარეობების შეკვეცით ჩვენს გრამატიკას შეგვიძლია დავუმატოთ A→τ სახის წესი.

ადვილი საჩვენებელია, რომ ასეთი სახის როგორი წესიც არ უნდა ავირჩიოთ, მივიღებთ $a^n b^n c^n$ ჯაჭვის გარდა აგრეთვე სხვა ჯაჭვებს, რომლებშიდაც a, b, და c -ს ხარისხები არ იქნებიან ტოლი.

ცხადია, რომ $L = \{a^n b^n c^n\}$ ენა რეკურსიულია. შესაბამისად KC -ენების კლასი საკუთრივი ქვეკლასია რეკურსიული ენების კლასისა.

სავარჯიშოები:

ააგეთ KC -გრამატიკა, რომლებიც წარმოქმნის $L_1 = \{a^n b^n c^b\}$ და $L_2 = \{a^m b^q c^q\}$ ენებს.

ააგეთ KC -გრამატიკა, რომელიც წარმოქმნის $\{x\,\tilde{x}\}$ ენას $\{a, b\}$ ალფავიტზე (შუაში ნიშნულის გარეშე).

ააგეთ KC -გრამატიკა, რომლებიც წარმოქმნის $L_1 = \{a^n b^n a^n \mid n \geq 1\}$ ენას.

დაამტკიცეთ, რომ $L_1 = \{a^p b^q c^r | p \leq q \leq r\}$ $L_2 = \{a^n b^n c b^n a^n \mid n \geq 1\}$ ენები არაა KC -ენები.

12. განვიხილოთ $V = V_A \cup V_T$ ალფავიტი და A→φსახის წესი; $A \epsilon V_A$. აქსიომებად ავიღოთ ყველა ელემენტი ერთმანეთის მიმდევრობით V_Aალფავიტიდან. ამ დროს შეიძლება თუ არა წარმოიქმნას სხვადასხვა ენა? ერთ-ერთი მათგანი ასეთია:

$$\left| \begin{array}{c} A \to AC \\ B \to BC \\ A \to Aa \\ A \to a \\ B \to Bb \\ B \to b \\ C \to AA \\ \end{array} \right|$$

13. ცარიელი ჯაჭვის ელიმინაცია. ვთქვათ G არის KC -გრამატიკა და ვთქვათ V_0 სიმბოლოების სიმრავლეა $A \epsilon V_A$-დან ისეთი, რომ მათთვის გვაქვს A→E სახის წესი. შევადგინოთ V_1 სიმრავლე V_0-ის ყველა ისეთი $A \epsilon V_A$-ს დამატებით,

რომელთათვისაც არსებობს A→φ სახის წესი, სადაც$\varphi\epsilon V_0^*$, ანალოგიურად განვსაზღვროთ V_2 და ა.შ.

განვიხილოთ მიღებული სიმრავლეების მიმდევრობა. ჩამოვაყალიბოთ აუცილებელი და საკმარისი პირობა, რომ $E\in L(G)$. G-დან გამომდინარე ავაგოთ G' KC -გრამატიკა ისეთი, რომ $L(G')=L(G)\backslash\{E\}$. .

შევნიშნოთ, რომ ამისათვის არასაკმარისია უბრალოდ ყველა A→E სახის წესის ამოგდება. მართლაც, განვიხილოთ გრამატიკა

$$G:\begin{vmatrix} S\to aAa \\ A\to E \\ A\to bA \\ A\to c \end{vmatrix}$$

თუ G-დან ამოვაგდებთ A→E წესს, მაშინ G არ წარმოქმნის არც aa სახის, არც ab^na სახის ჯაჭვებს. ცხადია, რომ ამ ჯაჭვების წარმოქმნის წესები შენარჩუნებული უნდა იყოს.

14. არაპროდუქტიული წესების ელიმინაცია. ცარიელი ენა. ვთქვათ G არის KC გრამატიკა. ჩავთვალოთ, რომ G -ში არ არის $A\to E$ სახის წესები. ავღნიშნოთ w_0 -ით V_T ტერმინალური ალფავიტისა და ისეთი A სიმბოლოების გაერთიანება, რომ$A\epsilon V_A$ და გვქონდეს A→τ სახის წესები, სადაც τ -ტერმინალური ჯაჭვია. შევქმნათ w_1 სიმრავლე w_0 -თან ყველა ისეთი $A\,\epsilon\, V_A$ -ს დამატებით, რომელთათვისაც არსებობს $A\to\varphi,\ \varphi\ \epsilon\ W_0^*\backslash E$ სახის წესები.

ანალოგიური მოქცევით წარმოიქმნება w_i სიმრავლეთა მიმდევრობა. w_i სიმრავლის გამოყენებით ჩამოაყალიბეთ აუცილებელი და საკმარისი პირობა იმისათვის, რომ L(G) ენა იყოს ცარიელი.

მიუთითეთ პროცედურაზე, რომელიც საშუალებას მოგვცემს გრამატიკიდან ამოვაგდოთ ყველა "არაპროდუქტიული" წესი.

შევნიშნოთ, რომ

$$G:\begin{vmatrix} S\to Bab \\ B\to Bb \\ B\to b \\ A\to ASA \\ A\to a \end{vmatrix}$$

გრამატიკისათვის w_0 და w_1 სიმრავლეები შესაბამისად არის {A,B} და $\{A, B, S\}$. მაგრამ თუ S არის აქსიომა, მაშინ A სიმბოლო “არაპროდუქტიულია”, რადგანაც S - დან მისი მიღება შეუძლებელია. საქმე სულ სხვაგვარად იქნება, თუ ამ გრამატიკას დავუმატებთ $S \rightarrow \varphi A \psi$ სახის წესს.

წინა სავარჯიშოს გამოყენებით მიუთითეთ პროცედურა, რომელიც საშუალებას მოგვცემს გამოვიცნოთ გრამატიკებით წარმოქმნილი ტერმინალური ჯაჭვების სიმრავლე უსასრულოა თუ არა.

ჩაკეტილობის თვისება. ოპერაციები KC - ენებზე

იტყვიან, რომ სიმრავლე ჩაკეტილია რაიმე ოპერაციის მიმართ, თუ ამ ოპერაციის გამოყენების შედეგი ნებისმიერ ელემენტარულ სიმრავლეზე (უნარული ოპერაციის შემთხვევაში) ან ნებისმიერ ელემენტთა წყვილზე (ბინარული ოპერაციის შემთხვევაში) და ა.შ. შედის ამ სიმრავლეში. ამ თვალსაზრისით განვიხილოთ ზოგიერთი მარტივი ოპერაცია KC -ენებზე.

1. KC -ენების გაერთიანება.

ვთქვათ L_1 და L_2 KC -ენებია, წარმოქმნილი შესაბამისად G_1 და G_2 გრამატიკებით და S_1, S_2 აქსიომებით. ზოგიერთი ან ყველა დამხმარე სიმბოლოების შეცვლით, თუ საჭიროა, შეიძლება მივაღწიოთ იმას, რომ

$$V_{A_1} \cap V_{A_2} = \emptyset.$$

განვიხილოთ გრამატიკა $S \notin V_{A_1} \cup V_{A_2}$ აქსიომით, რომლის წესებიც იქნება G_1 და G_2 გრამატიკების წესები და დამატებით $S \rightarrow S_1$ და $S \rightarrow S_2$ წესები. ცხადია, რომ ესაა KC -გრამატიკა და იგი წარმოშობს $L_1 \cup L_2$ ენას. მაშასადამე, ორი KC -ენების გაერთიანება არის - KC ენა.

2. KC - ენების ნამრავლი.

იგივე აღნიშვნების შენარჩუნებით განვსაზღვროთ $L_1 L_2$ ნამრავლი. როგორც ზემოთ, ამჟამადაც ისე მოვიქცეთ, რომ დამხმარე ალფავიტების თანაკვეთა G_1 და G_2

გრამატიკებისა ცარიელი იყოს. შემდეგ აქსიომის როლში ავიღოთ ახალი სიმბოლო S, გავაერთიანოთ ორივე გრამატიკის წესები და დავუმატოთ ამ გაერთიანებას $S \to S_1S_2$ წესი. ცხადია, რომ ასეთი გზით ჩვენ ავაგეთ KC -გრამატიკა, რომელიც წარმოქმნის L_1L_2 ენას.

მაშასადამე, ორი KC -ენის ნამრავლი არის KC -ენა.

3. KC -ენების იტერაცია

ზემოთა შედეგი საშუალებას გვაძლევს KC -გრამატიკით, რომელიც წარმოქმნის L ენას, ავაგოთ KC -გრამატიკები, რომლებიც წარმოქმნიან $L^2, L^3, \dots$ ენებს.

განვიხილოთ $L^*\backslash\{E\} = L \cup L^2 \cup L^3 \cup \dots \cup L^n \cup \dots$ ენა.

ვთქვათ S არის G გრამატიკის აქსიომა, რომელიც წარმოქმნის L ენას. დავამატოთ G -ს ახალი T სიმბოლო და $T \to ST$, $T \to S$ წესები. გარდა ამისა T გამოვაცხადოთ აქსიომად S -ის ნაცვლად. ახალი წესები საშუალებას იძლევა წარმოიქმნას $S, S^2, \dots S^n \dots$ ჯაჭვები, რომლებიდანაც G გრამატიკის წესებით მიიღება $L, L^2, \dots, L^n \dots$ენები. მაშასადამე, KC -ენების იტერაცია არის - KC ენა (ცარიელ ჯაჭვამდე სიზუსტით).

4. KC -ენების თანაკვეთა.

ა) განვიხილოთ KC -ენები $L_0 = \{a^nca^n\}$ და $L_M = \{xc\tilde{x} | x \epsilon \{a,b\}^*\}$.

ცხადია, რომ $L_0 \cap L_m = L_0$.

ბ) ახლა განვიხილოთ KC-ენები $L_1 = \{a^nb^nc^q\}$ და $L_2 = \{a^mb^qc^q\}$.

გვაქვს $L_1 \cap L_2 = \{a^nb^nc^n\}$. მაგრამ ეს ენა, როგორც ვნახეთ, არ არის KC -ენა.

მოყვანილი მაგალითებიდან ჩანს, რომ ორი KC -ენის თანაკვეთა არ არის აუცილებელი იყოს KC -ენა.

5. KC -ენის დამატება

$V_T{}^*$ ენა არის KC -ენა, რადგანაც იგი შეიძლება მიღებული იქნეს ისეთი გრამატიკისაგან, რომელიც შეიცავს $S \to aS,\ S \to Sa$ და $S \to S$ სახის ყველა შესაძლო წესს.

L ენის $\bar{L}$ დამატების ქვეშ იგულისხმება დამატება $V_T{}^*$ -ის მიმართ. ცნობილია, რომ $\overline{L_1 \cap L_2} = \overline{L_1} \cup \overline{L_2}$.

ამიტომ, თუ დამატების ოპერაცია შეინარჩუნებს იყოს KC -ენა, მაშინ თანაკვეთაც შეინარჩუნებს მას. შესაბამისად KC -ენის დამატება არა აუცილებელი იყოს KC -ენა.

6. KC -ენის შებრუნება

შებრუნების ოპერაცია, გამოყენებული L KC -ენაზე გრამატიკით $G: \{A \to \varphi\}$, შეუსაბამებს მას $\tilde{L}$ ენას გრამატიკით $\tilde{G}: \{A \to \tilde{\varphi}\}$. ცხადია, რომ KC -ენის შებრუნების შედეგი არის KC -ენა.

7. KC -ენის ჩასმა

ვთქვათ გვაქვს $L\ KC$ -ენა, წარმოქმნილი $V_A \cup V_T$ ალფავიტის G გრამატიკით, სადაც $V_T = \{a_i | 1 \leq i \leq n\}$. ვთქვათ გვაქვს n რაოდენობის $L_1, L_2, \dots, L_i, \dots, L_n$ KC –ენა (ამ ენების რაოდენობა იმდენია, რამდენი სიმბოლოცაა V_T-ში). ვთქვათ აგრეთვე, რომ $G, G_1, \dots, G_n$ KC -გრამატიკებია, რომლებიც წარმოქმნიან ამ ენებს. მოვიქცეთ ისე, რომ $G, G_1, \dots, G_n$-გრამატიკების დამხმარე ალფავიტების ყველა წყვილ-წყვილად თანაკვეთა იყოს ცარიელი და a_i გავაიგივოთ G_i გრამატიკის აქსიომასთან. $G, G_1, \dots, G_n$ გრამატიკების გაერთიანება არის KC -გრამატიკა, რომელიც წარმოქმნის ენას, მიღებულს $L, L_1, \dots, L_n$-დან ისეთი ოპერაციით, რომელსაც ეწოდება $L_1, \dots, L_n$ ენების ჩასმა L -ში. მაშასადამე, KC -ენების ჩასმა KC -ენაში, არის KC -ენა.

მიღებული შედეგების რეზიუმე: KC- ენების სიმრავლე ჩაკეტილია გაერთიანების, გამრავლების, იტერაციის, შებრუნების და ჩასმის ოპერაციების მიმართ.

KC -ენების სპეციალური კლასები

KC - გრამატიკებზე დამატებითი შეზღუდვების დადებით შეიძლება განვსაზღვროთ KC -ენების სხვადასხვა კლასები. რადგანაც სხვადასხვა KC -

გრამატიკებმა შეიძლება წარმოქმნან ერთი და იგივე ენა, ამიტომ ენების კლასიფიკაციისას უნდა გამოვიჩინოთ გარკვეული სიფრთხილე.

ჩვენ განვსაზღვრავთ ამა თუ იმ (α) ტიპის გრამატიკას, რომელსაც უნდა ფლობდეს მისი წესები.

ვიტყვით, რომ L ენა ეკუთვნის (α) ტიპს, თუ არსებობს ასეთი ტიპის ერთი მაინც გრამატიკა, რომელიც წარმოქმნის აღნიშნულ ენას.

მაშასადამე, რომ დავამტკიცოთ, რომ L ენა არ ეკუთვნის (α) ტიპს, აუცილებელია ვაჩვენოთ, რომ იგი არ შეიძლება წარმოქმნილი იყოს (α) ტიპის არცერთი გრამატიკით.

1. **ავტომატური ენები**

შემდეგში ჩვენ განვიხილავთ გრამატიკებს, რომელთა ყველა არატერმინალურ წესს აქვთ სახე:

$$A \to aB\ ;\quad A, B \in V_A\ ,\ \ a \in V_T.$$

ასეთი სახის გრამატიკებს ეწოდება ავტომატური გრამატიკები (A- გრამატიკები) , ხოლო მისგან წარმოქმნილ ენას ავტომატური ენა (A- ენა).

2. **წრფივი ენები**

გრამატიკებს, რომელთა ყველა არატერმინალურ წესს აქვთ

$$A \to xBy\ ;\ \ A, B \in V_A\ ;\ \ x, y \in V_T^*$$

სახე, ეწოდებათ წრფივი გრამატიკა.

ენას ეწოდება წრფივი, თუ ის წარმოქმნილია წრფივი გრამატიკით. A- ენები შეადგენენ წრფივი ენების ქვეკლასს. მეორე საინტერესო ქვეკლასს წარმოადგენს მინიმალური წრფივი ენები, რომელთათვისაც V_A შედგება მხოლოდ ერთი S სიმბოლოსაგან, ხოლო ტერმინალური წესების ერთობლიობა მხოლოდ ერთი $S \to c$ წესისაგან, ამავე დროს c სიმბოლო არ უნდა გვხვდებოდეს არცერთ სხვა წესში. L_m ენა (1. მაგალითი 19-ე გვერდზე) წარმოადგენს მინიმალურ წრფივ ენას.

3. **მეტაწრფივი ენები**

KC -გრამატიკას ეწოდება მეტაწრფივი, თუ მისი წესების მარჯვენა მხარე არ შეიცავს აქსიომებს და ყველა წესს, რომელთა მარცხენა მხარეები განსხვავდებიან აქსიომებისაგან, აქვთ ისეთივე სახე, როგორც წრფივი გრამატიკის წესებს.

ენას ეწოდება მეტაწრფივი, თუ არსებობს მისი წარმომქმნელი მეტაწრფივი გრამატიკა.

მაგალითი. განვიხილოთ ენა შედგენილი $V_A = \{S, T\},\ V_T = \{a, b, c\}$ ალფავიტებითა (S -აქსიომაა) და

$$G_{2m}: \begin{vmatrix} S \to TT \\ T \to aTa \\ T \to bTb \\ T \to c \end{vmatrix}$$

გრამატიკით. G_{2m} მეტაწრფივი გრამატიკაა. იგი წარმოქმნის $L_{2m} = \{xc\bar{x}yc\bar{y}\}$ „ორმაგ სარკულ ენას“.

კონტრმაგალითი. $\begin{vmatrix} S \to aSS \\ S \to b \end{vmatrix}$ გრამატიკა არაა მეტაწრფივი. მისგან წარმოქმნილი ენა შედგება ყველანაირი გამოსახულების უსასრულო ჩანაწერის „ჩონჩხისაგან“, რომელიც წარმოქმნილია საგნობრივი სიმბოლოებისა და ბინარული ოპერაციების სიმბოლოებისაგან. შეიძლება იმის ჩვენება, რომ ეს ენა არ წარმოადგენს მეტაწრფივ ენას.

4. მიმდევრობითი გრამატიკები

გრამატიკას ეწოდება მიმდევრობითი, თუ მისი არატერმინარული სიმბოლოები შეიძლება განვალაგოთ $A_1, A_2, \ldots\ldots, A_k$ მიმდევრობით ისეთნაირად, რომ წესი A_1 მარცხენა ნაწილით არ შედის A_1 -საგან განსხვავებული არატერმინალური სიმბოლოების მარჯვენა ნაწილში, წესი A_2 მარცხენა ნაწილით არ შეიცავს A_1 -სა და A_2 -საგან განსხვავებულ არატერმინალურ სიმბოლოებს და ა.შ.

ენას ეწოდება მიმდევრობითი, თუ ის წარმოქმნილია მიმდევრობითი გრამატიკით.

მაგალითები. $L_m = \{xc\bar{x}\}$ ენა არის მიმდევრობითი ენა, როგორც ყველა ენა წარმოქმნილი KC - გრამატიკითა და ერთი არატერმინალური სიმბოლოთი.

L_{2m} ენა აგრეთვეა მიმდევრობითი ენა : G_{2m} გრამატიკაში შეიძლება დავუშვათ $A_1 = T\ ,\ A_2 = S$.

ამის საწინააღმდეგოდ

$$\begin{vmatrix} S \to adTda \\ S \to aSa \\ S \to aca \\ T \to bdSdb \\ T \to bTb \end{vmatrix}$$

გრამატიკა არაა მიმდევრობითი. შესაბამისად მისგან წარმოქმნილი ენაც არაა მიმდევრობითი ენა.

ავტომატები და ენები

არსებობს ავტომატების სპეციალური კლასი, რომლებიც შეესაბამებიან KC - ენების კლასს, ე. წ. ავტომატები მაღაზიური მეხსიერებით (МП -ავტომატი). МП-ავტომატი არის სისტემა , რომელიც შედგება შემდეგი სამი კომპონენტისაგან: 1. შემავალი E ლენტა, რომელიც დაყოფილია უჯრებად და პოტენციურად შემოუსაზღვრელია ორივე მხრიდან. 2. სამუშაო ლენტა, ანუ მაღაზიური მეხსიერების M ლენტა, რომელიც შემოსაზღვრულია ერთი მხრიდან (ჩავთვალოთ მარცხნიდან). იგი ასევე დაყოფილია უჯრებად და პოტენციურად არაა შემოსაზღვრული მეორე მხრიდან. 3. მმართველი A მოწყობილობა, რომელსაც აქვს ორი წამკითხველი. შემავალი წამკითხველი კითხულობს სიმბოლოებს E შემავალ ლენტაზე. სამუშაო წამკითხველს შეუძლია წაიკითხოს სიმბოლოები M სამუშაო ლენტაზე, აგრეთვე მოახდინოს მასზე სიმბოლოების წაშლა და ჩაწერა.

შემავალი ლენტა. МП -ავტომატის მუშაობის დაწყებამდე შემავალ ლენტაზე შეიძლება ჩაიწეროს რაიმე ინფორმაცია-თითოეული სიმბოლო თითოეულ უჯრაში. ამ ინფორმაციის ჩასაწერად გამოიყენება შემავალი ალფავიტი $V_E = \{a_1, a_2, \dots, a_p\}$. ამ ალფავიტს ემატება ერთეულოვანი e სიმბოლო. შედეგად მივიღებთ

←

(E)										
	B B	B	a_{i_1}	a_{i_2}	a_{i_3}	a_{i_4}	a_{i_5}	a_{i_6}	B	B

(M)										
	B σ		A_k			A_j		B	B	B

→

←

e ერთეულოვანი სიმბოლო არ უნდა ავურიოთ ცარიელ სიმბოლოსთან, რომელიც აღინიშნება B-თი. ჯაჭვი, რომელიც შედგება V_E შემავალი ალფავიტისაგან, E

ლენტაზე ჩაიწერება ისე, რომ მარცხნივ და მარჯვნივ რჩება სიცარიელე. ჩაწერა ხორციელდება მარცხნიდან მარჯვნივ.

სამუშაო ლენტა. ჩაწერა სამუშაო ლენტაზე წარმოებს თვით ავტომატის მიერ სამუშაო ალფავიტის საშუალებით

$$V_M = \{A_1, A_2, \ldots, A_q\}.$$

ეს ალფავიტი შეიძლება ჩავრთოთ V_E- შიც. V_M ალფავიტს მიუერთდება ერთეულოვანი e სიმბოლო და კიდევ ერთი სიმბოლო, რომელიც არ გამოიყენება ავტომატის მიერ მისი მუშაობის დროს. მივიღებთ ალფავიტს $\overline{V_M} = \{e\} \cup V_M \cup \{\sigma\}$.

მმართველი მოწყობილობა. მას შეუძლია მიიღოს სასრული რაოდენობის $S_0, S_1, \ldots, S_i, \ldots, S_n$ შიგა მდგომარეობა.

$M\Pi$ ავტომატის ფუნქციონირება

MП -ავტომატის ფუნქციონირების აღსაწერად დაგვჭირდება შემდეგი ცნებები:

1. $\overline{V_M}^*$- ¯-ზე განსაზღვრული $\lambda(q)$ ფუნქცია ისეთი, რომ

$$\lambda(\sigma) = -1$$
$$\lambda(e) = 0$$
$$\lambda(\varphi) = |\varphi|, \quad \text{თუ} \quad \varphi \epsilon V_M^*.$$

2. ავტომატის ფიზიკური მდგომარეობა არის

$$\{a_i, S_i, A_K\}$$

სამეული, სადაც a_iარის E ლენტაზე ჩაწერილი სიმბოლო, S_i -ავტომატის მდგომარეობა; A_K - სიმბოლო, რომელიც ჩაწერილია M-ზე.

ავტომატის სიტუაცია - ეს არის ან ავტომატის ფიზიკური სიტუაცია, ან სამეული, მიღებული რაიმე ფიზიკური სიტუაციიდან a_i ან A_K (ან ორივე ერთად) სიმბოლოების შეცვლით e სიმბოლოთი. სწორედ აქ მჟღავნდება MП ავტომატის არადეტერმინირებული ხასიათი: რაიმე მოცემულ სიტუაციაში ავტომატმა შეიძლება “დახუჭოს” ის “თვალი”, რომლითაც იგი აღიქვამდა a_i-ს, ან ის “თვალი”, რომლითაც აღიქვამდა A_K -ს. მაშინ იგი ამ “თვალით” “ხედავს” ერთეულოვან სიმბოლოს.

თუ ავტომატი იმყოფება $\{a_i, S_i, A_K\}$ ფიზიკურ სიტუაციაში, მაშინ იგი ერთდროულად იმყოფება შემდეგ ოთხ სიტუაციაში:

$$\Sigma_1 = (a_i, S_i, A_K),$$
$$\Sigma_2 = (e, S_i, A_K),$$
$$\Sigma_3 = (a_i, S_i, e),$$
$$\Sigma_4 = (e, S_i, e).$$

ბრძანება არის $\Sigma \to (S_m, x)$ სახის გამოსახულება, სადაც Σ სიტუაციაა, S_m-- მდგომარეობა, ხოლო x აღნიშნავს ან e -ს, ან σ -ს, ან ჯაჭვს V_M ალფავიტში.

5. აბსტრაქტული МП-ავტომატი ეწოდება სასრული რაოდენობის $\Sigma_e \to (S_{m_e}, x_e)$ სახის ბრძანებებს. რადგანაც განსახილველი ავტომატები არ არიან დეტერმინირებული ავტომატები, შეიძლება ბრძანებები იყოს თანხვედრილი მარცხენა ნაწილებით (ე.ი. ერთნაირი სიტუაციებით). ახლა უკვე შეგვიძლია დავაზუსტოთ, თუ რა იგულისხმება გამოთვლის პროცესის ქვეშ.

გამოთვლის პროცესი. გამოთვლა МП-ავტომატში მოიცემა V_E ალფავიტში ჯაჭვით, რომელიც ჩაწერილია შემავალ ლენტაზე და ორივე მხრიდან არის შემოსაზღვრული ჰარით (ე.წ. „პრობელებით"). გამოთვლის დაწყების წინ მმართველი მოწყობილობა იმყოფება საწყის S_0 მდგომარეობაში, ხოლო შემავალი წამკითხველი ხედავს შემავალი ჯაჭვის ყველაზე მარცხენა სიმბოლოს. ამ დროს სამუშაო ლენტის ყველაზე მარცხენა უჯრაში უნდა იყოს ჩაწერილი σ სიმბოლო და ამ უჯრას უნდა უყურებდეს სამუშაო წამკითხველი.

მაშასადამე, გამოთვლა იწყება (a_i, S_0, σ) ფიზიკური სიტუაციით და მიმდინარეობს შემდეგნაირად.

დავუშვათ, რომ МП-ავტომატი გამოთვლის რომელიმე ბიჯზე მოხვდა Σ სიტუაციაში. თუ ავტომატის პროგრამაში არაა ბრძანებები Σ -მიმართ, მაშინ ავტომატი გაჩერდება. თუ ასეთი ბრძანება ან ბრძანებები არსებობს, მაშინ ავტომატი ამ ბრძანებებიდან ირჩევს ერთ-ერთს. ვთქვათ არჩეულია $\Sigma \to (S_m, x)$ ბრძანება.

ამ ბრძანებასთან დაკავშირებით შემდეგი რამ ხდება. ა). მმართველი მოწყობილობა გადადის S_m მდგომარეობაში. ბ) თუ მოცემული სიტუაციის პირველი სიმბოლო არის e , მაშინ შემავალი წამკითხველი E ლენტას უძრავს დატოვებს, ხოლო თუ პირველი სიმბოლოა a_i მაშინ წამკითხველი გადააადგილებს ლენტას ისე, რომ დაინახოს a_i-ს მარჯვნივ მყოფი უახლოესი სიმბოლო. გ) x -ის მნიშვნელობისაგან

დამოუკიდებლად სამუშაო წამკითხველი ლენტასთან შემდეგნაირად მოქმედებს: თუ $\lambda(\sigma) = -1$,, მაშინ გადავწიოთ ლენტა უფრო მარჯვნივ ერთი უჯრით, რომ წამკითხველი დავაყენოთ იმ სიმბოლოს პირდაპირ , რომელიც წინამდებარე სიმბოლოს მარცხნივაა, წაიშალოს წინამდებარე სიმბოლო; თუ $\lambda(e) = 0$, მაშინ ლენტა არ გადავაადგილოთ, არაფერი არ ჩავწეროთ; თუ $\lambda(\varphi) = m$ ჩავწეროთ $x = \varphi$ მიმდევრობა უკანასკნელი სიმბოლოს მარჯვნივ. გადავწიოთ ლენტა m უჯრით მარცხნივ ისე, რომ წამკითხველი ხედავდეს ჩაწერილი სიმბოლოებიდან ყველაზე მარჯვენას.

ენათა კლასი, დასაშვები MП-ავტომატში და წარმოქმნილი MП-ავტომატის მიერ

ვაჩვენოთ, რომ ნებისმიერ KC გრამატიკაში შეიძლება ავაგოთ MП-ავტომატი, რომლისთვისაც L(G) ენა დასაშვებია ან წარმოქმნილია მის მიერ.

განვიხილოთ G KC -გრამატიკა:

$$V_A = \{A_i | 1 \leq i \leq m\};\ \ V_T = \{a_j | 1 \leq j \leq n\};$$

$$S = A_1 \text{ აქსიომა}$$

$$G: \left| \begin{matrix} A_i \to w_{i,1}, \dots, A_i \to w_{i,k_i} \\ 1 \leq i \leq m \end{matrix} \right|$$

ამ გრამატიკით ავაგოთ MП-ავტომატი. მისი შესასვლელი $\overline{V_E}$ ალფავიტი იქნება V_T- ს დამატებული e სიმბოლო. $\overline{V_M}$ ალფავიტი იქნება $V_T \cup V_A$ გაერთიანება დამატებული σ და e სიმბოლოები. მისი მდგომარეობები იქნება $[0], [1], [A], \dots [A_m]$ სიმბოლოები. $[0]$ მდგომარეობა იქნება საწყისი. $[1]$ მდგომარეობას ვუწოდოთ "წინასინტაქსური", ხოლო $[A_i]$მდგომარეობებს –"სინტაქსური".

ავტომატის ბრძანებები შეიცავს:

1. საწყის ბრძანებას $(e, [0], \sigma) \to ([1], S)$

2. სინტაქსურ მდგომარეობაში გადასვლის ბრძანებებს

$$(e, [1], A_i) \to ([A_i], \sigma)$$

ამ ბრძანებების რაოდენობა ემთხვევა სინტაქსურ მდგომარეობათა რაოდენობას (ე.ი. G გრამატიკის დამხმარე სიმბოლოების რაოდენობას).

3. წარმოქმნის (ან დაშვების) ბრძანებები

$$(e, [A_i], e) \to ([1], w_{i,k_i}),$$

რომელთა რიცხვი ტოლია საწყისი გრამატიკების წესების რაოდენობისა.

4. სიმბოლოების წაკითხვის ან კოპირების ბრძანებები

$$\left(a_j, [1], a_j \right) \to ([1], \sigma),$$

რომელთა რიცხვი ტერმინალური სიმბოლოების ტოლია.

5. დამამთავრებელი ბრძანება

$$(e, [1], \sigma) \to ([0], \sigma).$$

ავტომატის ფუნქციონირება ენის წარმოქმნისას

ვთქვათ ჩვენს ავტომატს მიცემული აქვს სწორი საწყისი პირობები. მაშინ იგი გადავა [1] წინასინტაქსურ მდგომარეობაში, ჩაიწერს სამუშაო ლენტაზე აქსიომას და დაიმახსოვრებს მას. ე.ი. გადადის შესაბამის სინტაქსურ მდგომარეობაში და ამავდროულად წაშლის აქსიომას.

$[A_i]$ სინტაქსურ მდგომარეობაში ყოფნისას ავტომატი ირჩევს რაიმე $A \to w_{1,k_1}$ წესს, ჩაწერს მის მარჯვენა მხარეს სამუშაო ლენტაზე, დააყენებს წამკითხველს ისე, რომ იგი ხედავდეს მის მიერ ჩაწერილ ბოლო სიმბოლოს და ბრუნდება [1] წინასინტაქსურ მდგომარეობაში.

საზოგადოდ, თუ ავტომატი სამუშაო M ლენტაზე კითხულობს რაიმე წესის მარცხენა ნაწილს, მაშინ იგი მას წაშლის, გადადის შესაბამის სინტაქსურ მდგომარეობაში, ჩაიწერს M-ზე იმავე წესის მარჯვენა ნაწილს და, დაბრუნდება რა წინასინტაქსურ მდგომარეობაში, დაინახავს მის მიერ M-ზე ჩაწერილ უკანასკნელ სიმბოლოს.

თუ ავტომატი M ლენტაზე კითხულობს ტერმინალურ სიმბოლოს, მაშინ იგი მას წაშლის, წინასწარ მოახდენს რა მის კოპირებას E ლენტაზე.

პროცესი დამთავრდება, როდესაც ავტომატი მომდევნო კოპირების შემდეგ დაინახავს M ლენტაზე ყველაზე მარცხენა σ სიმბოლოს. ცხადია, რომ აღწერილი პროცესი ეკვივალენტურია KC -გრამატიკაში გამომდინარეობების აგებისა მიღებული ტერმინალური სიმბოლოების კოპირებით (ცხადია, შებრუნებული მიმდევრობით).

ავტომატის ფუნქციონირება ენის დაშვებისას

ავტომატის ფუნქციონირება ამ შემთხვევაში გარანტირებულია MП -ავტომატის "შებრუნებადობით".

მაგალითი. ავაგოთ ზემოთაღწერილი წესის მიხედვით L_m ენისათვის MП-ავტომატი.

ვთქვათ გვაქვს KC -გრამატიკა $V_A = \{S\}$, $V_T = \{a, b, c\}$ ალფავიტებით და $S \to aSa$, $S \to bSb$, $S \to c$ წესებით.

რადგან ამ გრამატიკაში არის მხოლოდ ერთი დამხმარე სიმბოლო - S აქსიომა, ამიტომ ჩვენს ავტომატს აქვს მხოოლოდ ერთი წინასინტაქსური მდგომარეობა. მოცემული გრამატიკის შესაბამისი ავტომატის პროგრამა მდგომარეობს შემდეგში:

1: $(e, [0], \sigma) \to ([1], S)$

2: $(e, [1], S) \to ([S], \sigma)$

3: $(e, [S], e) \to \begin{cases} ([1], aSa) \\ ([1], bSb) \\ ([1], c) \end{cases}$

4: $\left.\begin{matrix} (a, [1], a) \\ (b, [1], b) \\ (c, [1], c) \end{matrix}\right\} \to ([1], \sigma)$

5: $(e, [1], \sigma) \to ([0], \sigma)$

გრამატიკები, რომლებიც წარმოქმნიან МП-ავტომატისათვის დასაშვებ ენებს

KC -გრამატიკების აგება, რომლებიც წარმოქმნიან МП-ავტომატში დასაშვებ ენას, ასევეა შესაძლებელი, ოღონდ მისი შესრულება უფრო რთული პროცესებითაა შესაძლებელი. კერძოდ, ამისათვის საჭიროა ავტომატის ბრძანებების წინასწარი “ნორმალიზაცია”.

მაშასადამე, გვაქვს შემდეგი თეორემა.

თეორემა. МП-ავტომატის მიერ წარმოქმნილი და მასში დასაშვები ენების კლასი ემთხვევა KC -ენების კლასს.

ავტომატური გრამატიკები

ავტომატური ენები წარმოადგენენ კერძო შემთხვევას KC -ენებისა, რომელთაც აგრეთვე უწოდებენ კლინის ენას, ან რეგულარულ ენას ან კიდევ ენას მდგომარეობათა სასრული სიმრავლით.

ავტომატურ გრამატიკას, ანუ A-გრამატიკას აქვს: V=$V_A\cup V_T$ ალფავიტი, სადაც V_A-დამხმარე, ხოლო V_T-ძირითადი ალფავიტებია. ამავე დროს

$V_A\cap V_T=\emptyset$, $V_A=\{A_i|1\le i\le n\}$ და A_1 აქსიომაა, $V_T=\{a_j|0\le j\le m\}$, მარჯვენა და მარცხენა სახის წესები:

$$A_i\to a_jA_k\text{- (მარჯვენა)},$$

$$A_i\to A_ka_j\text{ (მარცხენა)},$$

(თითოეულ გრამატიკაში მხოლოდ მარჯვენა ან მხოლოდ მარცხენა წესია), აგრეთვე

$$A_i\to a_j,$$

$$A_i\to E$$

(ანბანური ანუ ტერმინალური წესები).

ისარი X→Y-ში აღნიშნავს “ ჩავსვათ Y X -ის ნაცვლად”.

მაგალითი. $V=\{P,Q\}\cup\{a,b\};$ P-აქსიომა

$$< G_{\text{მარჯ}} >: \begin{vmatrix} P \to aP \\ P \to aQ \\ Q \to bQ \\ Q \to b \end{vmatrix}.$$

ეს გრამატიკა მოგვცემს ენას, შედგენილს ყველა შესაძლო ჯაჭვისაგან შემდეგნაირად: მიმდევრობა a -დან, ხოლო მის მიყოლებით მიმდევრობა b -დან, ე.ი.

$$\{a^m b^n | m, n > 0\}$$

იგივე ენა შეიძლება აღიწეროს მეორე გრამატიკით

$$< G_{\text{მარცხ}} >: \begin{vmatrix} P \to Pb \\ P \to Qb \\ Q \to Qa \\ Q \to a \end{vmatrix}.$$

მაგალითის სახით მოვიყვანოთ a^3b^2ჯაჭვის გამოყვანა $< G_{\text{მარჯ}} >$ გრამატიკაში:

$$P \to ap \to aap \to aaaQ \to aaabQ \to aaabb$$

შენიშვნა. ნებისმიერი სასრული ენა წარმოადგენს A-ენას. სიმრავლე, რომელსაც $\{f_1, f_2, \dots, f_n\}, f_i = a_{i,1} \dots a_{i,k_i}$ ჯაჭვი შეიცავს შეიძლება აღწერილი იქნეს შემდეგი A -გრამატიკით:

$$G = \{P \to a_{i,1}A_{i,1};\ A_{i,1} \to a_{i,2}A_{i,2}; \dots;\ A_{i,k_{i-2}} \to a_{i,k_{i-1}};\ A_{i,k_{i-1}} \to a_{i,k_i} | 1 \le i \le n\}$$

ნებისმიერი G გრამატიკისათვის მოიძებნება ისეთი ავტომატური G^A გრამატიკა, რომელიც იგივე ენას წარმოქმნის (ცარიელი e სიტყვის სიზუსტემდე), რასაც $G: L(G^A) = L(G)\backslash\{e\}$.

განვიხილოთ ისეთი ავტომატები, რომლებსაც აქვთ შიგა მდგომარეობათა სასრული რაოდენობა. ავტომატის მიერ "ელემენტარული მოქმედების" შესრულება შეიძლება სხვადასხვა ხასიათის იყოს სხვადასხვა ავტომატისათვის. სასრულ ავტომატებს გააჩნიათ მდგომარეობათა სასრული რაოდენობა და შემავალი სიმბოლოების სასრული რაოდენობა. როგორც კი შემავალი სიმბოლო შევა სასრულ ავტომატში, ავტომატი ერთი მდგომარეობიდან გადავა მეორეში და ეს გადასვლა დეტერმინირებული იქნება. გარდა ამისა მდგომარეობათა სიმრავლეში გამოიყოფა საწყისი მდგომარეობა და დამამთავრებელ მდგომარეობათა სიმრავლე. ჯაჭვი დასაშვებია სასრული ავტომატისათვის, თუ საწყის მდგომარეობაში მუშაობის დაწყებისას და ამ ჯაჭვის ბოლომდე წაკითხვის შემდეგ ავტომატი გადადის ერთ-ერთ დამამთავრებელ მდგომარეობაში.

მაშასადამე, თითოეული სასრული ავტომატი მუშაობს როგორც მოწყობილობა, რომელიც გამოიცნობს რაიმე ჯაჭვების სიმრავლეს (იგი დაუშვებს მხოლოდ ისეთ ჯაჭვებს, რომლებიც ამ სიმრავლეს ეკუთვნიან).

განმარტება. სასრული ავტომატი ეწოდება მოწესრიგებულ $(K, \Sigma, \delta, P_0, F)$ ხუთეულს, სადაც:

K -მდგომარეობათა არაცარიელი სიმრავლეა.

Σ -შემავალი სიმბოლოების სიმრავლე (ალფავიტი).

δ – K*Σ სიმრავლის K სიმრავლეში ასახვა (გადასვლის ფუნქცია).

P_0 - საწყისი მდგომარეობა (P_0 ∊K).

F - დამამთავრებელ მდგომარეობათა სიმრავლე (F⊂K).

როგორც ზემოთ ავღნიშნეთ, ავტომატების მიერ წარმოდგენილ ენას ეწოდება აგრეთვე **რეგულარული ენა** , ან უბრალოდ, **ხდომილება**.

ფორმალური გრამატიკების კლასიფიკაცია

განვიხილოთ $V_T = \{a_1, a_2, \dots, a_n\}$ - ტერმინარული ანუ ძირითადი სასრული ალფავიტი და $V_A = \{b_1, b_2, \dots, b_m\}$ - დამხმარე სასრული ალფავიტი. $D = V_T \cup V_A$ სიმრავლიდან შევქმნათ სასრული სიგრძის ყველა შესაძლებელი ჯაჭვი. ასეთი ჯაჭვების G სიმრავლე უსასრულოა (თვლადი). განვიხილოთ $\varphi \to \xi$ სახის წესი, სადაც φ და ξ ნებისმიერი ჯაჭვებია G- დან. ვთქვათ Q არის $\varphi \to \xi$ სახის წესით ტერმინალური სიმბოლოებისაგან შედგენილი სიმრავლე. მაშინ Q არის ენა, წარმოქმნილი ფორმალური G გრამატიკით.

შემოვიღოთ ფორმალური გრამატიკების შემდეგი კლასიფიკაცია.

1. ალგორითმული გრამატიკები, რომლებიც წარმოქმნიან ნულოვანი ტიპის Q_0 ენას;
2. კონტექსტური გრამატიკები, რომლებიც წარმოქმნიან პირველი ტიპის Q_1 ენას;
3. უკონტექსტო გრამატიკები, რომლებიც წარმოქმნიან მეორე ტიპის Q_2 ენას;
4. ავტომატური გრამატიკები, რომლებიც წარმოქმნიან მესამე ტიპის Q_3 ენას.

თუ $\varphi \to \xi$ სახის წესებზე არავითარი შეზღუდვა არაა დაწესებული (გარდა იმისა, რომ φ არ შეიძლება შედგებოდეს მხოლოდ ძირითადი სიმბოლოებისაგან), მაშინ G გრამატიკას ეწოდება ალგორითმული გრამატიკა. თუ ამის გარდა მოითხოვება, რომ

ξ- ს სიგრძე (სიმბოლოების რაოდენობა ჯაჭვში) არ იყოს ნაკლები φ-ს სიგრძეზე, მაშინ G გრამატიკას ეწოდება კონტექსტური გრამატიკა.

თუ ამის გარდა წესებს აქვთ $b_j \to \xi$ სახე (ე.ი. φ შედგება ერთი დამხმარე სიმბოლოსაგან), მაშინ G გრამატიკას ეწოდება უკონტექსტო გრამატიკა. და ბოლოს, თუ ზემოთ ჩამოთვლილ შეზღუდვათა გარდა ξ-ს აქვს a_l ან $a_l b_k$ სახე (ე.ი. წესებს აქვთ $b_j \to a_l$ ან $b_j \to a_l b_k$ სახე), მაშინ G გრამატიკას ეწოდება ავტომატური გრამატიკა.

თუ $\{Q_i\}$ -ით ავღნიშნავთ i-ური ტიპის (i=0,1,2,3) ენათა სიმრავლეს, მაშინ, როგორც გრამატიკების განმარტებებიდან გამომდინარეობს, $\{Q_0\} \supseteq \{Q_1\} \supseteq \{Q_2\} \supseteq \{Q_3\}$. შეიძლება იმის ჩვენება, რომ სინამდვილეში ადგილი აქვს $\{Q_0\} \supset \{Q_1\} \supset \{Q_2\} \supset \{Q_3\}$ მკაცრ ჩართვას.

ზოგადი ცნებები სასრული ავტომატების შესახებ

სასრული ავტომატების თეორიას საქმე აქვს ისეთ მათემატიკურ მოდელებთან, რომლებიც გამოიყენებიან ფიზიკური ან აბსტრაქტული მოვლენების მიახლოებითი ასახვისათვის. სასრული ავტომატების თეორიის იდეა და ტექნიკა გამოიყენება ისეთი პრობლემების გადასაწყვეტად, როგორიცაა, მაგალითად, ადამიანის ნერვული სისტემის ფუნქციონირების გამოკვლევა, ინგლისური ენის სინტაქსის ანალიზი, ელექრონული გამომთვლელი მანქანების პროექტირება და სხვა.

უამრავი პრობლემა, რომელიც გვხვდება მეცნიერებასა და ტექნიკაში, შეიძლება დავყოთ ორ კატეგორიად: ანალიზის ამოცანები და სინთეზის ამოცანები.

ანალიზის ამოცანა მდგომარეობს მოცემული განსაზღვრული სისტემის ქცევის პროგნოზირებაში, ხოლო სინთეზის ამოცანა - მოცემული ქცევის მოხედვით სისტემის აგებაში.

ავტომატების ზოგადი თეორია იყოფა ორ ნაწილად: - ავტომატების აბსტრაქტული თეორია და ავტომატების სტრუქტურული თეორია.

ავტომატების აბსტრაქტულ თეორიას არ აინტერესებს არც ავტომატის სტრუქტურა და არც მასში შემავალი და გამომავალი სიგნალები. აბსტრაქტული სტრუქტურა სწავლობს მხოლოდ იმ გადასვლებს, რომლებსაც ის განიცდის შემავალი სიგნალების ზემოქმედებისას და იმ გამომავალ სიგნალებს, რომლებსაც იგი ამ დროს გამოსცემს.

ავტომატების აბსტრაქტული თეორიისაგან განსხვავებით, ავტომატების სტრუქტურული თეორია, უპირველეს ყოვლისა, შეისწავლის როგორც თვით ავტომატს, ასევე მასში შემავალი და გამომავალი სიგნალების სტრუქტურას.

დინამიური სისტემა

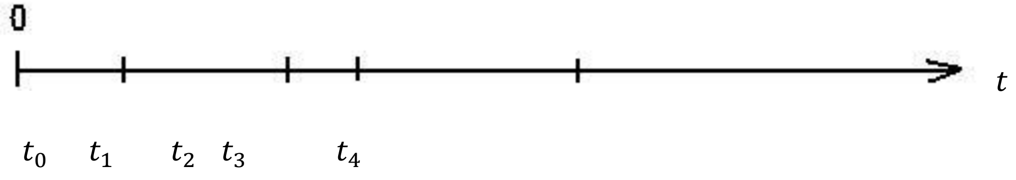
ვთქვათ $\{x\}_i$ $(i = 1,2, \dots, n)$ და $\{y\}$ არის ალფავიტები, რომლებიც შეიცავენ სასრული რაოდენობის რაღაც სიმბოლოებს. მაშინ

$$y = f(x_1, x_2, \dots, x_n) \qquad (1)$$

ფუნქციური დამოკიდებულება $\{x\}_1, \{x\}_2, \dots, \{x\}_n$ალფავიტებიდან ერთ-ერთს შეუსაბამებს სიმბოლოს $\{y\}$ ალფავიტიდან.

წარმოვიდგინოთ იდეალური მოწყობილება, რომელიც რეალიზაციას უკეთებს (1) ფუნქციურ დამოკიდებულებას. ამ მოწყობილობას აქვს n შესასვლელი და ერთი გამოსასვლელი. $x_1, x_2, \dots, x_n$ შესასვლელებთან ერთდროულად გამოჩნდებიან სიმბოლოები $x_1, x_2, \dots, x_n$ალფავიტებიდან. იმავე წამს გამოსასვლელზე გამოჩნდება სიმბოლო $\{y\}$ ალფავიტიდან (1) ტოლობის შესაბამისად. ასეთ მყისიერად მოქმედ იდეალურ მოწყობილობას ვუწოდოთ ფუნქციონალური გარდამქმნელი. კერძო შემთხვევაში, როცა თითოეული $\{x\}_1, \{x\}_2, \dots, \{x\}_n$ და $\{ y \}$ალფავიტებიდან შეიცავს მხოლოდ ორ სიმბოლოს, ე.ი. როცა $x_1, x_2, \dots, x_n$ და y ლოგიკური ცვლადებია, ხოლო f ლოგიკური ფუნქცია, მოწყობილობას ეწოდება ლოგიკური გარდამქმნელი.

ზემოთ აღნიშნული ფუნქციონალური დამოკიდებულებისას ჩვენ არ განგვიხილავს დრო. ამიტომაც ჩავთვალოთ, რომ ფუნქციონალური გარდამქმნელი მოქმედებს მყისიერად. ახლა კი შემოვიტანოთ განსახილველად დრო. ზოგადად, დროის შესახებ საუბრისას ვიგულისხმოთ, რომ იგი იცვლება მხოლოდ ერთი მიმართულებით (მომავლისაკენ) და უწყვეტად გაირბენს ყველა შესაძლო მნიშვნელობას რიცხვით ნახევარღერძზე. ასეთი დროისაგან განსხვავებით დისკრეტული მოქმედების მოწყობილობის შესწავლისას ხელსაყრელია შემოვიღოთ დისკრეტული დრო. წარმოვიდგინოთ, რომ დროის უწყვეტი რიცხვითი ნახევარღერძი რაღაცნაირად დაყოფილია უსასრულო როდენობის სასრულ ინტერვალებად, რომლებიც თავის მხრივ არ არის აუცილებელი ერთმანეთის ტოლი იყოს (ნახ.1)

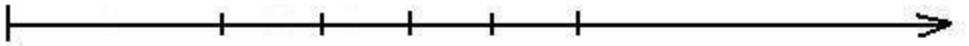


ნახ.1

$t = 0 -$ დან $t = \infty$- კენ მოძრაობისას ინტერვალების გამყოფი წერტილები ავღნიშნოთ t_0, t_1, t_2, ... ეს წერტილები ადგენენ თვლად სიმრავლეს. $t_0, t_1, t_2, \ldots$ დროის მომენტები ავღნიშნოთ 0,1,2…რიცხვებით და ვუწოდოდ ციკლები, ხოლო 0,1,2… რიცხვები განვიხილოთ როგორც სიმბოლოები, რომლებიც შეადგენენ $\{t\}$ ალფავიტს.

მიმდინარე დრო, რომელიც შეესაბამება აწმყო (მიმდინარე) მომენტს, ავღნიშნოთ t-თი. მაშინ $(t-1, t-2, \ldots)$ და $(t+1, t+2, \ldots)$ იქნებიან წარსული (წინამდებარე) და მომავალი (შემდეგი) ციკლები (ნახ.2).

(1) ტოლობაზე საუბრისას ვგულისხმობთ, რომ ცვლადები ფიქსირებულები არიან. ახლა დავუშვათ, რომ სიმბოლოები, რომლებიც შეესაბამებიან $x_1, x_2, \ldots, x_n$ ცვლადებს, იცვლებიან დისკრეტულ დროში. სხვაგვარად, რომ ვთქვათ, ყოველ t =0,1,2,… ციკლში მოცემულია ცვლადების განსაზღვრული მნიშვნელობა და ამით განსაზღვრულია ფუნქციები, მოცემული t = 0,1,2,… რიცხვთა მიმდევრობაზე და რომლებიც იღებენ მნიშვნელობებს $\{x\}_i$ $(i = 1,2, \ldots, n)$ ალფავიტიდან.



$t-2$ $t-1$ t $t+1$ $t+2$

ნახ. 2

1)-ის თანახმად მაშინ მათ შეესაბამებათ ფუნქცია:

$$y(t) = f[x_1(t), x_2(t), \ldots, x_n(t)], \qquad (2)$$

რომელიც მოცემულია ნატურალურ რიცხვთა მიმდევრობაზე და ღებულობს მნიშვნელობას $\{y\}$ ალფავიტიდან. ასეთი გვარის სისტემა მუშაობს დისკრეტულ დროში, მაგრამ არ აქვს მეხსიერება იმ აზრით, რომ ნებისმიერ t მომენტში y ”გამოსავლის” მნიშვნელობა დამოკიდებულია იმავე t მომენტში x_i “შესასვლელის” მნიშვნელობაზე.

შეიძლება წარმოვიდგინოთ ისეთი სისტემები, რომლებიც აგრეთვე მუშაობენ დისკრეტულ დროში და რომელთა კოორდინატებიც (შესასვლელი და გამოსასვლელი) ღებულობენ სიმბოლოების მნიშვნელობებს სასრული ალფავიტებიდან, მაგრამ უფრო რთული დამოკიდებულებაა შემავალი და გამომავალი კოორდინატების მნიშვნელობებს შორის. მაგალითად, ვთქვათ:

$$y(t) = f[x_1(t-1), x_2(t-1), \dots, x_n(t-1), x_1(t), x_2(t), \dots, x_n(t)]\,. \qquad (3)$$

სხვაგვარად რომ ვთქვთ, ნებისმიერ t ციკლში y სიმბოლო იმავე t ციკლში არამარტო ყველა x_i სიმბოლოზე და მნიშვნელობაზეა დამოკიდებული, არამედ დამოკიდებულია აგრეთვე ამავე სიმბოლოების მნიშვნელობაზე წინამდებარე $t-1$ ტაქტში.

დინამიური სისტემის ტერმინის ქვეშ განიხილება ისეთი სისტემები ტექნიკაში, ბუნებაში, ცხოვრებაში და სხვა, რომლებშიდაც პროცესი ვითარდება დროში. დინამიური სისტემის მდგომარეობას ყოველ მომენტში ახასიათებენ რაღაც განზოგადოებული (სასრული ან უსასრულო) კოორდინატების რიცხვით.

პროცესები დინამიურ სისტემაში ხასიათდება განზოგადოებული კოორდინატების ცვლილებით დროში და აღიწერება სხვადასხვა ტიპის განტოლებებით.

დინამიური სისტემები შეიძლება დავყოთ რამოდენიმე კლასად შემდეგი ფაქტორების გათვალისწინებით:

ითვლება თუ არა მიმდინარე დრო უწყვეტად ან დისკრეტულად.

აქვს თუ არა სისტემას განზოგადებული კოორდინატების სასრული ან უსასრულო სიმრავლე.

განზოგადოებული კოორდინატებიდან თითოეულის შესაძლებელი მნიშვნელობათა სიმრავლის სიმძლავრე, ე.ი. არის თუ არა ეს სიმრავლე სასრული, თვლადი.

დინამიური სისტემების განსაკუთრებულ კლასს შეადგენს სისტემები, რომელთათვისაც დრო დისკრეტულია, ე.ი. იცვლება თვლად სიმრავლეზე, ხოლო განზოგადებული კოორდინატები (რომელთა რიცხვი შეიძლება იყოს სასრულიც და უსასრულოც) იცვლებიან სასრულ სიმრავლეზე.

ყოველგვარი დინამიური სისტემა შეიძლება განიცდიდეს გარეშე ზემოქმედებას. ეს ზემოქმედებებიც შეიძლება მოცული იყოს კონტინუუმზე, თვლად სიმრავლეზე ან სასრულ სიმრავლეზე. დინამიურ სისტემებში, რომლებიც შეიძლება აღწერილი იქნენ დიფერენციალური ან სხვაობიანი განტოლებებით, ზოგადად განიხილავენ გარე ზემოქმედებათა სასრულ რაოდენობას, რომელთაც თავის მხრივ შეუძლიათ მიიღონ ნებისმიერი მნიშვნელობა სასრული კონტინუუმიდან.

დინამიური სისტემების განხილვის დროს, რომელთა განზოგადოებული კოორდინატები მოიცემა სასრულ სიმრავლეზე, ბუნებრივია ჩავთვალოთ, რომ გარე ზემოქმედებათა რიცხვი სასრულია და თითოეული მათგანი ასევე მოცემულია სასრულ სიმრავლეზე.

ისეთ დინამიურ სისტემებს, რომელთათვისაც დრო მოიცემა სასრულ სიმრავლეებზე და გარე ზემოქმედებებისა და კოორდინატების რიცხვი სასრულია, ვუწოდოთ სასრული დინამიური სისტემები. ასეთ სასრული დინამიური სისტემების კლასს ეკუთვნის სასრული ავტომატი.

სასრული ავტომატები

ვთქვათ$x = \{x_1, x_2, \dots, x_m\}$ და $y = \{y_1, y_2, \dots, y_n\}$ არის ელემენტთა ნებისმიერი სიმრავლე, რომელთაც ვუწოდოთ ალფავიტი, ხოლო მის ელემენტებს - ალფავიტის ასოები. ასოების სასრულ მოწესრიგებულ მომდევრობას ვუწოდოთ სიტყვა მოცემულ ალფავიტში. ავღნიშნოთ $\sigma(x)$ და $\sigma(y)$-ით შესაბამისად x და y ალფავიტებში სიტყვათა სიმრავლე. მაშინ დისკრეტული ინფორმაციის ნებისმიერი გარდაქმნა შეიძლება მოცემული იქნეს როგორც ცალსახა f ასახვა $\sigma(x)$ სიტყვათა სიმრავლისა $\sigma(y)$ სიტყვათა სიმრავლეში. f გარდაქმნას (ასახვას) უწოდებენ ალფავიტურ ასახვას ან ალფავიტურ ოპერატორს, ხოლო x და y ალფავიტს შესაბამისად f ოპერატორის შესასვლელ და გამოსასვლელ ალფავიტებს. თითოეულ შემავალ $p = x_{i_1} x_{i_2} \dots x_{i_k}$ სიტყვას f ოპერატორი შეუსაბამებს $r = y_{i_1} y_{i_2} \dots y_{i_e}$ გამომავალ სიტყვას. ამიტომ ყოველი $r \epsilon \sigma(y)$-თვის არსებობს ისეთი $r = f(p)$, რომ $p \epsilon \sigma(x)$. ამ შემთხვევაში f არის ფუნქცია, რომლის განსაზღვრის არეა $\sigma(x)$, ხოლო მნიშვნელობათა სიმრავლე $\sigma(y)$.

ნებისმიერი აბსტრაქტული ავტომატი რეალიზებას უკეთებს რომელიმე f ოპერატორს, ანუ ინდუცირებს რომელიმე f ასახვას. ავღნიშნოთ ზოგიერთი დაშვება, რომლებიც დაკავშირებულია აბსტრაქტული ავტომატის ცნებასთან:

1. ავტომატის განსხვავებული მდგომარეობების ნებისმიერი რაოდენობა და მომენტალური გადასვლა ერთი მდგომარეობიდან მეორეში.
2. ერთი მდგომარეობიდან მეორეში გადასვლა შესაძლებელია მხოლოდ დროის დისკრეტულ ინტერვალებში.
3. შემავალი და გამომავალი სიგნალების რაოდენობა სასრულია (განსხვავებული სიგნალების).
4. შემავალი სიგნალები არის ავტომატის ერთი მდგომარეობიდან მეორეში გადასვლის მიზეზი, ხოლო გამომავალი სიგნალები - ავტომატის რეაქცია იმ შემავალ სიგნალებზე, რომლებიც ეკუთვნის დროის იმ მომენტებს, რომლებიც განისაზღვრება ავტომატის შესაბამისი სიგნალებით.

ამ მდგომარეობის გათვალისწინებით შეიძლება ვთქვათ, რომ აბსტრაქტული ავტომატი ფუნქციონირებს დროის დისკრეტულ t =0,1,2,… მომენტებში. ყოველ $x(t)$ შემავალ სიგნალზე ავტომატი რეაგირებს გამომავალი $y(t)$ სიგნალით.

განასხვავებენ ორი სახის ავტომატს: სინქრონულს და ასინქრონულს. სინქრონულ ავტომატებში მდგომარეობიდან მდგომარეობაში გადასვლა ხდება დროის თანაბარ შუალედებში, ასინქრონულში კი არა. შემდგომში ჩვენ განვიხილავთ სინქრონულ ავტომატებს.

ავტომატის $q(t)$ მდგომარეობა t მომენტში ცალსახად განისაზღვრება წინამდებარე $q(t-1)$ მდგომარეობითა და შემავალი $x(t)$ სიგნალით. ამიტომ შეიძლება დავწეროთ:

$$q(t) = \phi\left(q(t-1), \left(x(t)\right)\right), \qquad (1)$$

სადაც ϕ არის ფუნქცია, რომელიც განსაზღვრავს ავტომატის მომდევნო მდგომარეობას, აღინიშნება უბრალოდ $\phi(x, y)$-ით და ეწოდება გადასვლის ფუნქცია.

რეალური ავტომატის გამომავალი $y(t)$ სიგნალი ყოველთვის გამოჩნდება შემავალი $x(t)$ სიგნალის შემდეგ. მაგრამ ავტომატის $q(t-1)$ მდგომარეობიდან $q(t)$ მდგომარეობაში გადასვლის დროს t მომენტის მიმართ $y(t)$ გამომავალი სიგნალი შეიძლება ამ მომენტზე ადრე ან გვიან გამოჩნდეს. ამიტომ სამართლლიანია გამოსახულებები:

$$y(t) = \psi\big(q(t-1), x(t)\big), \qquad (2)$$
$$y(t) = \psi\big(q(t), x(t)\big), \qquad (3)$$

სადაც $\psi(x, y)$ არის (2)-ის შემთხვევაში ჩვეულებრივი, ხოლო (3)-ის შემთხვევაში წანაცვლებული გამოსავლის ფუნქცია. იგი ცალსახად განსაზღვრავს ავტომატიდან გამომავალ ასოს წინამდებარე დროის მომენტში $q(t-1)$ მდგომარეობასა და $x(t)$ შემავალ სიგნალზე დამოკიდებულებით, თუ იგი არის ჩვეულებრივი გამოსავლის ფუნქცია, ხოლო წანაცვლებული ფუნქციის შემთხვევაში $q(t)$ მდგომარეობასა და შემავალ სიგნალზე დამოკიდებულებით.

რეალური ავტომატების მუშაობის გათვალისწინებით განასხვავებენ ორი სახის ავტომატებს, რომლებიც მოიცემიან შემდეგი ფორმულებით:

პირველი გვარის ავტომატები ეწოდება ავტომატებს, რომლებიც განისაზღვრებიან

$$q(t) = \phi\big(q(t-1), x(t)\big),$$
$$y(t) = \psi\big(q(t-1), x(t)\big)$$

ფორმულებით და განსაზღვრავენ ამ ავტომატების ფუნქციონირების კანონს, ხოლო მეორე გვარის ავტომატები ეწოდება ავტომატებს, რომელთა ფუნქციონირების კანონი მოიცემა განტოლებებით:

$$q(t) = \phi\big(q(t-1), x(t)\big),$$
$$y(t) = \psi\big(q(t), x(t)\big),$$

სადაც $t = 1,2,3, \dots$

ნებისმიერი გვარის აბსტრაქტულ ავტომატს ეწოდება წესიერი, თუ გამომავალი $y(t)$ სიგნალი ცხადად არ არის დამოკიდებული შემავალ $x(t)$ სიგნალზე და განისაზღვრება მხოლოდ $q(t-1)$ ან $q(t)$ მდგომარეობით. პირველი გვარის ავტომატებს ეწოდებათ აგრეთვე **მილის** ავტომატები, ხოლო მეორე გვარის ავტომატებს - **მური**ს ავტომატები.

ავტომატების ფუნქციონირების აღწერა

არსებობს აბსტრაქტული ავტომატების აღწერის სხვადასხვა საშუალებები. ეს არის ანალიზური, გეომეტრიული და მატრიცული.

იტყვიან, რომ მოცემულია აბსტრაქტული A ავტომატი, თუ მოცემულია ხუთი ობიექტის ერთობლიობა: ავტომატში შემავალი $x = \{x_i\}$, $i\epsilon I = \{1,2, \dots, m\}$ალფავიტი, ავტომატიდან გამომავალი $y = \{y_j\}$, $j\epsilon J = \{1,2, \dots, n\}$ ალფავიტი, მდგომარეობათა Q ალფავიტი, რომლის $q_1\epsilon Q$ ელემენტი არის ავტომატის საწყისი მდგომარეობა, F ასახვა, რომელიც Q-ს ასახავს თავის თაში და რომელიც ნებისმიერ $q\epsilon Q$ და ყოველ შემავალ $x\epsilon X$ ასოს შეუსაბამებს $q_k\epsilon Q$ მდგომარეობას და რომელიც განსაზღვრავს გადასვლის $\phi(x, y)$ ფუნქციას, გამოსავალი $y\epsilon Y$ ასო, რომელიც განსაზღვრავს ჩვეულებრივ ან წანაცვლებულ $\phi(x, y)$გამოსავლის ფუნქციას. შესაბამისად $A = (X, Q, Y, q_1 \in Q,$ $F(x\epsilon X/y\epsilon Y))$ აღნიშნავს ნებისმიერ აბსტრაქტულ ავტომატს. თუ A არის პირველი გვარის აბსტრაქტული ავტომატი, მაშინ $\phi(x, y)$ - გამოსავლის ჩვეულებრივი ფუნქციაა, ხოლო თუ A არის მეორე გვარის, მაშინ $\psi(x, y)$) - გამოსავლის წანაცვლებული ფუნქციაა.

თუ Q სიმრავლე სასრულია, მაშინ A ავტომატი სასრულია, წინააღმდეგ შემთხვევაში A ავტომატი უსასრულოა. ყველა რეალური ავტომატი სასრულია.

ვაჩვენოთ, თუ როგორ შეიძლება განვსაზღვროთ ასახვა, რომელიც ინდუცირებულია მოცე-მული სასრული A ავტომატის მიერ. როგორც ცნობილია, ავტომატი ფუნქციონირებს t=0,1,2,.... დისკრეტულ დროში. იგულისხმებს, რომ t=0 საწყის მომენტში ავტომატი იმყოფება $q_1\epsilon Q$ საწყის მდგომარეობაში. დროის ყოველ მომენტში, საწყისი მომენტისაგან განსხვავებით, ავტომატის შესასვლელზე

ყოველთვის მიეწოდება $x(t)$ შემავალი სიგნალი, რომელიც წარმოადგენს შემავალი X ალფავიტის ნებისმიერ ასოს, ხოლო გამოსასვლელზე წარმოიქმნება რაღაც გამოსასვლელი $y(t)$ სიგნალი- Y გამოსავალი ალფავიტის ასო. ვთქვათ $\sigma(x)$ და $\sigma(y)$ შესაბამისად არის A ავტომატის შემავალი და გამომავალი სიტყვების სიმრავლე და $p = x_{i_1}x_{i_2} \dots x_{i_k}$ ნებისმიერი შემავალი სიტყვაა, ე.ი. $p\epsilon\sigma(x)$. A ავტომატის შესასვლელზე, რომელიც იმყოფება საწყის მდგომარეობაში, $x(1) = x_{i_1}$ $x(2) = x_{i_2} \dots x(k) = x_{i_k}$ სასრული მომდევრობის მიწოდება F ასახვის საფუძველზე იწვევს ავტომატის გამოსასვლელზე $y(1) = y_{j_1}$ $y(2) = y_{j_2} \dots y(k) = y_{j_k}$ცალსახად განსაზღვრული სასრული მიმდევრობის გამოჩენას, რომელიც შეესაბამებამება $r = y_{j_1}y_{j_2} \dots y_{j_k}$ გამომავალ სიტყვას $\sigma(y)$-დან. ამიტომ $r = f(p)$. თუ ყოველ შესასვლელ სიტყვას მივაკუთნებთ მის შესაბამის $r = \sigma(y)$ გამოსასვლელ სიტყვას, მივიღებთ საძიებელ f ასახვას, რომელიც წარმოადგენს ასახვას, ინდუცირებულს სასრული A ავტომატის მიერ.

ცხადია, რომ F ასახვა Q სიმრავლისა Q -ში ცალსახად იძლევა $\phi(x, y)$ და $\psi(x, y)$ ფუნქციებს, რომლებიც განსაზღვრავენ A ავტომატის ფუნქციონირების კანონს და პირიქით. ეს ფუნქციები შეიძლება წარმოვიდგინოთ მართკუთხოვანი ცხრილის სახით, რომლის სტრიქონები შეესაბამებიან შემავალი X ალფალიტის სხვადასხვა ასოებს, ხოლო სვეტები - Q ალფავიტის განსხვავებულ მდგომარეობებს . X_i სტრიქონისა და q_k სვეტის გადაკვეთაზე $\phi(x, y)$-ის გადასვლის ცხრილში იწერება q მდგომარეობა, რომელშიც ის გადადის q მდგომარეობიდან x_i შემავალი სიგნალის ზემოქმედების შედეგად, ხოლო $\psi(x, y)$-ის გამოსასვლელის ცხრილში იწერება ასო, რომელიც გამოჩნდება ავტომატის გამოსასვლელზე.

ორ აბსტრაქტულ ავტომატს, A-სა და B-ს, ერთნაირი x შემავალი და y გამომავალი ალფავიტებით ეწოდება **ეკვივალენტური,** თუ ისინი ინდუცირებენ ერთსა და იმავე f ასახვას $\sigma(x)$ სიმრავლისა $\sigma(y)$ სიმრავლეში.

დავამყაროთ კავშირი პირველ და მეორე გვარის ავტომატებს შორის. ვთქვათ მოცემულია მეორე გვარის $A = (X, Q, Y, q_1 \in Q,\ F(x\epsilon X/y\epsilon Y))$ ავტომატი. დავწეროთ $\phi(x, y)$ გადასვლის ფუნქცია და გამოსავლის $\psi(x, y)$ წანაცვლებული ფუნქცია. ეს უკანასკნელი ასე გამოისახება: $y(t) = (\psi(q, t), x(t))$. ამ გამოსახულებაში ჩავსვათ $q(t) = \phi(q(t-1), x(t))$ მნიშვნელობა. მაშინ მივიღებთ განტოლებას

$$y(t) = \psi\Big(\phi\big(q(t-1), x(t)\big), x(t)\Big) = \psi^I(q(t-1), x(t)),$$

რომელიც განსაზღვრავს გამოსავლის ნებისმიერ $\psi(q, x)$ ფუნქციას და რომელიც ახასიათებს პირველი გვარის ავტომატს. მაშასადამე, მეორე გვარის ავტომატის გამოსასვლელის წანაცვლებულ $\psi(q, x)$ ფუნქციაში თუ ჩავსვამთ გადასვლის $\psi(q, x)$

ფუნქციას, მივიღებთ პირველი გვარის $A = (X, Q, Y, q_1 \in Q,\ F(x\epsilon X/y\epsilon Y))$ ავტომატს. რომელიც ინდუცირებს იმავე სახის ასახვას, როგორსაც A ავტომატი.

მეორე გვარის ავტომატის ასეთი სახით დაყვანას მისი ეკვივალენტური პირველი გვარის ავტომატზე, ეწოდება მეორე გვარის ავტომატის ინტერპრეტაცია პირველი გვარის ავტომატით. უფრო რთულია ჩვენება იმისა, რომ პირველი გვარის ავტომატისათვის შეიძლება აიგოს მისი ეკვივალენტური მეორე გვარის ავტომატი.

მაშასადამე, ჩვენ განვიხილეთ ავტომატების მოცემის ანალიზური ხერხი. ავტომატების მოცემის გეომეტრიული ხერხი დაიყვანება შესაბამისი ორიენტირებული გრაფის გამოსახულებაზე, რომლის წვეროებს წარმოადგენს ავტომატის მდგომარეობები, აღნიშნული სხვადასხვა $q\,\epsilon Q$ ასოებით, ხოლო ყოველი (q_k, q_e) გვერდის გასწვრივ იწერება ასო შემავალი $x_i\epsilon X$ ალფავიტიდან, რომელიც იწვევს ავტომატის q_k მდგომარეობიდან q_e-ში გადასვლას და გამომავალი ალფავიტის y_j ასო, რომელიც გამოჩნდება ავტომატის გამოსასვლელზე. თუ ავტომატი პირველი გვარისაა, მაშინ გამომავალი y_j ასო განისაზღვრება (q_k, x_e) წყვილით, ხოლო მეორე გვარის ავტომატის შემთხვევაში y_j დამოკიდებულია (q_e, x_i)-ზე. ავტომატის საწყისი მდგომარეობა აღინიშნება $q_1\epsilon Q$ ასოთი. მაშასადამე, ასეთი ორიენტირებული გრაფი ცალსახად იძლევა რომელიღაც აბსტრაქტულ ავტომატს. გრაფებს დატვირთული გვერდებით (გვერდებზე წონითი ფუნქციებით) ეწოდება გრაფოიდები. ამიტომ ორიენტირებული გრაფოიდი არის აბსტრაქტული ავტომატის გეომეტრიული ინტერპრეტაცია.

გრაფოიდიდან ადვილად შეიძლება გადასვლა მოცემულ ავტომატზე დატვირთული წინარე ხის საშუალებით, რომელიც წარმოადგენს სტრუქტურული გრაფის მაგალითს. გრაფოიდის მიხედვით, წინარე ხის აგება ხდება შემდეგნაირად: ფიქსირდება წვერო, რომელიც შეესაბამება საწყის $q_1\epsilon Q$ მდგომარეობას. იგი არის პირველი რანგის წვერო ან წინარე ხის ძირი. პირველი რანგის წვეროდან გაივლება ორი რკალი, რომელთაც ეწოდებათ პირველი რანგის რკალები და რომელთა რაოდენობაც ტოლია ავტომატის შემავალი ალფავიტის ასოთა რაოდენობისა. ყოველი რკალი შედის მეორე რანგის წვეროში (m რაოდენობის წვერო). მეორე რანგის ყოველი წვეროდან გამოდის m რაოდენობის რკალი, რომლებიც შედის მესამე რანგის წვეროებში ($m_i{}^2$ წვერო) და ა.შ. პირველი და მომდევნო რანგის თითოეულ რკალს მიეწერება შესავალი $x_i\epsilon X$ ასო და ფრჩხილებში გამოსავალი $y_j\epsilon Y$ ასო შესაბამისი ალფავიტებისა, ხოლო მეორე და მომდევნო რანგის წვეროებს $q_k\epsilon Q$ ასო. მაშასადამე, ავტომატის წინარე ხე არის გრაფი, რომლის ყოველ წვეროში, გარდა q_1 ფუძისა, შედის ერთი რკალი და რომელიც არ შეიცავს კონტურებს.

ეხლა განვიხილოთ ავტომატების მოცემის მატრიცული ხერხი. მის ერთ-ერთ სახეს წარმოადგენს ზემოთ აღწერილი მართკუთხა მატრიცები, რომელთაც ეწოდებათ გადასვლებისა და გამოსასვლელების ცხრილები და განსაზღვრავენ $\phi(q,x)$ და $\psi(q,x)$ ფუნქციებს. რადგანაც ამ მატრიცებში მონაწილეობს ყველა ასო შესასვლელი და გამოსასვლელი ალფავიტებისა და აგრეთვე მდგომარეობათა ალფავიტისა, ამიტომ ისინი სრულად და ცალსახად იძლევიან რაღაც ავტომატს.

უფრო ხშირად გამოიყენება კვადრატული მატრიცები, რომელთაც ეწოდებათ ავტომატის შეერთების მატრიცა და იგი შემდეგნაირად აიგება. შეერთების მატრიცის სტრიქონები და სვეტები შეესაბამებიან ავტომატის სხვადასხვა მდგომარეობას. პირველი სტრიქონი და პირველი სვეტი შეესაბამება საწყის $q_1 \epsilon Q$ მდგომარეობას. q_k-ური სტრიქონისა და q_e-ური სვეტის გადაკვეთაზე იწერება შემავალი ალფავიტის $x_i \epsilon X$ ასო ან შემავალი ასოების დიზიუნქცია, რომელიც იწვევს ავტომატის გადასვლას q_k მდგომაეობიდან q_e-ში, ხოლო ფრჩხილებში იწერება გამომავალი ალფავიტის $y_j \epsilon Y$ ასო ან გამომავალი ასოების დიზიუნქცია, რომლებიც გამოჩნდებიან ავტომატის გამოსასვლელზე. თუ არცერთ ასოს შემავალი ალფავიტისა არ გადაყავს ავტომატი q_k მდგომარეობიდან q_e-ში, მაშინ შესაბამის გადაკვეთაზე იწერება ნული.

შეერთების მატრიცას გააჩნია ის თვისება, რომ მის ნებისმიერ სტრიქონში შემავალი ალფავიტის ყოველი ასო მხოლოდ ერთხელ უნდა შეგვხვდეს. ეს პირობა დაკავშირებულია ავტომატის გადასვლისა და გამოსავლის ფუნქციების ცალსახობასთან და ეწოდება ცალსახობის პირობა.

მაგალითზე ვაჩვენოთ ავტომატის მოცემის სხვადასხვა მეთოდები და მეორე გვარის ავტომატის ინტერპრეტაცია პირველი გვარის ავტომატით.

მაგალითი 1: ვთქვათ მოცემულია აბსტრაქტული ავტომატი $A=(X,Q,Y,q_1 \in Q, F(x\epsilon X/y\epsilon Y))$, სადაც $X=\{x_1,x_2,x_3\}$, $Q=\{q_1,q_2,q_3,q_4\}$, $Y=\{y_1,y_2\}$, ხოლო Q სიმრავლის Q სიმრავლეში ასახვა განისაზღვრება შემდეგნაირად:

$$Fq_1=\{q_2(x_1/y_1),q_4(x_2/y_1),q_1(x_3/y_2)\},$$
$$Fq_2=\{q_1(x_1/y_2),q_3(x_2/y_1),q_4(x_3/y_1)\},$$
$$Fq_3=\{q_1(x_1/y_1),q_4(x_2/y_2),q_2(x_3/y_2)\},$$
$$Fq_4=\{q_4(x_1/y_2),q_1(x_2/y_1),q_3(x_3/y_1)\}.$$

ვიგულისხმოთ, რომ A ავტომატი არის პირველი გვარის. A ავტომატის შესასვლელზე, რომელიც თავის მხრივ იმყოფება q_1 საწყის მდგომარეობაში, $p_1=x_1x_1x_2x_3x_2x_3$ სიტყვის მიწოდება იწვევს გამოსასვლელზე $r_1=y_1y_1y_1y_2y_2y_1y_1$ სიტყვის გამოჩენას, ხოლო $p_2=x_3x_2x_2x_1x_1x_3x_1x_2$ შესასვლელი სიტყვა იწვევს გამოსასვლელზე $r_2=y_2y_1y_1y_1y_2y_2y_1y_1$ სიტყვის გამოჩენას. F ასახვაზე

დამოკიდებულებით (Q-ს ასახვა თავის თავში) ადვილად შეიძლება ავაგოთ გადასვლებისა და გამოსასვლელების ცხრილები A ავტომატისათვის. გადასვლის ცხრილს, რომელსაც განსაზღვრავს გადასვლის $\phi(q,x)$ ფუნქცია, აქვს (1) სახე, ხოლო გამოსასვლელის ცხრილს, რომელსაც განსაზღვრავს გამოსასვლელი $\psi(q,x)$ ჩვეულებრივი ფუნქცია, (2) სახე.

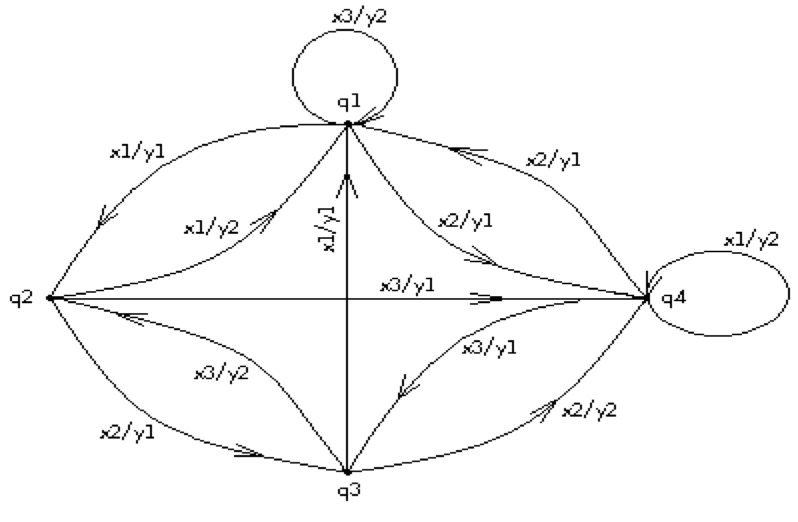
$X\backslash Q$	q_1	q_2	q_3	q_4
x_1	q_2	q_1	q_1	q_4
x_2	q_4	q_3	q_4	q_1
x_3	q_1	q_4	q_2	q_3

ცხრ.1

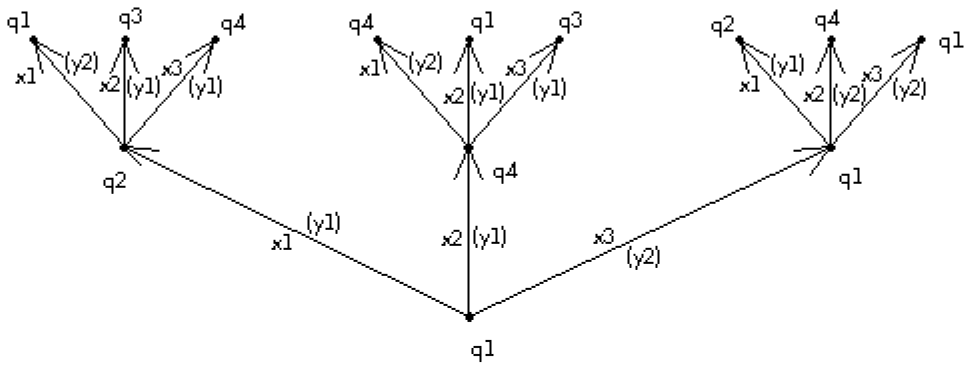
$X\backslash Q$	q_1	q_2	q_3	q_4
x_1	y_1	y_2	y_1	y_2
x_2	y_1	y_1	y_2	q_1
x_3	y_2	y_1	y_2	y_1

ცხრ.2

ავტომატის გრაფოიდი გამოსახულია 1 ნახაზზე, ხოლო დატვირთული წინარე ხის პირველი სამი იარუსი, აგებული გრაფოიდის მიხედვით, ნაჩვენებია 2-ე ნახაზზე.



ნახ.1



ნახ.2

ავტომატის შეერთების მატრიცას აქვს სახე:

$$R = \begin{array}{c} \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{array} \begin{array}{c} \begin{array}{cccc} q_1 & q_2 & q_3 & q_4 \end{array} \\ \left\| \begin{array}{cccc} x_3(y_2) & x_1(y_1) & 0 & x_2(y_1) \\ x_1(y_2) & 0 & x_2(y_1) & x_3(y_1) \\ x_1(y_1) & x_3(y_2) & 0 & x_2(y_2) \\ x_2(y_1) & 0 & x_3(y_1) & x_1(y_2) \end{array} \right\| \end{array} \qquad (4)$$

ე.ი. შეერთების R მატრიცა მიიღება q_1, q_2, q_3, q_4 ასახვებიდან. ვთქვათ ავტომატი არის მეორე გვარის ავტომატი. მაშინ (1) ცხრილი განსაზღვრავს გადასვლის $\phi(q,x)$ ფუნქციას, ხოლო მე-2 ცხრილი გამოსასვლელის $\psi(q,x)$ წანაცვლებულ ფუნქციას. მე-(2) გვარის ავტომატის გრაფოიდი მოცემულია (1)-ნახაზზე, ხოლო შეერთების მატრიცას აქვს (4) სახე. თუ ავტომატი იმყოფება q_1 საწყის მდგომარეობაში და მის შესასვლელზე მოვა $p_1 = x_1x_1x_2x_3x_2x_3$ და $p_2 = x_3x_2x_2x_1x_1x_3x_1x_2$ სიტყვები, ისეთები როგორებიც პირველი გვარის ავტომატის შესასვლელზე, მაშინ გამოსასვლელზე

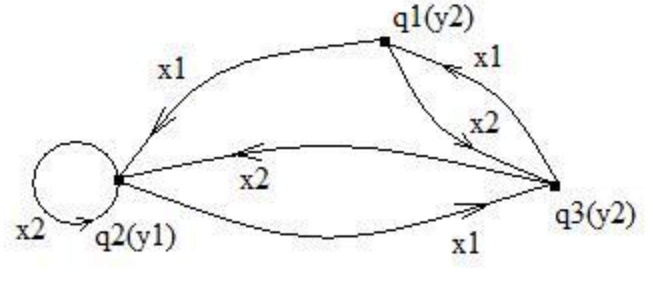
მივიღებთ $r_1 = y_2y_1y_1y_2y_1y_2$ და $r_2 = y_2y_1y_1y_2y_1y_2y_2y_2$ სიტყვებს, რომლებიც განსხვავდებიან პირველი გვარის ავტომატის გამოსასვლელი სიტყვებისგან. ამიტომ პირველი გვარის ავტომატის f ასახვა განსხვავდება მეორე გვარის ავტომატის ასახვისგან.

როგორც ცნობილია, მურის ავტომატის გამოსასვლელი სიგნალი დამოკიდებულია მხოლოდ მის მდგომარეობაზე ე.ი. $y(t) = \psi(q(t))$. ამიტომ ყოველი $q \in Q$ მდგომარეობა აღნიშნულია გამოსასვლელი $y \in Y$ ასოთი, რომელიც F ასახვაში ისმება ფრჩხილებში Q ელემენტის გვერდით. ამიტომ ანალიზურად მურის ავტომატი შეიძლება მოცემული იყოს ფორმით: $B = (X, Q, Y, q_1 \in Q, F(x \in X))$, სადაც F ასახვა Q-სათვის თავში ნებისმიერი $q \in Q$ და ყოველ $x \in X$ ასოს შეუსაბამებს $q_k \in Q$ მდგომარეობას, რომელიც განსაზღვრავს გამოსასვლელის წანაცვლებულ $\varphi(q)$ ფუნქციას. მაშასადამე, მურის ავტომატის გამოსასვლელის ცხრილი დაიყვანება ერთ სტრიქონამდე, რომელიც განლაგდება მდგომარეობათა გადასვლის ცხრილის თავზე. მურის ავტომატის გეომეტრიული ინტერპრეტაციის გამოსასვლელი ალფავიტის $y \in Y$ ასოები იწერება მდგომარეობათა $q \in Q$ ალფავიტის გვერდით ფრჩხილებში, ხოლო შეერთების მატრიცაში იწერება შესაბამისი სვეტის თავზე.

მაგალითი 2: ვთქვათ მოცემულია მურის $B = (X, Q, Y, q_1 \in Q, F(x \in X))$ ავტომატი, სადაც $X = \{x_1, x_2\}$, $Q = \{q_1, q_2, q_3\}$, $Y = \{y_1, y_2\}$, ხოლო

$$Fq_1(y_2) = \{q_2(x_1), q_3(x_2)\},$$
$$Fq_2(y_1) = \{q_3(x_1), q_2(x_2)\},$$
$$Fq_3(y_2) = \{q_1(x_1), q_2(x_2)\}.$$

ავტომატის გრაფოიდს აქვს სახე (ნახ.4).



ნახ.4

ამ ნახაზის მიხედვით ადვილად ავაგებთ გადასვლის ცხრილებს, რომლებიც განსაზღვრავენ მურის ავტომატის ფუნქციონირების კანონს ($\varphi(q,x)$ და $\psi(q)$ ფუნქციებს).

y	y_2	y_1	y_2
X\Q	q_1	q_2	q_3
x_1	q_2	q_3	q_1
x_2	q_3	q_2	q_2

(ცხრ.4)

X\Q	q_1	q_2	q_3
x_1	y_1	y_2	y_2
x_2	y_2	y_1	y_1

(ცხრ.5)

შეერთების R_B მატრიცას მურის ავტომატისათვის აქვს სახე:

$$\begin{matrix} (y_2) & (y_1) & (y_2) \end{matrix}$$

$$R_B= \left\| \begin{matrix} 0 & x_1 & x_2 \\ 0 & x_2 & x_1 \\ x_1 & x_2 & 0 \end{matrix} \right\| .$$

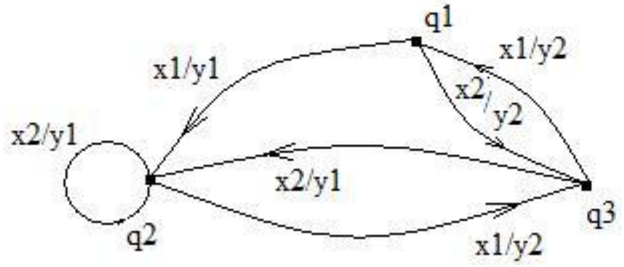
ავაგოთ ახლა მილის ავტომატი, რომელიც იქნება მურის B ავტომატის ინტერპრეტაცია. (4) ცხრილის საშუალებით ავაგოთ ψ(q,x) გამოსასვლელის ფუნქცია (ცხრ.5), რომელიც განსაზღვრავს მილის A = (X, Q, Y, q_1∈Q, F(x∈X / y∈Y)) ავტომატს:

$$Fq_1 =\{q_2(x_1/y_1),\ q_3(x_2/y_2)\},$$
$$Fq_2 =\{q_3(x_1/y_2),\ q_2(x_2/y_1)\},$$
$$Fq_3 =\{q_1(x_1/y_2),\ q_2(x_2/y_1)\},$$

შეერთების R_A მატრიცას ექნება სახე:

$$\boldsymbol{R_A}= \left\| \begin{array}{ccc} 0 & x_1(y_1) & x_2(y_2) \\ 0 & x_2(y_1) & x_1(y_2) \\ x_1(y_2) & x_2(y_1) & 0 \end{array} \right\| ,$$

ხოლო მილის ავტომატის გეომეტრიული ინტერპრეტაცია მოცემულია მე-5 ნახაზზე:



ნახ.5

მაგ. $p = x_1x_2x_2x_1$, $r = y_1y_1y_1y_2$, $p = x_2x_1x_2x_1x_1$,, $r = y_2y_2y_2y_2y_1$. ადვილი შესამოწმებელია, რომ მილის ავტომატი σ(x) სიმრავლის სიტყვებს ასახავს σ(y)-ში ისევე, როგორც მურის B ავტომატი.

აქამდე ვიხილავდით პირველი და მეორე გვარის ავტომატებს, რომელთათვისაც მდგომარეობათა Q სიმრავლის თავის თავში F ასახვა განსაზღვრული იყო ნებისმიერი (q,x) წყვილისათვის, სადაც q∈Q x∈X . ასეთ ავტომატებს უწოდებენ სრულად განსაზღვრულებს. ავტომატები (ნებისმიერი გვარის), რომელთათვისაც Q სიმრავლის თავის თავში F ასახვა განსაზღვრული არ არის ყველა q∈Q და x∈X წყვილისათვის, უწოდებენ ნაწილობრივ ავტომატებს. ნაწილობრივი ავტომატის ცნება შეიძლება განვმარტოთ, თუ გამოვიყენებთ ავტომატის გამოსასვლელისა და შესასვლელის ფუნქციებს. **ნაწილობრივი ავტომატი** ეწოდება ავტომატს, რომლისთვისაც გადასვლის ფუნქცია ან გამოსასვლელის ფუნქცია ან ორივე ეს ფუნქცია განსაზღვრული არ არის q∈Q და x∈X ელემენტთა წყვილის ყველა მნიშვნელობისათვის. გადასვლისა და გამოსვლის ცხრილების და აგრეთვე ავტომატის შეერთების მატრიცის იმ ადგილებზე, რომლებშიდაც შესაბამისი გადასვლები და გამოსვლები არ არის განსაზღვრული, ჩაიწერება ნული ან ტირე.

განვიხილოთ ასახვა, ინდუცირებული ნაწილობრივი ავტომატით. ავტომატის შესასვლელზე მივაწოდოთ შემავალი ალფავიტის სხვადასხვა სიტყვები. ვიგულისხმოთ, რომ რაღაც სიტყვის P=$x_{i_1}\, x_{i_2} \dots x_{i_k}$ შემავალი x_{i_k} ასოს მიწოდებისას

მისი შესაბამისი გამოსასვლელი სიგნალი არ არის განსაზღვრული. ამ შემთხვევაში იტყვიან, რომ P შესასვლელი სიტყვა აკრძალულია მოცემული ნაწილობრივი ავტომატისათვის. ყველა ასეთი სიტყვების აკრძალული ერთობლიობა შეადგენს მოცემული ავტომატის აკრძალულ არეს. სიტყვა, რომელიც არ არის აკრძალული, ეწოდება დასაშვები, ხოლო მათ სიმრავლეს, **დასაშვებ სიტყვათა სიმრავლე.**

განვსაზღვროთ დამაკავშირებელი ავტომატის ცნება, რომელიც მეტად მნიშვნელოვან როლს თამაშობს ნაწილობრივი ავტომატის შესწავლისას.

აბსტრაქტული ავტომატის $q_e \in$Q მდგომარეობას ეწოდება **მიღწევადი**, თუ იგი ემთხვევა საწყის მდგომარეობას ან თუ არსებობს ისეთი $q_k \in$Q მიღწევადი მდგომარეობა და შემავალი ალფავიტის ისეთი$x_i \in X$ ასო, რომ $q_l(x_i) \in Fq_k$. წინააღმდეგ შემთხვევაში q_l მდგომარეობას ეწოდება არამიღწევადი. აბსტრაქტულ ავტომატს, რომელთა ყველა მდგომარეობა მიღწევადია, ეწოდება **დამაკავშირებელი ავტომატი.**

ადვილი დასანახია, რომ საწყისი მდგომარეობიდან ავტომატი ვერ გადავა არამიღწევად მდომარეობაში, ამიტომ ავტომატის შეერთების მატრიცაში ის სტრიქონები და სვეტები, რომლებიც შეესაბამებიან არამიღწევად მდგომარეობებს, შეიძლება ამოვშალოთ. ამ შემთხვევა-ში ავტომატის მიერ ინდუცირებული ასახვა არ შეიცვლება, ამიტომ შემდეგში განვიხილავთ დამაკავშირებელ ავტომატებს.

ახლა განვიხილოთ ავტონომიური ავტომატი. ვთქვათ მოცემულია მილის A=(X, Q, Y, $q_1 \in$Q, F(x∈X/y∈Y)) ავტომატი. განასხვავებენ მილის ავტონომიური ავტომატის ორ სახეს: ავტონომიური ავტომატი შესასვლელით და ავტონომიური ავტომატი გამოსასვლელით.

ავტონომიური ავტომატი შესასვლელი x∈X ასოთი ეწოდება ავტომატს, რომელიც აღინიშნება $A_x = (\{x\}, Q, q_1 \in Q, F(x))$ სიმბოლოთი და რომლის შესასვლელი ალფავიტი შედგება მხოლოდ {x} სიმრავლისაგან, ხოლო F ასახვა Q სიმრავლისა თავის თავში განისაზღვრება ისე, რომ ნებისმიერი q∈Q მდგომარეობას შეესაბამება ისეთი $q_k \in$Q მდგომარეობა, რომელშიდაც გადასვლა განხორციელდება მხოლოდ x შემავალი ასოთი. A_x ავტომატის გამომავალი ასოები არ განიხილება.

თუ მოცემულია A ავტომატის გრაფოიდი, მასზე ყველა გვერდი უნდა წავშალოთ, რომელზედაც აღნიშნულია x -გან განსხვავებული შემავალი ასოები, მივიღებთ A_x ავტონომიური ავტომატის გრაფს. ავღნიშნოთ, რომ ავტონომიური ავტომატი მოიცემა ზერჟის გრაფით და არა გრაფოიდით. A_x ავტომატის შეერთების მატრიცა მიიღება A ავტომატის შეერთების მატრიციდან იმ ელემენტების ნულით შეცვლით, რომლებიც x ასოსაგან განსხვავდებიან და გამომავალი ასოების წაშლით, რომლებიც ფრჩხილებით x ასოს გვერდით დგას.

(6) ნახაზზე მოცემულია მილის ავტონომიური ავტომატის გრაფი შესაბამისად x_1 და x_2 შემავალი ასოს მიხედვით.

q2 x_1 -ით q3 q2 x_2-ით

ნახ. 6

A **ავტონომიური ავტომატი** y∈Y გამომავალი ასოს მიხედვით ეწოდება ავტომატს (აღინიშნება Ay=(Q, {y}, q_1∈Q, F(y)) სიმბოლოთი), რომლის გამომავალი ალფავიტი შედგება {y} სიმრავლისგან, ხოლო F ასახვა Q სიმრავლისა თავის თავში ისე განისაზღვრება, რომ ნებისმიერ q∈Q მდგომარეობას შეესაბამება ისეთი q_l∈Q მდგომარეობა, რომელშიდაც გადასვლა გამოიწვევს მხოლოდ გამომავალი y ასოს გამოჩენას. A_y ავტომატის შემავალი ასოები მხედველობაში არ მიიღებიან.

მაშასადამე, მილის A ავტომატი შეიძლება მოცემული იყოს როგორც ერთობლიობა ავტონომიური ავტომატებისა შემავალი და გამომავალი ალფავიტების ყველა ასოს მიხედვით. მურის ავტომატის შემთხვევაში აზრი აქვს ვისაუბროთ მხოლოდ ავტონომიურ ავტომატებზე შესასვლელი ასოების მიხედვით.

(7) ნახაზზე, მოცემულია მილის ავტონომიური ავტომატების გრაფი გამომავალი y_1 და y_2 ასოების მიხედვით, რომლის გრაფოიდიც (5)-ე ნახაზზეა მოცემული.

y_1-ით q_2 y_2-ით q_3

ნახ.7

ავტომატის ცნება , ავტომატის მუშაობის პრინციპი

ავტომატის ცნება განიხილება ორ ასპექტში:

1. ავტომატი - მოწყობილობა, რომელიც ასრულებს რაღაც ფუნქციებს ადამიანის უშუალო ჩარევის შედეგად. ავტომატი არის რეალური მოწყობილობა, რომელიც თუ რატომ და როგორ მუშაობს, გასაგებია სულ მცირე იმ ადამიანთათვის მაინც, ვინც ის დააპროექტა და დაამზადა. ავტომობილი, ტრაქტორი, თვითმფრინავი, ტელევიზორი, ტელეფონი, შუქნიშანი - ავტომატებია. ამ ასპექტში ეგმ უნდა განვიხილოთ როგორც ავტომატი, რომელიც მუშაობს ადამიანის მიერ შედგენილი პროგრამის მიხედვით.

2. ავტომატი - მათემატიკური ცნება, რომელიც აღნიშნავს რეალური ტექნიკური მოწყობილობის მათემატიკურ მოდელს. ავტომატი არის აბსტრაქტული მოწყობილობა, გაუგებარი თუ რატომ და რისთვის მუშაობს, და , საზოგადოდ, რატომ შეუძლია მას მუშაობა. ამ ასპექტით ავტომატი არის „შავი ყუთი“, რომელსაც თეორიულად შეუძლია ჩაატაროს რაიმე მოქმედებები. მათემატიკის აზრით კი აბსოლუტურად არაა აუცილებელი იმის ცოდნა, თუ როგორ და რისთვის ასრულებს ამა თუ იმ მოქმედებას.

ნებისმიერ ავტომატს უნდა გააჩნდეს რაღაც რაოდენობის შესასვლელი, რაღაც რაოდენობის გამოსასვლელი და რაღაც რაოდენობის შიგა მდგომარეობები.

ავტომატი არის რაიმე მოწყობილობის აბსტრაქტული მოდელი, რომელიც ფუნქციონირებს დისკრეტულ დროში. იგი შემავალი სიგნალების (სტიმულების) სასრულ მიმდევრობას გადაამუშავებს და გადააქცევს მას გამომავალი სიგნალების (რეაქციების) სასრულ მიმდევრობად.

სასრული ავტომატის მუშაობის პროცესში ხდება მისი სასრული რაოდენობის შიგა მდგომარეობების მიმდევრობითი ცვლა. ამავე დროს ავტომატის მდგომარეობა დროის განსაზღვრულ მომენტში ცალსახად განისაზღვრება შემავალი და გამომავალი სიგნალებით. ასეთი ავტომატები თანამედროვე გამოთვლითი ტექნიკისა და ყველა შესაძლებელი ავტომატური კონტროლისა და მართვის დისკრეტული სისტემების საფუძველს წარმოადგენენ.

ავტომატებთან, როგორც მექანიკურ მოწყობილობებთან, დაკავშირებული ზოგიერთი ისტორიული ფაქტი.

1. გერმანელმა ფილოსოფოსმა და ალქიმიკოსმა ალბერტ დიდმა 1216 – 1246 წ.წ. შექმნა „რკინის“ მსახური - ავტომატი, რომელიც სახლში მოახლის (привратник) მოვალეობას ასრულებდა.

2. ასტრონომმა იოჰან რეგიამონტანამ (1436-1476) შექმნა მექანიკური არწივი, რომელიც თავის დახრითა და ფრთების რხევით ესალმებოდა რომის იმპერატორის მაქსიმილიან II - ის ნიურბერგში შესვლას.

3. მექანიკოსი ჟან დე ვაკანსონი (1709 – 1782) მსოფლიოში პირველი ავტომატური სართავი დაზგის ავტორია. მან შექმნა მექანიკური იხვი, ზუსტი ასლი თავისი ცოცხალი ორეულისა, რომელიც ცურავდა, იწმენდდა ფრთებს, ყლაპავდა ხელის გულიდან მარცვლებს. მის მიერ შექმნილი მექანიკური ფლეიტისტი კი ასრულებდა თერთმეტ მუსიკალურ

4. პირველი პრიმიტიული არითმომეტრის კონსტრუქცია შეიმუშავა ბლეზ პასკალმა 1641 წ. ამ აღმოჩენისაკენ მას უბიძგა მამამისის განუწყვეტელმა წვალებამ, რომელიც საგადასახადო ინსპექტორად მუშაობდა და დღე და ღამე ასრულებდა დიდ რიცხვით გამოთვლებს. არითმომეტრის გამოგონებით თვრამეტი წლის ბიჭმა მამა გაათავისუფლა რთული გამოთვლებისაგან, ხოლო მსოფლიოს აჩუქა პირველი კალკულიატორი, რომელიც ასრულებდა რიცხვების შეკრებასა და გამოკლებას.

5. პირველი შახმატის ავტომატი აგებულ იქნა 1890 წ. ესპანელი ინჟინრის ტორეს კევედოს მიერ.

6. პირველი გამომთვლელი მანქანა ავტომატური მართვით შექმნა ჩარლზ ბაბადჟმა 1822წ. მან დააპროექტა არითმომეტრი, რომელსაც ჰქონდა დამამახსოვრებელი და არითმეტიკული მოწყობილობა, რომლებიც შემდეგში გახდა თანამედროვე ეგმ - ის

ავტომატების კლასიფიკაცია

ავტომატების კლასიფიკაცია შეიძლება მოვახდინოთ სხვადასხვა ნიშნის მიხედვით

1. საქმიანობის მიხედვით ავტომატები არის: ინფორმაციული, მმართავი და გამომთვლელი.

ინფორმაციულ ავტომატებს განეკუთვნება სხვადასხვა საცნობარო ცხრილი, საინფორმაციო ტაბლო სტადიონზე, ავარიული სიგნალიზაციის მოწყობილობა.

მმართავ ავტომატებს განეკუთვნება მოწყობილობები, რომლებიც განკუთვნილია რაიმე პროცესების მართვისათვის (ლიფტი, კონვეიერი, დაზგა, შლაგბაუმი).

გამომთვლელ ავტომატებს განეკუთვნება მიკროკალკულატორი, პროცესორი ეგმ -ში და მსგავსი მოწყობილობები, რომლებიც ასრულებენ გამთვლებს.

მრავალი ავტომატი იმდენად რთული სისტემაა, რომ ისინი ერთდროულად წარმოადგენენ როგორც გამომთვლელ, ისე მმართავ და ინფორმაციულ ავტომატებს.

2. სასრული ავტომატები ინფორმატიკის აზრით არის ისეთი ავტომატები, რომლებიც წარმოადგენენ ინფორმაციის დისკრეტულ გარდამქმნელებს. ამ დისკრეტულ გარდამქმნელებში როგორც შემავალი, ასევე გამომავალი სიგნალების სიმრავლეები სასრულია. სასრულია აგრეთვე შიგა მდგომარეობათა სიმრავლე.

3. ციფრული ავტომატები - ავტომატები, რომლებიც ციფრულ ინფორმაციას გარდაქმნიან. ასეთ ავტომატში შემავალი სიგნალები მოიცემა სასრული რაოდენობის მყისიერი სიმბოლოების სახით: ამ სიგნალების ხანგრძლიობა იმდენად მცირეა, რომ იგი შეიძლება უგულელვყოთ. ფიქსირებული დროის განმავლობაში მიმდინარეობს შემავალი სიმბოლოების გარდაქმნა, ხოლო გამოსასვლელზე მიმდინარეობს ნახტომისებრი გადასვლა ერთი მდგომარეობიდან მეორე მდგომარეობაში.

4. აბსტრაქტული ავტომატები - ავტომატები, რომლებიც შემავალი X ალფავიტის სიტყვათა სიმრავლეს ასახავს გამომავალი Y ალფავიტის სიტყვათა სიმრავლეში.

აბსტრაქტული ავტომატი არის:

~ მათემატიკური მოდელი,

~ კოდური მიმდევრობის რაიმე გარდაქმნის მოქმედების ალგორითმი,

~ შემავალი ალფავიტის გამომავალ ალფავიტში გარდაქმნის კანონი.

5. სინქრონული და ასინქრონული ავტომატები

6. სასრული და უსასრულო ავტომატები

7. დეტერმინირებული და ალბათური (სტოქასტური) ავტომატები

8. უნივერსალური ავტომატი. ავტომატების თეორიაში დამტკიცებულია, რომ სხვადასხვა სახის ინფორმაციის გარდაქმნისათვის საკმარისია პროგრამისა და შესაბამისი კოდირების საშუალებით აიგოს უნივერსალური ავტომატი, რომელსაც შეუძლია გადაწყვიტოს ნებისმიერი ამოცანა.

9. სინქრონული და ასინქრონული ავტომატები

10. კომბინაციური ავტომატები არის ისეთი ავტომატები, რომელშიდაც გამომავალი სიმბოლო არაა დამოკიდებული მის მდგომარეობაზე და განისაზღვრება მხოლოდ მიმდინარე შემავალი სიგნალებით, ე. ი. ასეთ ავტომატში ყველა მდგომარეობა ეკვივალენტურია. ასეთ ავტომატში გადასვლის ფუნქცია გადაგვარებულია და ფუნქციონირების პროცესში უცვლელია. ამიტომ მინიმალურ კომბინაციურ ავტომატს აქვს მხოლოდ ერთი მდგომარეობა.

სასრული ავტომატების მაგალითები

აქ მოყვანილ მაგალითებში X აღნიშნავს შემავალ ალფავიტს, Z-გამომავალ ალფავიტს, ხოლო S-მდგომარეობათა სიმრავლეს. თითოეული მაგალითისთვის მოცემული იქნება მისი სიტყვიერი აღწერა, რომელიც წარმოადგენს ძირითად საფუძველს მდგომარეობათა სიმრავლის არჩევისათვის.

მაგალითი 1. მოცემულია: ორგანიზმი აღიგზნება ორი სტიმულით: „დადებითი“ და „უარყოფითი“. ორგანიზმი არ რეაგირებს უარყოფით სტიმულზე და რეაგირებს დადებითი სტიმულების მონაცვლეობაზე.

X={დადებითი სტიმული; უარყოფითი სტიმული}

Z={რეაქცია; რეაქციის არარსებობა}

S={რეაქცია ბოლო დადებით სტიმულზე, არ არის რეაქცია ბოლო დადებით სტიმულზე}

მიმდინარე მომენტში „რეაქცია ბოლო დადებით სტიმულზე“ მდგომარეობისას და შესასვლელზე „დადებითი სტიმულის“ შემთხვევაში გამოსასვლელზე გამოჩნდება „არ არის რეაქცია“, ხოლო შემდეგი მდგომარეობა იქნება „არ არის რეაქცია ბოლო დადებით სტიმულზე“. მიმდინარე მომენტში სისტემის „არ არის რეაქცია ბოლო დადებით სტიმულზე“ მდგომარეობაში ყოფნისას და შესასვლელზე „დადებითი სიმულის“ არსებობისას წარმოიქმნება გამოსასვლელი რეაქცია და

შემდეგი მდგომარეობა „რეაქცია ბოლო დადებით სტიმულზე“. შესავლელზე „უარყოფითი სტიმულის“ დროს წარმოიქმნება გამოსასვლელი „არ არის რეაქცია“ მიმდინარე მომენტში მდგომარეობაზე დამოუკიდებლად, ხოლო სისტემის მდგომარეობა არ შეიცვლება.

მაგალითი 2. მოცემულია: ინგლისურ ტექსტში, შედგენილი ალფავიტის 26 ასოსგან და ჰარისაგან („პრობელებისაგან“), უნდა დავთვალოთ სიტყვათა რეაოდენობა, რომელიც იწყება un-ით და მთავრდება d-თი (მაგ.: united და სხვ.) სიმარტივისთვის პრობელი აღვნიშნოთ Π ასოთი, ხოლო ყველა დანარჩენი ასო d, n და u -საან განსხვავებული λ ასოთი.

X={ d, n, u, Π, λ }

Z={დავთვალოთ; არ დავთვალოთ}

S={ახალი სიტყვა, ველოდოთ ახალ სიტყვას, u-ს გამოჩენა, u-n-ის გამოჩენა,

u-n-d-ს გამოჩენა}

შესასვლელზე Π-ის გამოჩენისას შემდეგ მომენტში, არსებული მდგომარეობისაგან დამოუკიდებლად, გვექნება მდგომარეობა „ახალი სიტყვა“. მიმდინარე მომენტში „u-n-d-ს გამოჩენა“ მდგომარეობისას და შესასვლელზე Π-ის გამოჩენისას წარმოიშვება გამოსასვლელი „დავთვალოთ“. ყველა დანარჩენ შემთხვევაში გამოსასვლელზე წარმოიშვება „ არ დავთვალოთ“. მიმდინარე მდგომარეობა თუ არის „ახალი სიტყვა“, ხოლო შესასვლელზე u, მაშინ შემდეგი მდგომარეობა იქნება „u-ს გამოჩენა“, ხოლო შესასვლელზე d, n ან λ-ს შემთხვევაში შემდეგი მდგომარეობა იქნება „ველოდოთ ახალ სიტყვას“. მიმდინარე მომენტში მდგომარეობა თუ არის „u-ს გამოჩენა“, ხოლო შესასვლელზე-n, შემდეგ მომენტში მდგომარეობა იქნება „u-n-ის გამოჩენა“, ხოლო შესასვლელზე d, u ან λ-ს შემთხვევაში, შემდეგ მომენტში მდგომარეობა იქნება „ველოდოთ ახალ სიტყვას“. თუ მიმდინარე მომენტში მდგომარეობაა „u-n-ის გამოჩენა“ ან „u-n-d-ის გამოჩენა“, d შესასვლელის შემთხვევაში, მაშინ შემდეგ მომენტში გვექნება მდგომარეობა „u-n-d-ის გამოჩენა“, ხოლო n და u ან λ შესავლელის შემთხვევაში მდგომარეობა „u-n-ის გამოჩენა“. მდგომარეობა ველოდაოთ ახალ სიტყვას, Π-გან განსხვავებული შესასვლელის შემთხვევაში, დარჩება უცვლელი.

მაგალითი 3. მოცემულია: ბორბლის ბრუნვა, რომელიც მოძრაობაში მოჰყავს ძრავას, განისაზღვრება ორპოზიციანი ქანჩით: ქანჩის მარჯვენა მდგომარეობა

შეესაბამება საათის ისრის მიმართულებით ბრუნვას, მარცხენა კი საათის ისრის საწინააღმდეგოდ. ბრუნვის მიმართულების შეცვლის მომენტში ჩაირთვება ინდიკატორული ნათურა.

X={ მარჯვნივ, ნარცხნივ }

Z={ნათურა ჩართულია, ნათურა გამორთულია}

S={საათის ისრის მიმართულებით, საათის ისრის საწინააღმდეგოდ}

მიმდინარე მომენტში მდგომარეობა თუ იმყოფება „საათის ისრის მიმართულებით“, ხოლო შესავლელზე გვაქვს „მარჯვნივ“ ან მდგომარეობა თუ არის „საათის ისრის საწინააღმდეგო მიმართულებით“, ხოლო შესასვლელზე „მარცხნივ“, მდგომარეობა გვექნება უცვლელი და გვექნება გამოსასვლელი „ნათურა გამორთულია“. მიმდინარე მომენტში მდგომარეობა თუ იმყოფება „საათის ისრის მიმართულებით“, ხოლო შესასვლელზე გვაქვს „მარცხნივ“, ან პირიქით „საათის ისრის საწინააღმდეგომ მიმართულებით“ მდგომარეობა, ხოლო სესასვლელზე „მარჯვნივ“-მდგომარეობა შეიცვლება და გამოსასვლელზე გვექნება „ნათურა ჩართულია“.

განხილული მაგალითები გვიჩვენებს, რომ სასრულ ავტომატად შეიძლება წარმოვიდგინოთ თამაში, ენა, ალგორითმი, გადამრთველი, ცოცხალი ორგანზმი, ოგანიზაცია და ა.შ. არსებითად, ნებისმიერი სისტემა, რომელშიც მიმდინარე მომენტში გამომავალი სიგნალი დამოკიდებულია მიმდინარე მომენტში მდგომარეობაზე და შემავალ სიგნალზე, ხოლო მდგომარეობა შემდეგ მომენტში დამოკიდებულია მიმდინარე მომენტში მდგომარეობაზე და შემავალ სიგნალზე, შეიძლება წარმოვადგინოთ სასრულ ავტომატად.

ხდომილებათა წარმოდგენა ავტომატებში

აბსტრაქტული ავტომატის მუშაობის პირობებიდან გამომდინარე, შეიძლება მოგვეჩვენოს, რომ ავტომატური დროის $t=1,2,\ldots$ ყოველ მომენტში შემავალი $x(t)$ სიგნალის მოსვლისა და გამომავალი $y(t)$ სიგნალის გამოსვლის მოთხოვნა არ სრულდება რეალური ავტომატების შემთხვევაში, მაგრამ იგი შეიძლება ადვილად დავაკმაყოფილოთ, თუ შემოვიღებთ სპეციალურ e ასოს, რომელსაც ცარიელი ასო ეწოდება და მას განვიხილავთ შემავალი და გამომავალი ალფავიტების ასოების თანასწორად.

ავტომატის შესასვლელზე e ასოს გამოჩენა ნიშნავს, რომ ავტომატის შესასვლელზე სინამდვილეში არაფერი არ მიეწოდა, ხოლო გამოსასვლელზე ცარიელი ასოს გამოჩენა ნიშნავს ავტომატის გამოსასვლელზე რომელიღაც სიგნალის არარსებობას.

ვთქვათ X და Y შესაბამისად შესასვლელი გამოსასვლელი ალფავიტებია ნებისმიერი აბსტრაქტული ავტომატისა, ხოლო σ(x) და σ(y) ამ ალფავიტების შესასვლელი და გამოსასვლელი სიტყვების სიმრავლე. ვიგულისხმოთ, რომ აბსტრაქტული ავტომატი ინდუცირებს ნაწილობრივ f ასახვას σ(x) სიმრავლისა σ (y) სიმრავლეში.

f ასახვის განმარტებიდან გამომდინარეობს, რომ იგი აკმაყოფილებს პირობებს:

1. ნებისმიერ დასაშვებ შემავალ p$\in \sigma$ (x) სიტყვას f ასახვა შეუსაბამებს f(p)$\in \sigma$(y) გამომავალ სიტყვას, რომლის სიგრძე p სიტყვის სიგრძის ტოლია.

2. თუ p არის დასაშვები შემავალი სიტყვა, ხოლო p_1 კი p სიტყვის დასაწყისის რაიმე მონაკვეთი, მაშინ $f(p_1)$ არსებობს და თანხვდება f(p) სიტყვის საწყის მონაკვეთს.

აღნიშნულ პირობებს ეწოდება ასახვის ავტომატურობის პირობები, ხოლო ნებისმიერი ალფავიტური ასახვა, რომლებიც ამ პირობებს აკმაყოფილებს, წარმოადგენს ავტომატურ ასახვას.

ერთი შეხედვით გვეჩვენება, რომ ავტომატურობის პირობები აბსტრაქტული ავტომატების მიერ ინდუცირებულ ალფავიტურ ასახვებზე ადებს მკაცრ შეზღუდვებს. მაგალითად, ნებისმიერი (ცხადია ცალსახა) ალფავიტური ასახვისათვის შემავალი და გამომავალი სიტყვების ტოლობა როგორც წესი, არ სრულდება. მიუხედავად ამისა, ნებისმიერი ალფავიტური ასახვა შეიძლება გადაიქცეს ავტომატურ ასახვად შემდეგი სტანდარტული წესის გამოყენებით:

ვიგულისხმოთ, რომ f არის ნაწილობრივი ასახვა σ (x) სიტყვათა სიმრავლისა σ(y) სიმრავლეში. x და y ალფავიტებში შევიტანოთ ცარიელი e ასო, რომელიც თავდაპირველად მათში არაა და შევქმნათ ახალი ალფავიტები x'=x∪{e} და y'=y∪{e}. განვიხილოთ ნებისმიერი p$\in \sigma$ (x) სიტყვა, რომლის სიგრძეა n და რომელსაც f ასახვა შეუსაბამებს r=f(p) სიტყვას, რომლის სიგრძეც m -ის ტოლია. აღვნიშნოთ p'$\in \sigma$ (x')-ით სიტყვა x' ალფავიტში, რომელიც მიიღება p სიტყვისგან მარჯვნივ m რაოდენობის e ასოს მიწერით, ხოლო r'$\in \sigma$(y')-ით აღვნიშნოთ სიტყვა y' ალფავიტში, რომელიც მიიღება R სიტყვისაგან მარცხნივ n რაოდენობის e ასოს მიწერით. შედეგად მივიღებთ ერთი და იმავე m+n სიგრძის p' და r' სიტყვებს. ეს მეთოდი განმეორდება თითოეული p'_i სიტყვისათვის რომელზედაც განსაზღვრულია f' ასახვა და ვიგულისხმოთ, რომ

$p'_i \in \sigma$(x'). მივიღებთ ახალ ნაწილობრივ r'=f'(p') f' ასახვას σ (x') სიმრავლისა σ (y') სიმრავლეში .

განვაზოგადოთ f' ასახვა ნებისმიერი p'$\in \sigma$ (x') სიტყვის ყველა საწყის p'_i მონაკვეთებზე. ამ ოპერაციას ეწოდება f' ასახვის შეკრება. მისი აზრი შემდეგში მდგომარეობს: თუ p'_1 არის p' სიტყვის რაღაც მონაკვეთი f' ასახვის განსაზღვრის არიდან, მაშინ f'(p'_1) ტოლია f'(p') სიტყვის საწყისი მონაკვეთისა, რომლის სიგრძე მონაკვეთის სიგრძის ტოლია. f' ასახვის მიმართ შეერთების ოპერაციის გამოყენების შედეგად ვიღებთ ახალ f'' ასახვას, რომლის განსაზღვრის არე აკმაყოფილებს სისრულის პირობებს. თუ f'' ასახვა არის ცალსახა ასახვა, მაშინ ცხადია, რომ იგი აკმაყოფილებს ავტომატურობის პირობებს და მაშასადამე წარმოადგენს საძიებელ ავტომატურ ასახვას, მაგრამ მთელ რიგ შემთხვევებში ასახვის შევსება არაცალსახაა, რადგანაც ერთიდაიგივე სიტყვა შეიძლება შედიოდეს საწყისი მონაკვეთის სახით რამდენიმე განსხვავებულ სიტყვაში. ვაჩვენოთ, რომ ჩვენს მიერ აგებული f'' ასახვა ცალსახაა.

ვთქვათ p'=pe... არის ნებისმიერი სიტყვა σ(x')-დან, ამავე დროს p$\in \sigma$(x) სიტყვას აქვს სიგრძე და არ შეიცავს e ასოს, ხოლო r'$\in \sigma$(y') არის p'-ის სახე ასახვის მიმართ, რომელსაც აქვს სახე r'=e...er, სადაც r$\in \sigma$(y) არის m სიგრძის სიტყვა, რომელიც არ შეიცავს e ასოს. p'_1 იყოს p' სიტყვის ნებისმიერი საწყისი მონაკვეთი. განვიხილოთ ორი შესაძლო შემთხვევა.

1. დავუშვათ, რომ p'_1 შეიცავს ერთს მაინც e ასოს. მაშინ მას აქვს სახე p'=pe... . ასეთი სახის ნებისმიერი სიტყვის მოცემა ცალსახად განსაზღვრავს p სიტყვას და შესაბამისად p' სიტყვასაც. მაშასადამე p'_1 სიტყვა ემთხვევა p' სიტყვის მხოლოდ ერთ საწყის მონაკვეთს და თუკი p'_1 სიტყვის საწყისი მონაკვეთი სხვა g'=ge.... სიტყვისა, მაშინ რადგანაც g არ შეიცავს e ასოებს, p' და g' სიტყვები უნდა ემთხვეოდნენ ერთმანეთს. ამიტომ სახე განისაზღვრება ცალსახად.

დავუშვათ ახლა, რომ p'=pe... არ შეიცავს არცერთ e ასოს. ამ შემთხვევაში p'_1 სიტყვის სიგრძე არ აღემატება p სიტყვის სიგრძეს და სიგრძეთა გატოლების ოპერაციას. განსაზღვრის ძალით, r'_1=f''(p'_1) სიტყვის საწყისი მონაკვეთი შედგება მხოლოდ ცარიელი e ასოებისგან. ცხადია, რომ ასე იქნება არა მარტო p' სიტყვისათვის, არამედ ნებისმიერი სიტყვისათვის, რომლისთვისაც საწყის მონაკვეთს წარმოადგენს p' სიტყვა. ამიტომ არაცალსახობის წარმოშობა ამ შემთხვევაში გამორიცხულია. მაშასადამე, f'' ნაწილობრივი ასახვა, აგებული ნებიმიერი ალფავიტური ასახვიდან, თავისი აგების მეთოდიდან გამომდინარე აკმაყოფილებს ავტომატურობის პირობებს და წარმოადგენს ნაწილობრივ ავტომატურ ასახვას.

მოვიყვანოთ ახლა თეორემა ნებისმიერი ავტომატური ასახვისათვის აბსტრაქტული ავტომატის არსებობის შესახებ.

თეორემა 1: ნებისმიერი ნაწილობრივი ავტომატური f ასახვა შეიძლება ინდუცირებული იქნას რაიმე აბსტრაქტული ავტომატის საშუალებით.

დამტკიცება: ვთქვათ x={x_1, x_2, ... , x_m} და y={y_1, y_2, ... , y_n} შესაბამისად შემავალი და გამომავალი ალფავიტებია ნაწილობრივი ავტომატური f ასახვისა, რომელიც σ(x) სიმრავლეს ასახავს σ(y) სიმრავლეში. ავაგოთ პირველი გვარის აბსტრაქტული A ავტომატი, რომლის შესასვლელი ალფავიტია x, გამომავალი ალფავიტი y, ხოლო მდგომარეობათა სიმრავლეს წარმოადგენს შემავალი ალფავიტის ყველა დასაშვებ სიტყვათა სიმრავლე, რომელსაც აღვნიშნავთ Q-თი. A ავტომატის საწყის მდგომარეობად ავიღოთ ცარიელი e სიტყვა (e∈Q). A ავტომატის გადასვლის φ(q,x) ფუნქცია განვსაზღვროთ შემდეგნაირად: თუ q_k∈Q ავტომატის ∀ მდგომარეობაა (დასაშვები სიტყვა), ხოლო x_i ნებისმიერი შემავალი ასოა, მაშინ ჩავთვლით, რომ φ(q_k, x_i) ტოლია $q_k x_i$=q_e სიტყვისა, თუ q_e ∈Q (q_e დასაშვები სიტყვაა) და რომ φ(q_k, x_i) არაა განსაზღვრული საწინააღმდეგოდ q_e ∉Q. გამოსავლის ψ(q,x) ფუნქცია, რომლითაც მოიცემა A ავტომატის გამოსასვლელი y∈Y ასოები, განვსაზღვროთ თანაფარდობით ψ(q_k, x_i)= y_j , სადაც y_j არის $f(q_k, x_i)$ სიტყვის ბოლო ასო. ასე რომ, აგებულია აბსტრაქტული A=(x, Q, Y, e ∈ Q, F(x∈X, y∈Y)) ავტომატი, რომელიც ინდუცირებს f ასახვას.

იმ შემთხვევაში, როდესაც f ასახვის განსაზღვრის არე სასრულია, A ავტომატი სასრულია, წინააღმდეგ შემთხვევაში კი უსასრულო. ადვილი დასანახია, რომ გეომეტრიული ინტერპრეტაციისას აბსტრაქტული ავტომატი მოიცემა სასრული ან უსასრულო დატვირთული წინარე ხით, რომლის ფუძეა e .

გადავიდეთ ახლა ავტომატური ასახვის ფორმებზე, რომელთა მოცემაც შესაძლებელია ხდომილებათა საშუალებით.

ვთქვათ X={x_1, x_2, ... , x_m} ნებისმიერი ალფავიტია, ხოლო σ (x) ყველა სიტყვათა სიმრავლე ამ ალფავიტში. σ(x) სიმრავლის ნებისმიერ ქვესიმრავლეს ეწოდება ხდომილება X ალფავიტში.

ვთქვათ A არის აბსტრაქტული ავტომატი X={x_1, x_2, ... , x_m} შემავალი ალფავიტითა და გამომავალი y={y_1, y_2, ... , y_n} ალფავიტით. ავტომატი ინდუცირებს ნაწილობრივ f ასახვას σ (x) სიმრავლისა σ (y) სიმრავლეში.

მოვიყვანოთ ახლა თეორემა ნებისმიერი ავტომატური ასახვისათვის აბსტრაქტული ავტომატის არსებობის შესახებ.

თეორემა 2: ნებისმიერი ნაწილობრივი ავტომატური f ასახვა შეიძლება ინდუცირებული იქნას რაიმე აბსტრაქტული A ავტომატის საშუალებით.

დამტკიცება: ვთქვათ X={x_1, x_2, ... , x_m} და y={y_1, y_2, ... , y_n} შესაბამისად შემავალი და გამომავალი ალფავიტებია ნაწილობრივი ავტომატური f ასახვისა, რომელიც σ(x) სიმრავლეს ასახავს σ(y) სიმრავლეში. ავაგოთ პირველი გვარის აბსტრაქტული A ავტომატი, რომლის შესასვლელი ალფავიტია x, გამომავალი ალფავიტი y, ხოლო მდგომარეობათა სიმრავლეს წარმოადგენს შემავალი ალფავიტის ყველა დასაშვებ სიტყვათა სიმრავლე, რომელსაც აღვნიშნავთ Q-თი. A ავტომატის საწყის მდგომარეობად ავიღოთ ცარიელი e სიტყვა . A ავტომატის გადასვლის φ(q,x) ფუნქცია განვსაზღვროთ შემდეგნაირად: თუ q_k∈Q ავტომატის ∀ მდგომარეობაა (დასაშვები სიტყვა), ხოლო x_i ნებისმიერი შემავალი ასოა, მაშინ ჩავთვლით, რომ φ(q_k, x_i) ტოლია $q_k x_i$=q_e სიტყვისა, თუ q_e∈Q (ე.ი. q_e დასაშვები სიტყვაა) და რომ φ(q_k, x_i) არაა განსაზღვრული საწინააღმდეგოდ (e. i. q_e∉Q). გამოსავლის ψ(q,x) ფუნქცია, რომლითაც მოიცემა A ავტომატის გამოსასვლელი y∈Y ასოები, განვსაზღვროთ თანაფარდობით ψ(q_k, x_i)= y_j, სადაც y_j არის $f(q_k, x_i)$ სიტყვის ბოლო ასო. ასე რომ, აგებულია აბსტრაქტული A=(x,Q,Y,e∈Q, F(x∈X, y∈Y)) ავტომატი, რომელიც ინდუცირებს f ასახვას.

ვთქვათ A არის აბსტრაქტული ავტომატი X={x_1, x_2, ... , x_m} შემავალი ალფავიტითა და გამომავალი y={y_1, y_2, ... , y_n} ალფავიტით. ავტომატი ინდუცირებს ნაწილობრივ f ასახვას σ (x) სიმრავლისა სიმრავლეში. R_j ხდომილება, რომელიც A ავტომატში წარმოდგენილია y_j, j =1,2,...,n გამომავალი სიგნალით, ეწოდება ამ ავტომატის ყველა P∈σ(x) სიტყვის ერთობლიობას (სიმრავლეს), რომელთათვისაც f(p) სიტყვა განსაზღვრულია და მთავრდება y_j ასოთი. თუ N⊆Y არის გამომავალი სიგნალების რაღაც ქვესიმრავლე, მაშინ ხდომილებას, რომელიც წარმო-დგენილია A ავტომატში N სიმრავლით, ეწოდება ხდომილებათა გაერთიანება წარმოდგენილი ამ სიმრავლის ყველა ელემენტით. იმ შემთხვევაში, როცა N ემთხვევა Y ალფავიტს, მის შესაბამის ხდომილებას ეწოდება A ავტომატის R_1, R_2, ... , R_n კანონიკურ ხდომილებათა სიმრავლე.

თეორემა 3: ნებისმიერი აბსტრაქტული A ავტომატისთვის X={x_1, x_2, ... , x_m} შემავალი და y={y_1, y_2, ... , y_n} გამომავალი ალფავიტებით ნაწილობრივი ავტომატური F ასახვა მოიცემა δ(x) სიმრავლისა σ (y) სიმრავლეში ეკვივალენტურია მოცემული ავტომატის R_1, R_2, ... , R_n კანონიკურ ხდომილებათა სიმრავლის მოცემისა.

ამ თეორემიდან გამომდინარეობს, რომ ნებისმიერი ავტომატური ასახვა შეიძლება მოცემული იქნეს შემავალი ალფავიტის ყველა სიტყვათა σ (x) სიმრავლის დაყოფით სასრული რაოდენობის წყვილწყვილად არაგადამფარავ ხდომილებებად. ხდომილებების სასრულ ან უსასრულო კლასების ეფექტურად აღწერისთვის განვიხილოთ ე.წ. ხდომილებათა ალგებრა.

როგორც ცნობილია, ალგებრა ეწოდება ელემენტთა ყოველგვარ M სიმრავლეს, რომელზედაც განსაზღვრულია n -არული ალგებრულ ოპერაციათა სასრული ან უსასრულო სისტემა. M-ზე მოცემულია n -არული ალგებრული ω ოპერაცია, ნიშნავს, რომ ნებისმიერი n ელემენტისთვის g_1, g_2, … , g_n - დან მივუთითოთ ელემენტი რომელიც წარმოადგენს g_1, g_2, … , g_n∈M -ელემენტებზე ω ოპერაციის შედეგს. ეს შემდეგნაირად ჩაიწერება: g=ω (g_1, g_2, … , g_n). ω ოპერაცია შეიძლება განსაზღვრული არ იყოს ყველა g_1, g_2, … , g_n∈M ერთობლიობისთვის ე.წ. ნაწილობრივ ალგებრაში, მაგრამ ω ოპერაციის მიმართ ცალსახობის მოთხოვნა აუცილებელია.

ხდომილებათა ალგებრა X ალფავიტში ეწოდება ამ ალფავიტში ყველა ხდომილებათა სიმრავლეს, რომელზედაც მოცემულია სამი ოპერაციის სისტემა: ორი ბინარული ოპერაცია, დიზუნქცია და გამრავლება, და ერთი უნარული ოპერაცია - იტერაცია.

R და S ხდომილებათა დიზიუნქცია ეწოდება P ხდომილებას, რომელიც აღინიშნება P=RVS და წარმოადგენს R და S ხდომილებათა თეორიულ-სიმრავლურ გაერთიანებას.

R და S ხდომილებათა ნამრავლი ეწოდება U=R S სიმრავლეს, რომელიც შედგება u= r s სახის ყველა სიტყვისაგან, სადაც u∈U, r∈R, s∈S. შესაბამისად, U სიტყვა წარმოიქმნება, თუ R სიტყვას მარჯვნივ მივაწერთ S სიტყვას და არა პირიქით.

R ხდომილების იტერაცია ეწოდება ხდომილებას, რომელიც აღინიშნება {R}* და განისაზღვრება როგორც დიზუნქცია ცარიელი e სიტყვისა, R ხდომილებისა, R R ხდომილებისა, R R R ხდომილებისა და ა.შ. უსასრულობამდე. ე.ი.

{R}*= e ⋁ R ⋁ R R ⋁ R R R ⋁…

მაგალითის სახით განვიხილოთ ალფავიტი $X = \{x_1, x_2\}$. ვთქვათ მოცემულია ხდომილებები R={x_1, x_2x_1} და S={x_2x_2, x_1x_2} X ალფავიტში. განსაზღვრის მიხედვით R ⋁ S, R S, {R}* ხდომილებები:

R ⋁ S={$x_1, x_2x_1, x_2x_2, x_1x_2$},

R S= {$x_1x_2x_2, x_1x_1x_2, x_2x_1x_2x_2, x_2x_1x_1x_2$},

{R}*={e, $x_1, x_2x_1, x_1x_1, x_1x_2x_1, x_2x_1x_1, x_2x_1x_2x_1, x_1x_1x_1, x_1x_2x_1x_1, x_2x_1x_1x_1, x_2x_1x_2x_1x_1,$
$x_1x_1x_2x_1, x_1x_2x_1x_2x_1, x_2x_1x_1x_2x_1, x_2x_1x_2x_1x_2x_1$,… }.

ერთელემენტიან ხდომილებას, ე.ი. ერთი სიტყვისგან შედგენილ ხდომილებას, აღვნიშნავთ ამ სიტყვის სიმბოლოთი. შევნიშნოთ, რომ e ხდომილება, რომელიც შედგენილია ცარიელი e სიტყვით, შედგება ნულოვანი სიგრძის ერთი სიტყვისაგან და დამხმარე როლს თამაშობს ავტომატების თეორიაში. შევთანხმდეთ, რომ არ

განვასხვავოთ ერთმანეთისგან ხდომილებები, რომლებიც ერთმანეთისაგან მხოლოდ ცარიელი e სიტყვით განსხვავდებიან. განვიხილავთ აგრეთვე ცარიელ ∅ ხდომილებას, რომელიც შედგება შემავალი ალფავიტის ასოთა ცარიელი სიმრავლისაგან და ამიტომაც წარმოადგენს ნებისმიერი ხდომილების ნაწილს.

ვთქვათ X={x_1, x_2, ... , x_m} ნებისმიერი ალფავიტია. შემოვიღოთ რეგულარული გამოსახულების ცნება X ალფავიტში შემდეგნაირად:

1. $x_1, x_2, \ldots, x_m$, e ∈ ∅ სიმბოლოები რეგულარული გამოსახულებებია.
2 . თუ R და S რეგულარული გამოსახულებებია, მაშინ ასეთივეა R ⋁ S, R S, {R}*.
3. არცერთი გამოსახულება არ არის რეგულარული თუ იგი არ არის მიღებული სასრული რაოდენობის 1 და 2 წესების გამოყენებით.

მაშასადამე, რეგულარული გამოსახულება არის ფორმულა ხდომილებათა ალგებრაში. ერთი და იგივე ხდომილება შეიძლება სხვადასხვანაირად იყოს გამოსახული ერთელემენტიანი ხდომილების და დიზუნქციის, გამრავლებისა და იტერაციის ოპერაციების საშუალებით.

იმ ხდომილებებს, რომლებთათვისაც არსებობენ რეგულარული გამოსახულებები, ეწოდებათ რეგულარული ხდომილებები. წინააღმდეგ შემთხვევაში ხდომილებას ეწოდება არარეგულარული.

ხდომილებათა ალგებრაში დიდი მნიშვნელობა ენიჭება რეგულარულ გამოსახულებათა ექვივალენტური გარდაქმნების კანონების დადგენას. განვიხილოთ დიზიუნქციის, გამრავლების და იტერაციის ოპერაციების ძირითადი თვისებები, რომლებიც გამომდინარეობს მათი განმარტებებიდან. ვთქვათ, R და S ნებისმიერი რეგულარული ხდომილებებია σ(x)-დან. მაშინ ადგილი აქვს შემდეგ თანაფარდობას:

P ⋁ ∅= ∅ ⋁ P,

P ∅= ∅ P=∅,

P e = e P = P,

{∅}* = e, {e}* = e.

იტერაციისა და დიზიუნქციის კომუტაციურობა:

P ⋁ R = R ⋁ P,

P {P}* = {P}* P.

დიზიუნქციისა და გამრავლების ასოციაციურობა:

$$P \vee (R \vee S) = (P \vee R) \vee S,$$

$$P\ (R\ S) = (P\ R)\ S.$$

გამრავლების მარცხენა და მარჯვენა დისტრიბუციულობა დიზიუნქციის მიმართ:

$$P\ (R \vee S) = (P\ R) \vee (P\ S),$$

$$(P \vee R)\ S = (P\ S) \vee (R\ S).$$

დიზუნქციისა და იტერაციის იდენპოტენციურობა:

$$P \vee P = P,$$

$$\{\{P\}^*\}^* = \{p\}^*.$$

იტერაციის დიზიუნქციური შთანთქმა: :

$$\{P\}^* \vee P = \{P\}^*.$$

იტერაციის მულტიპლიკაციური შთანთქმა:

$$\{P\}^*\ \{P\}^* = \{P\}^*.$$

იტერაციის აღსანიშნავად გამოყენებულ {}* სიმბოლოს ეწოდება იტერაციული ფრჩხილები. ხდომილებათა ალგებრაში ოპერაციათა შესრულების თანმიმდევრობისათვის გამოიყენება ჩვეულებრივი ფრჩხილები. მათი არარსებობის შემთხვევაში პირველად შესრულდება იტერაცია, შემდეგ გამრავლება და ბოლოს დიზიუნქცია.

რეგულარული გამოსახულების ციკლური სიღრმე ეწოდება ერთმანეთში ჩადგმული იტერაციული ფრჩხილების მაქსიმალურ რიცხვს. მაგალითად: $\{x_1\{x_2\{x_1\}^*x_1\}^*x_3\}^*\vee x_2$ გამოსახულების ციკლური სიღრმეა 3, ხოლო $(\{x_2\}^*\times\{x_1\ x_3\}^*)\times x_2\vee\{x_1\ \ x_2\}^*$ გამოსახულების ციკლური სიღრმეა 1. ყველა სასრულ გამოსახულებას აქვს ნულოვანი ციკლური სიღრმე.

რეგულარული ხდომილების ციკლური სიღრმის ქვეშ იგულისხმება მასში წარმოდგენილ რეგულარულ გამოსახულებათა მინიმალური ციკლური სიღრმე.

განვიხილოთ X=$\{x_1,\ x_2,\ x_3\}$ ალფავიტში რეგულარულ ხდომილებათა ზოგიერთი მაგალითი:

1. ხდომილება, რომელსაც ეწოდება უნივერსალური და შედგება X ალფავიტის ყველა ასოსაგან, შეიძლება ჩაიწეროს შემდეგი სახით: p = $\{x_1\vee x_2\vee x_3\}^*$

2. ხდომილება, რომელიც შეიცავს X ალფავიტის მხოლოდ სამ ასოიან სიტყვას, აქვს სახე: R = (x_1Vx_2Vx_3)×(x_1Vx_2Vx_3)×(x_1Vx_2Vx_3) .

3. ხდომილება, რომელიც შედგება ისეთი სიტყვებისაგან, რომელშიდაც ერთხელ მაინც გვხვდება x_1 x_2 x_1 მონაკვეთი, შეიძლება ჩაიწეროს შემდეგი ფორმით: S = {x_1Vx_2Vx_3}*×$x_1x_2x_1$×{x_1Vx_2Vx_3}*

4. ხდომილება, რომელიც შედგება ყველა იმ სიტყვისაგან, რომლებიც იწყება x_1 ან x_2 ასოთი და მთავრდება x_1 ან x_2 მონაკვეთით: T = (x_1Vx_3){ x_1Vx_2Vx_3}*x_1x_2 .

5. ხდომილება, რომელიც შედგება ყველა იმ სიტყვისაგან, რომელთა სიგრძე 2-ის ჯერადია: U = {(x_1∨x_2∨x_3)(x_1∨x_2∨x_3)}*

6. ისეთი სიტყვებისაგან შედგენილი ხდომილება, რომელთა საწყისია x_1 ან x_2 ასო, შემდეგ მოდის x_2 ან x_3-ისგან შედგენილი ორასოიანი სიტყვა და ბოლოს, დასასრული შეიცავს ერთ x_1 ასოს მაინც: Q = {x_1∨x_2}*×(x_2∨x_3)×(x_2∨x_3)× x_1{x_1}*.

ხდომილებათა ალგებრის განხილული ∨, ×, {}*ოპერაციებიდან გამომდინარეობს, რომ ყველა სასრული ხდომილება რეგულარულია. ეს გამომდინარეობს იქიდან, რომ ხდომილების ნებისმიერი სიტყვა გამოისახება ასოების ნამრავლით, ხოლო ნებისმიერი სასრული ხდომილება მისი შედგენილი სიტყვების დიზიუნქციით.

იტერაციის გამოყენება იწვევს უსასრულო რაოდენობის რეგულარულ ხდომილებათა გაჩენას. მაგრამ ადვილი შესამჩნევია, რომ უსასრულო ხდომილებათა უმრავლესობა არარეგულარულია. მართლაც, სიტყვათა სიმრავლე არაცარიელ სასრულ ალფავიტში თვლადია, მაშინ, როდესაც ყველა ხდომილებათა სიმრავლეს ასეთ ალფავიტში აქვს კონტინიუმის სიმძლავრე.

ავაგოთ არარეგულარული ხდომილების მაგალითი. ავირჩიოთ მთელ რიცხვთა უსასრულო ზრდადი მიდევრობა n_1, n_2, ...; n_i, ... ისე, რომ n_{i+1}-n_i (i=1,2,...) სხვაობა არაა შემოსაზღვრული ერთობლიობაში. ე.ი. ნებისმიერი რაგინდ დიდი N რიცხვისთვის მოიძებნება ისეთი i, რომ n_{i+1}-n_i>N. ამ პირობას აკმაყოფილებს, მაგალითად, ნატურალურ რიცხვთა რიგის კვადრატების მიმდევრობა, კუბების მიმდევრობა და ა.შ. ახლა ნებისმიერ შესასვლელ X ალფავიტში ავაგოთ S ხდომილება, რომელიც შედგება ყველა იმ სიტყვებისაგან, რომელთა სიგრძეებია n_1, n_2, ...; ვაჩვენოთ, რომ S აუცილებლად არარეგულარული ხდომილებაა. დავუშვათ საწინააღმდეგო და ჩავთვალოთ, რომ R არის რაღაც რეგულარული გამოსახულება S ხდომილებისა. რადგანაც S ხდომილება უსასრულოა, ეს გამოსახულება შეიცავს მხოლოდ ისეთ იტერაციულ ფრჩხილებს მაინც, რომელთა შიგნით მოთავსებული გამოსახულება განსხვავდება ცარიელი e სიტყვისაგან.

R გამოსახულებაში ყველა დანარჩენი იტერაციული ფრჩხილი შევცვალოთ ცარიელი სიტყვით, ხოლო გამოყოფილი ფრჩხილები {p}* გამოსახულებით, სადაც P ნებისმიერი არაცარიელი სიტყვაა S ხდომილებიდან. შედეგად მივიღებთ R_1 რეგულარულ გამოსახულებას რაღაც ხდომილებისათვის, რომელიც შედის S-ში. R_1--ის გამოსახულებიდან გამომდინარეობს, რომ S ხდომილებაში შედის rg, rpg, rppg, rpppg, … სახის სიტყვები, რომელთა სიგრძეც შეადგენს უსასრულოდ ზრდად არითმეტიკულ პროგრესიას. მაგრამ ეს ეწინააღმდეგება S ხდომილების აგების წესს. შესაბამისად, S ხდომილება არ შეიძლება წარმოდგენილი იქნეს არავითარი რეგულარული გამოსახულებით და იგი წარმოადგენს არარეგულარულ ხდომილებას.

რეგულარულ ხდომილებებთან დაკავშირებით სამართლიანია კლინის პირველი და მეორე თეორემა, რომლებიც შემდეგნაირად ჩამოყალიბდებიან:

თეორემა 1: ხდომილება, რომელიც წარმოდგენილია მილის ან მურის ნებისმიერ სასრულ ავტომატში გამომავალი სიგნალების ან მდგომარეობების რაიმე სიმრავლით, აუცილებლად რეგულარულია.

თეორემა 2: ნებისმიერი რეგულარული ხდომილება შეიძლება წარმოდგენილი იყოს მილის ან მურის სასრულ ავტომატში გამომავალი სიგნალების ან მდგომარეობების რაიმე სიმრავლით.

ეს ორი თეორემა შეიძლება გავერთიანოთ ერთ თეორემად:

თეორემა (კლინი): სასრულ ავტომატებში წარმოდგენილი ხდომილებათა კლასი ემთხვევა რეგულარულ ხდომილებათა კლასს.

ავტომატების ანალიზისა და სინთეზის ამოცანები

ავტომატების თეორიის ძირითად ამოცანებს წარმოადგენენ ავტომატების ანალიზისა და სინთეზის ამოცანები. ანალიზის ქვეშ იგულისხმება იმ ასახვის დადგენა, რომელიც რეალიზდება მოცემული A ავტომატის მიერ, ხოლო სინთეზის ქვეშ კი იგულისხმება ავტომატის სქემის აგება, რომელიც ინდუცირებს მოცემულ ფ ასახვას.

გრაფთა თეორიის მეთოდები საშუალებას იძლევა წარმატებით გადაიჭრას ავტომატების აბსტრაქტული და სტრუქტურული თეორიის ამოცანები. ავტომატების აბსტრაქტულ თეორიაში მნიშვნელოვანი როლი უკავით ავტომატების ანალიზისა და სინთეზის ამოცანებს, ხოლო სტრუქტურულ დონეზე პირველ ადგილზე დგას ავტომატების სინთეზის პრობლემა.

ავტომატების ანალიზისა და სინთეზის ამოცანები ურთიერთსაწინააღმდეგოა. განასხვავებენ ავტოამტების აბსტრაქტული ანალიზისა და სინთეზის კანონიკურ და ზოგად ამოცანებს.

ანალიზისა და სინთეზის კანონიკური ამოცანები ფორმულირდება შემდეგი სახით:

1) მილის ან მურის მოცემული A ავტომატისათვის ვიპოვოთ მისი შესაბამისი კანონიკურ ხდომილებათა M სიმრავლე

2) მოცემული ავტომატურ ხდომილებათა M სიმრავლის მიხედვით ვიპოვოთ მილის ან მურის A ავტომატი, რომლის კანონიკურ ხდომილებათა სიმრავლე ემთხვევა M სიმრავლეს.

ცნობილია, რომ კანონიკურ ხდომილებათა სიმრაველე ცალსახად განისაზღვრება აბსტრაქტული ავტომატის მიერ. ამ აზრით ავტომატების ანალიზის კანონიკური ამოცანა ცალსახაა. ამასთან ერთად, ერთი და იგივე ხდომილება შეიძლება წარმოდგენილი იქნას რამოდენიმე ეკვივალენტური ფორმის რეგულარული ხდომილებით. ამიტომ ანალიზის დროს მოითხოვება მოიძებნოს ისეთი რეგულარული გამოსახულება, რომელსაც გააჩნია მინიმალური ციკლური სიღრმე.

რაც შეეხება სინთეზის კანონიკურ ამოცანას, ის თავისი არსით არაცალსახაა. ეს აიხსენება იმით, რომ აბსტრაქტული ავტომატი ცალსახად არ განისაზღვრება მისი შესაბამისი კანონიკურ ხდომილებათა სიმრავლით, რადგანაც ერთი და იგივე ავტომატური ასახვა შეიძლება ინდუცირებული იყოს სხვადასხვა აბსტრაქტული ავტომატებით. ამიტომ სინთეზის კანონიკური ამოცანის ამოხსნისას ვპოულობთ ერთ შესაძლო ავტომატს, ხოლო შემდეგ ისმება ამოცანა აბსტრაქტული ავტომატების მინიმიზაციისა, რომელიც გულისხმობს ისეთი ავტომატის პოვნას, რომლის მდგომარეობათა რიცხვი მინიმალურია.

ავტომატების ანალიზისა და სინთეზის ზოგადი ამოცანები ფორმულირდება შემდეგნაირად:

1) მოცემული მილის ან მურის A აბსტრაქტული ავტომატით ვიპოვოთ ხდომილება, რომელიც წარმოდგენილია A ავტომატში გამოსასვლელი ასოების ნებისმიერი სიმრავლით ან ავტომატის მდგომარეობებით.

2) ნებისმიერი ხდომილებათა სასრული სიმრავლის მიხედვით ავაგოთ მილის ან მურის A ავტომატი, რომელიც ამ სიმრავლის ნებისმიერ ხდომილებას წარმოადგენს გამოსასვლელი ასოების რაღაც სიმრავლით ან მდგომარეობებით.

ადვილი დასნახია, რომ კანონიკური და ზოგადი ამოცანები ერთმანეთს ჰგავს.

სასრულ ავტოამატებში ხდომილებათა წარმოდგენის შესახებ სამართლიანია კლინის ფუნდამენტური თეორემა.

თეორემა (კლინი) : ხდომილებათა კლასი, წარმოდენილი სასრულ ავტომატებში, ემთხვევა რეგულარულ ხდომილებათა კლასს.

ეს თეორემა შედგება ორი თეორემისგან, რომელთაც ხშირად უწოდებენ კლინის პირველ და მეორე თეორემებს.

თეორემა (კლინი 1) : ხდომილება, წარმოდგენილი მილის ან მურის ნებისმიერ სასრულ ავტომატებში გამომავალი სიგნალების ნებისმიერი სიმრავლით ან მდგომარეობებით, აუცილებლად რეგულარულია.

თეორემა (კლინი 2) : ნებისმიერი რეგულარული გმაოსახულება შეიძლება წარმოდგენილი იქნას მილის ან მურის სასრულ ავტომატში გამომავალი სიგნალების რაიმე სიმრავლით ან მდგომარეობებით.

კლინის თეორემა წარმოადგენს ფუნდამენტურ თეორემას აბსტაქტული ავტოამტების თეორიაში, რადგანაც მისგან გამომდინარეობს, რომ რეგულარულ გამოსახულებათა ენა საკმარისია ნებისმიერი სასრული ავტომატის მიერ ინდუცირებული ასახვების აღწერისათვის.

რეგულარული გამოსახულების მინიმალური ფორმა

ავტომატების ანალიზის შედეგად მიიღება სხვადასხვა სახის რეგულარულ გამოსახულებათა ეკვივალენტური ფორმები წარმოდგენილი ავტომატში. ეს რეგულარული გამოსახულებები განსხვავებულები არიან თავიანთი სირთულით. ამიტომ წამოიჭრა რეგულარული გამოსახულების მინიმალური ფორმის პრობლემა.

რეგულარულ გამოსახულებას ეწოდება მინიმალური, თუ მას გააჩნია უმცირესი ციკლური S სიღრმე. ერთი და იგივე ციკლური სიღრმის რამოდენიმე რეგულარული გამოსახულების დროს მინიმალურ გამოსახულებად ჩაითვლება ის რეგულარული გამოსახულება, რომელსაც აქვს უმცირესი e სიგრძე, სადაც e არის ასოთა რაოდენობა, რომელიც შედის რეგულარულ გამოსახულებაში.

სასრული ავტომატის ანალიზის დროს მიიღება სასრული რაოდენობის რეგულარულ გამოსახულებათა ეკვივალენტური ფორმები. ამიტომ ერთი და იგივე ხდომილების რეგულარული გამოსახულების მინიმალური ფორმა შეიძლება მიღებული იქნეს ავტომატის გრაფოიდზე სასრული რაოდენობის ანალიზის ალგორითმის გამოყენებით.

იმისათვის, რომ მივიღოთ გამარტივებული (მინიმალურთან ახლო) ფორმა რეგულარული გამოსახულებისა ანალიზის ალგორითმის ერთჯერადი გამოყენების პროცესში, საჭიროა მთელი რიგის წესების გათვალისწინება.

შემოვიღოთ წინასწარ ზოგიერთი ცნება.

გრაფოიდს ეწოდება ტიპიური, თუ მას გააჩნია წყარო და შემკრები; სუსტად ტიპიური, თუ მას აქვს მხოლოდ წყარო და არატიპიური, თუ ის არ შეიცავს არც წყაროს და არც შემკრებს, ან აქვს მხოლოდ შემკრები.

გავითვალისწინოთ, რომ q_i წვეროში შესასვლელის ხარისხი არის იმ წვეროთა რაოდენობა, რომელთაგანაც q_i წვეროში შედის ერთი მაინც რკალი.

წესი 1. ტიპიური გრაფოიდისთვის, რომელსაც გააჩნია მხოლოდ პირდაპირი გზა, მინიმალური ფორმა რეგულარული გამოსახულებისა განისაზღვრება განტოლებათა სისტემაში უცნობთა (წვეროების) ნებისმიერი რიგით მიმდევრობით გამორიცხვის საშუალებით.

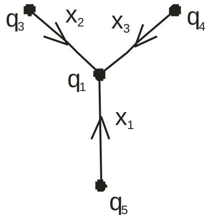
ასეთი ავტომატისათვის არსებობს მხოლოდ ერთი რეგულარული გამოსახულება.

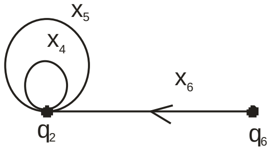
წესი 2. ტიპიური გრაფოიდისთვის, რომელსაც აქვს უკუკავშირის კონტური, უცნობთა გამორიცხვა იწყება იმ R_i უცნობით, რომელიც შეესაბამება უმცირესი შესასვლელის ხარისხის მქონე q_i წვეროს. შემდგომი გამორიცხვა უნდა მიმდინარეობდეს შესასვლელის ხარისხის ზრდადი მიმდევრობით.

წესი 3. სუსტად ტიპიური გრაფოიდისთვის უცნობთა გამორიცხვა უნდა დაიწყოს უმცირესი შესასვლელის მქონე q_i წვეროს შესაბამისი R_i უცნობით.

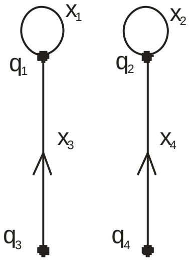
წესი 4. არატიპიური გრაფოიდისთვის, რომლის აღნიშნული წვერო არის შემკრები, უცნობთა გამორიცხვა იწყება მეორე წესით, როგორც ტიპიური გრაფოიდისთვის, ხოლო თუ აღნიშნული წვერო არის შემკრები, მაშინ მესამე წესით.

ზოგად შემთხვევაში შეიძლება აღმოჩნდეს, რომ სისტემის რამოდენიმე განტოლებაში შედის უცნობთა ერთიდაიგივე რიცხვი (გრაფოიდზე ეს შეესაბამება ერთდაიგივე შესასვლელის ხარისხის მქონე წვეროს არსებობას). მაშინ გაუგებარია თუ რომელი უცნობით (წვეროთი) უნდა იწყებოდეს გამორიცხვა. ასეთი შემთხვევის



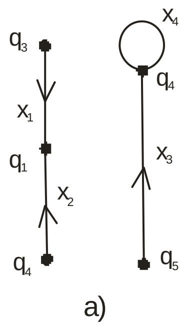


b)



g)

რაოდენობა სულ შეიძლება სამი აღმოჩნდეს. იმის მიხედვით, თუ რომელი წვეროა საწყისი, მივიღებთ შემდეგ სამ ვარიანტს. ამ ნახაზებზე მოცემულია რაღაც ავტომატის ნაწილები, რომელთათვისაც არცერთი წვერო არაა საწყისი მდგომარეობა.



a)

I. ა) ნახაზზე მოცემული გრაფოიდისთვის

$$\begin{cases} R_1 = R_3x_1 \vee R_4x_2 \\ R_2 = R_5x_3 \vee R_2x_4 \end{cases} \qquad (7)$$

საიდანაც

$R_2 = R_6x_6\{x_4 \vee x_5\}^*$

ამიტომ გამორიცხვა უნდა დაიწყოს R_2 უცნობით.

ბ) ნახაზზე მოცემული გრაფოიდისთვის

$$\begin{cases} R_1 = R_5x_1 \vee R_3x_2 \vee R_4x_3 \\ R_2 = R_2x_4 \vee R_2x_5 \vee R_6x_4 \end{cases} \qquad (8)$$

sadac

$R_2 = R_6x_6\{x_4 \vee x_5\}^*$

ამიტომ გამორიცხვა უნდა დაიწყოს R_2 უცნობით.

გ) ნახაზზე მოცემული გრაფოიდისთვის

$$\begin{cases} R_1 = R_1x_1 \vee R_3x_3 \\ \end{cases} \qquad (9)$$

$R_2 = R_2x_2 \vee R_4x_4$

და გამორიცხვა შეიძლება დაიწყოს ნებისმიერი მიმდევრობით.

დავუშვათ, რომ q_1 არის ავტომატის რაღაც ნაწილების საწყისი მდგომარეობა, მოცემული 6.12 ა), ბ), გ) ნახატებზე, მაშინ (7), (8), (9) განტოლებათა სისტემებს R_1-თვის დაემატება e ასო. ეს ართულებს R_1 განტოლებას. ამიტომ უცნობთა გამორიცხვა ყველა შემთხვევაში უნდა დაიწყოს R_2-ით.

დავუშვათ, რომ q_2 არის საწყისი წვერო (ნახ. 6.12 ა) ბ) გ)), მაშინ განტოლებათა სისტემა, რომლებიც აღწერს 6.12 ა) ბ) ნახაზზე მოცემულ გრაფოიდებს მიიღებს სახეს

$$\begin{cases} R_1 = R_3x_2 \vee R_4x_2, \\ R_2 = R_5x_3 \vee R_2x_4 \vee e \end{cases}$$

და

$$\begin{cases} R_1 = R_5x_1 \vee R_2x_4 \vee R_4x_3, \\ R_2 = R_2x_4 \vee R_2x_5 \vee R_6x_6 \vee e, \end{cases}$$

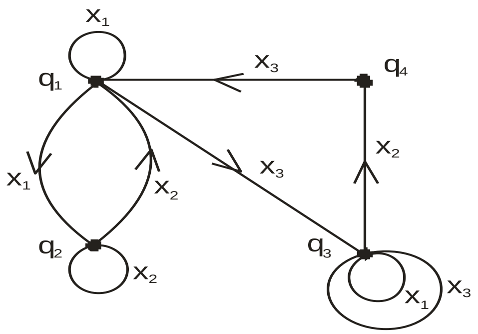
ხოლო მათი ამოხსნები შესაბამისად იქნება

$$R_2 = (R_5x_3 \vee e)\{x_4\}^*,$$

$$R_2 = (R_6x_6 \vee e)\{x_4 \vee x_5\}^*,$$

რომელთაგანაც უნდა დაიწყოს უცნობთა გამორიცხვა. 6.12 ნახატზე მოცემული გრაფოიდისათვის გამორიცხვა უნდა დაიწყოს R_1- ით, რადგან (9) განტოლება R_2–სთვის რთულდება e ასოს დამატებით.

მაგალითი. ვთქვათ მოცემულია A ავტომატი. (ნახ. 6.13.) ვიპოვოთ



nax.13

ხდომილების მინიმალური გამოსახულება წარმოდგენილი ავტომატში q_4 მდგომარეობით. შევადგინოთ გამოსახულებათა სისტემა :

$$\begin{cases} R_1 = R_1x_1 \vee R_2x_2 \vee R_4x_3 \vee e, \\ R_2 = R_1x_1 \vee R_2x_2 \\ R_3 = R_1x_3 \vee R_3x_3 \vee R_3x_1 \\ R_4 = R_3x_2 \end{cases}$$

მიმდევრობით გამოვრიცხოთ R_2, R_1, R_3 და განვსაზღვროთ R_4.

$R_2 = R_1x_1\{X_2\}^*$

$R_1 = R_1x_1 \vee R_1x_1\{x_2\}^*x_2 \vee R_4x_3 \vee e$

$R_1 = (R_4x_3 \vee e)\{x_1 \vee x_1\{x_2\}^*x_2\}^*$

$R_3 = (R_4x_3 \vee e)\{x_1 \vee x_1\{x_2\}^*x_2\}^*x_1\{x_2\}^*x_3 \vee R_3(x_3 \vee x_1)$

$R_4 = R_4x_3\{x_1 \vee x_1\{x_2\}^*x_2\}^*x_1\{x_2\}^*x_3\{x_3 \vee x_1\}^*x_2 \vee \{x_1 \vee x_1\{x_2\}^*x_2\}^*x_1\{x_2\}^*x_3\{x_3 \vee x_1\}^*x_2$

$R_4 = \{x_1 \vee x_1\{x_2\}^*x_2\}^*x_1\{x_2\}^*\{x_3 \vee x_1\}^*x_2\{x_3\}x_1 \vee x_1\{x_2\}^*\}^*x_1\{x_2\}^*x_3\{x_3 \vee x_1\}^*x_2.$

საძიებელი რეგულარული გამოსახულების სიგრძეა L=19, ხოლო ციკლური სიღრმე S=3.

ვიპოვოთ იგივე ხდომილების მინიმალური რეგულარული გამოსახულება, გამოვიყენოთ მე- 4 წესი და პირველ რიგში გამოვრიცხოთ R_4 უცნობი, შემდეგ R_2. შემდეგ კი მე-2 წესის მიხედვით გამოვრიცხავთ R_3 და R_1-ს, ვიპოვით R_4-ს:

$R_2 = R_1x_1\{x_2\}^*$

$R_3 = R_1x_1\{x_2\}^* \vee R_3x_3 \vee R_3x_1$

$R_3 = R_1x_1\{x_2\}^*\{x_3 \vee x_1\}^*$

$R_1 = R_1x_1 \vee R_1x_1\{x_2\}^* \vee R_1x_1\{x_2\}^*\{x_3 \vee x_1\}^*x_2x_3 \vee e$

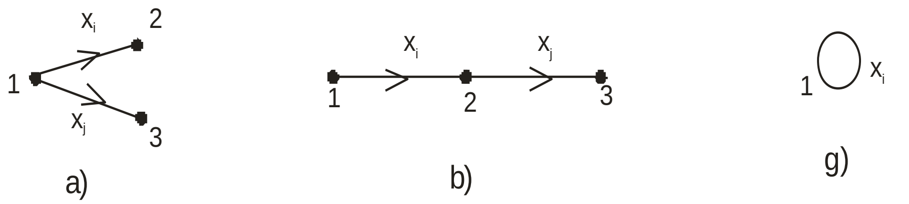
$R_1 = \{x_1 \vee x_1\{x_2\}^* \vee x_1\{x_2\}^*\{x_3 \vee x_1\}^*x_2x_3\}^*$

$R_4 = \{x_1 \vee x_1\{x_2\}^* \vee x_1\{x_2\}^*\{x_3 \vee x_1\}^*x_2x_3\}^*x_1\{x_2\}^*\{x_3 \vee x_1\}^*x_2$

ამ რეგულარული გამოსახულების სიგრძეა e =14, ხოლო ციკლური სიღრმე S=2.

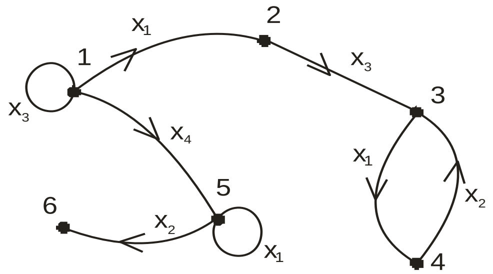
რეგულარული გამოსახულებების მოცემა გრაფების ფორმით

ნებისმიერი რეგულარული გამოსახულება შეიძლება წარმოვადგინოთ გრაფის სახით. ელემენტარული რეგულარული გამოსახულებების გრაფები (დიზიუნქცია $x_i V x_j$, , ნამრავლი $x_i x_j$, და იტერაცია $\{x_i\}^*$) ნაჩვენებია 6.14. ა,ბ,გ, ნახაზებზე.



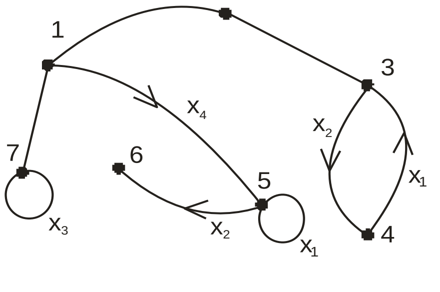
გრაფის საწყისი წვერო აღინიშნება 1-იანით, ხოლო დანარჩენები ნომრებით 2,3,4…სიმრავლიდან.

მაგალითად $R=x_1x_3\{x_2x_1\}^* \vee x_4\{x_1\}^*x_2 \vee \{x_3\}^*$ რეგულარულ გამოსახულებას აძლევს სახესსადაც საწყისია წვერო 1, ხოლო ბოლო – 3, 6 და 1 წვეროები. თითოეულ გზას



nax.6.15

რეგულარული გამოსახულების გრაფზე საწყისიდან ნებისმერ ბოლომდე უნდა შეესაბამებოდეს შემავალი ასოების მიმდინარეობა, რომელიც ეკუთვნის მოცემულ რეგულარულ გამოსახულებას და პირიქით, მიმდევრობა, რომელიც რეგულარული გამოსახულებით განსაზღვრავს გარკვეულ გზას რეგულარული გამოსახულების გრაფში. მაგრამ შესაძლოა რომ აგებულ გრაფში აღმოჩნდეს მცდარი მიმდევრობები, რომლებიც არაა მოცემული რეგულარული გამოსახულებით. მაგალითად, გზა გამოყოფილი მუქი ხაზებით 6.15 ნახ-ზე შეესაბამება $x_3x_4x_2$ სიტყვას, რომელიც არ ეკუთვნის R რეგულარულ გამოსახულებას. წარმოქმნილი მცდარი მიმდევრობები შეიძლება აღმოვფხვრათ გრაფზე ცარიელი ისრის შემოღებით, რომელიც აღნიშნავს მყისიერ გადასვლას გრაფის ერთი წვეროდან მეორეში. ცარიელი ისრის გამოყენებით 6.15 გრაფის შესწორებული გრაფის ნახატი მოცემულია 6.16 ნახ-ზე.



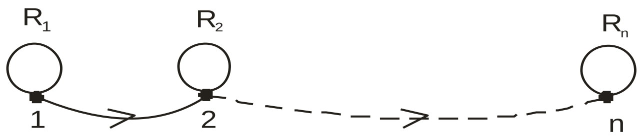
nax.6.16

მოვიყვანოთ წესები, რომელთა შემთხვევაშიც შესაძლებელია ცარიელი ისრების გამოყენება.

წესი 1. თუ გვაქვს ორი ან მეტი იტერაციის ნამრავლი

$$S = \prod \{R_i\}^*, \ i = \{1,2,\ldots,n\} ,$$

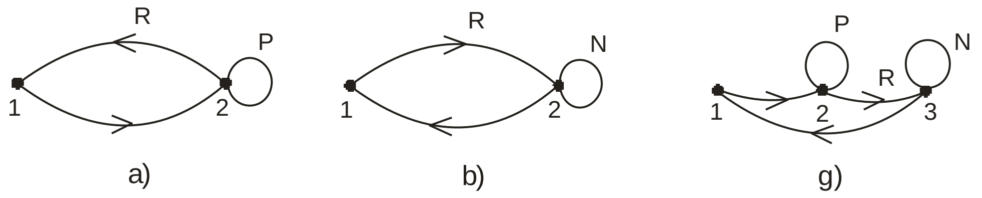
სადაც R_i - ნებისმიერი რეგულარული გამოსახულებაა. გეომეტრიული ინტერპრეტაცია მოცემულია ნახ. 6.17 - ზე.



nax.6.17

წესი 2. რეგულარულ გამოსახულებაში იტერაციულ ფრჩხილში დაწყება ან დამთავრება ხდება იტერაციით

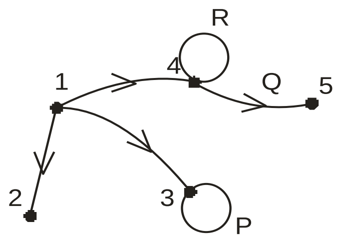
a) $S = \{ \{P\}^* R \}^*$
b) $S = \{ R \{N\}^* \}^*$
g) $S = \{ \{P\}^* R \{N\}^* \}^*$



ნახ.6.18

სადაც P, R, N – ნებისმიერი რეგულარული გამოსახულებებია

წესი 3. თუ რეგულარულ გამოსახულებაში ერთი მაინც დიზიუნქციური წევრი იწყება იტერაციით (ნახ. 6.19)

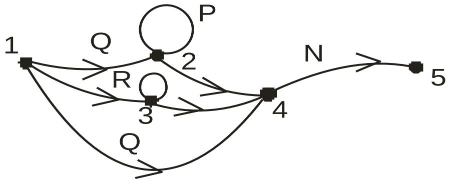


nax.6.19

S = {R}* Q V {P}*…V Q,

სადაც Q რეგულარული გამოსახულებაა, რომელიც არ შეიცავს იტერაციულ ფრჩხილებს.

წესი 4. თუ რეგულარულ გამოსახულებაში მარცხენა თანამამრავლი წარმოადგენს დიზიუნქციას, რომლის ერთი მაინც წევრი მთავრდება იტერაციით (ნახ. 6.20)



nax.6.20

S = (Q {P}* V {R}* V…V Q)N

შევნიშნოთ, რომ კერძო შემთხვევებში P, R, N, Q რეგულარული გამოსახულებები შეიძლება შედგებოდეს ერთი ასოსაგან.

1-4 სისტემა, ცარიელი ისრების შემოღების რეგულარული გამოსახულების გრაფში მცდარი გზების უკუსაგდებად, არის სრული სისტემა.

ენები და დაპროგრამების სისტემები

გამოთვლითი ტექნიკის განვითარების ქრონოლოგია

ზოგიერთი კომპიუტერული ფირმის მოკლე ისტორია

დაპროგრამების ენები მიეკუთვნებიან ფორმალურ ენათა კლასს. მათი დანიშნულებაა მონაცემთა (ინფორმაციის) აღწერა და მათი დამუშავება გამომთვლელ მანქანაზე. დაპროგრამების ენების თეორიულ საფუძვლებს შეადგენენ ალგორითმული სისტემები (რეკურსიული ფუნქციები, ნორმალური ალგორითმები და სხვ.) რომელთა დანიშნულებაცაა ალგორითმების აღწერა.

პირველი დაპროგრამების ენა იყო მანქანური ენები, რომლებიც ძალიან დაბალი დონის იყო. მათი დამახასიათებელი თავისებურება მდგომარეობდა ბრძანებების ციფრულ კოდირებაში. ამიტომ მანქანურ ენაში არ არსებობდა განსხვავება მონაცემების სიგა წარმოდგენასა და ბრძანებებისა, რომელთა საშუალებითაც აღიწერებიან პროგრამები. უფრო მაღალი დონის ენებში ასეთი თავისებურება აღარაა.

განასხვავებენ დაპროგრამების ენების რამდენიმე დონეს, რომელთა შორისაა ძირითადად მანქანურად-ორიენტირებული (ავტოკოდი), პროცედურულად ორიენტირებული, პრობლემურად-ორიენტირებული ენები და საერთო დანიშნულების ენები.

ავტოკოდები, (ანუ ენები ერთი-ერთზე [1:1]) გამოიყენება ბრძანებათა მისამართებისა და ოპერაციათა კოდების შეცვლით მათი სიმბოლური აღნიშვნით, ხოლო უფრო განვითარებულ (მაკროენებში) – გამომთვლელ მანქანებში ელემენტარული ოპერაციების გაფართოებისათვის.

სიმბოლური კოდირების ენებმა გამოყენება ჰპოვეს მომდევნო თაობის მანქანებში, რამაც გაამარტივა პროგრამირების პროცესი მეხსიერების განაწილების ავტომატიზაციის ხარჯზე. სიმბოლური კოდირების ენები საბაზო ენებია ოპერაციულ სისტემებში. მათ გამოიყენებენ პროგრამების აწყობაში, ამიტომაც მათ ასევე უწოდებენ ასემბლერულ ენებს ანუ ამწყობ ენებს.

ამოცანების სპეციფიკამ განაპირობა შექმნილიყო პროცედურულად ორიენტირებული ენები. ამ ენების განმასხვავებელი თავისებურება მდგომარეობს იმ კლასის ობიექტების გამოყოფაში, რომლებიც ექვემდებარებიან დამუშავებას, რთული გამოსახულებების (არითმეტიკული, ბულის, ტექსტური და სხვ.) გამოყენებაში,

ოპერატორებში, რომლებიც უზრუნველყოფენ პროგრამების ჩაწერის მოხერხებულობას (ფუნქციების, ციკლური გამოთვლების, პროცედურების და სხვ.)

დაპროგრამების ენებს შორის, რომლებიც ორიენტირებულია გამოთვლითი და სამენიერო ამოცანების კლასზე, უფრო ფართო გავრცელება ჰპოვა ენამ "ფორტრანმა".

1960 წლისათვის არსებობდა ასზე მეტი სხვადასხვა ორიენტაციის დაპროგრამების ენები და სისტემები. ამავე წელს შეიქმნა საერთაშორისო დაპროგრამების ენა ალგოლ-60. პროგრამირებაში საერთო იდეების განვითარებაში არსებითი გავლენა მოახდინა ალგოლ-60-ის სინტაქსის ფორმალურმა აღწერამ კონტექსტურად თავისუფალი ენების საშუალებით.

იმ ენებს შორის, რომლებიც ეკონომიკური ინფორმაციის დასამუშავებლადაა განუთვნილი, მთავარი ადგილი უჭირავს კობოლს. ენათა კლასს, რომლებიც გათვალისწინებულია ფორმულების ანალიზური გარდაქმნისათვის, დიფერენცირებისათვის, გამოსახულებათა ინტეგრებისათვის, ლინგვისტური ტექსტების დამუშავებისათვის და ა.შ. მიეკუთვნება ენები: ფორმაკი, ფორმულა-ალგოლ, სირიუსი და სხვ.

ჩანაწერებისა და სტრიქონების დამუშავებისათვის უფრო ცნობილია კომიტი, სნაბოლი, იპლ-V, ლისპ, ლისპ-2 ენები. ამ კლასის ენები მოსახერხებელია ლინგვისტური ტექსტების, ალგებრული გამოთვლების, ხელოვნური ინტელექტის კვლევის, თეორემების დამტკიცების, ავტომატიზაციის დამუშავებასა და ანალიზში.

პროცედურულად ორიენტირებული ენების მეტად საჭირო კლასს წარმოადგენს აგრეთვე ენები, რომლებიც განკუთვნილია საინფორმაციო-საცნობარო სისტემების მომსახურეობისათვის, დისკრეტული და უწყვეტი სისტემების მოდელირებისათვის.

დაპროგრამების ენების შექმნა, რომლებიც ამოცანების ფორმალიზაციისა და მათი ამოხსნის პრინციპებისთვის პრაქტიკულად იქნება ხელსაყრელი და მოსახერხებელი, ამჟამად ერთ-ერთ ცენტრალურ ადგილს იკავებს დაპროგრამების თეორიასა და პრაქტიკაში.

გამოთვლითი ტექნიკის განვითარების ქრონოლოგია

ყველაზე ადრინდელი გამომთვლელი მოწყობილობა იყო "აბაკი", რომელსაც იყენებდნენ ბაბილონელები ჩვენს წელთაღრიცხვამდე 4-5 ათასი წლის წინათ.

არისტოტელემ (384-322), ჩვ.წ. აღ-მდე 400 წლის წინათ, საფუძველი ჩაუყარა მათემატიკურ ლოგიკას.ლოგიკაში შემოიტანა ცვლადის ცნება და ასოები გამოიყენა ცნებათა აღსანიშნავად. IX საუკუნეში (820 წელს) შუააზიელმა მათემატიკოსმა ალ-ხორეზმმა (787-850), წიგნის "კიტაბ ალ-ჯებრ ალ-მუკაბალა"-ას ავტორმა, საიდანაც წარმოიშვა სიტყვა "ალგებრა", შემოიტანა ალგორითმის ცნება. IX-X საუკუნეში ევროპაში ვრცელდება თვლის არაბული სისტემა, რომელიც წარმოადგენს პოზიციურს და გააჩნია ნულის ცნება. თვლის არაბულმა სისტემამ ნელ-ნელა განდევნა რომაული ციფრები. ეს პროცესი გაგრძელდა XVII საუკუნემდე.

1457 წელს იოჰან გუტენბერგმა გამოიგონა საბეჭდი მოწყობილობა. 1612 წელს შოტლანდიელმა ლორდმა, მათემატიკოსმა ჯონ ნეპერმა (1550-1617) შემოიტანა ათობითი ლოგარითმის ცნება და ორი წლის შემდეგ შექმნა ლოგარითმული ცხრილი. 1622 წელს ინგლისელმა მეცნიერმა უილიამ ოუტრედმა გამოიგონა ლოგარითმული სახაზავი.

1623 წელს, გერმანიაში, ქალაქ შტუტგარდში, ვილჰელმ შიკარდის მიერ შეიქმნა პირველი გამოთვლითი მანქანა სახელწოდებით "გამომთვლელი საათი". შიკარდის მანქანა 6 თანრიგა იყო. მას შეეძლო რიცხვების შეკრება და გამოკლება. იგი თანრიგების გადავსებას ზარის დარეკვით იტყობინებოდა. ერთი პერიოდი შიკარდის მანქანა თავისი ნახაზებიანად დაიკარგა. იგი იპოვეს 1935 წელს, მაგრამ მეორე მსოფლიო ომის დროს ისევ დაიკარგა. 1956 წელს იპოვეს მხოლოდ ნახაზები. 1960 წელს აღნიშნული ნახაზების მიხედვით ენთუზიასტთა ჯგუფმა ააგო მანქანა და დაამტკიცა, რომ შიკარდის მანქანა ერთ-ერთი პირველი მუშა გამომთვლელი მანქანაა. აღნიშნული მანქანა დღესაც ინახება ქალაქ შტუტგარდის სამეფო ბიბლიოთეკაში.

1642 წელს პარიზში ფრანგმა მეცნიერმა ბლეზ პასკალმა (1623-1662) ააგო გამომთვლელი მანქანა სახელწოდებით "პასკალინი". პასკალის მანქანას შეეძლო მხოლოდ რიცხვების შეკრება. გამოკლებას კი ვერ აკეთებდა. შიკარდის "გამომთვლელ საათთან" შედარებით იგი ნაკლებად საიმედო იყო. მიუხედავად ამისა, გაურკვეველი მიზეზებით ისტორიას აწყობდა დაევიწყებინათ შიკარდის მანქანა და პასკალი გაეხადათ გამოთვლითი მანქანების კონცეპციის ფუძემდებლად.მიუხედავად ასეთი მცდელობისა, გამოთვლითი მანქანის პირველ გამომგონებლად შიკარდი დარჩა.

1674 წელს გერმანელი მეცნიერის ვილჰელმ ლაიბნიცის (1646-1716) მიერ შემუშავებულ იქნა "ბიჯური გამომთვლელი", რომელიც ააგო ფრანგმა ოლივერმა.

ლაიბნიცის მანქანას შეეძლო 5 და 12 ნიშნა რიცხვების გამრავლება 16 ნიშნა შედეგით. ლაიბნიცის მანქანამ გამოყენება ვერ ჰპოვა.

1812 წელს ინგლისელმა მეცნიერმა ჩარლზ ბებიჯმა (1791-1871) შეიმუშავა მექანიკური კალკულატორის პროექტი, რომლის მიხედვითაც 1823 წელს ააგო მთლიანად ავტომატიზებული მანქანა **Difference Engine**. ბებიჯის მანქანის საშუალებით შეიძლებოდა ექვსნიშნა რიცხვების შეკრება, გამოკლება, გამრავლება და გაყოფა. შემდეგ ბებიჯმა შექმნა ანალიტიკური მანქანა **Analytical Engine**. აღნიშნულ მანქანას გააჩნდა ცენტრალური პროცესორული მოწყობილობა, მეხსიერება და პერფობარათები, რომელზეც დაიტანებოდა გარკვეული პროგრამები. ბებიჯის მანქანას შეეძლო ოცნიშნა რიცხვებზე ოპერირება.

1820 წელს შარლ კოლმარმა შექმნა არითმომეტრი, რომელიც აგებული იყო ლაიბნიცის მანქანის პრინციპით და ასრულებდა რიცხვების გამრავლებას. იგი იყო პირველი კალკულატორი, რომელიც ჩაშვებულ იქნა მასიურ წარმოებაში. აღნიშნულმა მანქანამ რეალიზაციის თვალსაზრისით მსოფლიორეკორდი დაამყარა.

1842 წელს ცნობილი ინგლისელი პოეტის ლორდ ბაირონის ქალიშვილმა-ადა ბაირონ კინგმა (1815-1852) ბებიჯის მანქანისათვის შეიმუშავა პროგრამა. გამოთვლითი ტექნიკის ისტორიაში ადა ბაირონი კინგი ცნობილია, როგორც პირველი პროგრამისტი. მის სახელს ატარებს 1979 წელს შექმნილი პროგრამირების უნიკალური ენა **ADA**.

1847 წელს ირლანდიელმა მათემატიკოსმა ჟორჟ ბულმა (1815-1864) გამოუშვა წიგნი "**The Mathematical Analysis of Logic**" ("ლოგიკის მათემატიკური ანალიზი"), სადაც ჩამოაყალიბა ახალი ალგებრული სისტემა, რომელსაც შემდგომში "ბულის ალგებრა "ეწოდა.

1935 წელს კორპორაცია **IBM** მსოფლიოს წარუდგინა მანქანა **IBM 601,** რომელიც აგებული იყო რელეებზე და შეეძლო გამრავლება გაეკეთებინა 1 წამში. ოპერაციათა შესრულების ასეთმა სიჩქარემ **IBM 601-ს** დიდი პოპულარობა მოუტანა მეცნიერებსა და ბიზნესმენებს შორის.

1936 წელს ინგლისელმა მეცნიერმა ალან ტიურინგმა (1912-1954) შექმნა კომპიუტერის მათემატიკური მოდელი, რომელსაც "ტიურინგის მანქანა" ეწოდა. ტიურინგის მანქანაში" მოცემული იყო კომპიუტერის აგების პრობლემების გადაწყვეტის გზები.

1945 წელს ჯონ ფონ ნეიმანმა წამოაყენა გარე დამამახსოვრებელი მოწყობილობების იდეა პროგრამებისა და მონაცემების შესანახად. ამავე წელს ჯონ ფონ ნეიმანი გაერთიანდა ამერიკული კომპიუტერის **ENIAC**-ის დაპროექტებულ ჯგუფთან და აღწერა ახალი კომპიუტერის **EDVAC**-ის მოწყობილობები. აქვე გააკეთა შენახული პროგრამის კონცეფციის განსაზღვრის დეტალური აღწერა. ამ ნაშრომიდან დაიწყო ე.წ. "ჯონ ფონ ნეიმანის არქიტექტურა".

1947 წელს ცნობილ ამერიკელ მეცნიერს (კიბერნეტიკის ფუძემდებელი) ნორბერტ ვინერს (1894-1964) შემოაქვს ტერმინი "კიბერნეტიკა".

1955 წელს **IBM**-ს თანამშრომლებმა ჯონ ბეკუსის ხელმღვანელობით შექმნეს ალგორითმული ენა **FORTRAN (FORmuler TRANslator**-ფორმულების გარდამქმნელი), რომელიც გამოიყენებოდა სამეცნიერო ტექნიკური და საინჟინრო ამოცანების დასაპროგრამებლად.ამავე წელს ფირმა **Ferrant**-იმ გამოუშვა კომპიუტერი Pegasus, რომელშიც გამოყენება ჰპოვა საერთო დანიშნულების რეგისტრების კონცეპციამ. აღნიშნული კომპიუტერისათვის, აშშ-ს სამხედრო საზღვაო ფლოტის ოფიცერმა (შემდგომში, ადმირალმა,ერთადერთმა ქალმა აშშ-ს სამხედრო საზღვაო ფლოტში), ფირმა **Remington Rand**-ის პროგრამისტების ჯგუფის ხელმძღვანელმა, კაპიტანმა გრეის ხოპერმა პირველმა შეიმუშავა მატრანსილებელი პროგრამა, რომელსაც უწოდა კომპილატორი. ეს პროგრამა მოხერხებულ ალგებრულ ფორმაში ჩაწერილ პროგრამებს მანქანურ ენაზე უკეთებდა ტრანსლიაციას.

1960 წელს მეცნიერების ჯონ მაკართის,ფრიც ბაუერის,ჯონ ვეგშტეინის,ჯონ ბეკუსის,პიტერ ნეუერის და ალან პერლისის მიერ შეიქმნა ალგორითმული ენა **ALGOL (ALGOritmic Language**-ალგორითმული ენა), რომელშიც შემოტანილი იყო ახალი ცნებები. მაგალითად, ბლოკური სტრუქტურა. **ALGOL-60** კონცეპტუალური საფუძველი გახდა პროგრამირების მრავალი ენისათვის.

1962 წელს ფირმა **"Teletype"** -მა გამოუშვა კლავიატურა და მონიტორი.1964 წელს ამერიკელმა მეცნიერმა დუგლას ენჯელბერტმა შექმნა დიალოგიური პროგრამული ინტერფეისი, რომელიც რეაგირებდა სპეციალური მაჩვენებლის კურსორის მოქმედებაზე. კურსორის ჩასართავად ენჯელბერტმა შექმნა მანიპულატორი, რომელსაც შემდგომში "თაგვი" ეწოდა. მანიპულატორი "თაგვი" დაპატენტებულ იქნა 1970 წელს.

1965 წელს დორტმუნდის კოლეჯის პროფესორების ტომ კურტცისა და ჯონ კემენის მიერ შემუშავებულ იქნა ალგორითმული **ენა BASIC (Beginers all-purpouse**

simbolic instruction code-სიმბოლური ინსტრუქციების მრავალმიზნობრივი ენა დამწყებთათვის).

1967 წელს კორპორაცია **IBM**-მა გამოუშვა **Floppy** დისკი.

1969 წელს შეიქმნა პირველი კომპიუტერული ქსელი **ARPANET**-ი ეს ინტერნეტის შექმნის საფუძველი გახდა. ამავე წელს შეიქმნა პირველი ლაზერული პრინტერი. ფირმა **XEROX**–მა წარმოადგიმა ლაზერული ბეჭდვა ქსეროგრაფიული ბეჭდვის საფუძველზე.

1970 წელს ჰილბერტ ჰაიატმა დააპატენტა ერთ ნახევარგამტარულ კრისტალზე დამზადებული კომპიუტერის პროცესორის სქემა. ამ უკანასკნელმა კი საფუძველი ჩაუყარა მიკროკომპიუტერის შექმნის ერას.ამავე წელს ციურიხის პოლიტექნიკური უნივერსიტეტის პროფესორმა ნიკოლას ვირტმა შექმნა ალგორითმული ენა, რომელსაც ფრანგი მეცნიერის ბლეზ პასკალის პატივსაცემად **PASCAL**-ი უწოდა.

1971 წელს ამერიკული კომპანია BBN–ის მთავარმა ინჟინერმა რეი ტომლინსონმა ერთი კომპიუტერიდან მეორე კომპიუტერზე გაგზავნა შეტყობინება „**QWERTYUIOP**”-სტანდარტული კლავიატურის ზედა რეგისტრის სიმბოლოები, რომელშიც პირველმა გამოიყენა ნიშანი „@“. შეტყობინება ზუსტად იქნა მიღებული მეორე კომპიუტერზე. ამით დაიწყო ახალი ერა ადამიანთა ურთიერთობის ისტორიაში.

1974 წელს ბრაიან კერნიგმა და დენის რიტჩმა შექმნეს პროგრამირების ენა C.

1981 წელს კომპანია **IBM** უშვებს პერსონალურ კომპიუტერს **IBM PC**, რომელიც ხდება პერსონალური კომპიუტერის სტანდარტი. აქედან წამოვიდა ტერმინი “**IBM** თავსებადი კომპიუტერები“. ამავე წელს კომპანია **Microsoft** -მა დაიწყო მუშაობა **DOS**-ის გრაფიკულ გარსზე, სახელწოდებით **Interface manager**. აღნიშნული გარსი გახდა **Windows**-ის შექმნის საფუძველი. 1982 წელს კომპანია **Philips**-ი კომპიუტერისათვის უშვებს პირველ **CD ROM**-ს. 1985 წელს **Microsoft**-მა სავაჭრო ქსელში სარეალიზაციოდ გაუშვა **Windows** 1.0. 1990 წელს კომპანია **Microsoft**-მა დაიწყო **Windows** 3.0. 2002 წელს კომპანია **Microsoft**-ი უშვებს **Windows** 2002–ს ანუ **Windows XP Profesional-ს, Windows XP Home Edition-ს და Windows XP Server**-ს.

ზოგიერთი კომპიუტერული ფირმის მოკლე ისტორია

1. IBM- აი-ბი-ემ

1911 წელს ნიუ-იორკში რეგისტრაციაში გატარდა ფირმა სახელწოდებით **Computing-Tabuluting-Recordind (CTR).** კომპანიის საშტატო ერთეული შეადგენდა 1300. კომპანიის საქმიანობა იყო პერფობარათებზე მომუშავე ელექტრული ტაბულატორების გამოშვება. კომპანია **CTR**-მა სწრაფად მოიპოვა პოპულარობა და გახსნა თავისი ქვეგანყოფილებები, როგორც ამერიკის სხვადასხვა ადგილებში, ასევე კანადაში.

კომპანიაში საქმისადმი უსაზღვრო ერთგულებისა და განსაკუთრებული ოპტიმიზმის სული სუფევდა. ტომას უოტსონი თანამშრომლებს მიმართავდა : ”თქვენი გულები ეკუთვნის ჩვენს ბიზნესს”, და მართლაც, კომპანიის ყოველი წარმატება თითოეული თანამშრომლის სიამაყე იყო.

1924 წელს კომპანია CTR-მა სახელი შეიცვალა და მას **ეწოდება IBM (International Busines Machines Corporation). 1945 წელს IBM**-მა კოლუმბიის უნივერსიტეტში გახსნა სამეცნიერო-კვლევითი ლაბორატორია. ერთი წლის შემდეგ **IBM**-მა წარმოადგინა ელექტრონული გამომთვლელი მანქანის IBM Multiplier-ის პირველი კომერციული მოდელი. 1952 წელს **IBM**-ის პრეზიდენტად ირჩევენ თომას უოტსონ-უმცროსს. მისი ხელმძღვანელობით IBM-ი ხდება მსოფლიოს კომპიუტერული ბაზრის უპირობო ლიდერი. 1957 წელს **IBM**-ის წლიურმა შემოსავალმა გადააჭარბა ერთ მილიარდ დოლარს. ევროპის კომპიუტერების 90% იყო **IBM**-ის მარკით. 1959 წელს **IBM**-ი უშვებს პრინტერს **IBM** 1401-ს, რომლის სიჩქარე 4-ჯერ მეტი იყო მანამდე არსებული პრინტერების სიჩქარეზე.

1969 წელს **IBM**-ის კომპიუტერების დახმარებით **NASA**-მ პირველმა განახორციელა მთვარეზე ადამიანის გაფრენა. ამავე წელს იანვარში აშშ-ს იუსტიციის სამინისტრომ კომპანია **IBM**-ში დაიწყო მაშტაბური ანტიმონოპოლიური გამოძიება, რომელიც გაგრძელდა 13 წელიწადს. ამ დროისათვის **IBM**-ს მსოფლიო კომპიუტერული ბაზრის 70% ჰქონდა დაპყრობილი. იუსტიციის სამინისტროს დაგეგმილი ჰქონდა **IBM**-ი დაეყო რამოდენიმე წვრილ ფირმებად, რომლებიც შემდგომში ერთმანეთის კონკურენტები გახდებოდნენ.

80-იან წლებსა და 90-იან წლების დასაწყისში **IBM**-ი იძულებული გახდა გამოშვებული პროდუქციის ფასი დაეწია თვითღირებულებაზე დაბლა. აქედან გამომდინარე, 80-იანი წლების შუა პერიოდში **IBM** აღმოჩნდა მძიმე ფინანსურ

მდგომარეობაში. **IBM**-ს 10 წელზე მეტი დასჭირდა ფინანსური კრიზისიდან გამოსასვლელად.

1984 წელს **IBM**-მა შექმნა კომპიუტერული ქსელი, რომელსაც შეეძლო 1000-ზე მეტი კომპიუტერის გაერთიანება. ამავე წელს შექმნა მეხსიერების 1 მეგაბაიტიანი ჩიპი. 1994 წელს **IBM**-მა გამოუშვა სუპერკომპიუტერი, რომლის სიჩქარე იყო 136 მილიარდი ოპერაცია წამში.

1997 წელს შედგა ისტორიული მატჩი ჭადრაკში **IBM Deep Blue**- სა და გარი კასპაროვს შორის, რომელიც გროსმაისტერის მარცხით დამთავრდა.

2. INTL-ინტელი

1968 წლის ივნისში რობერტ ნოისმა გორდონ მურმა დააფუძნეს კორპორაცია **Intel(Integrated electronics)**. მათთან მივიდა ანდრეი გროვი, რომელიც **Intel**-ის დღევანდელი დირექტორთა საბჭოს თავჯდომარეა. 1974 წელს კორპორაცია **Intel**-ში მოვიდა მათი მომავალი პრეზიდენტი კრეიგ ბარეტი. იმ დღიდან **Intel** გადაიქცა მსოფლიოში მიკროპროცესორების უმსხვილეს მწარმოებელ კორპორაციად. მიკროპროცესორების წარმოების პირველი ცდები ტარდებოდა ჯერ ფირმა **Shockley Semiconductor Laboratory**-ში, შემდეგ კი -**Faichild Semyconductor**-ში. ნოისი და გორდონი ზემოთ ხსენებული ორივე ფირმის თანამშრომლები იყვნენ. როცა ნოისმა და გორდონმა შექმნეს ფირმა Intel-ი, მათ ორივე ფირმაში მუშაობის დიდი გამოცდილებაა ჰქონდათ.

ერთ მშვენიერ დღეს ნოისს დაებადა იდეა, ეს უწვრილესი დეტალი შეეერთებინა ერთმანეთთან სილიციუმის ერთ კრისტალზე გამტარების გარეშე. 1959 წელს ნოისმა გააკეთა განაცხადი თავისი ჩანაფიქრის განხორციელების შესახებ. ოდნავ მოგვიანებით ნოისმა ერთ კრისტალზე რამოდენიმე ელემენტის განლაგების შესაძლებლობა დაამტკიცა. ამ მომენტიდან რეალური ხდება ინტეგრალური სქემის შექმნა. ტრანზიტორების საფუძველზე შექმნილმა ინტეგრალურმა სქემამ საფუძველი ჩაუყარა მიკროკომპიუტერების შექმნას.

ორი წლის მუშაობის შემდეგ ნოისმა და მურმა შექმნეს მეხსიერების ჩიპი 1103, რომლის ზომა ორი ნაბეჭდი სიმბოლოს ზომას არ აღემატებოდა. თითოეული ჩიპი შეიცავდა 4 ათას ტრანზიტორს. აღნიშნული უწვრილესი ჩიპები ცვლიდნენ ათასობით კერამიკულ გულანებს და სწრაფად მუშაობდნენ. 1985 წელს იაპონელებმა

გამოუშვეს ბევრად უფრო იაფი მეხსიერების ჩიპები, რამაც Intel-ი აიძულა შეეწყვიტა კომპიუტერის მეხსიერების წარმოება. მურმა ამ მომენტს „პატარა წარუმატებლობა“ უწოდა.

1982 წელს ფირმა **Intel**-ისა და ფირმა **AMD**-ის (**Advanced Micro Devices**) გზები გადაიკვეთა იმის გამო, რომ 1991 წელს **AMD**-მ დაიწყო ფირმა ინტელის პროცესორის X86-ის გადაკეთებული ვარიანტის Am 386 პროცესორის გამოშვება, რომელიც საკმაოდ პოპულარული გახდა. ამან კი გამოიწვია **Intel**-სა და **AMD**-ს შორის სასამართლო პროცესების დაწყება, რაც Intel-ის გამარჯვებით დამთავრდა.

1997წელს **Intel**-ი უკვე მზად იყო 300 მეგაჰერციანი **Pentium II** გამოსაშვებად.

3. HEWLETT-PACARD- ჰიულეტ პაკარდი

1934 წელს დევიდ პაკარდმა (1912-1996) და იულიამ ჰიულეტმა დაამთავრეს სტენფორდის უნივერსიტეტის ელექტრო ფაკულტეტი და დიპლომის აღების აღსანიშნავად ერთად წავიდნენ სათევზაოდ. სწორედ იმ დღეს დაებადათ იდეა ერთად დაეწყოთ საკუთარი ბიზნესი. მეგობრებმა გარაჟში დაიწყეს თავიანთი პირველი მოწყობილობის აუდიო ოსცილიატორის **HP200A**-ს აწყობა, რომელიც განკუთვნილი იყო ბგერითი აპარატურის ტესტირებისათვის. 1939 წელს შეიქმნა კომპანია **Hewlett Packard**-ი. კომპანიის სახელწოდებაში მეგობრების სახელების თანმიმდევრობა გადაწყდა მონეტის აგდებით.

1942 წელს **Hewlett Packard**-მა პალო-ალტოში ააშენა პირველი საწარმო იმ გაანგარიშებით, რომ, თუ გაკოტრდებოდნენ, შენობა გადაქცეულიყო პროდუქტების მაღაზიად. მეორე მსოფლიო ომის დროს სახელმწიფო შეკვეთების მოცულობა ძალიან გაიზარდა და კომპანია **Hewlett Packard**-ი თავიანთი პროდუქციის „სასიგნალო გენერატორების ‘‘ წარმოებაში ლიდერები გახდნენ.

1959 წელს ჟენევაში იქმნება **Hewlett Packard**-ის ევროპული წარმომადგენლობა და პირველი საზღვარგარეთული წარმოება დასავლეთ გერმანიაში. 60-იან წლებში კომპანია **Hewlett Packard**-ი პირველად მოხვდა აშშ-ს 500 უმსხვილესი ფირმების სიაში. **Hewlett Packard**-მა იაპონიასთან შექმნა ერთობლივი საწარმო. 1966 წელს შექმნეს ზოგიერთი საზომი ხელსაწყოების მაკონტროლებელი პირველი კომპიუტერი **HP2116A**.

1968 წელს შექმნეს პირველი სამაგიდო გამომთვლელი **Hewlett Packard**-ი. 1969 წელს გამოუშვეს 16 მომხმარებლისათვის განკუთვნილი მიკროკომპიუტერის ოპერაციული სისტემა. ამან კი გამოიწვია დევიდ პაკარდის საკმაოდ მაღალ თანამდებობაზე გადაყვანა. იგი დაინიშნა აშშ-ს უშიშროების საბჭოს მდივნის მოადგილედ.

4. MICROSOFT-მაიქროსოფტი

უილიამ ჰენრი გეიტს III დაიბადა 1955 წელს, იგი სწავლობდა კერძო სკოლაში, სადაც მას აღმოაჩნდა მათემატიკისაკენ მიდრეკილება და სხვა განსაკუთრებული შესაძლებლობები. ბილ გეიტსმა იქვე გაიცნო პოლ ალენი. ორივე ფანტასტიკის მოყვარულები იყვნენ. მერვე კლასელი ბილ გეიტსი უკვე პროგრამებს წერდა თავისი პირველი **alma mater**-ისთვის, რაზეც მას მანქანური დროის მიხედვით ფულს უხდიდნენ. ბოლო კლასში ყოფნისას ბილი პროგრამისტად მოეწყო აეროკოსმოსურ კორპორაცია **TRW**-ში, სადაც წელიწადში 20 ათას დოლარს უხდიდნენ. 1975 წელს **Microsoft**-ი ხდება კერძო კორპორაცია, რომლის თავჯდომარეა ბილ გეიტსი, ხოლო პოლ ალენი აღმასრულებელი ვიცე-პრეზიდენტი.

ერთხელ ბილმა და ალენმა ჟურნალ **Popular Electronics**-ში ნახეს სტატია სამაგიდო კომპიუტერ **Altair**-ზე, სადაც მისი მწარმოებელი კომპანია **MITS**-ი ეძებდა კომპიუტერისათვის პროგრამირების ენას, რომლებიც სპეციალისტებს საშუალებას მისცემდა ემუშავათ **Altair**-ზე. პოლმა და ბილმა სასწრაფოდ დაამყარეს კავშირი კომპანია **MITS**-თან და განუცხადეს, რომ გააჩნიათ ალგორითმული ენა **BASIC**-ის საუკეთესო ვერსია. ერთადერთი რაც მათ ქონდათ, იყო სამი კვირა კომპანია **MITS**-ის ხელმძღვანელობასთან შეხვედრამდე. ბილმა კოდი გააკეთა. პოლმა პროგრამა დაწერა. მოკლედ, ორ კვირაში პროგრამული უზრუნველყოფა მზად იყო. კომპანიას არ ჰქონდა პროდუქციის გასაღების მენეჯერის აყვანის საშუალება. ამიტომ ეს საქმე თავისთავზე აიღო გეიტსის დედამ. იგი ყოველგვარი მორიდების გარეშე სთავაზობდა Microsoft-ის პროდუქციას ისეთ უზარმაზარ კორპორაციებს, როგორიცაა **IBM** და **AT&T**.

1977 წლის ივლისში **Microsoft**-ი იწყებს პროგრამირების მეორე ენის ForTran-ის გაყიდვას. 1978 წლის აპრილში კი პროგრამირების მესამე ენის **COBOL-80**-ის გაყიდვას, რომელიც განკუთვნილი იყო 8080, Z-80 და 8085 მიკროპროცესორული სისტემებისათვის.

BASIC 8080 პირველი პროგრამული პროდუქტია, რომელიც 1979 წლის 4 აპრილს ერთ მილიონიან პრემიას ღებულობს. პრემია, რომელიც გადაეცემა პოლ ალენს იყო **Microsoft**-ის პირველი ჯილდო.

1982 წლის 1 აპრილს **Microsoft**-ი ევროპაში ხსნის მარკეტინგისა და გასაღების წარმომადგენლობას. ასევე კომპანიის პირველ საზღვარგარეთულ განყოფილებას დიდ ბრიტანეთში. ამავე წლის ნოემბერში კომპიუტერების მწარმოებელმა 50-ზე მეტმა ფირმამ შეიძინა **Microsoft**-ის დისკური ოპერაციული სისტემის **MS DOS**-ის ლიცენზია.

1983 წელს **Microsoft**-მა პუბლიკას წარუდგინა MS DOS-თვის განკუთვნილი ტექსტური რედაქტორი **Microsoft Word**-ი. ამავე წელს იგი ქმნის ოპერაციულ სისტემა **Windows-ს.** 1984 წელს **Microsoft**-ი პროდუქტების მწარმოებელ ფირმებს შორის იკავებს პირველ ადგილს. ამავე წელს **Microsoft-ი Windows**-თვის ქმნის ელექტრონულ ცხრილს **Excel**-ს.

აშშ-ს ანტიმონოპოლიურ კანონმდებლობაში ერთ ერთ საკანონმდებლო აქტს წარმოადგენს შერმანის კანონი, რომელიც კრძალავს „ერთპიროვნულ კონტროლს ამა თუ იმ დარგში". აღნიშნული კანონი, რომელიც ბრწყინვალედ მუშაობდა სხვა დარგში, ვერ მუშაობდა პროგრამული უზრუნველყოფის ბაზარზე. აქედან გამომდინარე შეგვიძლია გავაკეთოთ დასკვნა, რომ **Microsoft**-ს მისი დაარსებიდან დღემდე აშშ-ს მთავრობა ფარულად მხარს უჭერდა, რადგან სახელმწიფოს შეუძლია მიიღოს საკმაოდ დიდი ეკონომიკური და პოლიტიკური სარგებლობა, კერძოდ:

1998 კორპორაციის მეპატრონე ქვეყნის პრესტიჟს მაღლა სწევს, რამდენადაც ის მსოფლიოში ყველაზე მდიდარი ადამიანია;
1999 კორპორაცია **Microsoft**-ი ქვეყანას უხდის უზარმაზარ გადასახადს, რადგანაც მონოპოლისტების გადასახადი ბევრად სჭარბობს ჩვეულებრივი საწარმოების გადასახადს.
2000 და, ბოლოს, ბილ გეიტსი არნახული შრომის მოყვარეა. იგი დღეღამეში 15 საათზე მეტს **Microsoft**-თვის ხარჯავს. **Microsoft**-ის შემყურე ქვეყანაში კი შრომისწარმოებლურობა მკვეთრად იზრდება.

5. SAMSUNG-სამსუნგი

ფირმა **samsung**-ი წარმოადგენს სამხრეთ-კორეისა და ჩინეთის ერთობლივ კომპანიას. "**Samsung**" ქართულად ნიშნავს „სამი ვარსკვლავი". იგი დაარსდა 1938

წელს ქალაქ ტეგუში ახალგაზრდა ბიზნესმენის ლბიუნ გოლის მიერ, ორმოც კაციანი შტატით, როგორც სავაჭრო ფირმა. იგი ვაჭრობდა ბრინჯით.

1969 წელს კომპანია იცვლის პროფილს და თითქმის ნულიდან იწყებს ელექტრონული პროდუქციის წარმოებას. ფირმა Sanyo-სთან ერთად იქმნება **Samsung Electronic Compani(SEC),** რომელიც უშვებდა ნახევარგამტარებს. რამდენიმე წლის შემდეგ ეს ფირმა Samsung-ის საკუთრებაში გადავიდა.

1983 წელს დაიწყო **64 M DRAM** მიკროსქემების გამოშვება, რომლის ტევადობა 64 მეგაბაიტი იყო. **Samsung**-ი იყო პირველი ფირმა, რომელმაც გამოუშვა **CD-ROM, VIDEO-CD,PHOTO-CD.**

Samsung-ი ელექტრონულ–სხივური მილაკების, კინოსკოპების **(Samsung Display Devices Co (SDD)** ერთ–ერთი უმსხვილესი მწარმოებელია. მას გააჩნია ქარხნები კორეაში, მალაიზიასა და გერმანიაში. **Samsung**-მა და **LG-Electronics**-მა გამოიგონეს ბიოტელევიზორები. ბიოტელევიზორების ერთ–ერთი მახასიათებელია ინფრა წითელი გამოსხივება.

სავარჯიშოები

1. ააგეთ გრამატიკა, რომელიც წარმოქმნის $L = \{a^n b^n\},\ n \in N$ ენას.

2. ააგეთ გრამატიკა, რომელიც წარმოქმნის $L = \{a^m b^n\},\ m, n \in N$ ენას.

3. ააგეთ გრამატიკა, რომელიც წარმოქმნის $L = \{a^m c\ b^n\},\ m, n \in N$ ენას.

4. მოცემულია V_A და V_T ალფავიტები და G გრამატიკა. ზოგადად როგორ ენას წარმოქმნის ეს გრამატიკა?

   $V_A = \{\mathrm{S}, \mathrm{T}\},\ V_T = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$, S - აქსიომაა. G: S → T T , T→ aTa, T→ bTb, T→ c.

5. მოცემულია V_A და V_T ალფავიტები და G გრამატიკა. ზოგადად როგორ ენას წარმოქმნის ეს გრამატიკა?

   $V_A = \{\mathrm{S}\},\ V_T = \{\mathrm{p}, \mathrm{q}, \mathrm{r}\}$ G: S → p S p, S→ qSq, S → r.

6. მოცემულია V_A და V_T ალფავიტები და G გრამატიკა. ზოგადად როგორ ენას წარმოქმნის ეს გრამატიკა?

$V_A = \{\mathrm{S}, \mathrm{T}, \mathrm{U}\},\ V_T = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\},$, S - აქსიომაა. G: S → T a , T→ aS, S→ Ub, U→ bS, S→ c.

7. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = ab\ V\ \{a\}^*\{c\}^*\ V\ b$$

8. მოცემულია ავტომატის მდგომარეობებს შორის გადასვლის ცხრილი. იპოვე ხდომილობა, რომელიც ავტომატს q_1 მდგომარეობიდან გადაიყვანს q_4 მდგომარეობაში. ავტომატი სრულადაა განსაზღვრული თუ ნაწილობრივ?

Q/S	a	b	c
q_1	q_4	q_3	q_1
q_2	q_3	q_4	q_2
q_3	q_1	q_3	q_2
q_4	q_2	q_4	q_1

9. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = \{a\}^*\{c\}^* \quad V\ b\ \ V\ ab$$

10. მოცემულია ავტომატის მდგომარეობებს შორის გადასვლის ცხრილი. იპოვე ხდომილობა, რომელიც ავტომატს q_1 მდგომარეობიდან გადაიყვანს q_4 მდგომარეობაში. ავტომატი სრულადაა განსაზღვრული თუ ნაწილობრივ?

Q/S	a	b	c
q_1	q_4	q_3	q_1
q_2	$-$	q_4	q_2
q_3	q_1	q_3	$-$
q_4	q_2	q_4	q_1

11. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = \{a\ b\}^*a\ \ V\ \{a\}^*\ b\ \{c\}^*\ \ V\ a.$$

12. მოცემულია ავტომატის მდგომარეობებს შორის გადასვლის ცხრილი. იპოვე ხდომილობა, რომელიც ავტომატს q_1 მდგომარეობიდან გადაიყვანს q_4 მდგომარეობაში. ავტომატი სრულადაა განსაზღვრული თუ ნაწილობრივ?

Q/S	a	b	c
q_1	q_4	q_3	q_1
q_2	q_3	q_4	q_2
q_3	q_4	q_3	q_2
q_4	q_2	q_3	q_1

13. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = b\ \ V\ \{a\}^*\{c\}^*\ \ V\ b\ \{a\ b\}^*.$$

14. მოცემულია ავტომატის მდგომარეობებს შორის გადასვლის ცხრილი. იპოვე ხდომილობა, რომელიც ავტომატს q_1 მდგომარეობიდან გადაიყვანს q_4 მდგომარეობაში. ავტომატი სრულადაა განსაზღვრული თუ ნაწილობრივ?

Q/S	a	b	c
q_1	q_4	q_3	q_1
q_2	q_3	q_4	q_2
q_3	q_1	q_3	q_2
q_4	q_2	q_4	q_1

15. ააგეთ გრამატიკა, რომელიც წარმოქმნის $L = \{x\ c\ \tilde{x}\}$ ენას, სადაც x და $\tilde{x}$ სარკული ანასახებია.

16. მოცემულია V_A და V_T ალფავიტები და G გრამატიკა. ზოგადად როგორ ენას წარმოქმნის ეს გრამატიკა?

$$V_A = \{T\},\ V_T = \{\mathrm{b,c}\} \quad \mathrm{G:}\ \mathrm{S} \rightarrow \mathrm{T\ T}\ ,\ \mathrm{T} \rightarrow cTc,\ \mathrm{T} \rightarrow bTb.$$

17. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = a\ (b\ \ V\ \{a\}^*\{c\}^*\ \ V\ a).$$

18. მოცემულია ავტომატის მდგომარეობებს შორის გადასვლის ცხრილი. იპოვე ხდომილობა, რომელიც ავტომატს q_1 მდგომარეობიდან გადაიყვანს q_4 მდგომარეობაში. ავტომატი სრულადაა განსაზღვრული თუ ნაწილობრივ?

Q/S	a	b	c
q_1	q_4	q_3	q_1
q_2	$-$	$-$	q_2
q_3	q_1	q_3	q_2
q_4	q_2	q_4	q_1

19. ააგეთ გრამატიკა, რომელიც წარმოქმნის $L = \{a\ b^n c\},\ n \in N$ ენას.

20. მოცემულია V_A და V_T ალფავიტები და G გრამატიკა. ზოგადად როგორ ენას წარმოქმნის ეს გრამატიკა?

$$V_A = \{\mathrm{T}\},\ V_T = \{\mathrm{a,b,c}\} \quad \mathrm{G:}\ \mathrm{S} \to \mathrm{T\ T}\ ,\ \mathrm{T} \to aTa,\ \mathrm{T} \to bTb,\ \mathrm{T} \to c.$$

21. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = a\ \{b\ c\ \}^*\ V\ \{c\}^*\ V\ cb.$$

22. მოცემულია ავტომატის მდგომარეობებს შორის გადასვლის ცხრილი. იპოვე ხდომილობა, რომელიც ავტომატს q_1 მდგომარეობიდან გადაიყვანს q_3 მდგომარეობაში. ავტომატი სრულადაა განსაზღვრული თუ ნაწილობრივ?

Q/S	a	b	c
q_1	q_4	q_3	q_1
q_2	q_3	q_4	q_2
q_3	q_1	q_3	q_2
q_4	$-$	q_4	q_1

23. მოცემულია რეგულარული გამოსახულებები: $R = \{ab, c\ \},\ S = \{cba, a\ \}$.

    იპოვეთ: $R\ V\ S,\ \ S * R,\ \{R\}^*\ ,\ \{S\}^*$.

24. მოცემულია რეგულარული გამოსახულებები: $R = \{a, bc\ \},\ S = \{cb, ba\ \}$.

    იპოვეთ: $R\ V\ S,\ \ S * R,\ \{R\}^*\ ,\ \{S\}^*, R * S$.

25. ააგეთ გრამატიკა, რომელიც წარმოქმნის $L = \{a^n b^n c\},\ n \in N$ ენას.

26. მოცემულია V_A და V_T ალფავიტები და G გრამატიკა. ზოგადად როგორ ენას წარმოქმნის ეს გრამატიკა?

$$V_A = \{\mathrm{S}\},\ V_T = \{\mathrm{a,b}\} \quad \mathrm{G:}\ \mathrm{S} \to \mathrm{a\ S}\ ,\ \ S \to Sb\ .$$

27. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = ab\ \ V\ b\ \{c\}^*\ \ V\ b.$$

28. მოცემულია ავტომატის მდგომარეობებს შორის გადასვლის ცხრილი. იპოვე ხდომილობა, რომელიც ავტომატს q_1 მდგომარეობიდან გადაიყვანს q_3 მდგომარეობაში. ავტომატი სრულადაა განსაზღვრული თუ ნაწილობრივ?

Q/S	a	b	c
q_1	q_4	q_3	$-$
q_2	q_3	q_4	q_2
q_3	q_1	q_3	q_2
q_4	q_2	q_4	q_1

29. ააგეთ გრამატიკა, რომელიც წარმოქმნის $L = \{a^n\ c\ b^n\},\ n \in N$ ენას.

30. მოცემულია V_A და V_T ალფავიტები და G გრამატიკა. ზოგადად როგორ ენას წარმოქმნის ეს გრამატიკა?

$V_a = \{\mathrm{S}, \mathrm{T}\},\ V_T = \{\mathrm{a}, \mathrm{b}\},$, S - აქსიომაა. G: S → T T , T→ aTa, T→ bTb .

31. მოცემულია რეგულარული გამოსახულება. ააგე შესაბამისი გრაფი.

$$R = (ab\ \ V\ \{a\}^*\{c\}^*\ )\ b.$$

32. ვთქვათ მოცემულია მურის $B = \big(X, Q, Y, q_1 \in Q, F(x \in X)\big)$ ავტომატი, სადაც $X = \{x_1, x_2\}$, $Q = \{q_1, q_2, q_3\}$, $Y = \{y_1, y_2\}$, ხოლო

$$Fq_1(y_1) = \{q_2(x_1), q_3(x_2)\},$$
$$Fq_2(y_1) = \{q_3(x_1), q_2(x_2)\},$$
$$Fq_3(y_2) = \{q_1(x_1), q_3(x_2)\}.$$

ა). ააგეთ გადასვლის ცხრილები, რომლებიც განსაზღვრავენ მურის ავტომატის ფუნქციონირების კანონს ($\varphi(q, x)$ და $\psi(q)$ ფუნქციებს).

ბ). ააგეთ შესაბამისი გრაფი.

გ). ააგეთ მილის ავტომატი, რომელიც იქნება მურის B ავტომატის ინტერპრეტაცია (მისი ეკვივალენტური ავტომატი).

33. ვთქვათ მოცემულია მურის $B = \left(X, Q, Y, q_1 \in Q, F(x \in X)\right)$ ავტომატი, სადაც $X = \{x_1, x_2\}$, $Q = \{q_1, q_2, q_3\}$, $Y = \{y_1, y_2\}$, ხოლო

$$Fq_1(y_1) = \{q_1(x_1), q_3(x_2)\},$$

$$Fq_2(y_2) = \{q_3(x_1), q_1(x_2)\},$$

$$Fq_3(y_2) = \{q_1(x_1), q_2(x_2)\}.$$

ა). ააგეთ გადასვლის ცხრილები, რომლებიც განსაზღვრავენ მურის ავტომატის ფუნქციონირების კანონს ($\varphi(q, x)$ და $\psi(q)$ ფუნქციებს).

ბ). ააგეთ შესაბამისი გრაფი.

გ). ააგეთ მილის ავტომატი, რომელიც იქნება მურის B ავტომატის ინტერპრეტაცია (მისი ეკვივალენტური ავტომატი).