

## Tiesioginio ir atbulinio išvedimo produkcijų sistemoje algoritmų realizacija Python programavimo kalba

Darbą atliko: Jonas Kristupas Rudžianskas, informatika: kompiuterių mokslas, 4 kursas, 2 grupė

### Turinys

1. Tiesioginio išvedimo algoritmo pseudokodas .....	1
2. Atbulinio išvedimo algoritmo pseudokodas .....	2
3. Tiesioginio išvedimo programos testai .....	2
1. Faktas konsekvėntė .....	2
2. Čyras vs Negnevitsky; Čyras laimi .....	4
3. Čyras vs Negnevitsky; Negnevitsky laimi .....	5
4. Tikslas tarp faktų .....	6
5. Kelias neegzistuoja .....	7
6. Negnevitsky pavyzdys (5 produkcijos, p.37) .....	8
4. Atbulinio išvedimo programos testai .....	9
1. Užmirštama šaka .....	9
2. Devynios produkcijos D,C .....	10
3. Devynios produkcijos C,D .....	11
4. Ciklas ir praleistas potikslis .....	12
5. Grafas su trumpu keliu .....	13
6. Grafas su ilgu keliu .....	14
7. Trys alternatyvos tikslui .....	16
8. Trys alternatyvos ir nepasiekiamas tikslas .....	17
9. Tikslas tarp faktų .....	18
10. Negnevitsky pavyzdys (5 produkcijos, p.39) .....	19
5. UML klasių diagrama .....	20
6. Literatūros sąrašas .....	20

### 1. Tiesioginio išvedimo algoritmo pseudokodas

Ivestis:

*rules* – produkcijų sąrašas.

*facts* – pradinių faktų sąrašas.

*goal* – tikslas.

*plan* – sąrašas, atsakymo produkcijoms kaupti.

Išvestis:

*boolean* – algoritmas grąžina *true* jei kelias iki tikslo egzistuoja ir sugrąžina *false* jei nepavyksta tokio rasti.

```
function forward_chaining(rules, facts, goal, plan) returns boolean
    if goal ∈ facts then                                     //1
        return true                                         //2
    end if
    repeat                                                  //3
        is_new_fact ← false                                  //4
        for all rule ∈ rules                                //5
            if ¬is_applied(rule) ∧ produces_new_fact(rule) ∧ is_satisfied(rule, facts) then //6
```

```

        if right_hand_side(rule) ∈ facts then //7
            set_produces_new_fact(rule, false) //8
        else //9
            new_fact ← right_hand_side(rule) //10
            set_is_applied(rule, true) //11
            facts ← facts ∪ { new_fact } //12
            plan ← plan ∪ { name(rule) } //13
            is_new_fact ← true //14
            if new_fact = goal then //15
                return true //16
            end if
            goto end for //17
        end if
    end if
end for
until ¬is_new_fact //18
return false //19
end function

```

## 2. Atbulinio išvedimo algoritmo pseudokodas

Ivestis:

*rules* – produkcijų sąrašas.

*facts* – pradinių faktų sąrašas.

*goal* – tikslas.

*new\_facts* – faktų išvestų duotuoju laiko momentu sąrašas.

*new\_goal* – bandomas išvesti tikslas duotuoju laiko momentu.

*current\_goals* – visų bandomų išvesti tikslų duotuoju laiko momentu sąrašas.

*plan* – sąrašas, atsakymo produkcijoms kaupti.

Išvestis:

*boolean* – algoritmas grąžina *true* jei kelias iki tikslo egzistuoja ir sugrąžina *false* jei nepavyksta tokio rasti.

```

function backward_chaining(rules, facts, goal, new_facts, new_goal, current_goals, plan) returns boolean //1
    if goal ∈ facts then //2
        return true
    end if
    current_new_facts ← save_state(new_facts) //3
    current_plan ← save_state(plan) //4
    for all rule ∈ rules //5
        if right_hand_side(rule) = goal then //6
            is_satisfied ← true //7
            for all fact ∈ left_hand_side(rule) //8
                if fact ∉ facts ∪ new_facts then //9
                    if fact ∈ new_goals then //10
                        is_satisfied ← false //11
                        goto end for //12
                    end if
                    new_goals ← new_goals ∪ { goal } //13
                    is_satisfied ← backward_chaining(rules, facts, goal, new_facts, fact, new_goals, plan) //14
                    new_goals ← new_goals \ { goal } //15
                    if ¬is_satisfied then //16
                        goto end for //17
                    end if
                    if goal ∈ new_facts then //18
                        return true //19
                    end if
                end if
            end for
        end if
        if is_satisfied then //20
            plan ← plan ∪ { name(rule) } //21
            new_facts ← new_facts ∪ { right_hand_side(rule) } //22
            return true //23
        end if
        new_facts ← restore_state(current_new_facts) //24
        plan ← restore_state(current_plan) //25
    end if
end for
return false //26
end function

```

## 3. Tiesioginio išvedimo programos testai

### 1. Faktas konsekvente

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//1 testas. Faktas konsekvente.
//1) Taisykles:
L A // R1: A -> L
K L // R2: L -> K
A D // R3: D -> A
M D // R4: D -> M
Z F B // R5: F, B -> Z
F C D // R6: C, D -> F
D A // R7: A -> D

//2) Faktai:
A B C

//3) Tikslas:
Z
```

## Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
R1: A -> L
R2: L -> K
R3: D -> A
R4: D -> M
R5: F, B -> Z
R6: C, D -> F
R7: A -> D

2) Faktai:
A, B, C

3) Tikslas:
Z

2 DALIS. Vykdymas

1 ITERACIJA
R1:A->L taikome. Pakeliame flag1. Faktai A, B, C ir L.

2 ITERACIJA
R1:A->L praleidziame, nes pakelta flag1.
R2:L->K taikome. Pakeliame flag1. Faktai A, B, C ir L, K.

3 ITERACIJA
R1:A->L praleidziame, nes pakelta flag1.
R2:L->K praleidziame, nes pakelta flag1.
R3:D->A netaikome, nes truksta D.
R4:D->M netaikome, nes truksta D.
R5:F,B->Z netaikome, nes truksta F.
R6:C,D->F netaikome, nes truksta D.
R7:A->D taikome. Pakeliame flag1. Faktai A, B, C, L ir K, D.

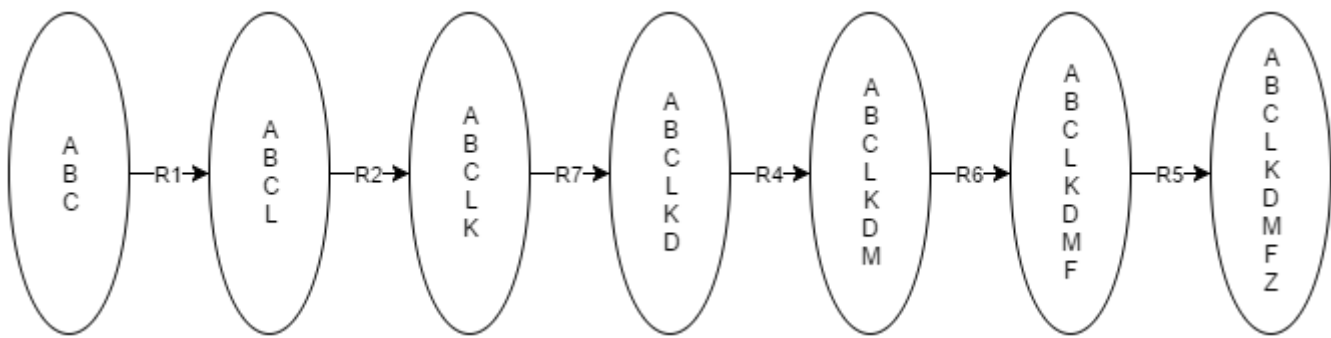
4 ITERACIJA
R1:A->L praleidziame, nes pakelta flag1.
R2:L->K praleidziame, nes pakelta flag1.
R3:D->A netaikome, nes konsekvantas faktuose. Pakeliame flag2.
R4:D->M taikome. Pakeliame flag1. Faktai A, B, C ir L, K, D, M.

5 ITERACIJA
R1:A->L praleidziame, nes pakelta flag1.
R2:L->K praleidziame, nes pakelta flag1.
R3:D->A praleidziame, nes pakelta flag2.
R4:D->M praleidziame, nes pakelta flag1.
R5:F,B->Z netaikome, nes truksta F.
R6:C,D->F taikome. Pakeliame flag1. Faktai A, B, C ir L, K, D, M, F.

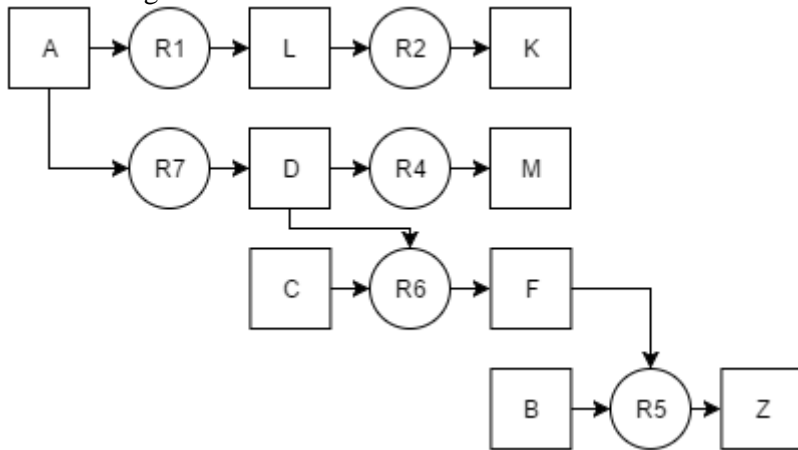
6 ITERACIJA
R1:A->L praleidziame, nes pakelta flag1.
R2:L->K praleidziame, nes pakelta flag1.
R3:D->A praleidziame, nes pakelta flag2.
R4:D->M praleidziame, nes pakelta flag1.
R5:F,B->Z taikome. Pakeliame flag1. Faktai A, B, C ir L, K, D, M, F, Z.

3 DALIS. Rezultatai
1) Tikslas Z isvestas.
2) Kelias R1, R2, R7, R4, R6, R5.
```

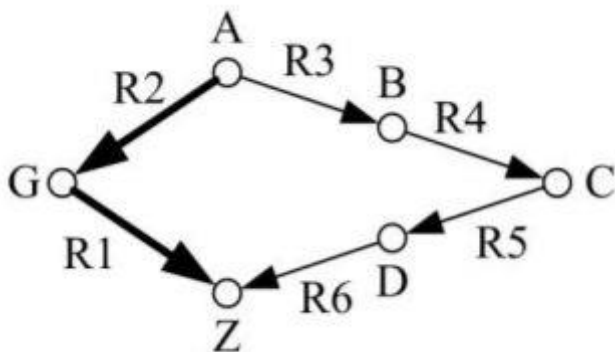
## Verifikavimo grafas:



Semantinis grafas:



## 2. Čyras vs Negnevitsky; Čyras laimi



Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//2 testas. Cyras vs Negnevitsky; Cyras laimi
//1) Taisykles:
Z G // R1: G -> Z
G A // R2: A -> G
B A // R3: A -> B
C B // R4: B -> C
D C // R5: C -> D
Z D // R6: D -> Z

//2) Faktai:
A

//3) Tikslas:
Z
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
R1: G -> Z
R2: A -> G
R3: A -> B
R4: B -> C
R5: C -> D
R6: D -> Z

2) Faktai:
A

3) Tikslas:
```

Z

2 DALIS. Vykdymas

1 ITERACIJA

R1:G->Z netaikome, nes truksta G.

R2:A->G taikome. Pakeliame flag1. Faktai A ir G.

2 ITERACIJA

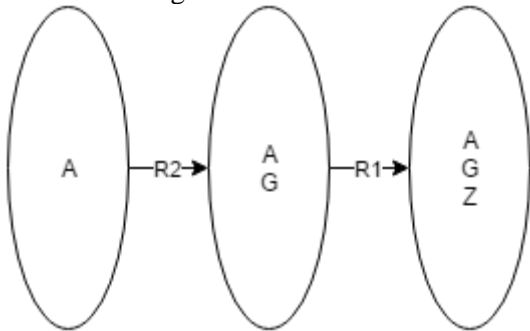
R1:G->Z taikome. Pakeliame flag1. Faktai A ir G, Z.

3 DALIS. Rezultatai

1) Tikslas Z isvestas.

2) Kelias R2, R1.

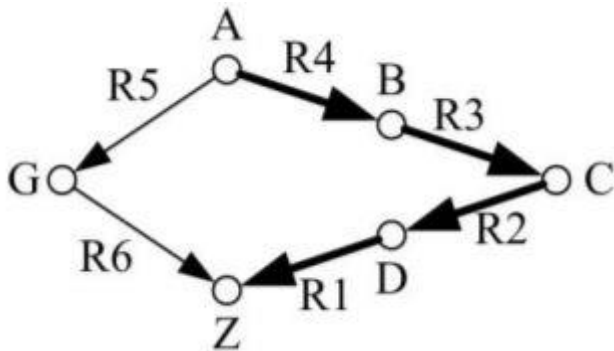
Verifikavimo grafas:



Semantinis grafas:



### 3. Čyras vs Negnevitsky; Negnevitsky laimi



Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//3 testas. Čyras vs Negnevitsky; Negnevitsky laimi
//1) Taisykles:
Z D // R1: D -> Z
D C // R2: C -> D
C B // R3: B -> C
B A // R4: A -> B
G A // R5: A -> G
Z G // R6: G -> Z

//2) Faktai:
A

//3) Tikslas:
Z
```

Išvestis:

1 DALIS. Duomenys

1) Taisykles:

R1: D -> Z

R2: C -> D

R3: B -> C

R4: A -> B

R5: A -> G

R6: G -> Z

2) Faktai:

A

```

3) Tikslas:
   Z

2 DALIS. Vykdymas

1 ITERACIJA
R1:D->Z netaikome, nes truksta D.
R2:C->D netaikome, nes truksta C.
R3:B->C netaikome, nes truksta B.
R4:A->B taikome. Pakeliame flag1. Faktai A ir B.

2 ITERACIJA
R1:D->Z netaikome, nes truksta D.
R2:C->D netaikome, nes truksta C.
R3:B->C taikome. Pakeliame flag1. Faktai A ir B, C.

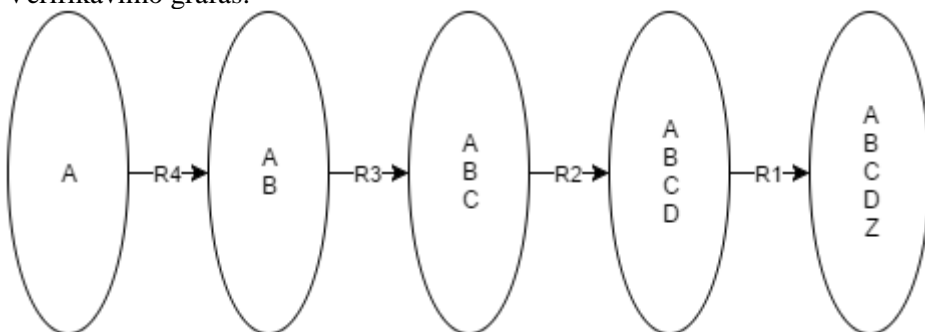
3 ITERACIJA
R1:D->Z netaikome, nes truksta D.
R2:C->D taikome. Pakeliame flag1. Faktai A ir B, C, D.

4 ITERACIJA
R1:D->Z taikome. Pakeliame flag1. Faktai A ir B, C, D, Z.

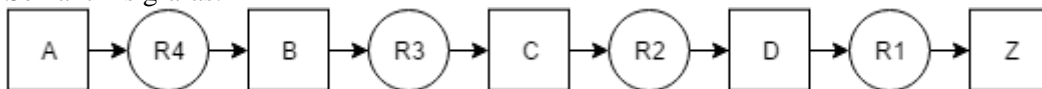
3 DALIS. Rezultatai
1) Tikslas Z isvestas.
2) Kelias R4, R3, R2, R1.

```

Verifikavimo grafas:



Semantinis grafas:



## 4. Tikslas tarp faktų

Ivestis:

```

//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//4 testas. Tikslas tarp faktų
//1) Taisyklės:
Z A // R1: A -> Z

//2) Faktai:
A Z

//3) Tikslas:
Z

```

Išvestis:

```

1 DALIS. Duomenys

1) Taisyklės:
R1: A -> Z

2) Faktai:
A, Z

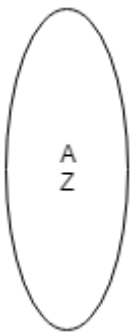
3) Tikslas:
Z

2 DALIS. Vykdymas

3 DALIS. Rezultatai
1) Tikslas Z tarp pradinių faktų

```

Verifikavimo grafas:



Semantinis grafas:



## 5. Kelias neegzistuoja

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//5 testas. Kelias neegzistuoja
//1) Taisykles:
B A // R1: A -> B
Z C // R2: C -> Z

//2) Faktai:
A

//3) Tikslas:
Z
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
  R1: A -> B
  R2: C -> Z

2) Faktai:
  A

3) Tikslas:
  Z

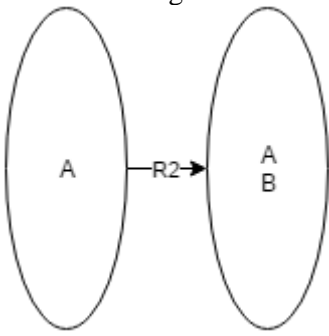
2 DALIS. Vykdymas

1 ITERACIJA
  R1:A->B taikome. Pakeliame flag1. Faktai A ir B.

2 ITERACIJA
  R1:A->B praleidziame, nes pakelta flag1.
  R2:C->Z netaikome, nes truksta C.

3 DALIS. Rezultatai
1) Tikslas Z neisvestas.
```

Verifikavimo grafas:



Semantinis grafas:



## 6. Negnevitsky pavyzdys (5 produkcijos, p.37)

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//6 testas. Negnevitsky pavyzdys (5 produkcijos, p.37)
//1) Taisykles:
Z Y D // R1: Y, D -> Z
Y X B E // R2: X, B, E -> Y
X A // R3: A -> X
L C // R4: C -> L
N L M // R5: L, M -> N

//2) Faktai:
A B C D E

//3) Tikslas:
Z
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
R1: Y, D -> Z
R2: X, B, E -> Y
R3: A -> X
R4: C -> L
R5: L, M -> N

2) Faktai:
A, B, C, D, E

3) Tikslas:
Z

2 DALIS. Vykdymas

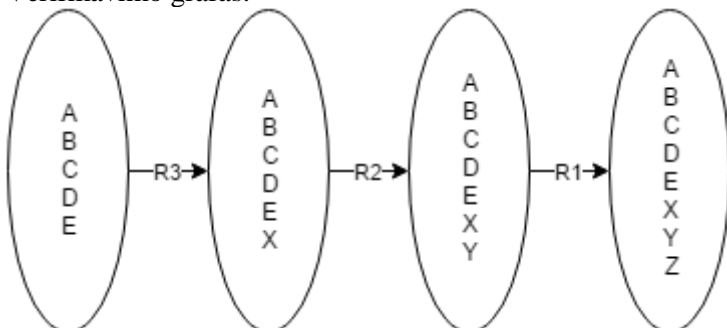
1 ITERACIJA
R1:Y,D->Z netaikome, nes trūksta Y.
R2:X,B,E->Y netaikome, nes trūksta X.
R3:A->X taikome. Pakeliame flag1. Faktai A, B, C, D, E ir X.

2 ITERACIJA
R1:Y,D->Z netaikome, nes trūksta Y.
R2:X,B,E->Y taikome. Pakeliame flag1. Faktai A, B, C, D, E ir X, Y.

3 ITERACIJA
R1:Y,D->Z taikome. Pakeliame flag1. Faktai A, B, C, D, E ir X, Y, Z.

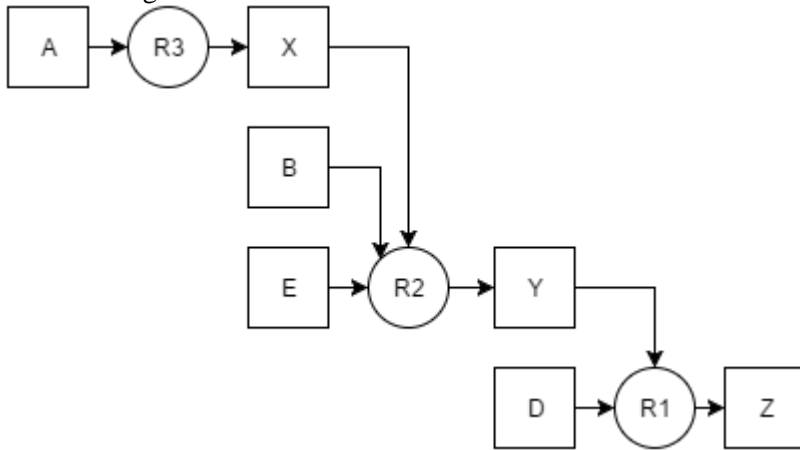
3 DALIS. Rezultatai
1) Tikslas Z išvestas.
2) Kelias R3, R2, R1.
```

Verifikavimo grafas:





Semantinis grafas:



## 4. Atbulinio išvedimo programos testai

### 1. Užmirštama šaka

Ivestis:

```

//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//1 testas. Uzmirstama saka.
//1) Taisykles:
Z C D // R1: C, D -> Z
C T   // R2: T -> C
Z T   // R3: T -> Z

//2) Faktai:
T

//3) Tikslas:
Z
  
```

Išvestis:

```

1 DALIS. Duomenys

1) Taisykles:
R1: C, D -> Z
R2: T -> C
R3: T -> Z

2) Faktai:
T

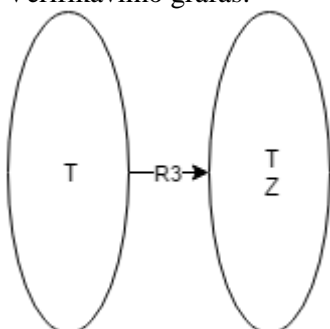
3) Tikslas:
Z

2 DALIS. Vykdymas

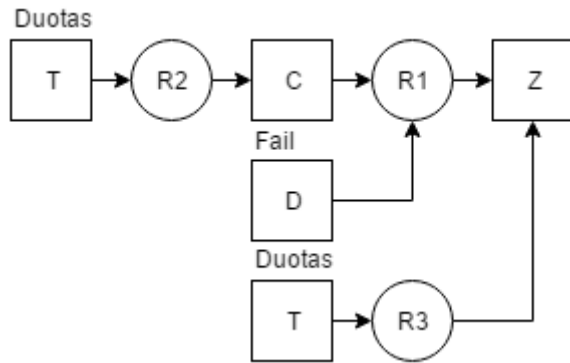
1).Tikslas Z. Randame R1:C,D->Z. Nauji tikslai C, D.
2)..Tikslas C. Randame R2:T->C. Nauji tikslai T.
3)...Tikslas T. Faktas (duotas), nes faktai T. Griztame, sekme.
4)..Tikslas C. Faktas (dabar gautas). Faktai T ir C. Griztame, sekme.
5)..Tikslas D. Nera taisykliu jo isvedimui. Griztame, FAIL.
6).Tikslas Z. Randame R3:T->Z. Nauji tikslai T.
7)..Tikslas T. Faktas (duotas), nes faktai T. Griztame, sekme.
8).Tikslas Z. Faktas (dabar gautas). Faktai T ir Z. Griztame, sekme.

3 DALIS. Rezultatai
1) Tikslas Z isvestas.
2) Kelias R3.
  
```

Verifikavimo grafas:



Semantinis grafas:



## 2. Devynios produkcijos D,C

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//2 testas. Devynios produkcijos D, C.
//1) Taisykles:
Z D C // R1: D, C -> Z
D C // R2: C -> D
C B // R3: B -> C
B A // R4: A -> B
A D // R5: D -> A
D T // R6: T -> D
A G // R7: G -> A
B H // R8: H -> B
C J // R9: J -> C

//2) Faktai:
T

//3) Tikslas:
Z
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
R1: D, C -> Z
R2: C -> D
R3: B -> C
R4: A -> B
R5: D -> A
R6: T -> D
R7: G -> A
R8: H -> B
R9: J -> C

2) Faktai:
T

3) Tikslas:
Z

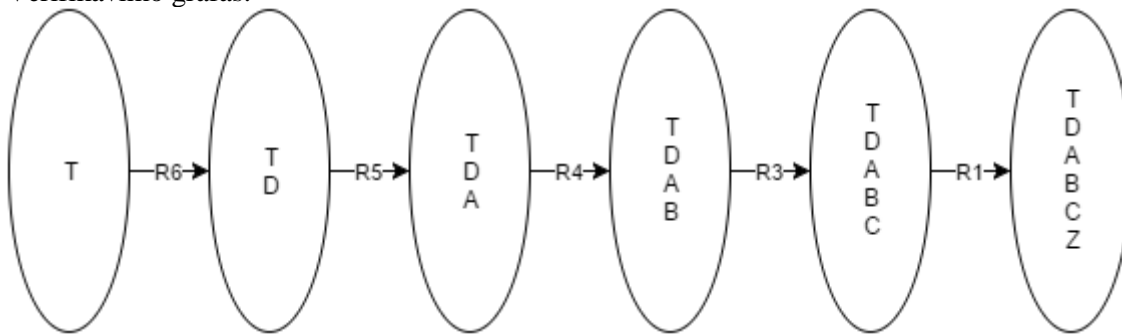
2 DALIS. Vykdymas

1)..Tikslas Z. Randame R1:D,C->Z. Nauji tikslai D, C.
2)..Tikslas D. Randame R2:C->D. Nauji tikslai C.
3)...Tikslas C. Randame R3:B->C. Nauji tikslai B.
4)...Tikslas B. Randame R4:A->B. Nauji tikslai A.
5)....Tikslas A. Randame R5:D->A. Nauji tikslai D.
6).....Tikslas D. Ciklas. Griztame, FAIL.
7)....Tikslas A. Randame R7:G->A. Nauji tikslai G.
8).....Tikslas G. Nera taisykliu jo isvedimui. Griztame, FAIL.
9)....Tikslas A. Nera daugiau taisykliu jo isvedimui. Griztame, FAIL.
10)...Tikslas B. Randame R8:H->B. Nauji tikslai H.
11)....Tikslas H. Nera taisykliu jo isvedimui. Griztame, FAIL.
12)....Tikslas B. Nera daugiau taisykliu jo isvedimui. Griztame, FAIL.
13)...Tikslas C. Randame R9:J->C. Nauji tikslai J.
14)...Tikslas J. Nera taisykliu jo isvedimui. Griztame, FAIL.
15)...Tikslas C. Nera daugiau taisykliu jo isvedimui. Griztame, FAIL.
16)..Tikslas D. Randame R6:T->D. Nauji tikslai T.
17)...Tikslas T. Faktas (duotas), nes faktai T. Griztame, sekme.
18)..Tikslas D. Faktas (dabar gautas). Faktai T ir D. Griztame, sekme.
19)...Tikslas C. Randame R3:B->C. Nauji tikslai B.
20)...Tikslas B. Randame R4:A->B. Nauji tikslai A.
21)....Tikslas A. Randame R5:D->A. Nauji tikslai D.
22).....Tikslas D. Faktas (buvo gautas), nes faktai T ir D. Griztame, sekme.
23)....Tikslas A. Faktas (dabar gautas). Faktai T ir D, A. Griztame, sekme.
24)...Tikslas B. Faktas (dabar gautas). Faktai T ir D, A, B. Griztame, sekme.
25)...Tikslas C. Faktas (dabar gautas). Faktai T ir D, A, B, C. Griztame, sekme.
26).Tikslas Z. Faktas (dabar gautas). Faktai T ir D, A, B, C, Z. Griztame, sekme.

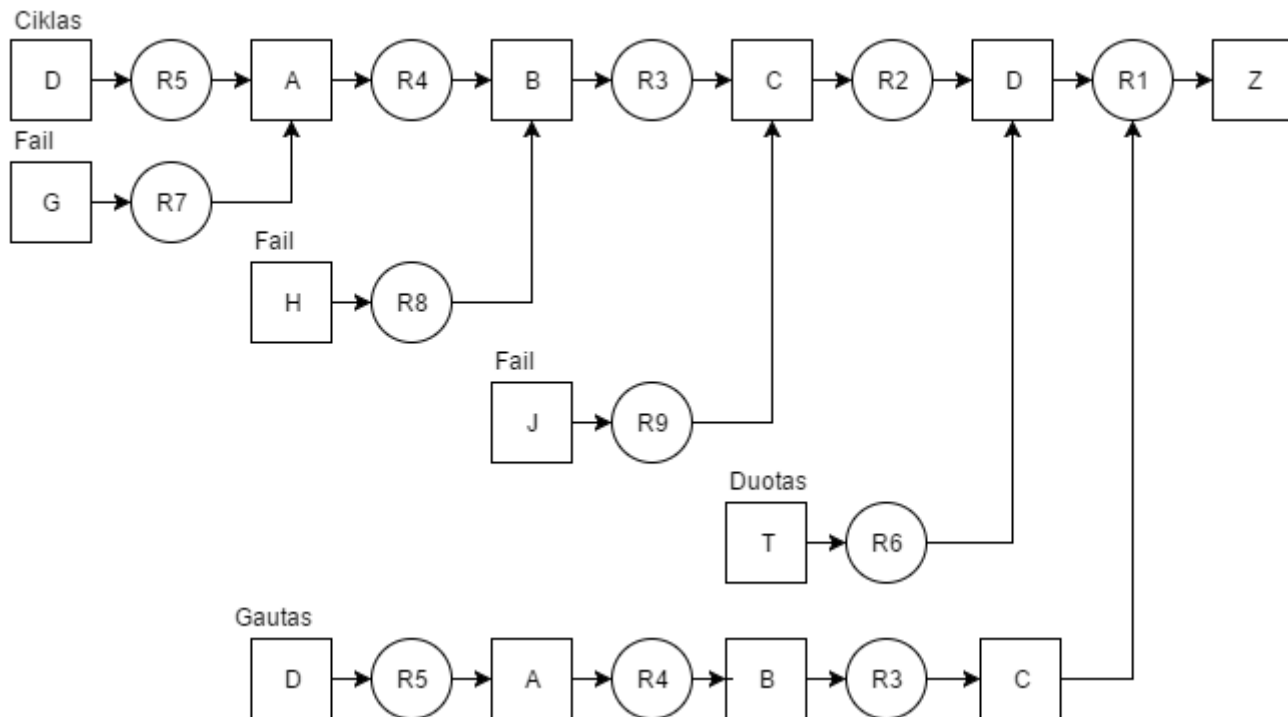
3 DALIS. Rezultatai
```

- 1) Tikslas Z išvestas.
- 2) Kelias R6, R5, R4, R3, R1.

Verifikavimo grafas:



Semantinis grafas:



### 3. Devynios produkcijos C,D

Ivestis:

```

//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//3 testas. Devynios produkcijos C, D.
//1) Taisykles:
Z C D // R1: C, D -> Z
D C // R2: C -> D
C B // R3: B -> C
B A // R4: A -> B
A D // R5: D -> A
D T // R6: T -> D
A G // R7: G -> A
B H // R8: H -> B
C J // R9: J -> C

//2) Faktai:
T

//3) Tikslas:
Z
  
```

Išvestis:

```

1 DALIS. Duomenys

1) Taisykles:
R1: C, D -> Z
R2: C -> D
R3: B -> C
R4: A -> B
R5: D -> A
  
```

```

R6: T -> D
R7: G -> A
R8: H -> B
R9: J -> C

```

2) Faktai:  
T

3) Tikslas:  
Z

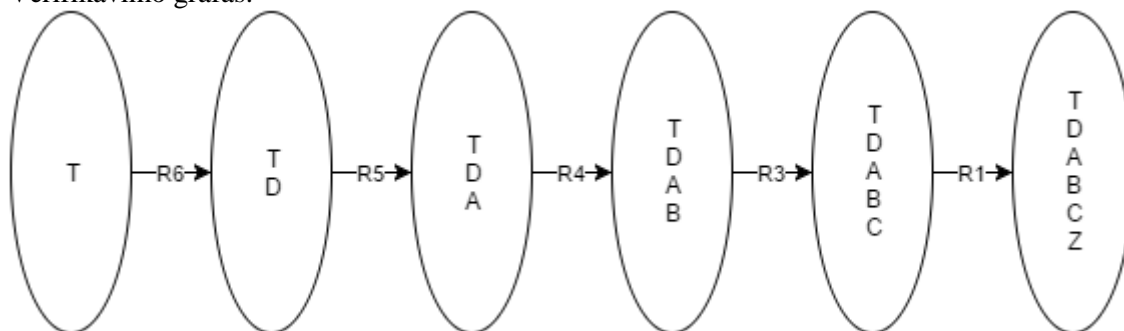
## 2 DALIS. Vykdymas

- 1).Tikslas Z. Randame R1:C,D->Z. Nauji tikslai C, D.
- 2)..Tikslas C. Randame R3:B->C. Nauji tikslai B.
- 3)...Tikslas B. Randame R4:A->B. Nauji tikslai A.
- 4)....Tikslas A. Randame R5:D->A. Nauji tikslai D.
- 5).....Tikslas D. Randame R2:C->D. Nauji tikslai C.
- 6).....Tikslas C. Ciklas. Griztame, FAIL.
- 7).....Tikslas D. Randame R6:T->D. Nauji tikslai T.
- 8).....Tikslas T. Faktas (duotas), nes faktai T. Griztame, sekme.
- 9).....Tikslas D. Faktas (dabar gautas). Faktai T ir D. Griztame, sekme.
- 10)....Tikslas A. Faktas (dabar gautas). Faktai T ir D, A. Griztame, sekme.
- 11)...Tikslas B. Faktas (dabar gautas). Faktai T ir D, A, B. Griztame, sekme.
- 12)..Tikslas C. Faktas (dabar gautas). Faktai T ir D, A, B, C. Griztame, sekme.
- 13)..Tikslas D. Faktas (buvo gautas), nes faktai T ir D, A, B, C. Griztame, sekme.
- 14).Tikslas Z. Faktas (dabar gautas). Faktai T ir D, A, B, C, Z. Griztame, sekme.

## 3 DALIS. Rezultatai

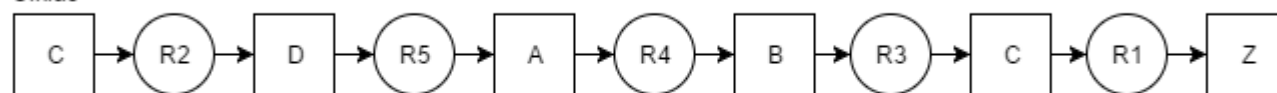
- 1) Tikslas Z isvestas.
- 2) Kelias R6, R5, R4, R3, R1.

## Verifikavimo grafas:

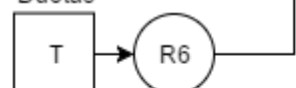


## Semantinis grafas:

### Ciklas



### Duotas



### Gautas



## 4. Ciklas ir praleistas potikslis

### Ivestis:

```

//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//4 testas. Penkios produkcijos ir praleistas potikslis.
//1) Taisykles:
Z A // R1: A -> Z
A B // R2: B -> A
B A C // R3: A, C -> B
B T // R4: T -> B
C T // R5: T -> C

//2) Faktai:
T

//3) Tikslas:
Z

```

### Išvestis:

# 1 DALIS. Duomenys

1) Taisyklės:  
R1: A -> Z  
R2: B -> A  
R3: A, C -> B  
R4: T -> B  
R5: T -> C

2) Faktai:  
T

3) Tikslas:  
Z

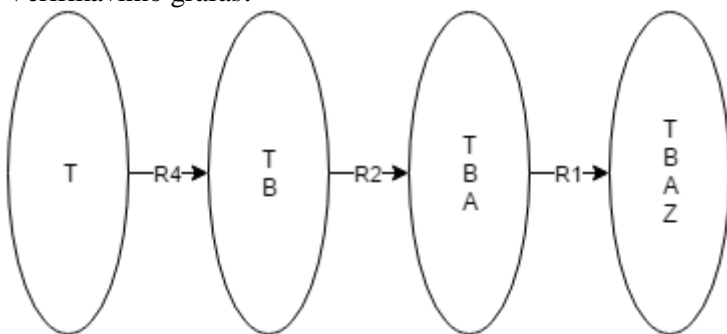
# 2 DALIS. Vykdymas

1).Tikslas Z. Randame R1:A->Z. Nauji tikslai A.  
2)..Tikslas A. Randame R2:B->A. Nauji tikslai B.  
3)...Tikslas B. Randame R3:A,C->B. Nauji tikslai A, C.  
4)....Tikslas A. Ciklas. Griztame, FAIL.  
5)...Tikslas B. Randame R4:T->B. Nauji tikslai T.  
6)....Tikslas T. Faktas (duotas), nes faktai T. Griztame, sekme.  
7)...Tikslas B. Faktas (dabar gautas). Faktai T ir B. Griztame, sekme.  
8)..Tikslas A. Faktas (dabar gautas). Faktai T ir B, A. Griztame, sekme.  
9).Tikslas Z. Faktas (dabar gautas). Faktai T ir B, A, Z. Griztame, sekme.

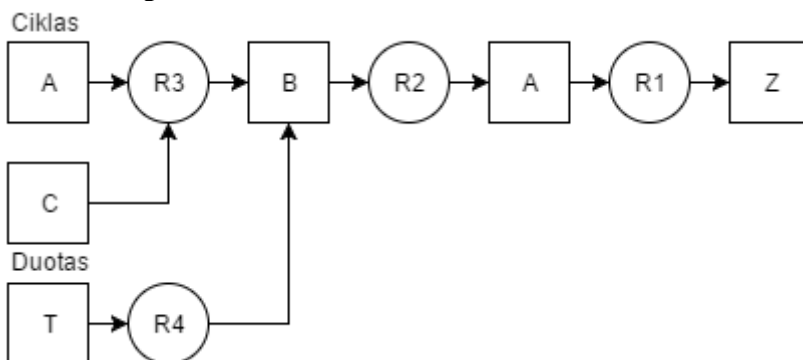
# 3 DALIS. Rezultatai

1) Tikslas Z isvestas.  
2) Kelias R4, R2, R1.

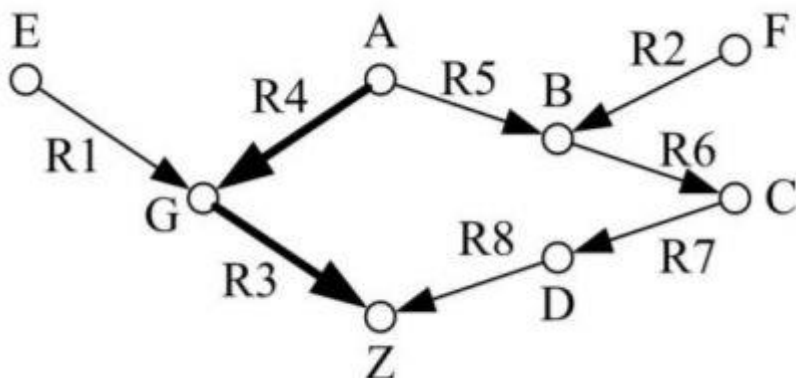
## Verifikavimo grafas:



## Semantinis grafas:



## 5. Grafas su trumpu keliu



## Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//5 testas. Grafas su trumpu keliu.
//1) Taisykles:
G E // R1: E -> G
B F // R2: F -> B
Z G // R3: G -> Z
G A // R4: A -> G
B A // R5: A -> B
C B // R6: B -> C
D C // R7: C -> D
Z D // R8: D -> Z

//2) Faktai:
A

//3) Tikslas:
Z
```

## Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
  R1: E -> G
  R2: F -> B
  R3: G -> Z
  R4: A -> G
  R5: A -> B
  R6: B -> C
  R7: C -> D
  R8: D -> Z

2) Faktai:
  A

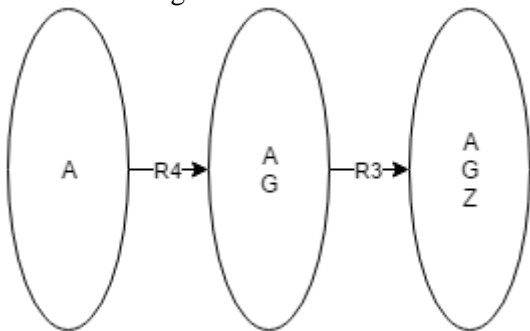
3) Tikslas:
  Z

2 DALIS. Vykdymas

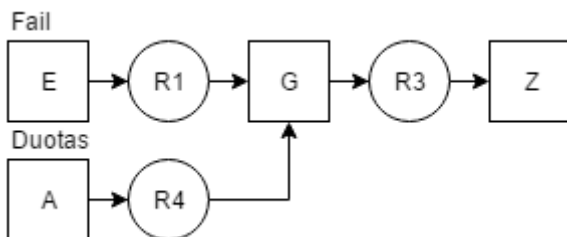
1).Tikslas Z. Randame R3:G->Z. Nauji tikslai G.
2)..Tikslas G. Randame R1:E->G. Nauji tikslai E.
3)...Tikslas E. Nera taisykliu jo isvedimui. Griztame, FAIL.
4)..Tikslas G. Randame R4:A->G. Nauji tikslai A.
5)...Tikslas A. Faktas (duotas), nes faktai A. Griztame, sekme.
6)..Tikslas G. Faktas (dabar gautas). Faktai A ir G. Griztame, sekme.
7).Tikslas Z. Faktas (dabar gautas). Faktai A ir G, Z. Griztame, sekme.

3 DALIS. Rezultatai
1) Tikslas Z isvestas.
2) Kelias R4, R3.
```

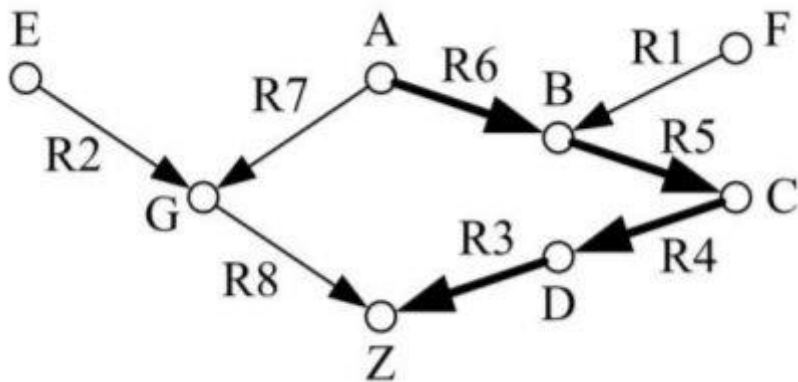
## Verifikavimo grafas:



## Semantinis grafas:



## 6. Grafas su ilgu keliu



#### Ivestis:

```

//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//6 testas. Grafas su ilgu keliu.
//1) Taisykles:
B F // R1: F -> B
G E // R2: E -> G
Z D // R3: D -> Z
D C // R4: C -> D
C B // R5: B -> C
B A // R6: A -> B
G A // R7: A -> G
Z G // R8: G -> Z

//2) Faktai:
A

//3) Tikslas:
Z

```

#### Išvestis:

```

1 DALIS. Duomenys

1) Taisykles:
  R1: F -> B
  R2: E -> G
  R3: D -> Z
  R4: C -> D
  R5: B -> C
  R6: A -> B
  R7: A -> G
  R8: G -> Z

2) Faktai:
  A

3) Tikslas:
  Z

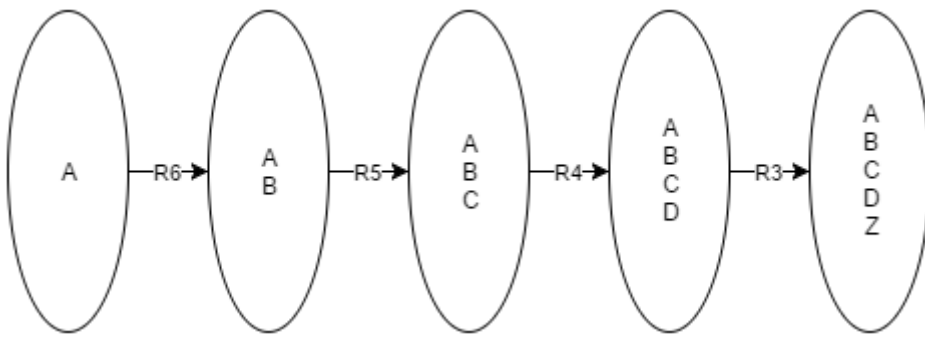
2 DALIS. Vykdymas

1).Tikslas Z. Randame R3:D->Z. Nauji tikslai D.
2)..Tikslas D. Randame R4:C->D. Nauji tikslai C.
3)...Tikslas C. Randame R5:B->C. Nauji tikslai B.
4)....Tikslas B. Randame R1:F->B. Nauji tikslai F.
5).....Tikslas F. Nera taisykliu jo isvedimui. Griztame, FAIL.
6)....Tikslas B. Randame R6:A->B. Nauji tikslai A.
7).....Tikslas A. Faktas (duotas), nes faktai A. Griztame, sekme.
8)....Tikslas B. Faktas (dabar gautas). Faktai A ir B. Griztame, sekme.
9)...Tikslas C. Faktas (dabar gautas). Faktai A ir B, C. Griztame, sekme.
10)..Tikslas D. Faktas (dabar gautas). Faktai A ir B, C, D. Griztame, sekme.
11).Tikslas Z. Faktas (dabar gautas). Faktai A ir B, C, D, Z. Griztame, sekme.

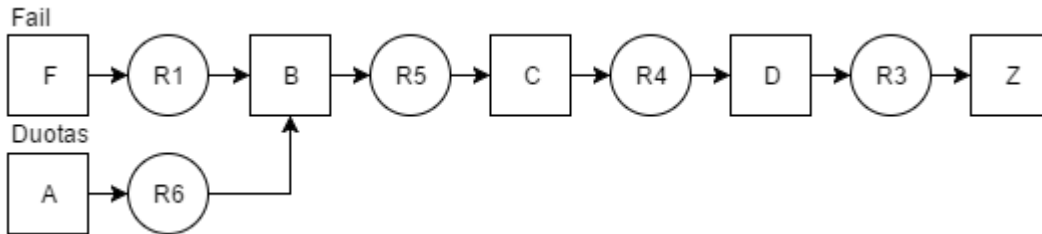
3 DALIS. Rezultatai
1) Tikslas Z isvestas.
2) Kelias R6, R5, R4, R3.

```

#### Verifikavimo grafas:



Semantinis grafas:



## 7. Trys alternatyvos tikslui

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//7 testas. Trys alternatyvos tikslui.
//1) Taisykles:
Z A // R1: A -> Z
Z B // R2: B -> Z
Z C // R3: C -> Z

//2) Faktai:
C

//3) Tikslas:
Z
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
  R1: A -> Z
  R2: B -> Z
  R3: C -> Z

2) Faktai:
  C

3) Tikslas:
  Z

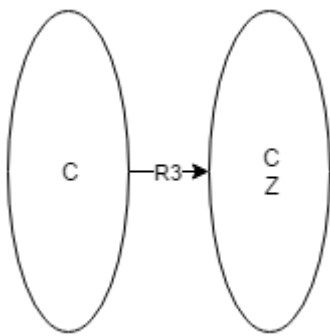
2 DALIS. Vykdymas

1).Tikslas Z. Randame R1:A->Z. Nauji tikslai A.
2)..Tikslas A. Nera taisykliu jo isvedimui. Griztame, FAIL.
3).Tikslas Z. Randame R2:B->Z. Nauji tikslai B.
4)..Tikslas B. Nera taisykliu jo isvedimui. Griztame, FAIL.
5).Tikslas Z. Randame R3:C->Z. Nauji tikslai C.
6)..Tikslas C. Faktas (duotas), nes faktai C. Griztame, sekme.
7).Tikslas Z. Faktas (dabar gautas). Faktai C ir Z. Griztame, sekme.

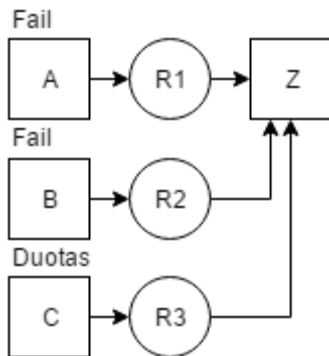
3 DALIS. Rezultatai
1) Tikslas Z isvestas.
2) Kelias R3.
```

Verifikavimo grafas:





Semantinis grafas:



## 8. Trys alternatyvos ir nepasiekiamas tikslas

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//8 testas. Trys alternatyvos ir nepasiekiamas tikslas.
//1) Taisykles:
Z A D // R1: A, D -> Z
Z B D // R2: B, D -> Z
Z C D // R3: C, D -> Z
Y C E // R4: C, E -> Y

//2) Faktai:
C D

//3) Tikslas:
Y
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
R1: A, D -> Z
R2: B, D -> Z
R3: C, D -> Z
R4: C, E -> Y

2) Faktai:
C, D

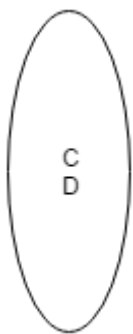
3) Tikslas:
Y

2 DALIS. Vykdydas

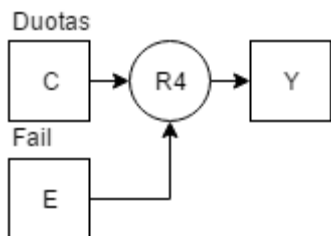
1).Tikslas Y. Randame R4:C,E->Y. Nauji tikslai C, E.
2)..Tikslas C. Faktas (duotas), nes faktai C, D. Griztame, sekme.
3)..Tikslas E. Nera taisykliu jo isvedimui. Griztame, FAIL.
4).Tikslas Y. Nera daugiau taisykliu jo isvedimui. Griztame, FAIL.

3 DALIS. Rezultatai
1) Tikslas Y neisvestas.
```

Verifikavimo grafas:



Semantinis grafas:



## 9. Tikslas tarp faktų

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//9 testas. Tikslas tarp faktų
//1) Taisykles:
Z A // R1: A -> Z

//2) Faktai:
A Z

//3) Tikslas:
Z
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
   R1: A -> Z

2) Faktai:
   A, Z

3) Tikslas:
   Z

2 DALIS. Vykdymas

3 DALIS. Rezultatai
1) Tikslas Z tarp pradiniu faktų
```

Verifikavimo grafas:



Semantinis grafas:



## 10. Negnevitsky pavyzdys (5 produkcijos, p.39)

Ivestis:

```
//Studentas Jonas Kristupas Rudzianskas, informatikos: kompiuteriu mokslo studiju programa, 4 kursas, 2 grupe
//10 testas. Negnevitsky pavyzdys (5 produkcijos, p.39)
//1) Taisykles:
Z Y D // R1: Y, D -> Z
Y X B E // R2: X, B, E -> Y
X A // R3: A -> X
L C // R4: C -> L
N L M // R5: L, M -> N

//2) Faktai:
A B C D E

//3) Tikslas:
Z
```

Išvestis:

```
1 DALIS. Duomenys

1) Taisykles:
R1: Y, D -> Z
R2: X, B, E -> Y
R3: A -> X
R4: C -> L
R5: L, M -> N

2) Faktai:
A, B, C, D, E

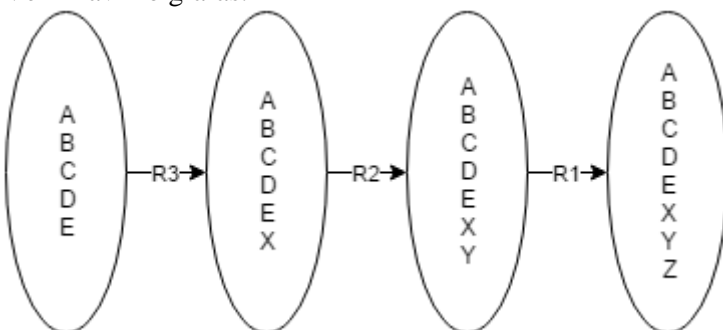
3) Tikslas:
Z

2 DALIS. Vykdymas

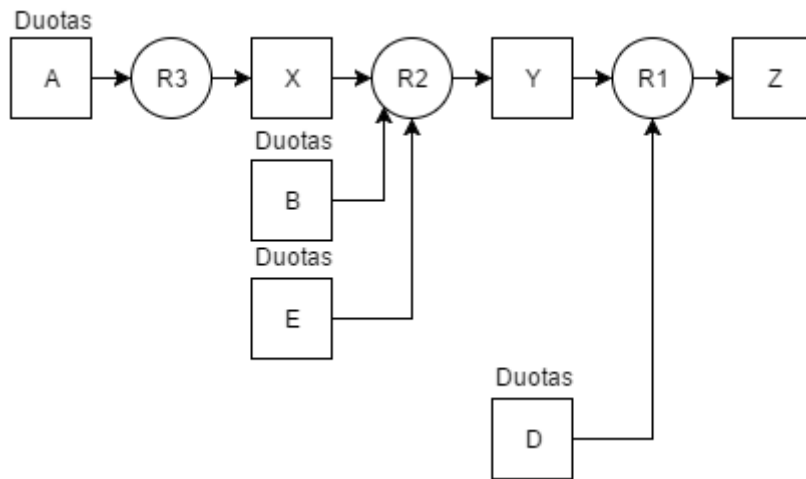
1).Tikslas Z. Randame R1:Y,D->Z. Nauji tikslai Y, D.
2)..Tikslas Y. Randame R2:X,B,E->Y. Nauji tikslai X, B, E.
3)...Tikslas X. Randame R3:A->X. Nauji tikslai A.
4)....Tikslas A. Faktas (duotas), nes faktai A, B, C, D, E. Griztame, sekme.
5)...Tikslas X. Faktas (dabar gautas). Faktai A, B, C, D, E ir X. Griztame, sekme.
6)...Tikslas B. Faktas (duotas), nes faktai A, B, C, D, E. Griztame, sekme.
7)...Tikslas E. Faktas (duotas), nes faktai A, B, C, D, E. Griztame, sekme.
8)..Tikslas Y. Faktas (dabar gautas). Faktai A, B, C, D, E ir X, Y. Griztame, sekme.
9)..Tikslas D. Faktas (duotas), nes faktai A, B, C, D, E. Griztame, sekme.
10).Tikslas Z. Faktas (dabar gautas). Faktai A, B, C, D, E ir X, Y, Z. Griztame, sekme.

3 DALIS. Rezultatai
1) Tikslas Z isvestas.
2) Kelias R3, R2, R1.
```

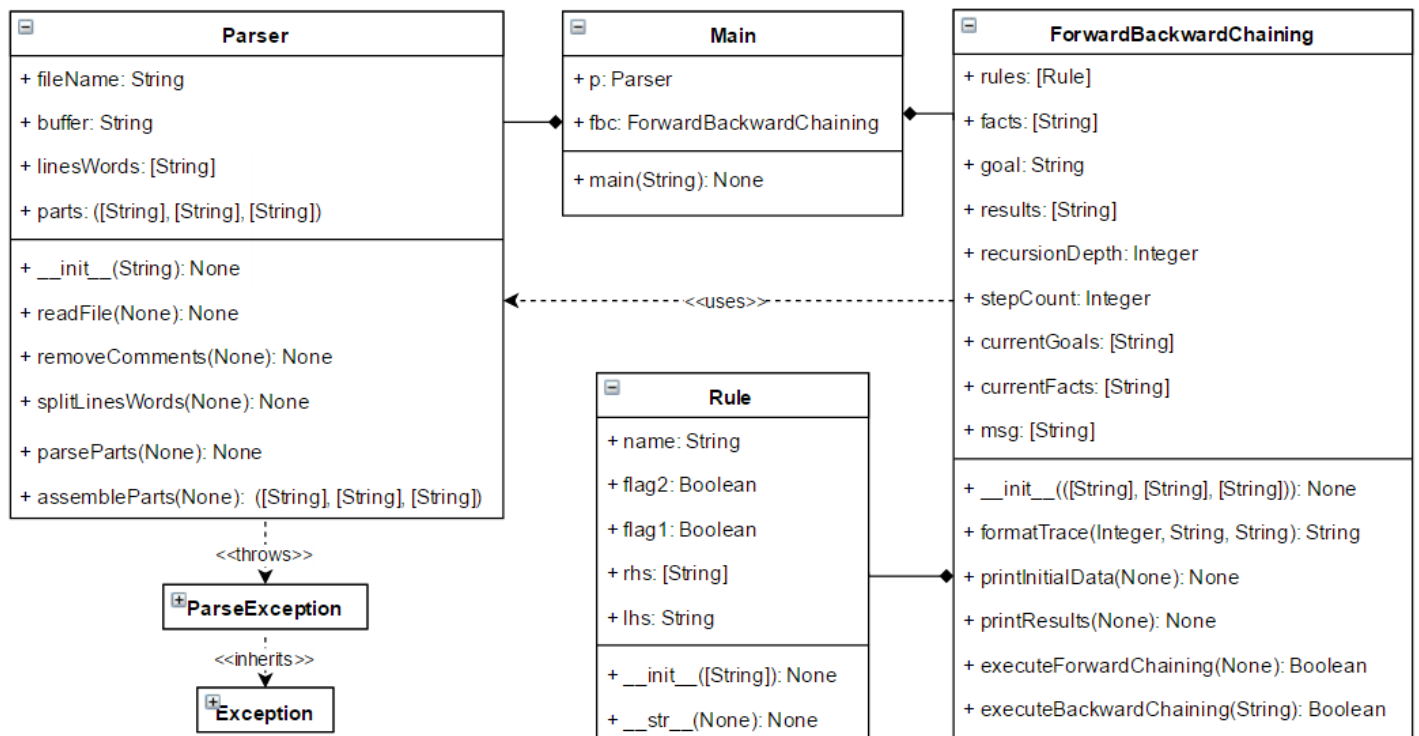
Verifikavimo grafas:



Semantinis grafas:



## 5. UML klasių diagrama



## 6. Literatūros sąrašas

1. V. Čyras „Intelektualios sistemos“. <http://www.mif.vu.lt/~cyras/AI/konspektas-intelektualios-sistemos.pdf>
2. M. Negnevitsky „Artificial Intelligence. A Guide to Intelligent Systems“, Pearson Education Limited, Harlow, 2005.