

# Algoritmi generiranja ciljnog programa

- Generiranje ciljnog programa na temelju *troadresnih naredbi*
- Generiranje ciljnog programa na temelju *postfiksног sustava oznaka*
- Generiranje ciljnog programa na temelju *sažetog sintaksnog stabla*

## Generiranje ciljnog programa na temelju troadresnih naredbi

- **Generator ciljnog programa koristi**
  - opisnik registara
  - opisnik podataka
- **Opisnik registara**
  - popis podataka spremljenih u pojedinim registrima
  - jedan register sadrži vrijednosti različitih podataka
    - vrijednost varijable  $y$  u registru  $Dx$
    - generira se naredba preslikavanja  $x:=y$
    - register  $Dx$  sadrži vrijednost varijabli  $x$  i  $y$
- **Opisnik podataka**
  - popis mesta zapisa podataka
  - vrijednosti podataka spremaju se u različite spremnike
    - register, priručne memorije, radnu memoriju

## Generiranje ciljnog programa na temelju troadresnih naredbi

$R := P \text{ op } D$

- **op** - oznaka operacije
- **R** - simboličko ime varijable rezultata
- **P** - simboličko ime varijable prvog operanda
- **D** - simboličko ime varijable drugog operanda
- različite operacije
  - aritmetičke, logičke

$$R := P \text{ op } D$$

### 1. Odredi mjesto $M$ zapisa rezultata $R$

- Prednost se daje registrima
- Nije li rezultat moguće spremiti u register, on se spremi u memoriju
- Sljedeći algoritam izabere mjesto  $M$  zapisa rezultata  $R$

```
ako (( Vrijednost varijable  $P$  je u registru  $Dx$  )
    && ( Registar  $Dx$  nema pridruženih drugih vrijednosti )
    && ( Varijabla  $P$  se ne koristi u nastavku programa ))
{
    Mjesto zapisa  $M$  rezultata  $R$  je register  $Dx$ ;
    Zapiši u opisnik podataka da varijabla  $P$  nije u registru  $Dx$ ;
}
```

$$R := P \text{ op } D$$

### 1. Odredi mjesto $M$ zapisa rezultata $R$

- Prednost se daje registrima
- Nije li rezultat moguće spremiti u registar, on se spremi u memoriju
- Sljedeći algoritam izabere mjesto  $M$  zapisa rezultata  $R$

inače ako ( Registr  $Dx$  je slobodan )  
Mjesto zapisa  $M$  rezultata  $R$  je registar  $Dx$ ;

$$R := P \text{ op } D$$

### 1. Odredi mjesto $M$ zapisa rezultata $R$

- Prednost se daje registrima
- Nije li rezultat moguće spremiti u registar, on se spremi u memoriju
- Sljedeći algoritam izabere mjesto  $M$  zapisa rezultata  $R$

inače ako (( Rezultat  $R$  se koristi u nastavku programa )

    || ( Operacija  $\text{op}$  zahtijeva uporabu registra ))

{

    Izaberi za mjesto zapisa  $M$  rezultata  $R$  jedan od zauzetih registara  $Dx$  u kojem je spremljena vrijednost  $V$ ;

$$R := P \text{ op } D$$

### 1. Odredi mjesto $M$ zapisa rezultata $R$

- Prednost se daje registrima
- Nije li rezultat moguće spremiti u registar, on se spremi u memoriju
- Sljedeći algoritam izabere mjesto  $M$  zapisa rezultata  $R$

ako (( Vrijednost  $V$  varijable  $Z$  nije spremljena u memoriju )  
    && ( Varijabla  $Z$  koristi se u nastavku programa ))

*GenerirajNaredbu( MOVE Dx, Z );*

$$R := P \text{ op } D$$

### 1. Odredi mjesto $M$ zapisa rezultata $R$

- Prednost se daje registrima
- Nije li rezultat moguće spremiti u registar, on se spremi u memoriju
- Sljedeći algoritam izabere mjesto  $M$  zapisa rezultata  $R$

```
{  
    Zapiši u opisnik podataka da varijabla  $Z$  nije u registru  $Dx$ ;  
    Zapiši u opisnik podataka da je varijabla  $Z$  u memoriji;  
}  
}
```

$$R := P \text{ op } D$$

### 1. Odredi mjesto $M$ zapisa rezultata $R$

- Prednost se daje registrima
- Nije li rezultat moguće spremiti u registar, on se spremi u memoriju
- Sljedeći algoritam izabere mjesto  $M$  zapisa rezultata  $R$

inače

{

Mjesto zapisa  $M$  rezultata  $R$  jest memorija;  
Zapiši u opisnik podataka da je varijabla  $R$  u memoriji;

}

$$R := P \text{ op } D$$

### 1. Odredi mjesto $M$ zapisa rezultata $R$

- Prednost se daje registrima
- Nije li rezultat moguće spremiti u registar, on se spremi u memoriju
- Sljedeći algoritam izabere mjesto  $M$  zapisa rezultata  $R$

### 2. U opisniku podataka pronadi sva mjesta $p$ gdje je spremljena vrijednost varijable $P$

- Prednost se daje registrima
- Ako je mjesto  $p$  različito od mesta zapisa rezultata  $M$  izabranog u koraku (1), onda se generira naredba:

MOVE  $p, M$

$$R := P \text{ op } D$$

3.U opisniku podataka pronađi sva mesta  $d$  gdje je spremljena vrijednost varijable  $D$ .

1.Prednost se daje registrima

2.Generira se naredba:

OP  $d, M$

- U opisnik podataka zapiše se da je varijabla  $R$  spremljena na mjestu  $M$ .
- Ako je  $M$  registar, onda se u opisnik registara zapiše da registar  $M$  sadrži vrijednost varijable  $R$ .
- U opisnike se zapiše da ostala mesta ne sadrže vrijednost varijable  $R$ , jer je naredba  $R := P \text{ op } D$  promijenila njezinu vrijednost.

$$R := P \text{ op } D$$

4. Ako se nakon izvedene naredbe  $R := P \text{ op } D$  vrijednosti varijabli  $P$  i  $D$  ne koriste u nastavku programa

i ako su njihove vrijednosti u registrima

onda se u opisnike zapiše da registri više ne sadrže vrijednosti tih varijabli

## Generiranje ciljnog programa na temelju troadresnih naredbi

- **Na sličan način**
  - unarni operator  $R := \text{op } P$
- **Naredba preslikavanja  $R:=P$** 
  - varijabla  $P$  je u registru  $Dx$ 
    - ne generira se nijedna naredba ciljnog programa
    - promijene se podaci u opisnicima
      - u opisnike se zapiše da je varijabla  $R$  isključivo u registru  $Dx$
      - ne koristi li se varijabla  $P$  u nastavku programa, u opisnike se zapiše da varijabla  $P$  nije u registru  $Dx$

## Generiranje ciljnog programa na temelju troadresnih naredbi

- **Naredba preslikavanja  $R:=P$** 
  - **varijabla  $P$  je isključivo u memoriji**
    - varijabla  $R$  se ne koristi neposredno nakon naredbe preslikavanja
      - generira se naredba **MOVE  $P, R$**
    - varijabla  $R$  se koristi neposredno nakon naredbe preslikavanja
      - izabere se odgovarajući registar  $Dx$  u koji se spremi vrijednost varijable  $P$
      - generira se naredba **MOVE  $P, Dx$**
      - u opisnike registara i podataka zapiše se da registar  $Dx$  sadrži vrijednost varijable  $R$
      - varijabla  $P$  se koristi u nastavku programa
        - » u opisnike se zapiše da je varijabla  $P$  u registru  $Dx$

# Generiranje ciljnog programa na temelju troadresnih naredbi

## **IZVORNI PROGRAM**

**$m = \underline{(k-n)} + \underline{(k+o)} + \underline{(k+o)}$  ;**

## **MEĐUKÔD**

```
pri1 := k-n
pri2 := k+o
pri3 := pri1 + pri2
m    := pri3 + pri2
```

*Naredbe  
međukôda*

*Generirani ciljni  
program*

*Opisnik  
registara*

*Opisnik podataka*

*k,n,o : Memorija*

*pri1 := k-n*

ako (( Vrijednost varijable **P** je u registru **Dx** )  
  && ( Registr **Dx** nema pridruženih drugih vrijednosti )  
  && ( Varijabla **P** se ne koristi u nastavku programa ))  
{  
    Mjesto zapisa **M** rezultata **R** je registr **Dx**;  
    Zapiši u opisnik podataka da varijabla **P** nije u registru **Dx**;  
}

*Naredbe  
međukôda*

*Generirani ciljni  
program*

*Opisnik  
registara*

*Opisnik podataka*

*k,n,o : Memorija*

*pri1 := k-n*

**inače ako ( Registrar **Dx** je slobodan )  
Mjesto zapisa **M** rezultata **R** je registrar **Dx**;**

*Naredbe  
međukôda*

*Generirani ciljni  
program*

*Opisnik  
registara*

*Opisnik podataka*

*k,n,o : Memorija*

*pri1 := k-n*

## 2.U opisniku podataka pronađi sva mesta *p* gdje je spremljena vrijednost varijable *P*

- Prednost se daje registrima
- Ako je mjesto *p* različito od mesta zapisa rezultata *M* izabranog u koraku (1), onda se generira naredba:

**MOVE *p, M***

*Naredbe  
međukôda*

*Generirani ciljni  
program*

*Opisnik  
registara*

*Opisnik podataka*

***k,n,o : Memorija***

***pri1 := k-n***

**MOVE k, D0**

*Naredbe  
međukôda*

*Generirani ciljni  
program*

*Opisnik  
registara*

*Opisnik podataka*

*k,n,o : Memorija*

*pri1 := k-n*

**MOVE k, D0**

**3.U opisniku podataka pronađi sva mesta *d* gdje je spremljena vrijednost varijable *D*.**

1.Prednost se daje registrima.

2.Generira se naredba:

*OP d, M*

- U opisnik podataka zapiše se da je varijabla *R* spremljena na mjestu *M*.
- Ako je *M* registar, onda se u opisnik registara zapiše da registar *M* sadrži vrijednost varijable *R*.
- U opisnike se zapiše da ostala mesta ne sadrže vrijednost varijable *R*, jer je naredba *R := P op D* promijenila njezinu vrijednost.

*Naredbe  
međukôda*

*Generirani ciljni  
program*

*Opisnik  
registara*

*Opisnik podataka*

**pri1 := k-n**

**MOVE k, D0  
SUB n, D0**

**k,n,o : Memorija**

<i>Naredbe međukôda</i>	<i>Generirani ciljni program</i>	<i>Opisnik registara</i>	<i>Opisnik podataka</i>
<i>pri1 := k-n</i>	<b>MOVE k, D0</b> <b>SUB n, D0</b>	<b>D0 : pri1</b>	<i>k,n,o : Memorija</i>
<i>pri2 := k+o</i>	<b>MOVE k, D1</b> <b>ADD o, D1</b>	<b>D0 : pri1</b> <b>D1 : pri2</b>	<i>k,n,o : Memorija</i> <i>pri1 : D0</i> <i>pri2 : D1</i>
<i>pri3 := pri1+pri2</i>	<b>ADD D1, D0</b>	<b>D0 : pri3</b> <b>D1 : pri2</b>	<i>k,n,o : Memorija</i> <i>pri3 : D0</i> <i>pri2 : D1</i>
<i>m := pri3+pri2</i>	<b>ADD D1, D0</b>	<b>D0 : m</b>	<i>k,n,o : Memorija</i> <i>m : D0</i>

## Generiranje ciljnog programa na temelju postfiksног sustava oznaka

- **Međukôd**

- **postfiksni sustav oznaka**

- sintaksno stablo djelomično izravnato
    - redoslijed operatora = redoslijed njihovog izvođenja
    - izravnava se sintaksno stablo
    - generiraju se naredbe ciljnog programa

- **izravnavanje sintaksnog stabla**

- ostvaruje se primjenom potisnog stoga
    - na stog se stavljaju operandi i operatori
    - postupak upravljanja potisnim stogom sličan je postupku računanja izraza zadanog primjenom postfiksног sustava oznaka

## Generiranje ciljnog programa na temelju postfiksnog sustava oznaka

### IZVORNI PROGRAM

```
ako ( a > b )
      c = b + d * 4;
inače
      c = a + d * 4;
```

### MEĐUKÔD

<i>Redni broj</i>	1	2	3	4	5	6	7	8	9	10	11	12
<i>Operandi i operatori</i>	a	b	-	15	JPP	c	b	d	4	*	+	:=

<i>Nastavak međukôda</i>	13	14	15	16	17	18	19	20	21	22
	22	JMP	c	a	d	4	*	+	:=	

## Generiranje ciljnog programa na temelju postfiksnog sustava oznaka

### 1) Pročitan operand

**Stavi**

pročitani znak međukôda na vrh stoga

**Pomakni**

glavu za čitanje na sljedeći znak

### 2) Pročitan operator

**Uzmi**

s vrha stoga zadani broj operanada

**Generiraj**

naredbe ciljnog programa

**Stavi**

rezultirajući operand na vrh stoga

## Generiranje ciljnog programa na temelju postfiksног sustava oznaka

- **Jednostavni generator ciljnog programa**

- koristi samo jedan podatkovni registar D0
- vrijednosti međurezultata  $priX$  spremi u memoriju
- ne koristi opisnike registara i podataka
- koristi najjednostavniji postupak generiranja ciljnog programa primjenom unaprijed definiranih uzoraka
- uzorak za binarne operacije
  1. na temelju oznake operanda spremljene na vrhu stoga generira se naredba dohvata vrijednosti prvog operanda iz memorije u registar D0
  2. na temelju sljedeće oznake operanda generira se naredba koja obavlja zadanu binarnu operaciju koristeći vrijednost drugog operanda spremljenu u memoriji
  3. generira se naredba koja spremi vrijednost rezultata u memoriju
- **velik broj prekomjernih pristupa podacima u memoriji**
  - optimirati se postupkom zasnovanim na prozorчиću



1	2	3	4	5	6	7	8	9	10	11	12
a	b	-	15	JPP	c	b	d	4	*	+	:=

*Naredbe ciljnog  
programa*

*Znak*

*Sadržaj stoga*

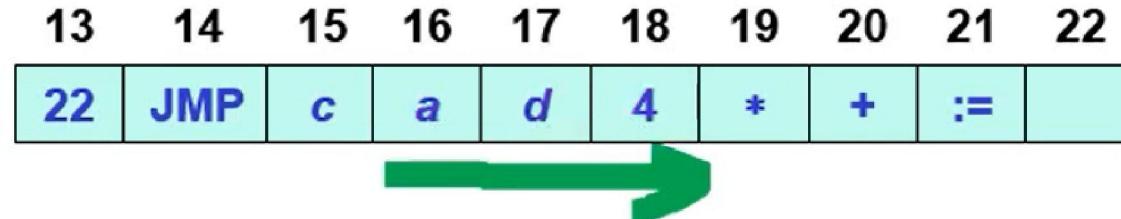
a		▽ a
b		▽ a b
-	MOVE b, D0 SUB a, D0 MOVE D0, pri1	▽ pri1
15		▽ pri1 15
JPP	BR POS, Ozn15	▽
c		▽ c
b		▽ c b
d		▽ c b d
4		▽ c b d 4

1 2 3 4 5 6 7 8 9 10 11 12

a	b	-	15	JPP	c	b	d	4	*	+	:=
---	---	---	----	-----	---	---	---	---	---	---	----

Znak *Naredbe ciljnog  
programa* *Sadržaj stoga*

*	<b>MOVE #4, D0</b> <b>MUL d, D0</b> <b>MOVE D0, pri2</b>	<span style="color: green;">▽ c b</span> <span style="color: green;">▽ c b pri2</span>
+	<b>MOVE pri2, D0</b> <b>ADD b, D0</b> <b>MOVE D0, pri3</b>	<span style="color: green;">▽ c</span> <span style="color: green;">▽ c pri3</span>
:=	<b>MOVE pri3, D0</b> <b>MOVE D0, c</b>	<span style="color: green;">▽</span> <span style="color: green;">▽</span>



*Naredbe ciljnog  
programa*

*Znak*

*Sadržaj stoga*

22		$\nabla 22$
JMP	BR Ozn22	$\nabla$
c	Ozn15:	$\nabla c$
a		$\nabla c a$
d		$\nabla c a d$
4		$\nabla c a d 4$

13 14 15 16 17 18 19 20 21 22

22	JMP	c	a	d	4	*	+	:=	
----	-----	---	---	---	---	---	---	----	--

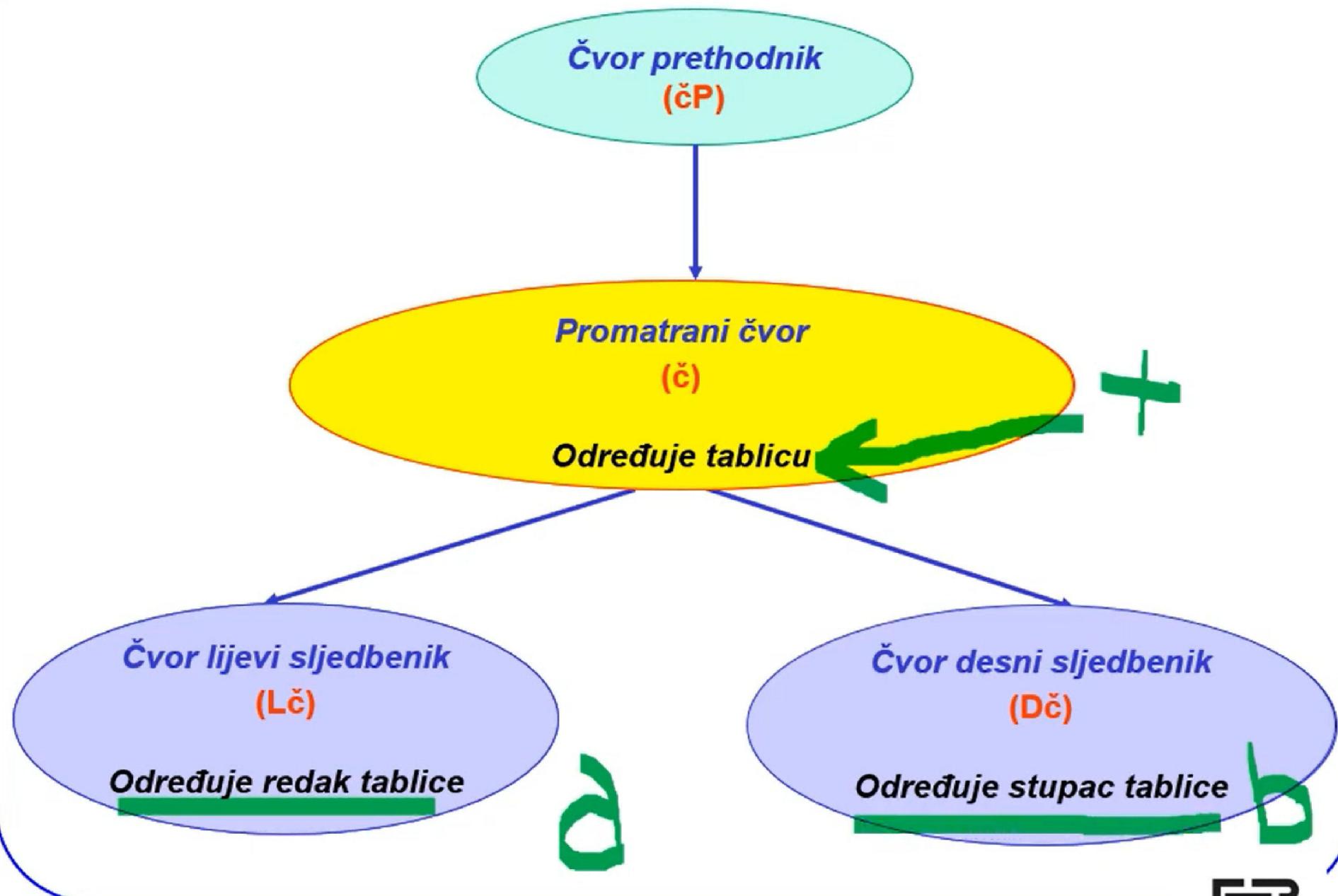
Znak *Naredbe ciljnog programa* *Sadržaj stoga*

*	MOVE #4, D0 MUL d, D0 MOVE D0, pri2	▽ c a ▽ c a pri2
+	MOVE pri2, D0 ADD b, D0 MOVE D0, pri3	▽ c ▽ c pri3
:=	MOVE pri3, D0 MOVE D0, c	▽ ▽

- **Tijekom generiranja ciljnog programa**
  - **izravna se sažeto sintaksno stablo**
    - odredi se redoslijed izvođenja operacija
    - odredi se redoslijed primjene operanada
  - **obilazak čvorova sažetog sintaksnog stabla**
    - započinje korijenom
    - generiraju se naredbe ciljnog programa
    - mijenjaju se oznake unutrašnjih čvorova
    - početne oznake čvorova stabla
      - svi unutrašnji čvorovi označeni su operatorima
      - listovi su označeni varijablama
    - nakon generiranja ciljnog programa za podstablo
      - korijen podstabla označi se imenom регистра u kojem je spremljen rezultat
    - vrijednost регистра spremi se u memoriju
      - korijen podstabla označi se imenom varijable

- **Rad generatora ciljnog programa**
  - upravlja se primjenom tablica
  - binarne operacije
    - koriste dvodimenzionalne tablice
      - retci i stupci su označeni varijablom, operatorom ili registrom
      - oznaka čvora određuje operaciju
      - čvorovi sljedbenici određuju redak i stupac tablice

## Generiranje ciljnog programa na temelju sažetog sintaksnog stabla



- **U elementima tablice zadaje se:**
  1. Naredbe ciljnog programa koje se generiraju
  2. Put obilaska sažetog sintaksnog stabla
    - u elementu tablice zadano je koji se čvor sljedeći obilazi
    - moguće je ponovo obići
      - isti čvor  $o(c)$
      - lijevi čvor sljedbenik  $o(Lc)$
      - desni čvor sljedbenik  $o(Dc)$
      - čvor prethodnik  $o(cP)$

### 3. Način označavanja čvorova

- u elementu tablice zadana je promjena oznake čvora
- moguće je promijeniti oznaku
  - promatranog čvora
  - lijevog čvora sljedbenika
  - desnog čvora sljedbenika
- promjena oznake čvora zapisuje se na način **čvor = Oznaka**
  - **čvor**
    - » jedna od oznaka **č**, **Lč** ili **Dč**
  - **oznaka**
    - » ime privremene varijable ili naziv registra
  - **č = D0**
    - » nova oznaka čvora je registar **D0**
  - **Dč = pri5**
    - » nova oznaka desnog čvora sljedbenika je ime privremene varijable **pri5**

Oznaka promatranog čvora č

Operacije \*, +

OP=ADD ili OP=MUL

Desni čvor sljedbenik Dč

Lijevi čvor sljedbenik Lč

		Varijabla Var2	Operator	Registar
Varijabla Var1	Generiraj	MOVE Var1, D0 OP Var2, D0	---	OP Var1, D0
Operator	Označi	č = D0	---	č = D0
Registar	Obidi	o(čP)	o(Dč)	o(čP)
	Generiraj	---	---	MOVE D0, priX
	Označi	---	---	Dč = priX
	Obidi	o(Lč)	o(Dč) ili o(Lč)	o(č)
	Generiraj	OP Var2, D0	MOVE D0, priX	Pogreška
	Označi	č = D0	Lč = priX	
	Obidi	o(čP)	o(č)	

Oznaka promatranog čvora č

Operacije –, /

OP=SUB ili OP=DIV

Desni čvor sljedbenik Dč

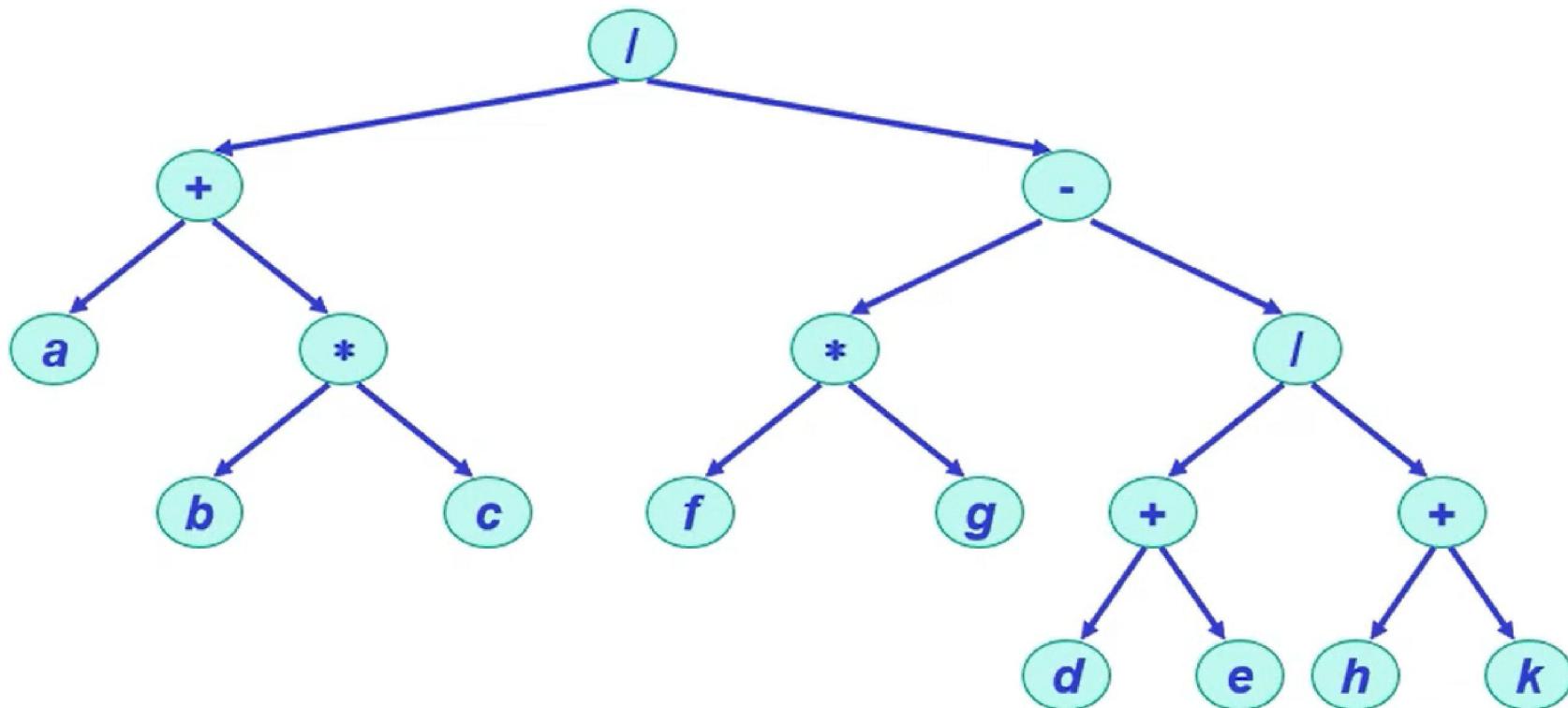
Lijevi čvor sljedbenik Lč

	Varijabla Var1	Varijabla Var2	Operator	Registar
Varijabla Var1	Generiraj	MOVE Var1, D0 OP Var2, D0	---	MOVE D0, priX
Operator	Označi	č = D0	---	Dč = priX
Registar	Obidi	o(čP)	o(Dč)	o(č)
	Generiraj	---	---	MOVE D0, priX
	Označi	---	---	Dč = priX
	Obidi	o(Lč)	o(Dč)	o(č)
	Generiraj	OP Var2, D0	Pogreška	
	Označi	č = D0		
	Obidi	o(čP)		

## IZVORNI PROGRAM

(*a* + *b* \* *c*) / (*f* \* *g* - ( *d* + *e* ) / (*h* + *k* ) ;

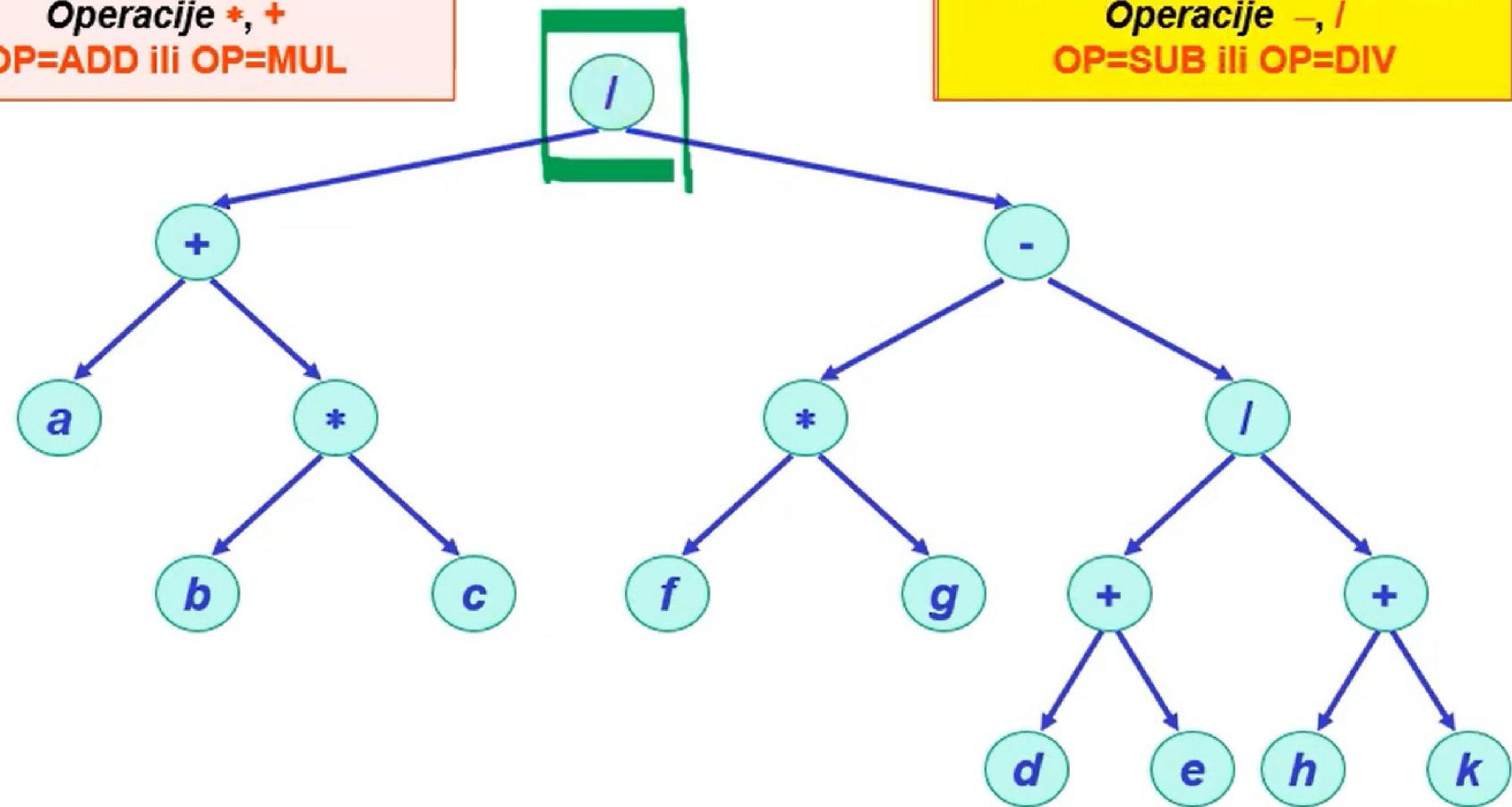
## MEĐUKÔD



Oznaka promatranog čvora č  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora č  
Operacije -, /  
OP=SUB ili OP=DIV

Ciljni program



Oznaka promatranog čvora č

Operacije –, /

OP=SUB ili OP=DIV

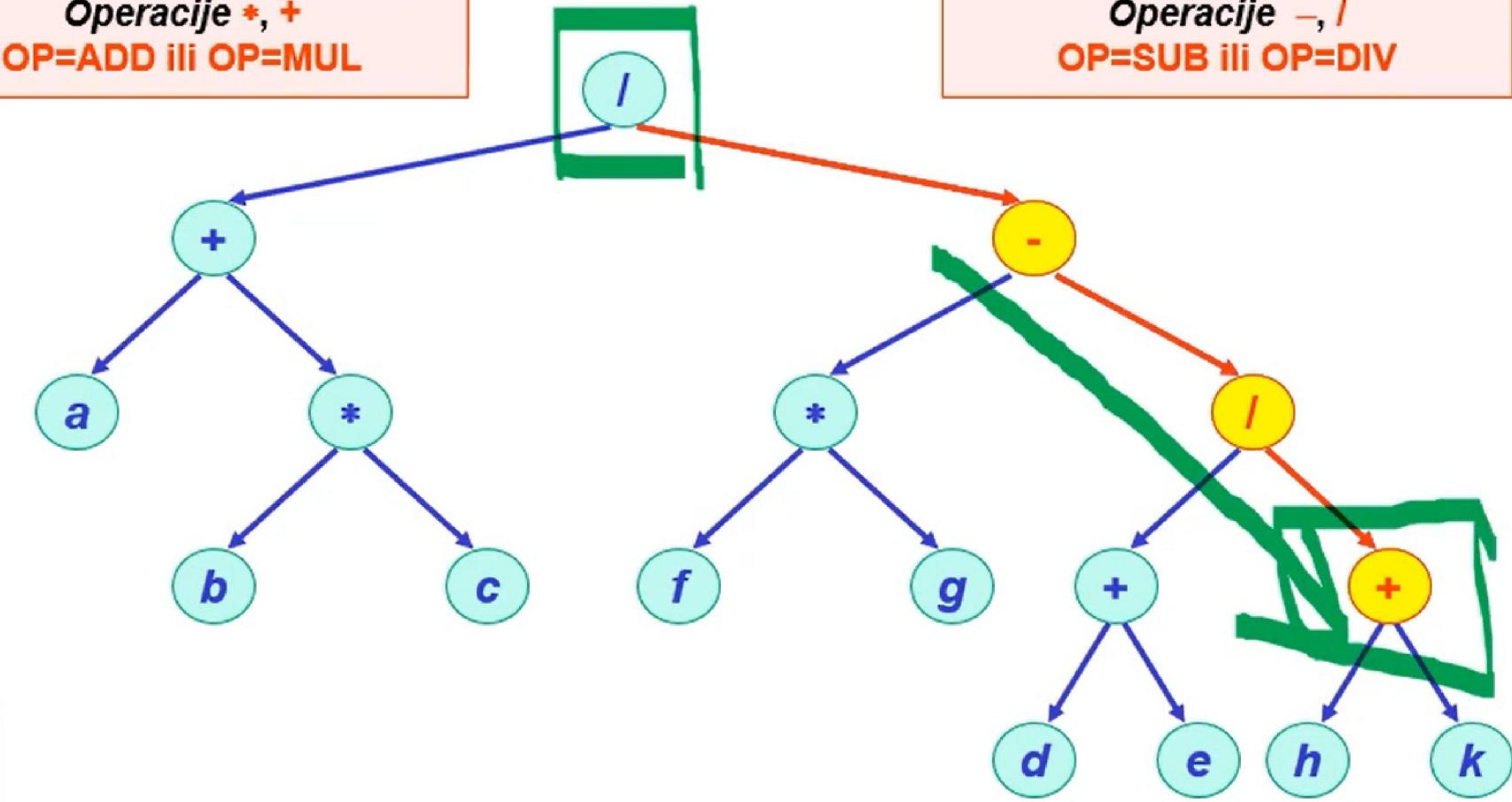
Desni čvor sljedbenik Dč

Lijevi čvor sljedbenik Lč		Varijabla Var1	Operator	Registar
Varijabla Var1	Generiraj	MOVE Var1, D0 OP Var2, D0	---	MOVE D0, priX
	Označi	č=D0	---	Dč=priX
	Obidi	o(čP)	o(Dč)	o(č)
Operator	Generiraj	---	---	MOVE D0, priX
	Označi	----	----	Dč=priX
	Obidi	o(Lč)	o(Dč)	o(č)
Registar	Generiraj	OP Var2, D0	Pogreška	
	Označi	č=D0		
	Obidi	o(čP)		

Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV

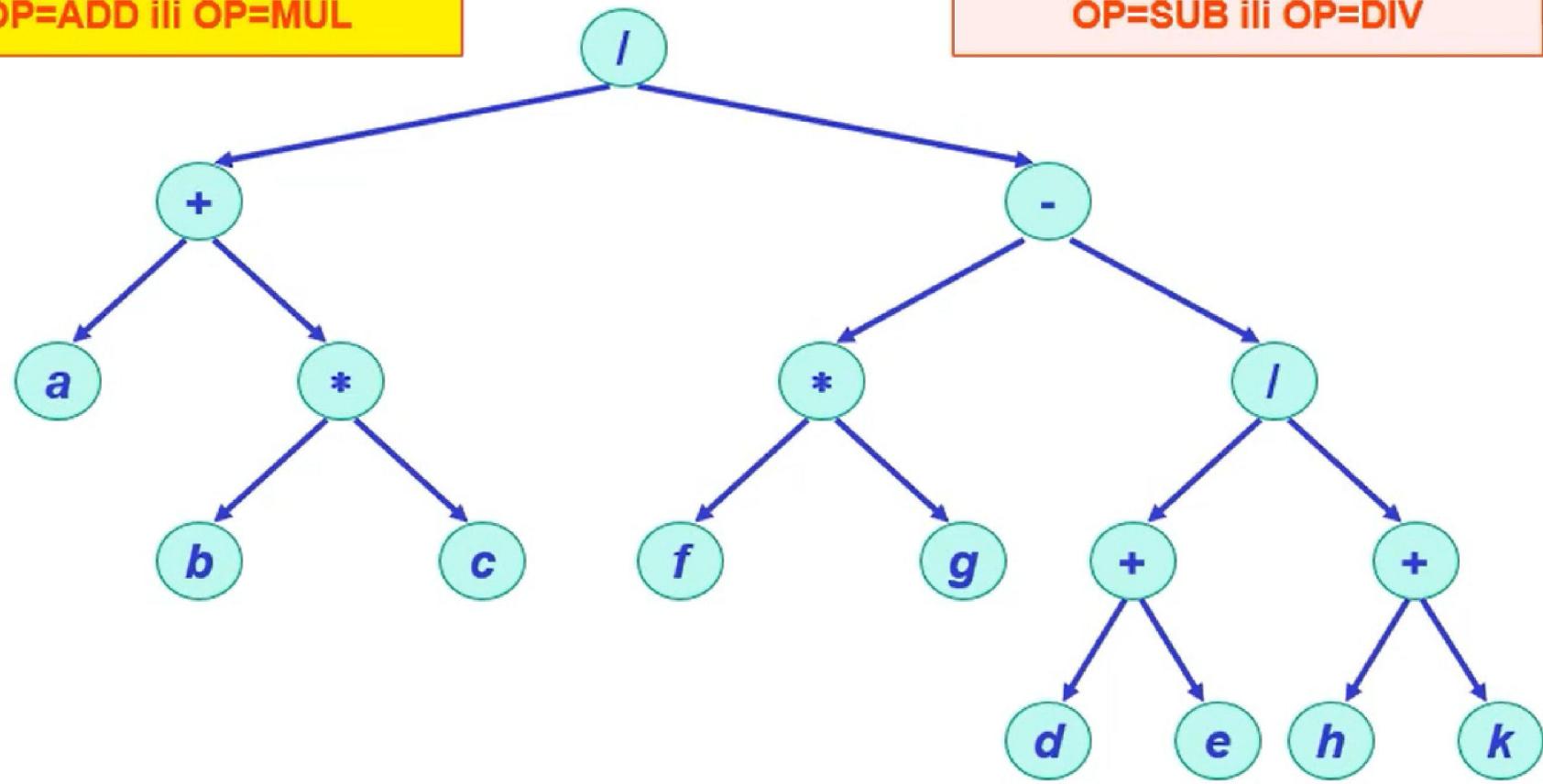
Ciljni program



Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV

Ciljni program



Oznaka promatranog čvora č

Operacije \*, +

OP=ADD ili OP=MUL

Desni čvor sljedbenik Dč

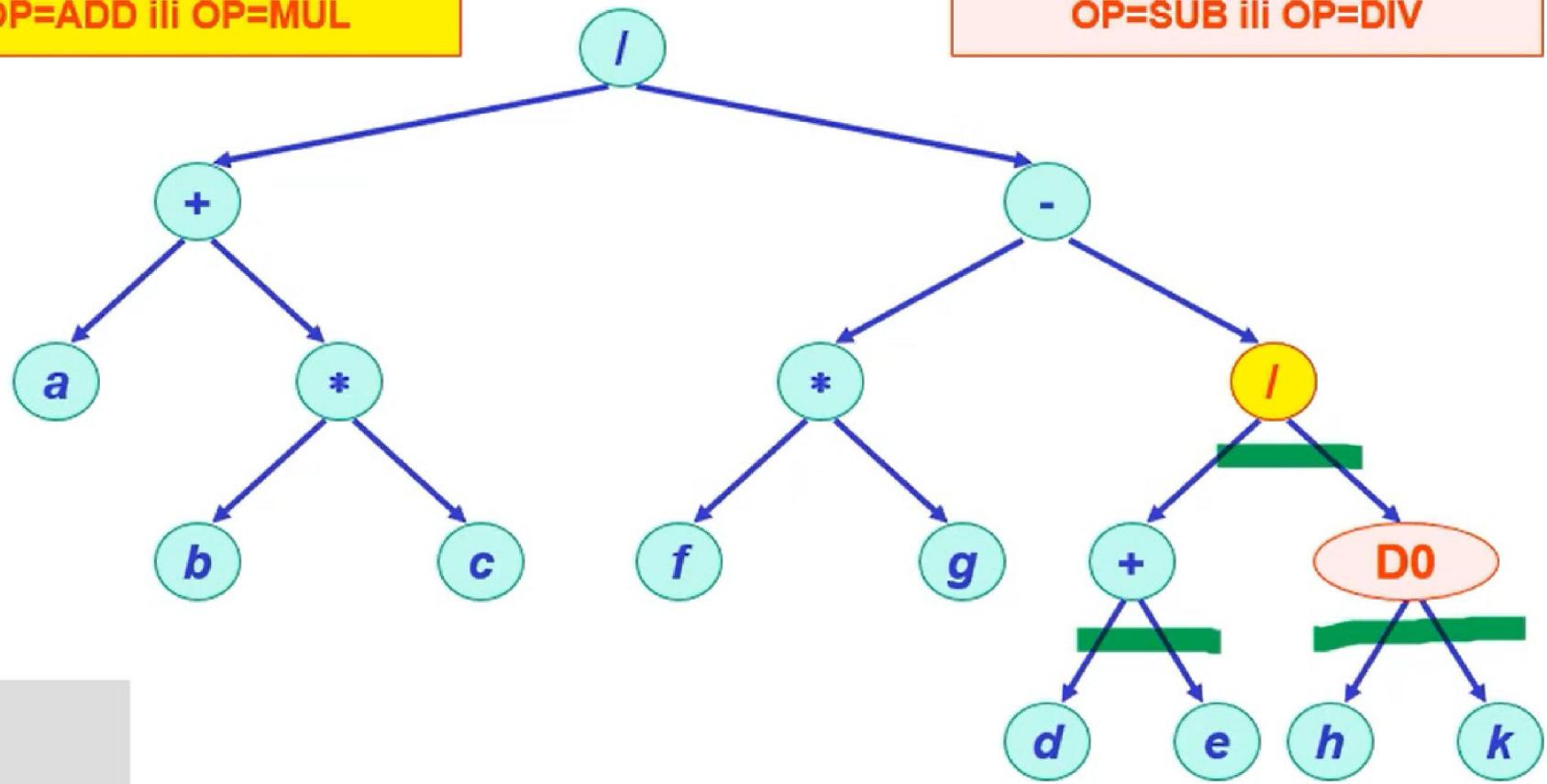
Lijevi čvor sljedbenik Lč

Varijabla Var1	Generiraj	Varijabla Var2	Operator	Registar
Operator	Označi	MOVE Var1, D0 OP Var2, D0	---	OP Var1, D0
	Obidi	o(čP)	o(Dč)	o(čP)
	Generiraj	---	---	MOVE D0, priX
Registar	Označi	---	---	Dč=priX
	Obidi	o(Lč)	o(Dč) ili o(Lč)	o(č)
	Generiraj	OP Var2, D0	MOVE D0, priX	Pogreška
Registar	Označi	č=D0	Lč=priX	
	Obidi	o(čP)	o(č)	



Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV



Ciljni program

MOVE h, D0  
ADD k, D0

Oznaka promatranog čvora č

Operacije –, /

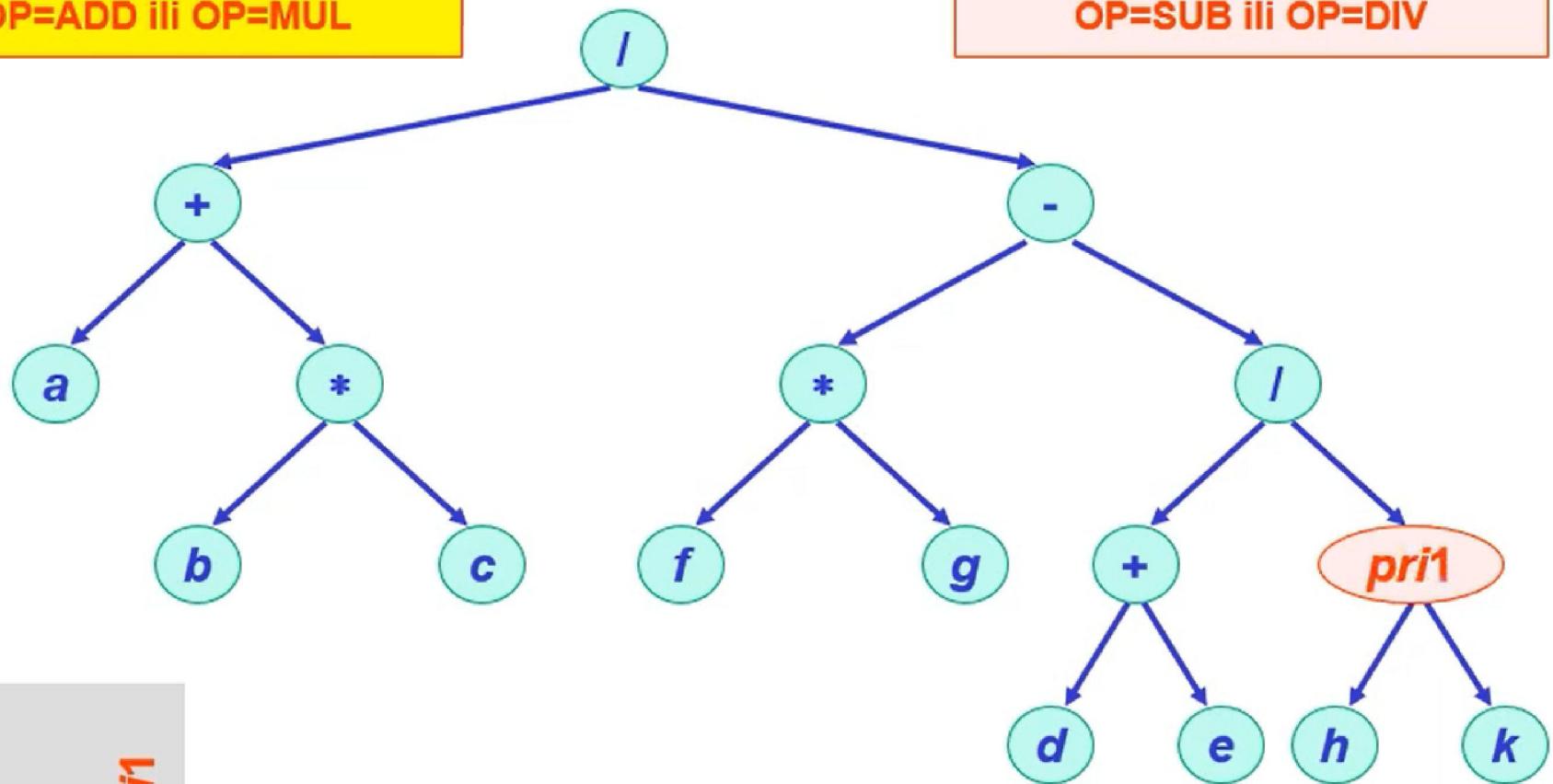
OP=SUB ili OP=DIV

Desni čvor sljedbenik Dč

Lijevi čvor sljedbenik Lč		Varijabla Var2	Operator	Registar
Varijabla Var1	Generiraj	MOVE Var1, D0 OP Var2, D0	---	MOVE D0, priX
	Označi	č=D0	---	Dč=priX
	Obidi	o(čP)	o(Dč)	o(č)
Operator	Generiraj	---	---	MOVE D0, priX
	Označi	---	---	Dč=priX
	Obidi	o(Lč)	o(Dč)	o(č)
Registar	Generiraj	OP Var2, D0	Pogreška	Pogreška
	Označi	č=D0		
	Obidi	o(čP)		

Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV



Ciljni program

MOVE h, D0  
ADD k, D0  
**MOVE D0, pri1**

Oznaka promatranog čvora č

Operacije –, /

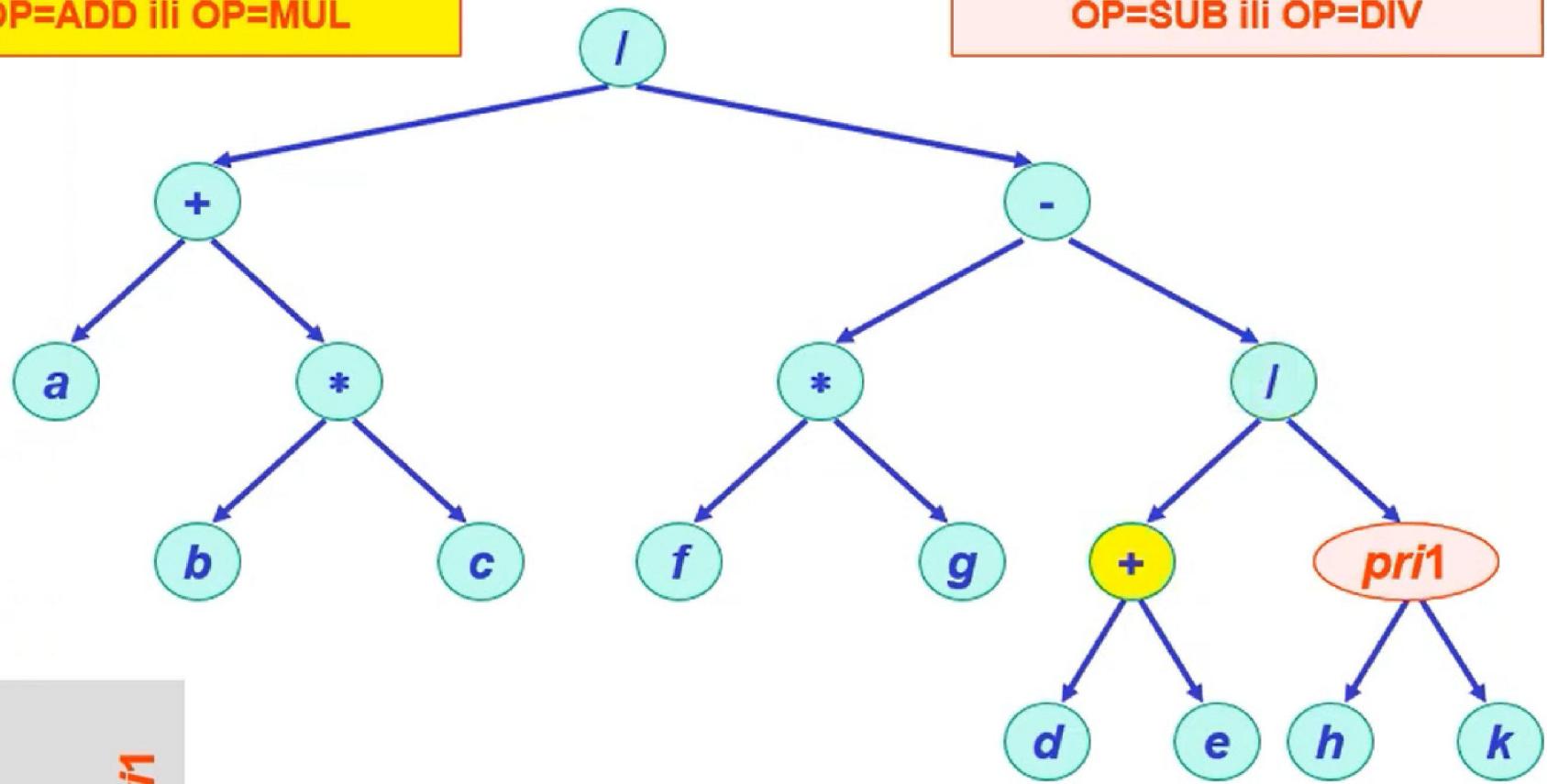
OP=SUB ili OP=DIV

Desni čvor sljedbenik Dč

Lijevi čvor sljedbenik Lč		Varijabla Var2	Operator	Registar
Varijabla Var1	Generiraj	MOVE Var1, D0 OP Var2, D0	---	MOVE D0, priX
	Označi	č=D0	---	Dč=priX
	Obidi	o(čP)	o(Dč)	o(č)
Operator	Generiraj	---	---	MOVE D0, priX
	Označi	---	---	Dč=priX
	Obidi	o(Lč)	o(Dč)	o(č)
Registar	Generiraj	OP Var2, D0	Pogreška	
	Označi	č=D0		
	Obidi	o(čP)		

Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV

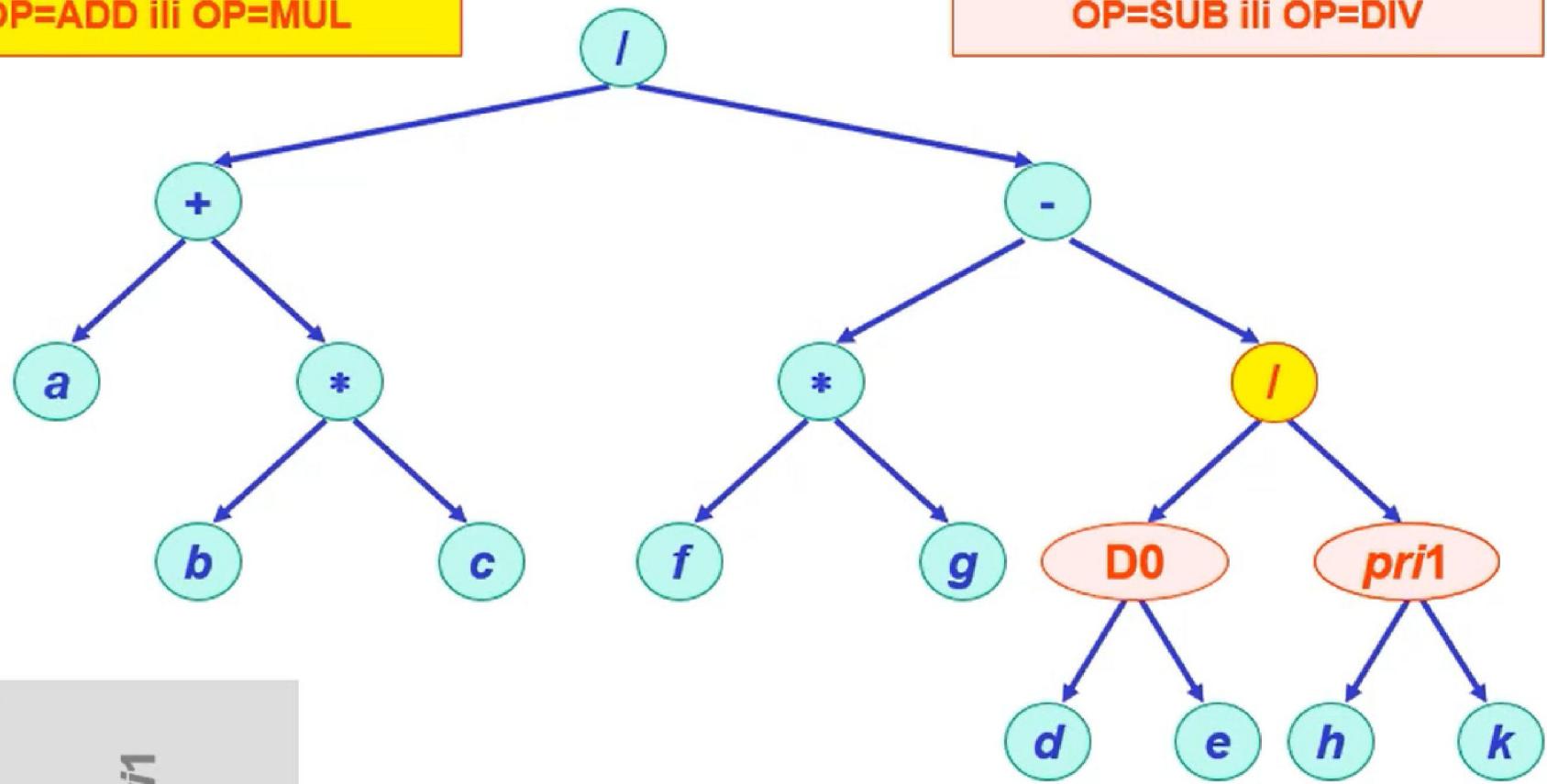


Ciljni program

MOVE h, D0  
ADD k, D0  
**MOVE D0, pri1**

Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV

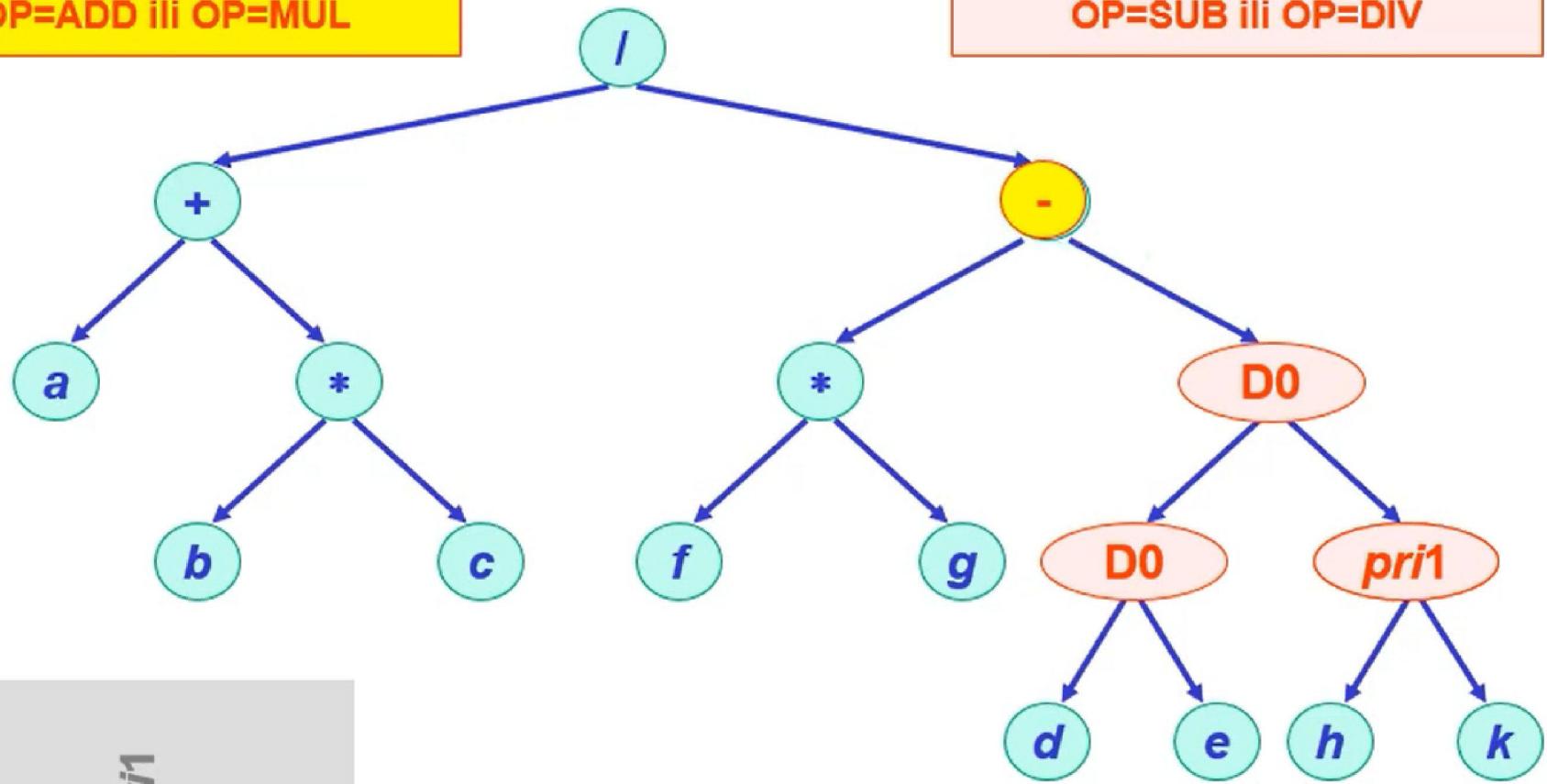


### Ciljni program

```
MOVE h, D0  
ADD k, D0  
MOVE D0, pri1  
MOVE d, D0  
ADD e, D0
```

Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV

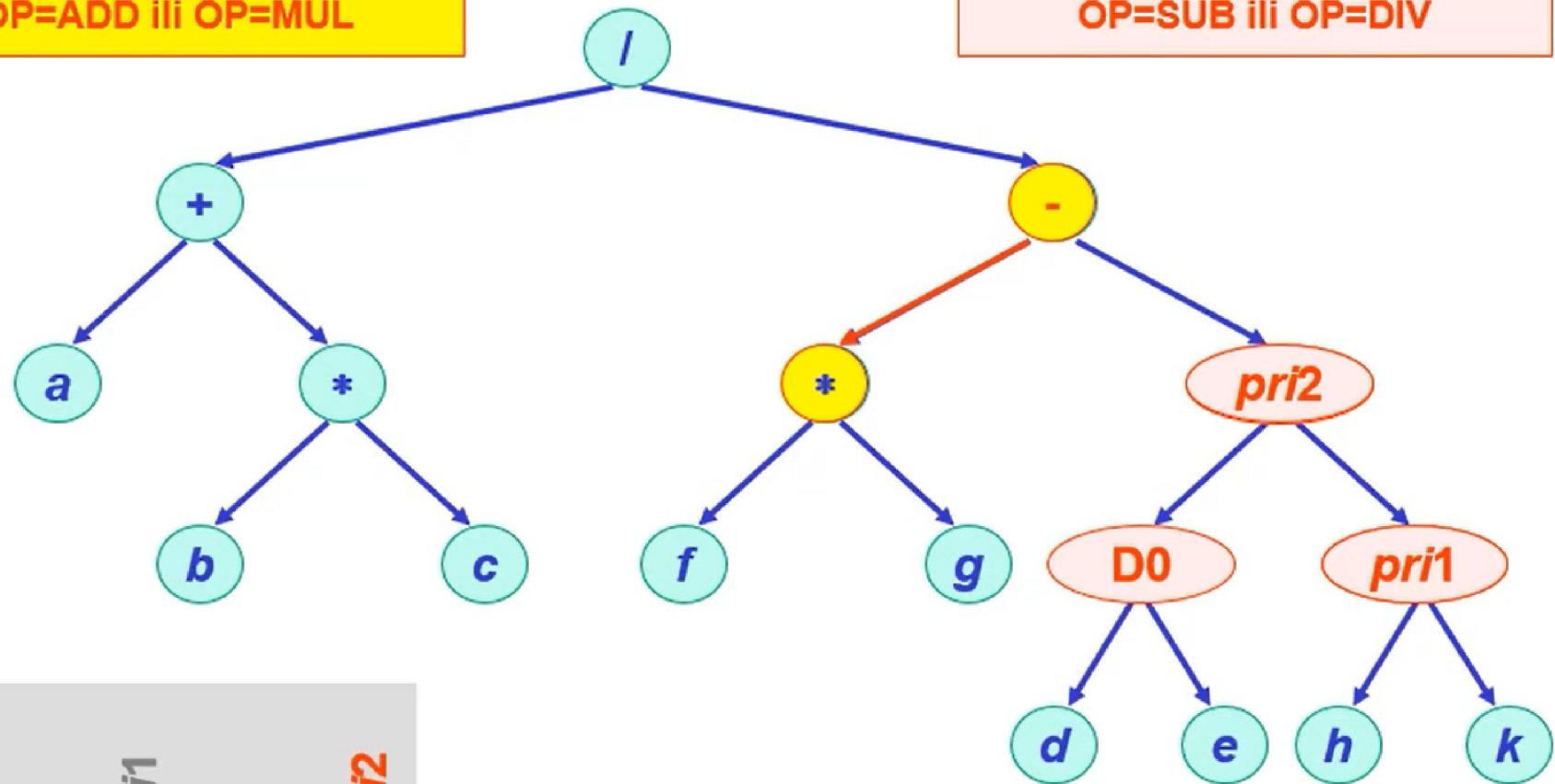


### Ciljni program

```
MOVE h, D0  
ADD K, D0  
MOVE D0, pri1  
MOVE d, D0  
ADD e, D0  
DIV pri1, D0
```

Oznaka promatranog čvora č  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora č  
Operacije -, /  
OP=SUB ili OP=DIV

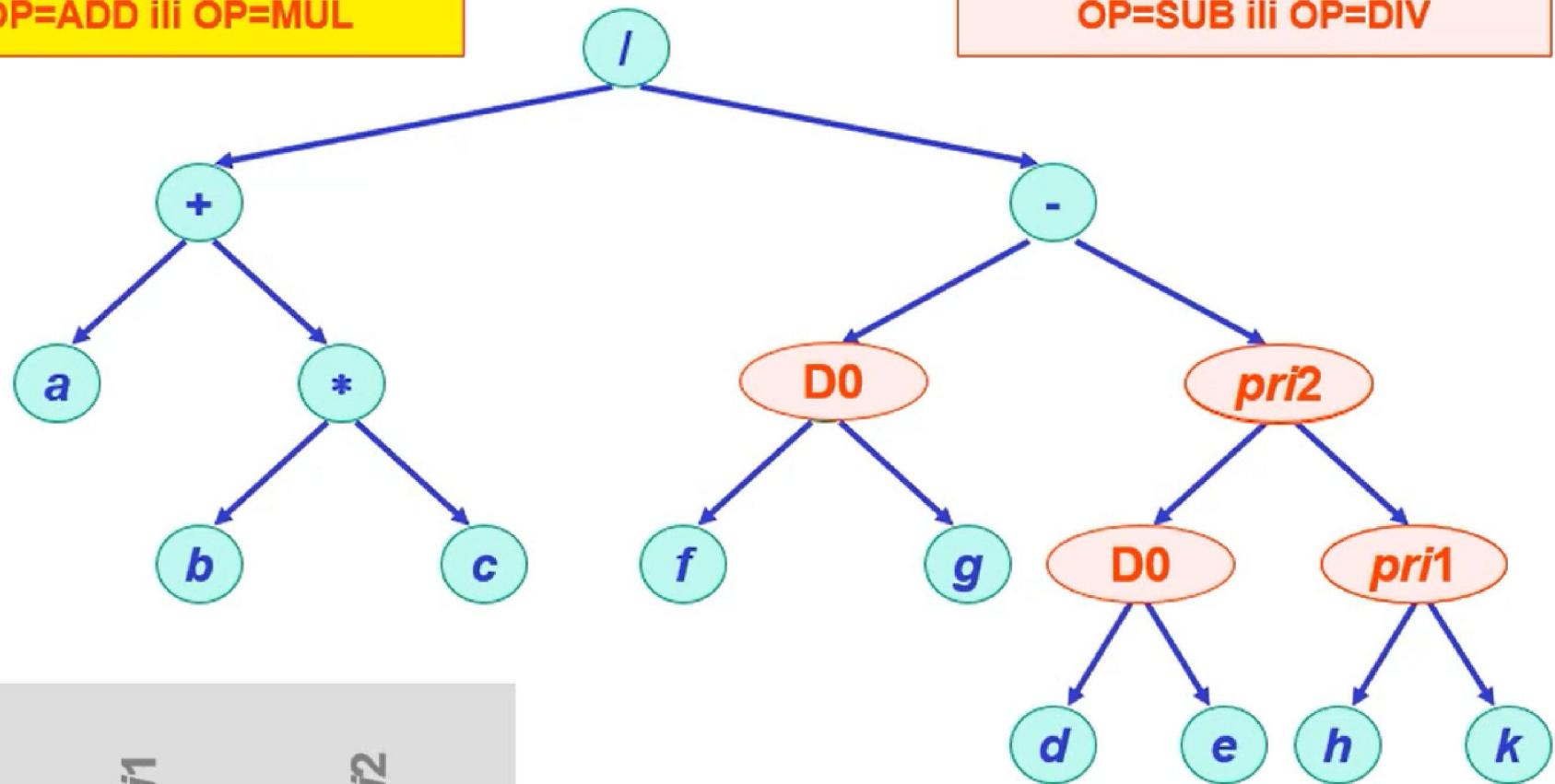


### Ciljni program

MOVE h, D0  
ADD k, D0  
MOVE D0, pri1  
MOVE d, D0  
ADD e, D0  
DIV pri1, D0  
**MOVE D0, pri2**

Oznaka promatranog čvora **č**  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
Operacije -, /  
OP=SUB ili OP=DIV

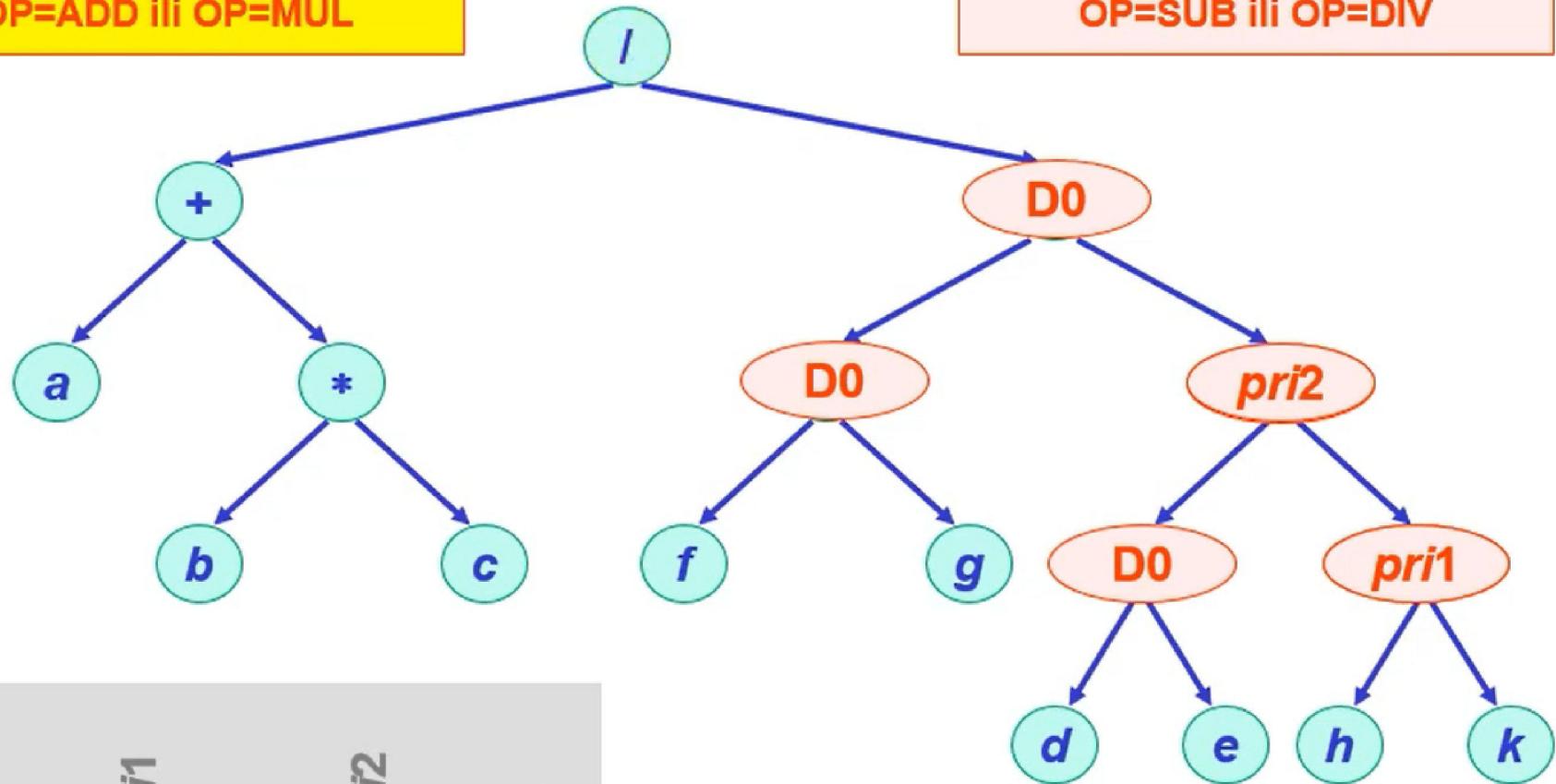


### Ciljni program

```
MOVE h, D0  
ADD k, D0  
MOVE D0, pri1  
MOVE d, D0  
ADD e, D0  
DIV pri1, D0  
MOVE D0, pri2  
MOVE f, D0  
MUL g, D0
```

Oznaka promatranog čvora č  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora č  
Operacije -, /  
OP=SUB ili OP=DIV



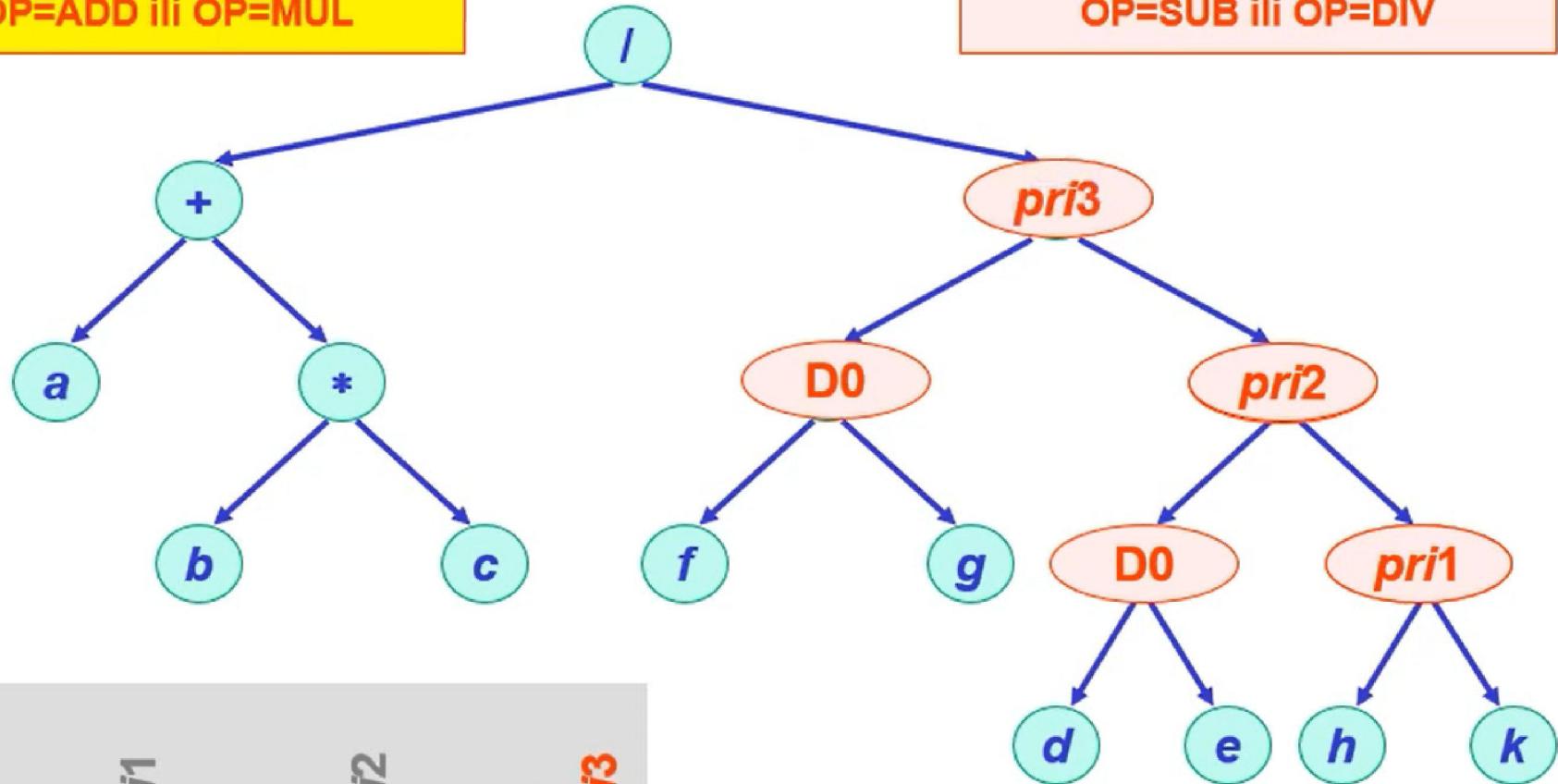
### Ciljni program

```

MOVE h, D0
ADD k, D0
MOVE D0, pri1
MOVE d, D0
ADD e, D0
DIV pri1, D0
MOVE D0, pri2
MOVE f, D0
MUL g, D0
SUB pri2, D0
  
```

Oznaka promatranog čvora č  
Operacije \*, +  
OP=ADD ili OP=MUL

Oznaka promatranog čvora č  
Operacije -, /  
OP=SUB ili OP=DIV



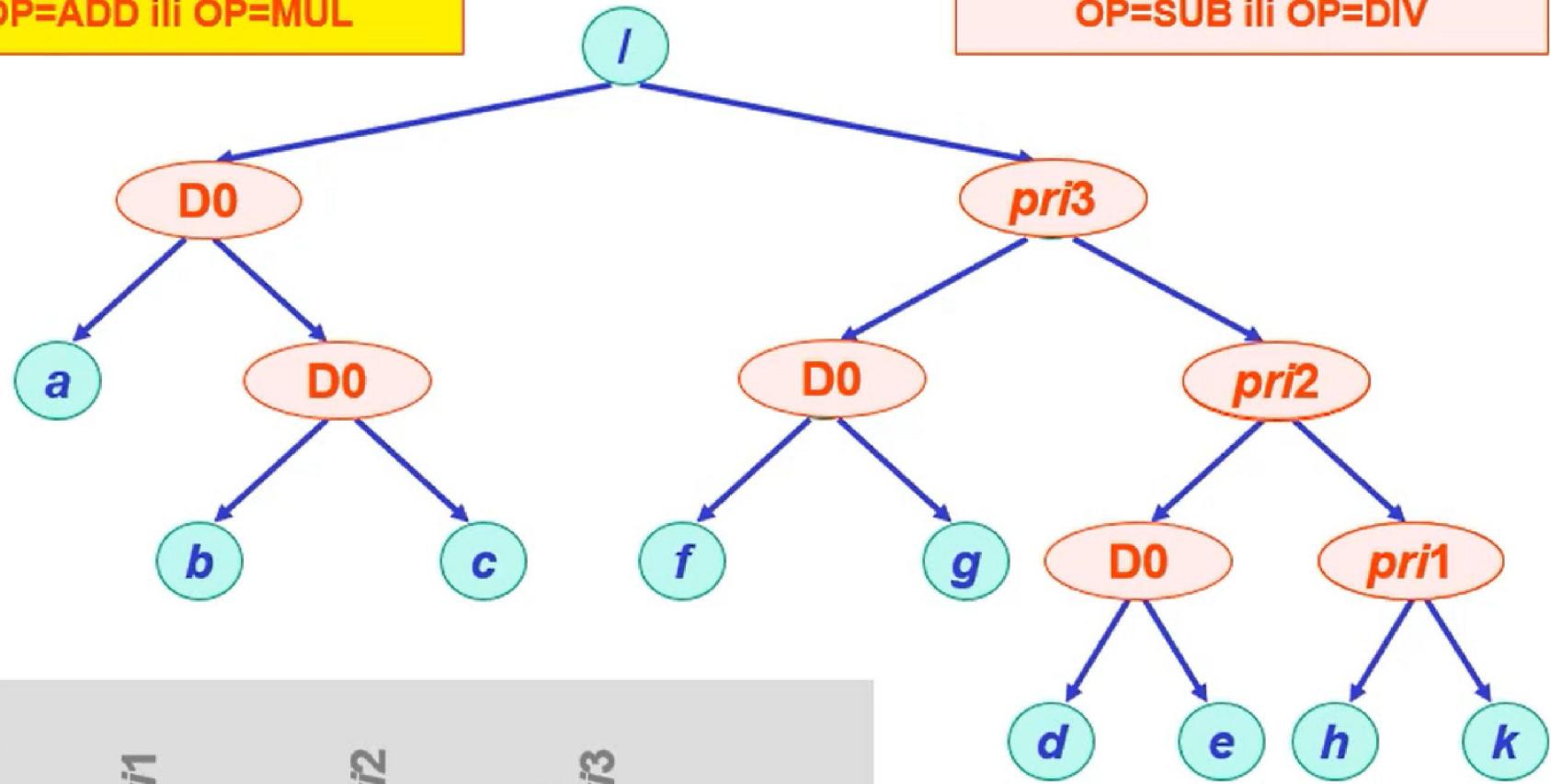
### Ciljni program

```

MOVE h, D0
ADD k, D0
MOVE D0, pri1
MOVE d, D0
ADD e, D0
DIV pri1, D0
MOVE D0, pri2
MOVE f, D0
MUL g, D0
SUB pri2, D0
MOVE D0, pri3
  
```

Oznaka promatranog čvora **č**  
 Operacije \*, +  
 OP=ADD ili OP=MUL

Oznaka promatranog čvora **č**  
 Operacije -, /  
 OP=SUB ili OP=DIV



### Ciljni program

```

MOVE h, D0
ADD k, D0
MOVE D0, pri1
MOVE d, D0
ADD e, D0
DIV pri1, D0
MOVE D0, pri2
MOVE f, D0
MUL g, D0
SUB pri2, D0
MOVE D0, pri3
MOVE b, D0
MUL c, D0
ADD a, D0
DIV pri3, D0
  
```

# Generator generatora ciljnog programa

- **Postupak generiranja ciljnog programa**
  - prevodenje međukôda u ciljni program
  - atributna prijevodna gramatika
  - primjena programa Yacc

# Generator generatora ciljnog programa

- Označavanje čvorova stabla i generiranje naredbi ciljnog programa
  - moguće je zadati produkcijama atributne prijevodne gramatike
    - međukôd u prefiksnom sustavu oznaka

# Generator generatora ciljnog programa

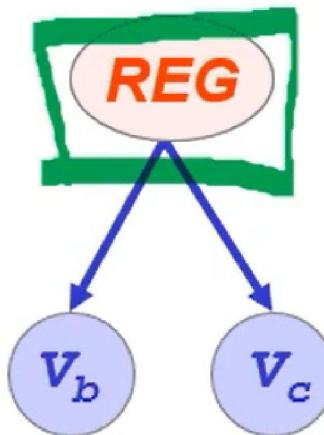
## IZVORNI PROGRAM

$(a + b * c) / (f * g - (d + e) / (h + k)) ;$



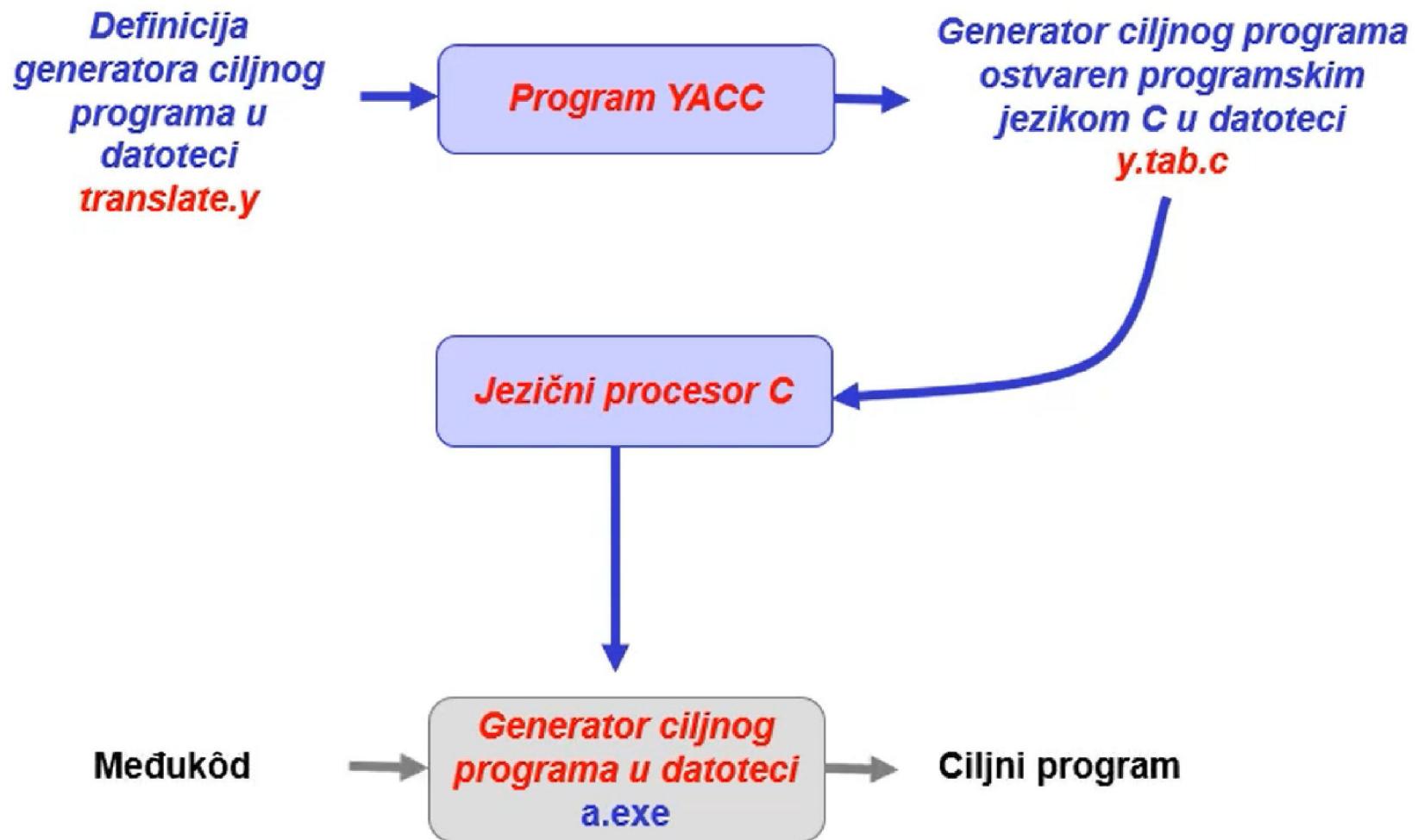
## MEĐUKÔD

$/ - / + V_h V_k + V_d V_e * V_f V_g + V_a *$



**MOVE var1, D0**  
**MUL var2, D0**

# Generator generatora ciljnog programa



# Generator generatora ciljnog programa

(1)  $V_{priX} \rightarrow REG$

{ Generiraj(MOVE D0, priX); }

(2)  $REG \rightarrow + V_{var1} V_{var2}$

{ Generiraj((MOVE var1, D0) || (ADD var2, D0)); }

(3)  $REG \rightarrow * V_{var1} V_{var2}$

{ Generiraj((MOVE var1, D0) || (MUL var2, D0)); }

(4)  $REG \rightarrow - V_{var1} V_{var2}$

{ Generiraj((MOVE var1, D0) || (SUB var2, D0)); }

(5)  $REG \rightarrow / V_{var1} V_{var2}$

{ Generiraj((MOVE var1, D0) || (DIV var2, D0)); }

(6)  $REG \rightarrow + V_{var1} REG$

{ Generiraj(ADD var1, D0); }

(7)  $REG \rightarrow + REG V_{var1}$

{ Generiraj(ADD var1, D0); }

(8)  $REG \rightarrow * V_{var1} REG$

{ Generiraj(MUL var1, D0); }

(9)  $REG \rightarrow * REG V_{var1}$

{ Generiraj(MUL var1, D0); }

(10)  $REG \rightarrow - V_{var1} REG$

{ Generiraj(SUB var1, D0); }

(11)  $REG \rightarrow / V_{var1} REG$

{ Generiraj(DIV var1, D0); }



# Generator generatora ciljnog programa

*Postupak parsiranja međukôda  
prefiksnog sustava oznaka*

*Primjena  
redukcije*

*Generirani  
ciljni program*

$$I - I + V_h \ V_k + V_d \ V_e * V_f \ V_g + V_a * V_b \ V_c$$

(2)      **REG**  $\rightarrow + V_{var1} \ V_{var2}$       { Generiraj((MOVE var1, D0) || (ADD var2, D0)); }

# Generator generatora ciljnog programa

Postupak parsiranja međukôda  
prefiksnog sustava oznaka

Primjena  
redukcije

Generirani  
ciljni program

$I - I + V_h V_k + V_d V_e * V_f V_g + V_a * V_b V_c$

(2)

MOVE h, D0  
ADD k, D0

$I - I [REG] + V_d V_e * V_f V_g + V_a * V_b V_c$

(1)  $V_{priX} \rightarrow REG$  { Generiraj(MOVE D0, priX); }

# Generator generatora ciljnog programa

*Postupak parsiranja međukôda prefiksnog sustava oznaka*

*Primjena redukcije*

*Generirani ciljni program*

$I - I + V_h V_k + V_d V_e * V_f V_g + V_a * V_b V_c$	(2)	<b>MOVE h, D0 ADD k, D0</b>
$I - I \text{REG} + V_d V_e * V_f V_g + V_a * V_b V_c$	(1)	<b>MOVE D0, pri1</b>
$I - I V_{pri1} + V_d V_e * V_f V_g + V_a * V_b V_c$	(2)	<b>MOVE d, D0 ADD e, D0</b>
$I - I V_{pri1} \text{REG} * V_f V_g + V_a * V_b V_c$	(11)	<b>DIV pri1, D0</b>
$I - \text{REG} * V_f V_g + V_a * V_b V_c$	(1)	<b>MOVE D0, pri2</b>
$I - V_{pri2} * V_f V_g + V_a * V_b V_c$	(3)	<b>MOVE f, D0 MUL g, D0</b>

# Generator generatora ciljnog programa

$/ - V_{pri2} REG + V_a * V_b V_c$	(10)	<b>SUB pri2, D0</b>
$/ REG + V_a * V_b V_c$	(1)	<b>MOVE D0, pri3</b>
$/ V_{pri3} + V_a * V_b V_c$	(3)	<b>MOVE b, D0</b> <b>MUL c, D0</b>
$/ V_{pri3} + V_a REG$	(6)	<b>ADD a, D0</b>
$/ V_{pri3} REG$	(11)	<b>DIV pri3, D0</b>
$REG$	(1)	<b>MOVE D0, pri4</b>
$V_{pri4}$		