



# Estrutura de Dados e Algoritmos

Construção de Árvores Binárias

- Construção:
  - Ler notação linear;
  - Alocar e ligar os nós;

ABG...C.DE.F....



Estrutura de Dados e Algoritmos

Prof. Nilton Correia da Silva

ABG..C.DE.F....

```
void Constroi(Tno *No)
{
    Dado = LerProximo();
    if(Dado != '-')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

A B G . . C . D E . F . . .

```
? | A | ?  
void Constrói(Raiz)  
{  
    Dado = 'A';  
    void main()  
    {  
        if(Dado != ':')  
        {  
            TNo *Raiz;  
            Raiz = AloqueNo();  
            Raiz->Dado = Dado;  
            Constrói(Raiz->Esq);  
            Constrói(Raiz->Dir);  
        }  
        else  
            Raiz = NULL;  
    }
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

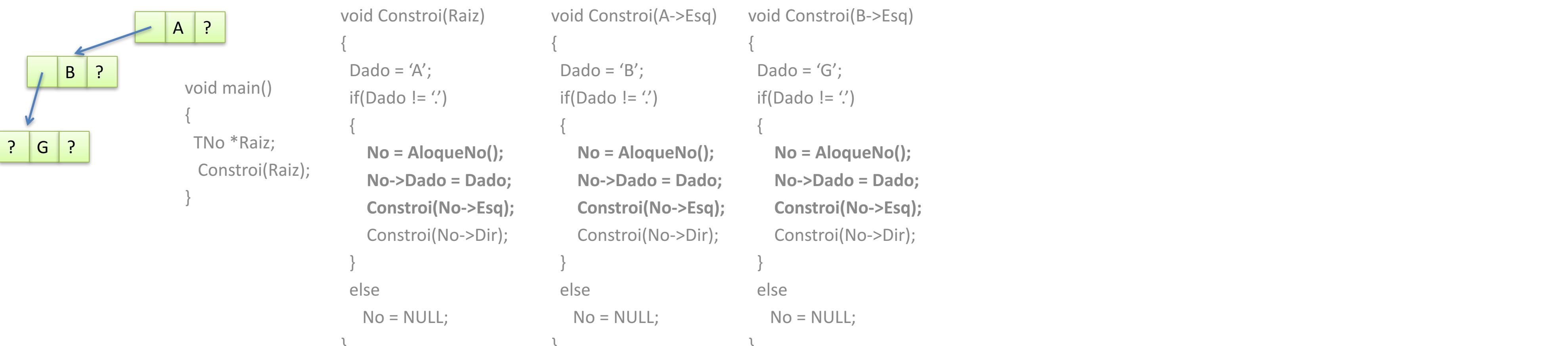
```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

```
void Constrói(Raiz)  
{  
    Dado = 'A';  
    if(Dado != ':')  
    {  
        No = AloqueNo();  
        No->Dado = Dado;  
        Constrói(No->Esq);  
        Constrói(No->Dir);  
    }  
    else  
        No = NULL;  
}
```

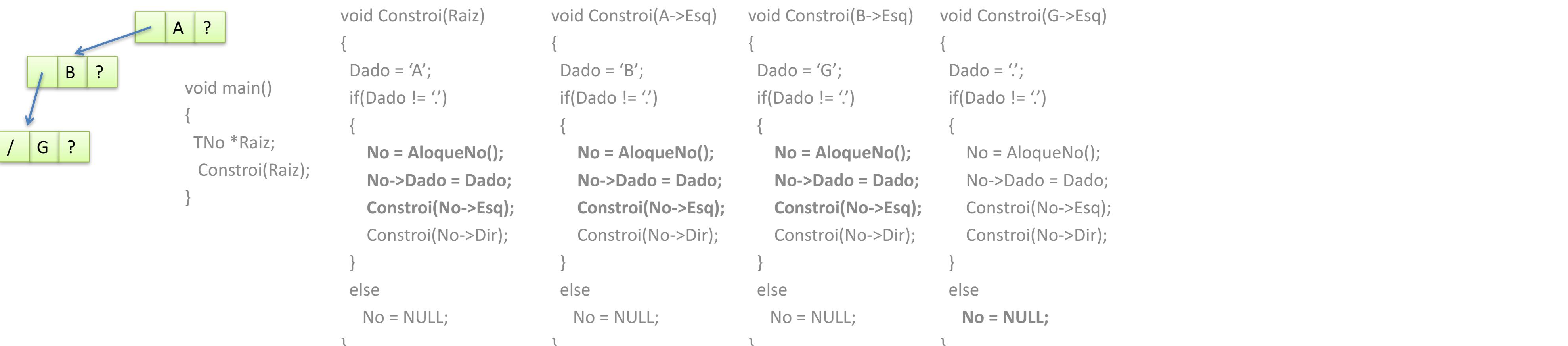
# ABG...C.DE.F....



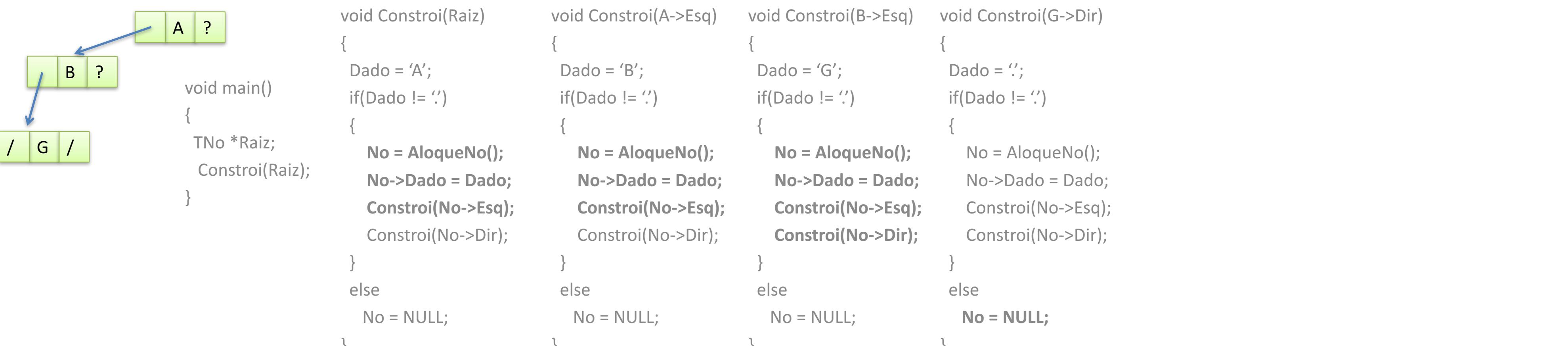
A B G . . C D E F . . .



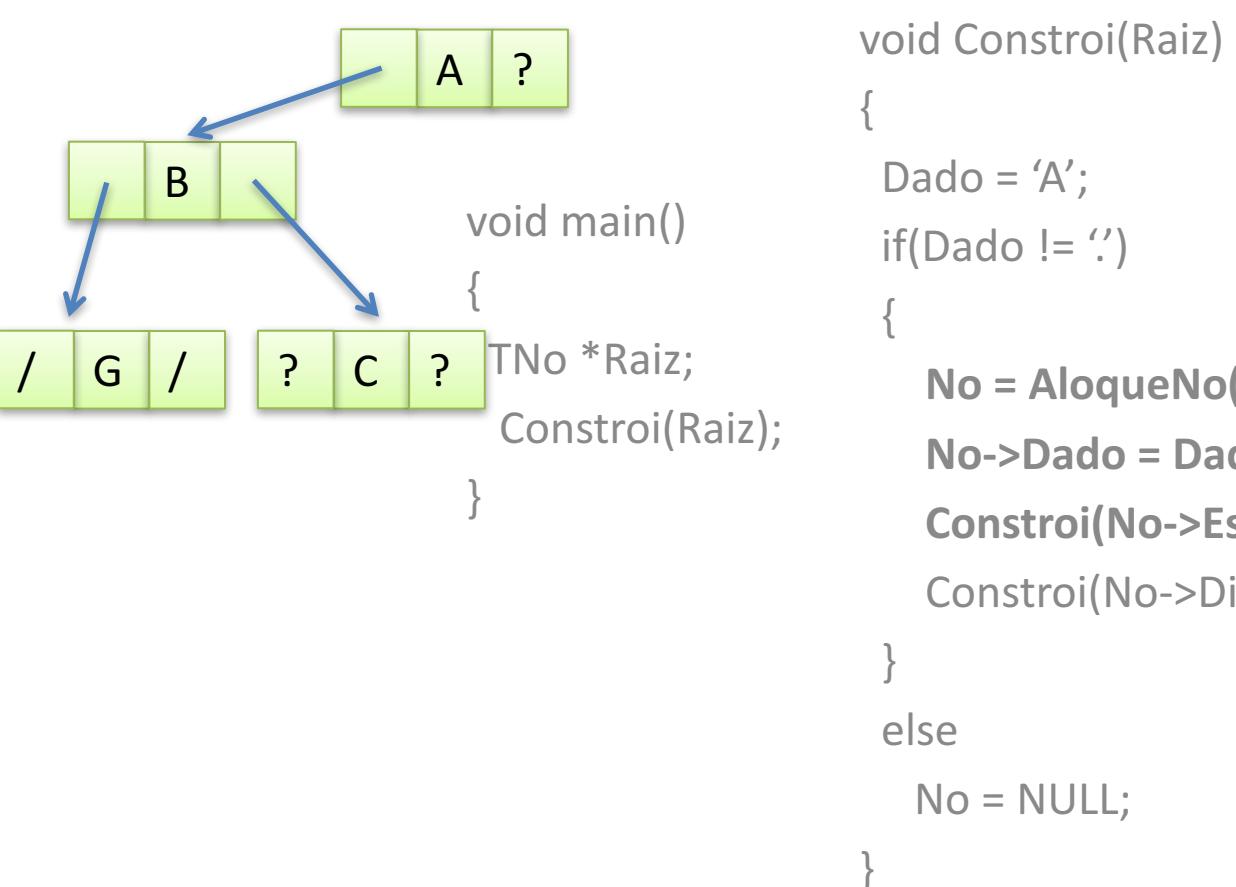
A B G . . C D E F . . .



A B G . . C D E F . . .



A B G . . C D E . F . . .



```
void main()
{
    TNo *Raiz;
    Constroi(Raiz);
}
```

```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

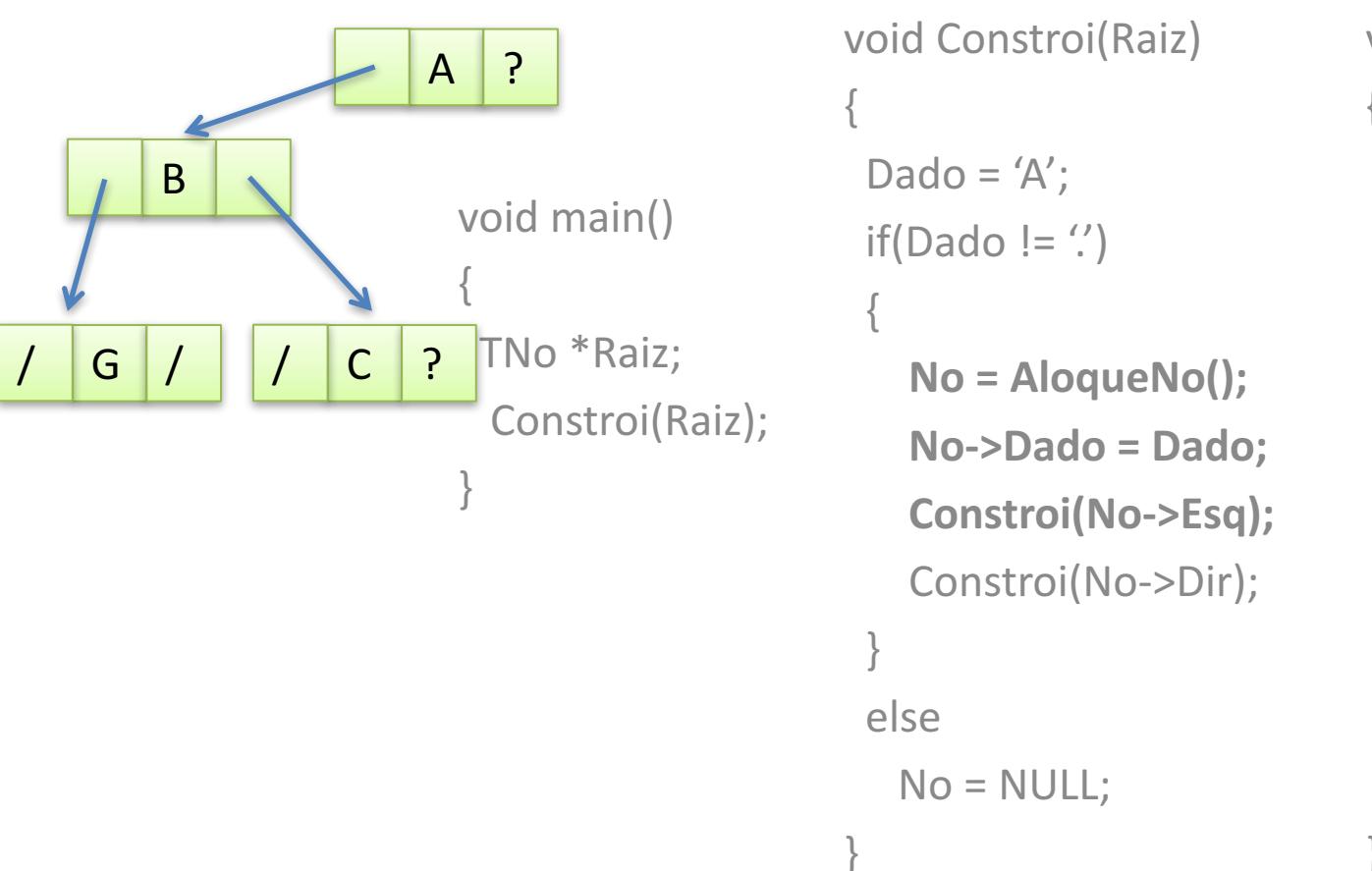
```
void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(B->Dir)
{
    Dado = 'C';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

Prof. Nilton Correia da Silva

Estrutura de Dados e Algoritmos

A B G . . C . D E . F . . .



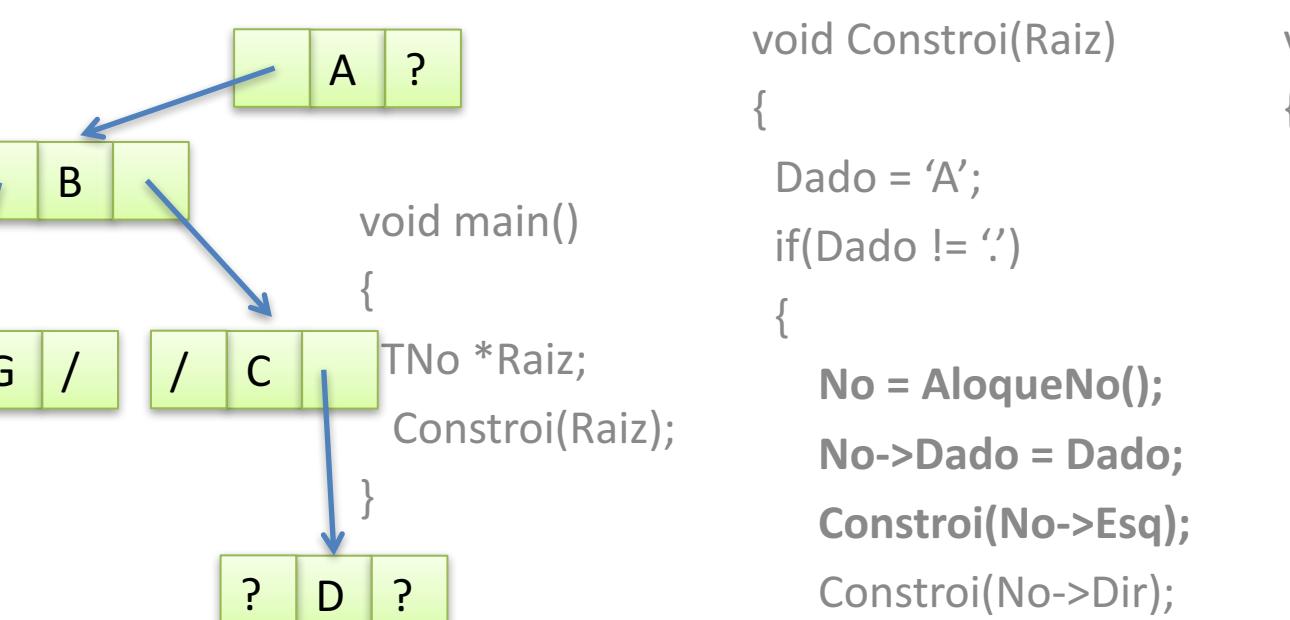
```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(C->Esq)
{
    Dado = '';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

A B G . . C D E . F . . .



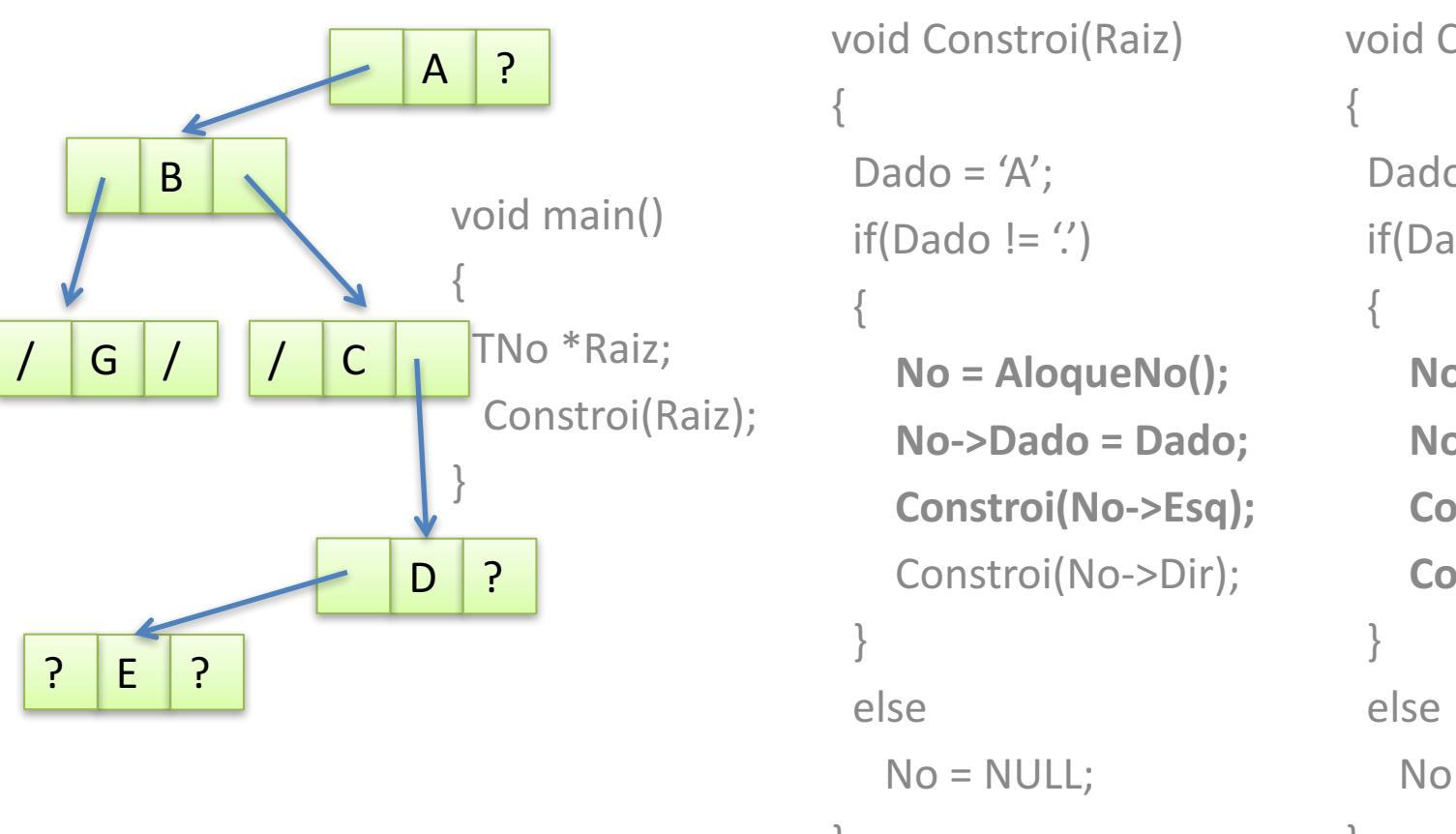
```
void Constrói(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constrói(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constrói(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constrói(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

A B G . . C D E . F . . .



```
void Constrói(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

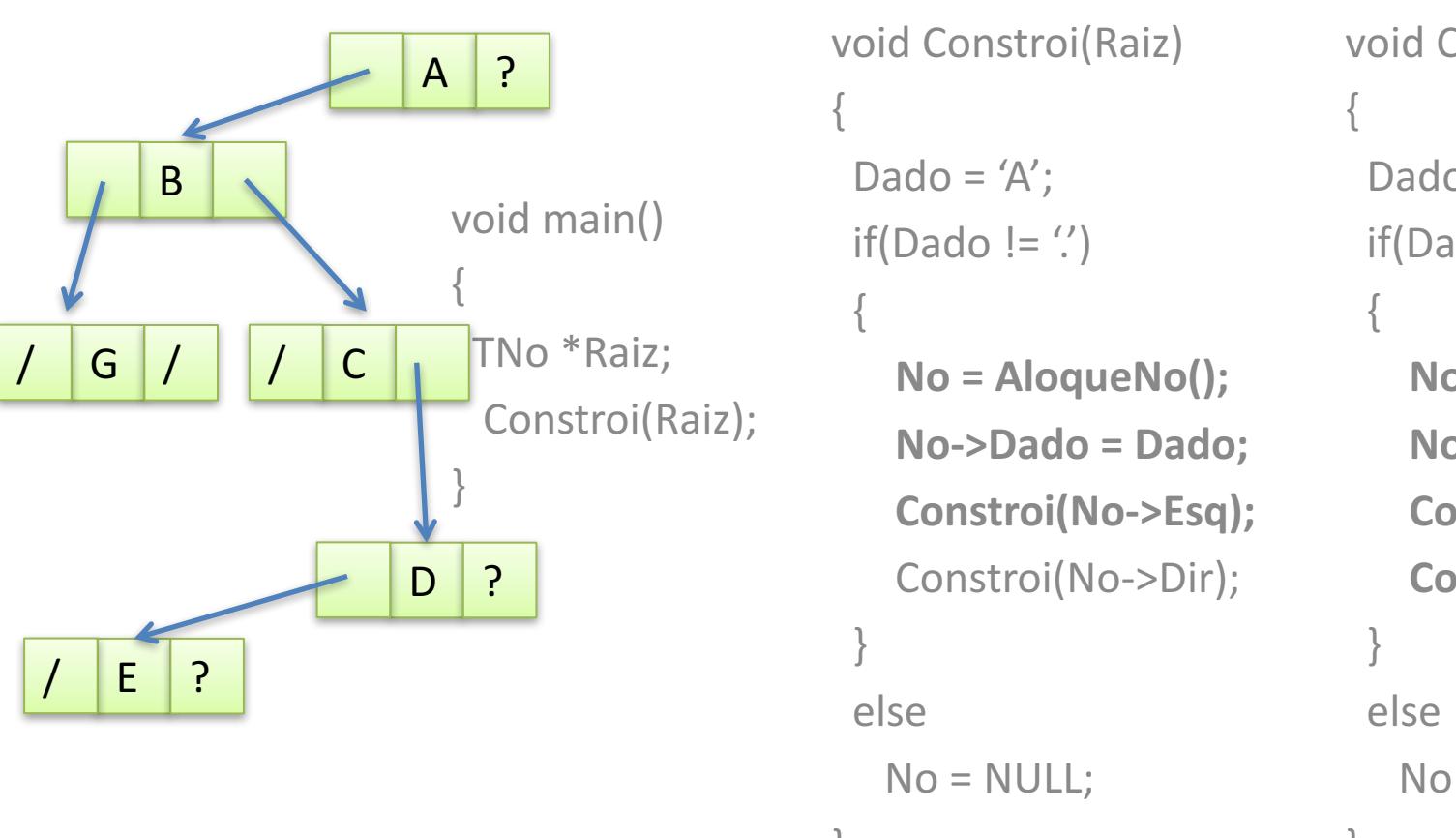
void Constrói(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(D->Esq)
{
    Dado = 'E';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

Estrutura de Dados e Algoritmos  
Prof. Nilton Correia da Silva

A B G . . C D E . F . . .



```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

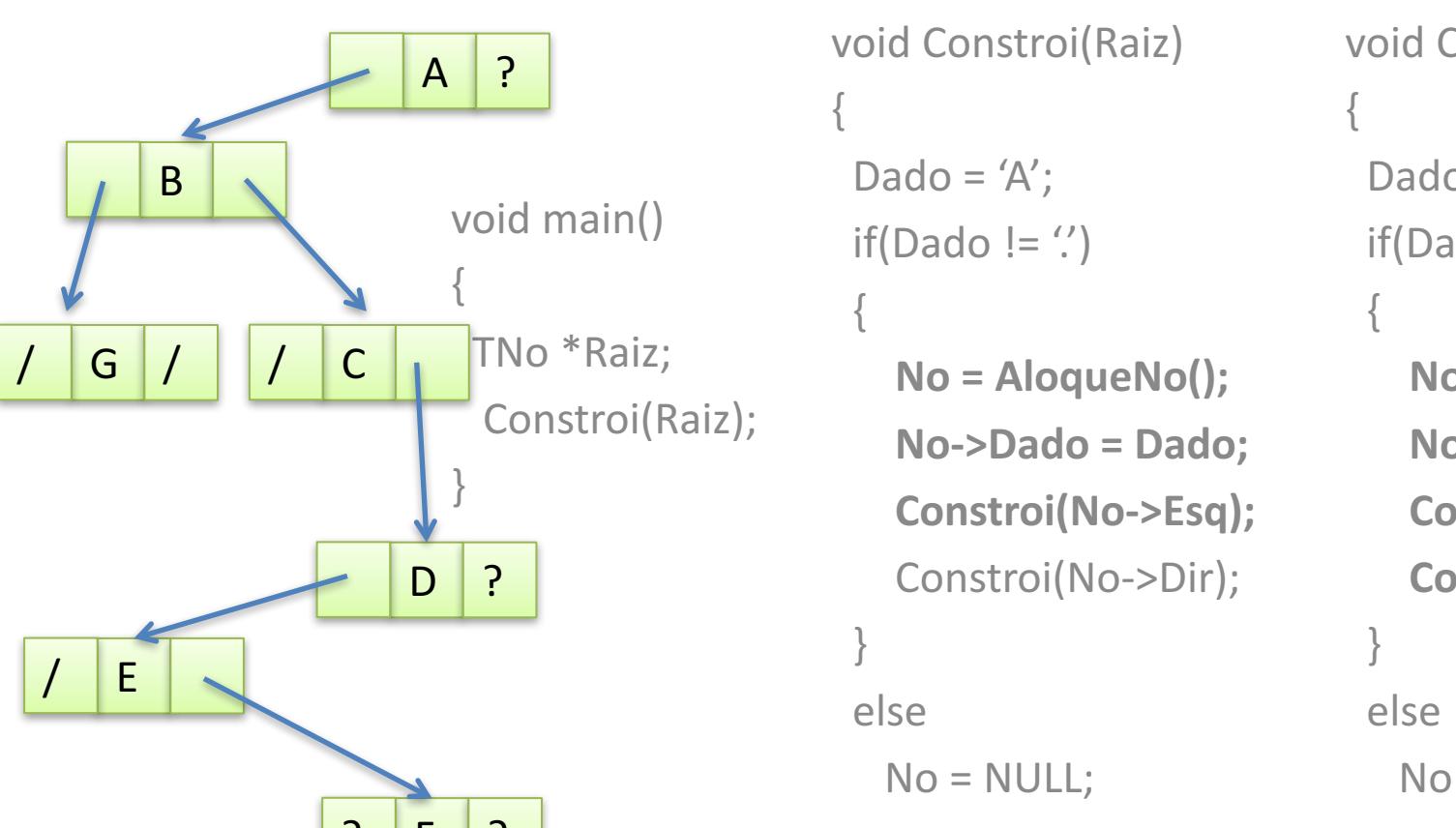
void Constroi(B->Dir)
{
    Dado = 'C';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(C->Dir)
{
    Dado = 'D';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(D->Esq)
{
    Dado = 'E';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(E->Esq)
{
    Dado = '/';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

A B G . . C D E . F . . .



```
void Constrói(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

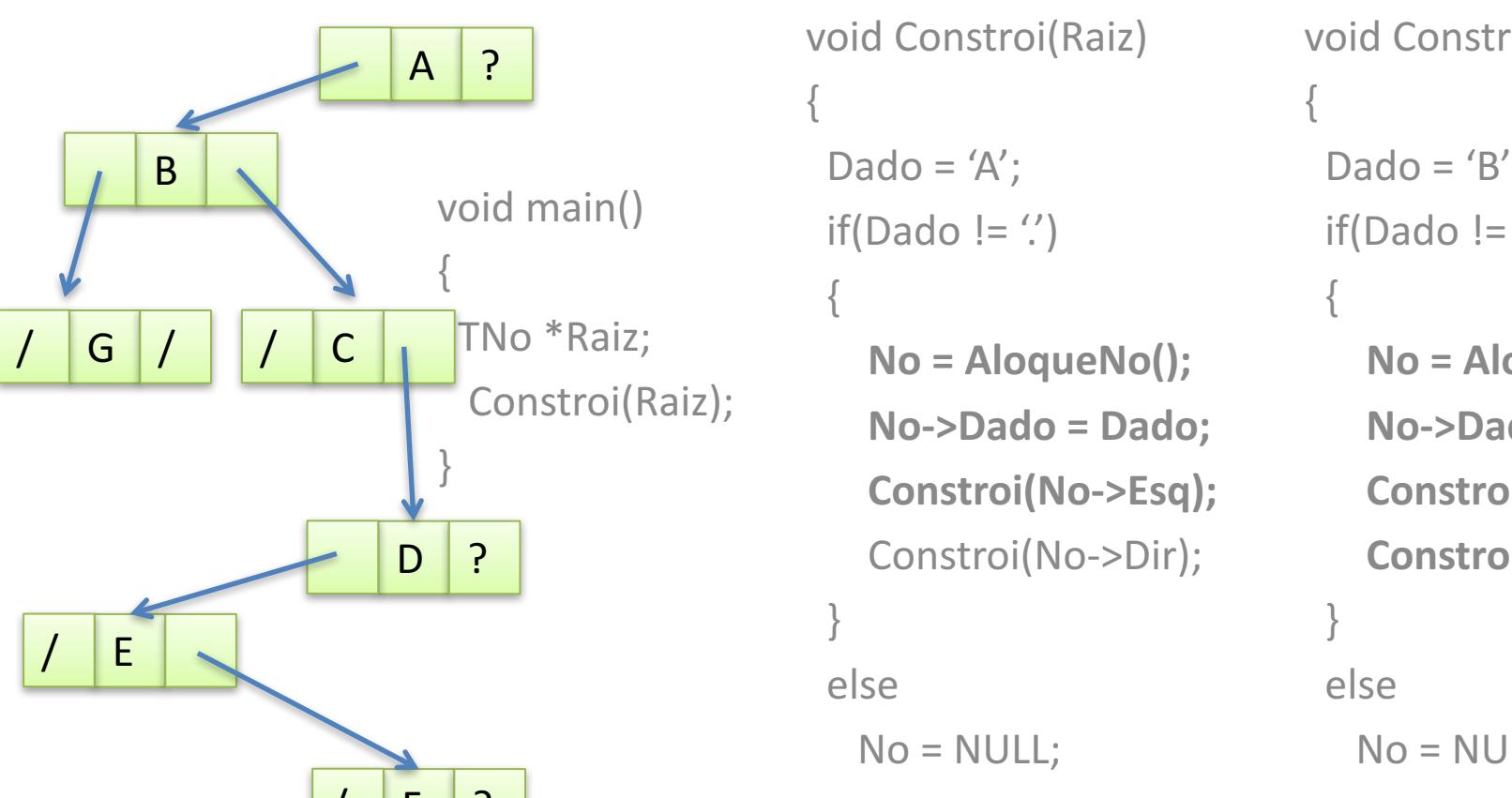
void Constrói(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(D->Esq)
{
    Dado = 'E';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(E->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

# ABG...C.D.E.F....



```

void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

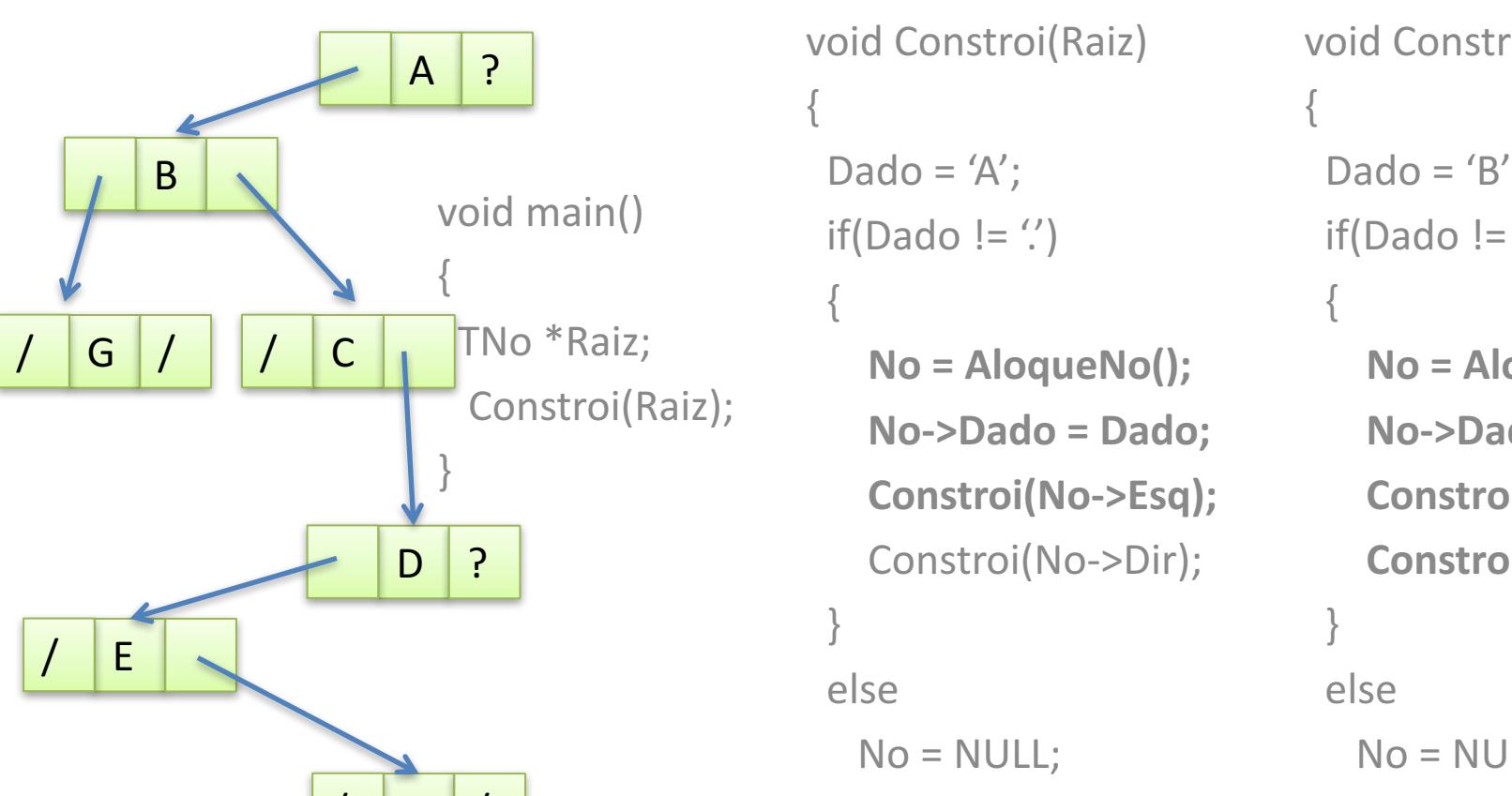
void Constroi(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(D->Esq)
{
    Dado = 'E';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(E->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(F->Esq)
{
    Dado = '';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
  
```

A B G . . C D E . F . . .



```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

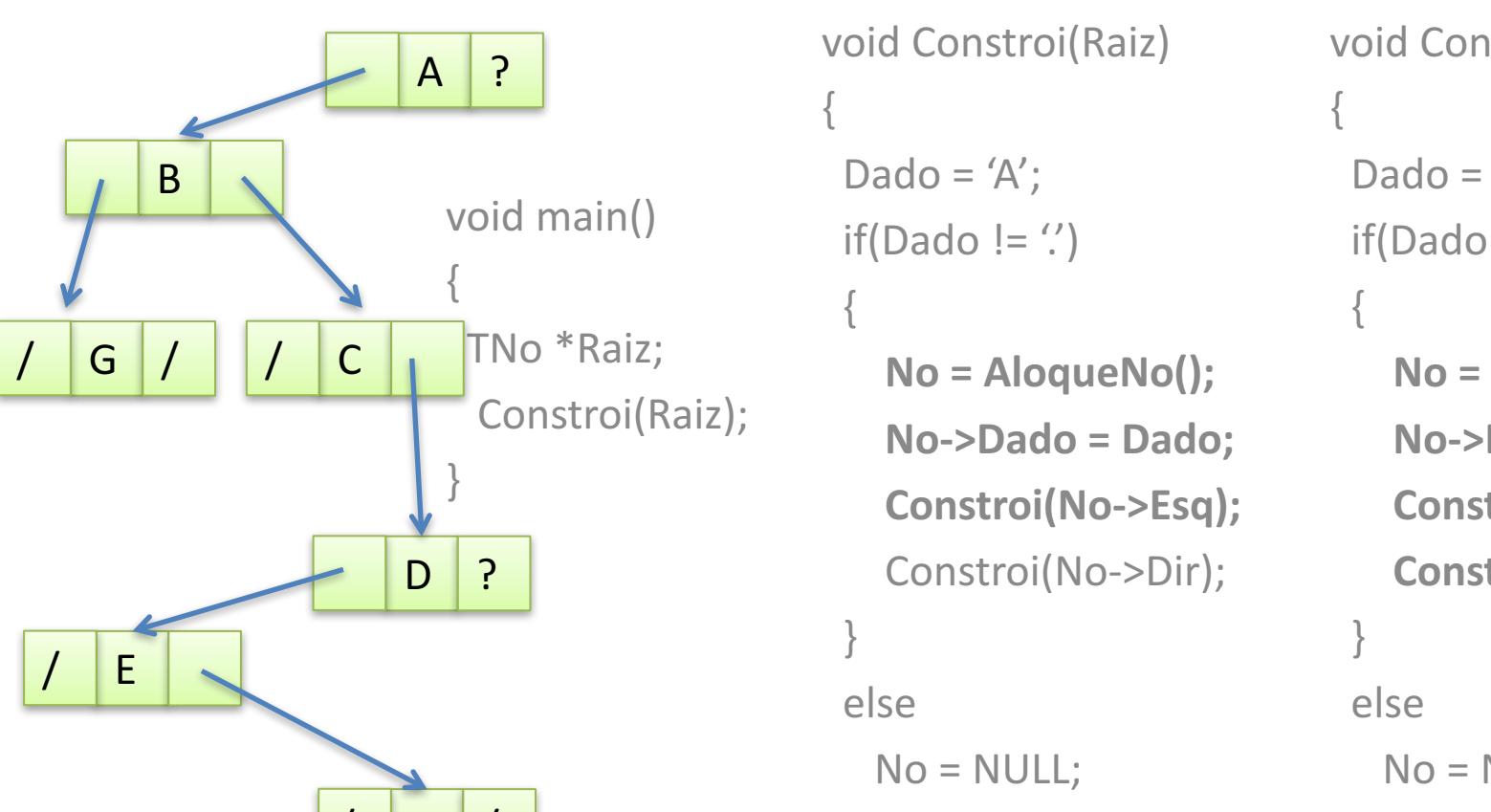
void Constroi(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(D->Esq)
{
    Dado = 'E';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(E->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}

void Constroi(F->Dir)
{
    Dado = '';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

A B G . . C D E . F . . .



```
void Constrói(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

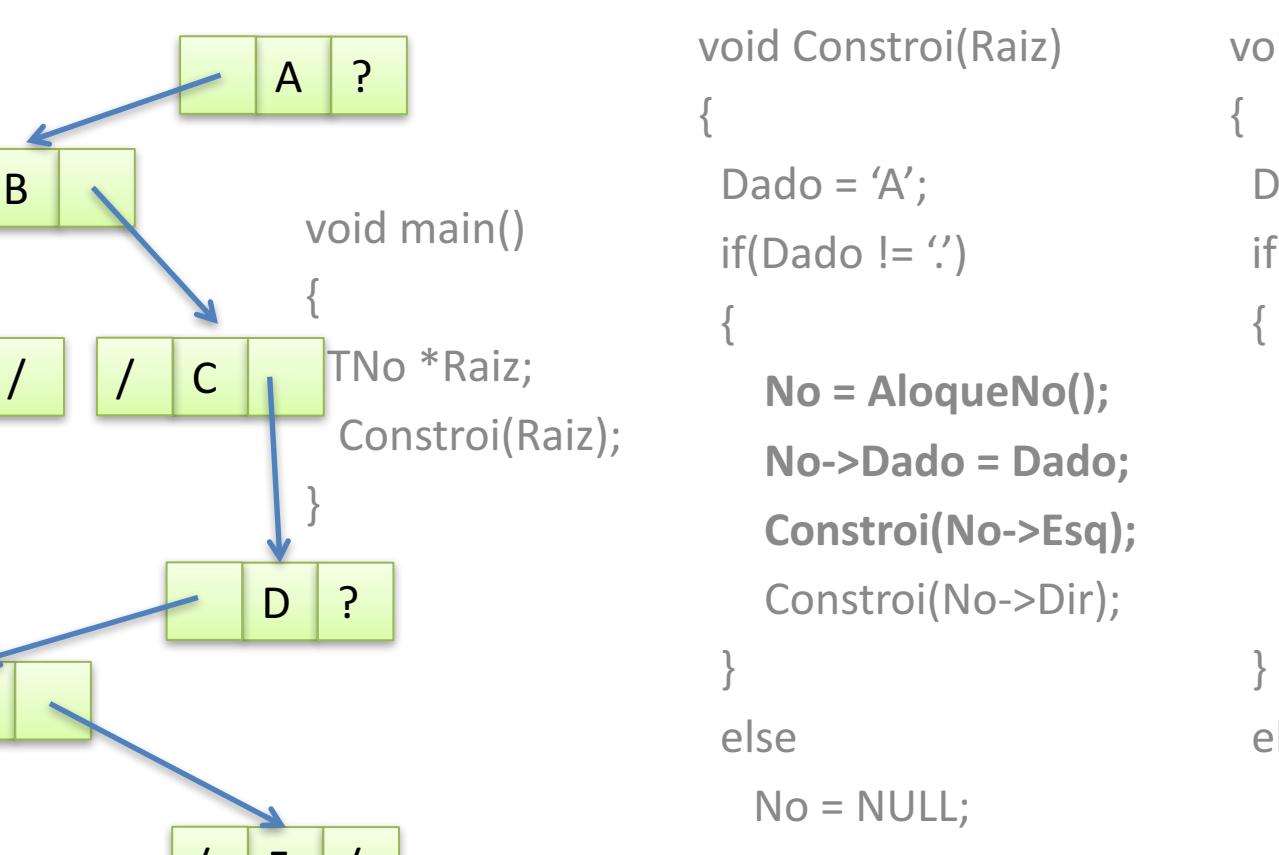
void Constrói(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(D->Esq)
{
    Dado = 'E';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(E->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

A B G .. C . D E . F . . . .



```
id Constroi(A->Esq)    void Constroi(B->Dir)    void Constroi(C->Dir)    void Constroi(D->Esq)
{
ado = 'B';
(Dado != '.')
{
No = AloqueNo();
No->Dado = Dado;
Constroi(No->Esq);
Constroi(No->Dir);
}
else
No = NULL;
}

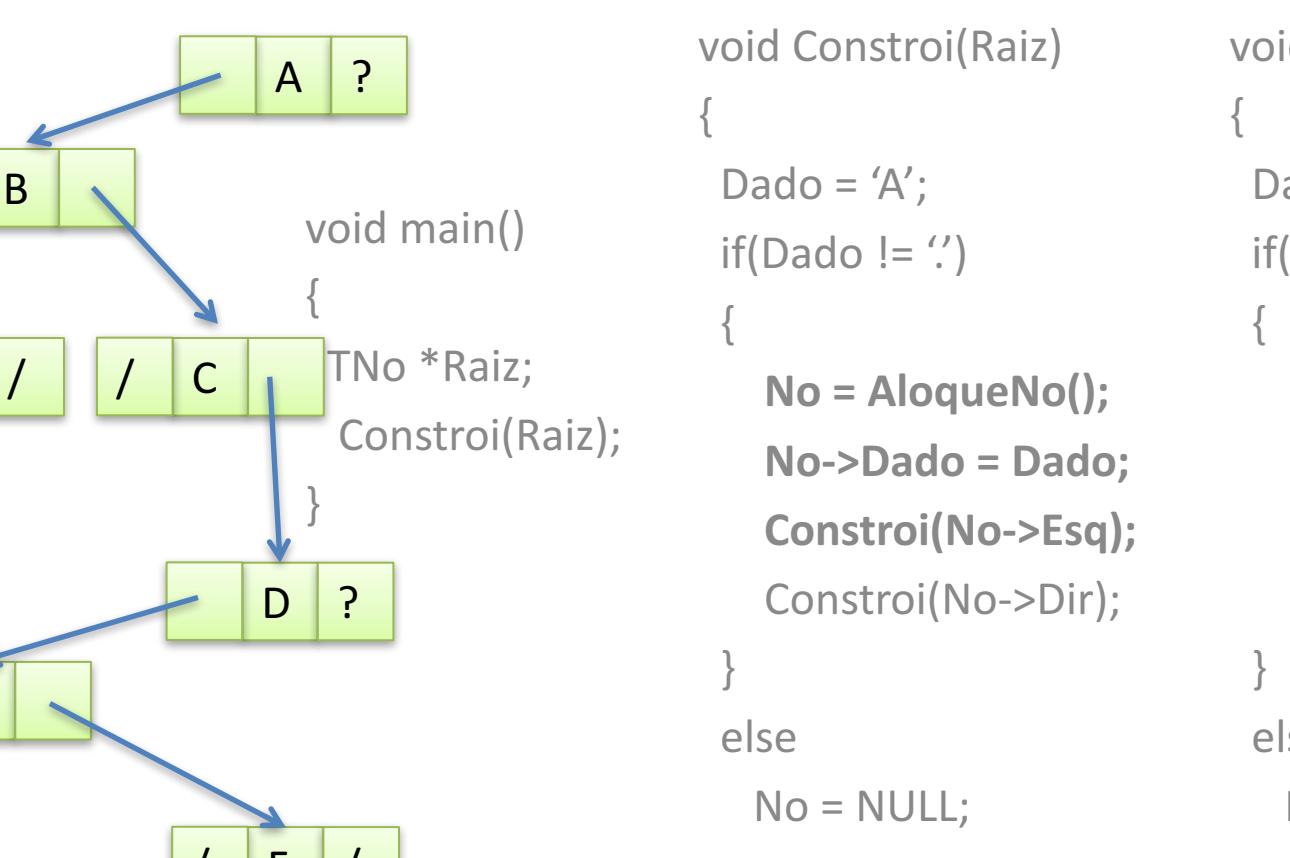
void Constroi(B->Dir)
{
Dado = 'C';
if(Dado != '.')
{
No = AloqueNo();
No->Dado = Dado;
Constroi(No->Esq);
Constroi(No->Dir);
}
else
No = NULL;
}

void Constroi(C->Dir)
{
Dado = 'D';
if(Dado != '.')
{
No = AloqueNo();
No->Dado = Dado;
Constroi(No->Esq);
Constroi(No->Dir);
}
else
No = NULL;
}

void Constroi(D->Esq)
{
Dado = 'E';
if(Dado != '.')
{
No = AloqueNo();
No->Dado = Dado;
Constroi(No->Esq);
Constroi(No->Dir);
}
else
No = NULL;
}
```

*de Dados e Algoritmos*

A B G . . C . D E . F . . . .



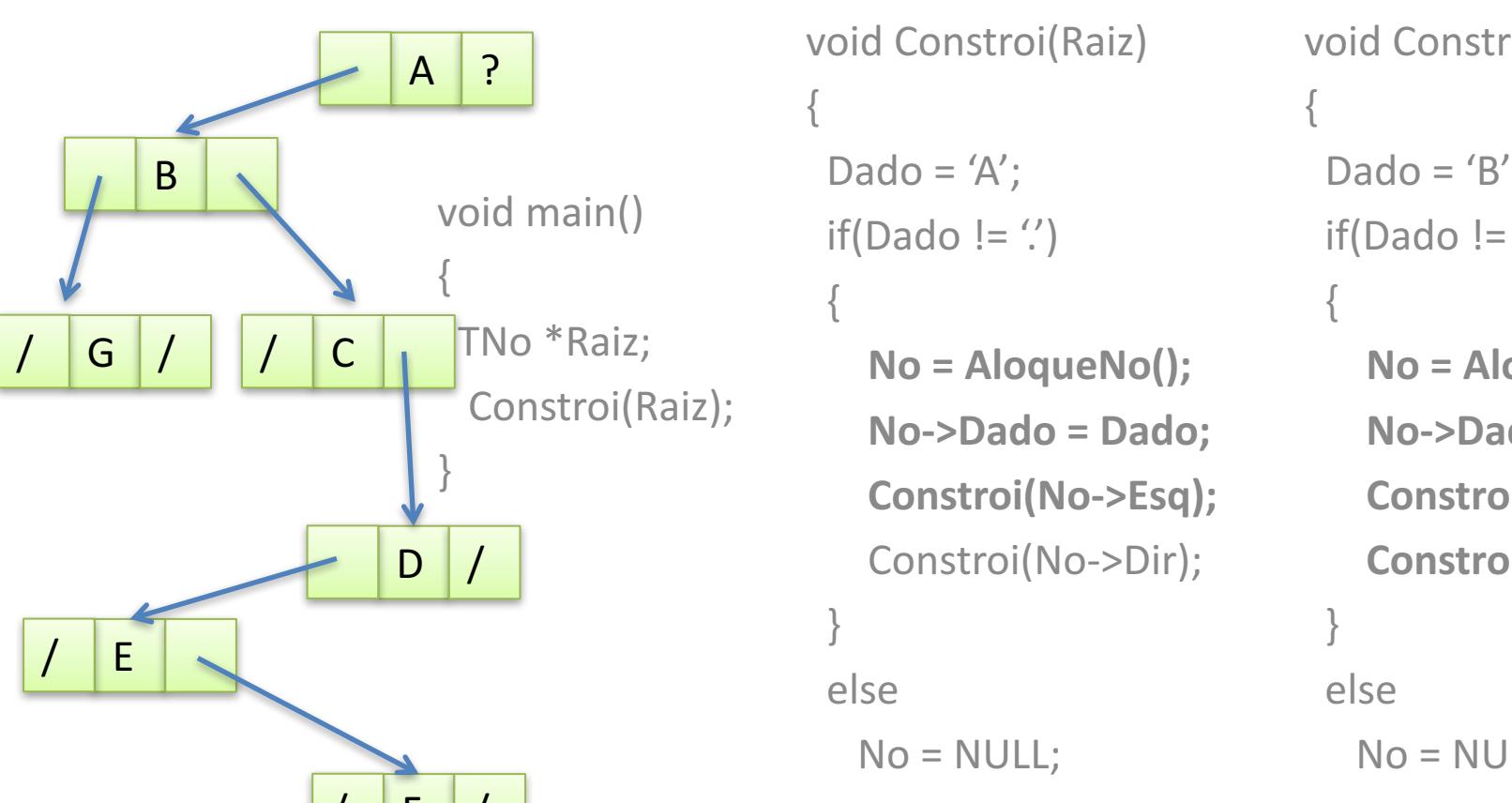
```
d Constroi(A->Esq)    void Constroi(B->Dir)    void Constroi(C->Dir)
{
ado = 'B';
Dado != '.'
{
No = AloqueNo();
No->Dado = Dado;
Constroi(No->Esq);
Constroi(No->Dir);
}
se
No = NULL;
}

Dado = 'C';
if(Dado != '.')
{
No = AloqueNo();
No->Dado = Dado;
Constroi(No->Esq);
Constroi(No->Dir);
}
else
No = NULL;

Dado = 'D';
if(Dado != '.')
{
No = AloqueNo();
No->Dado = Dado;
Constroi(No->Esq);
Constroi(No->Dir);
}
else
No = NULL;
```

Estrutura de Dados e Algoritmos

A B G . . C D E . F . . .



```
void Constrói(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

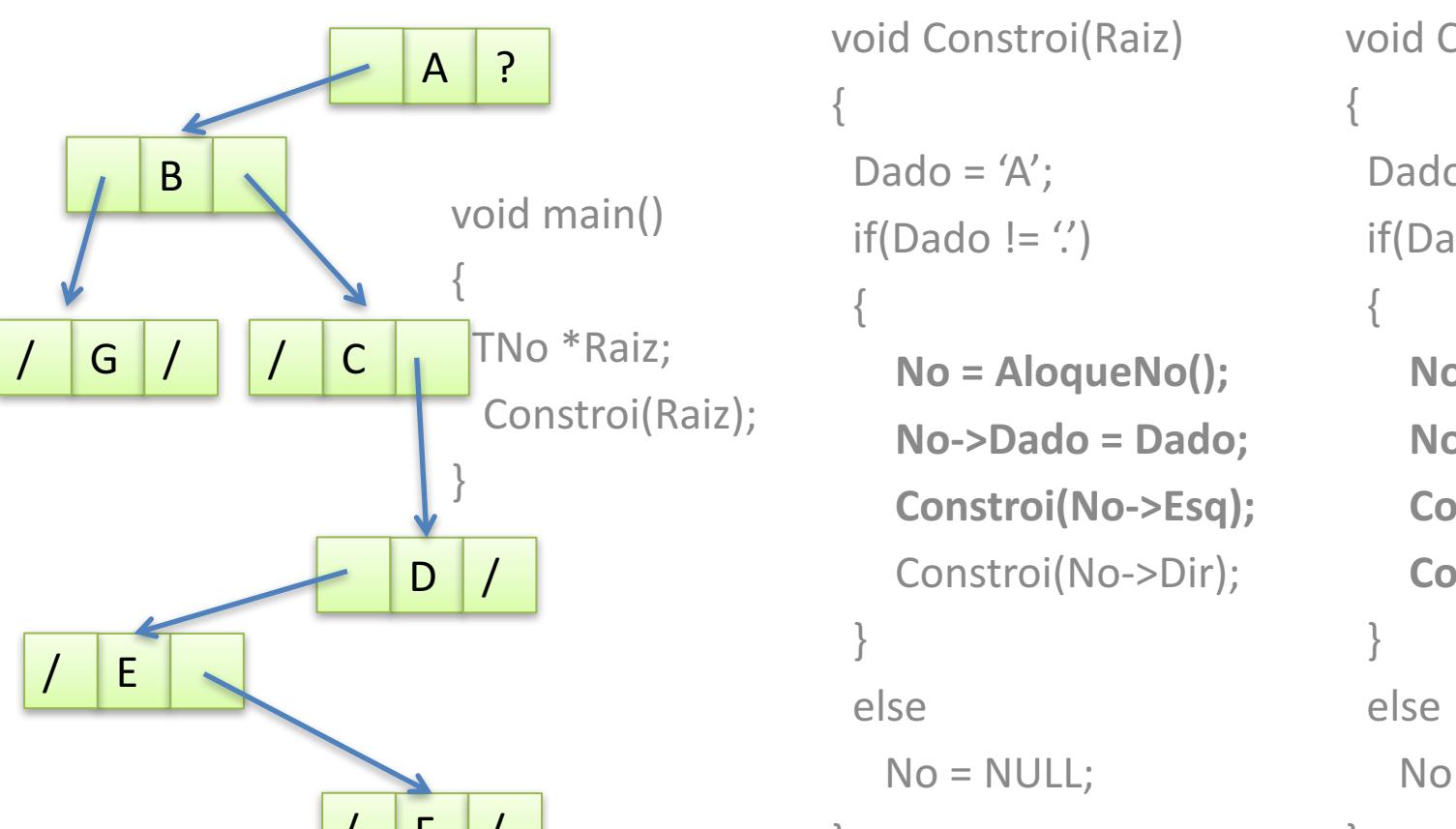
void Constrói(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}

void Constrói(D->Dir)
{
    Dado = '';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

A B G . . C D E . F . . .



```
void main()
{
    TNo *Raiz;
    Constroi(Raiz);
}
```

```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(C->Dir)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(D->Dir)
{
    Dado = 'E';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(E->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

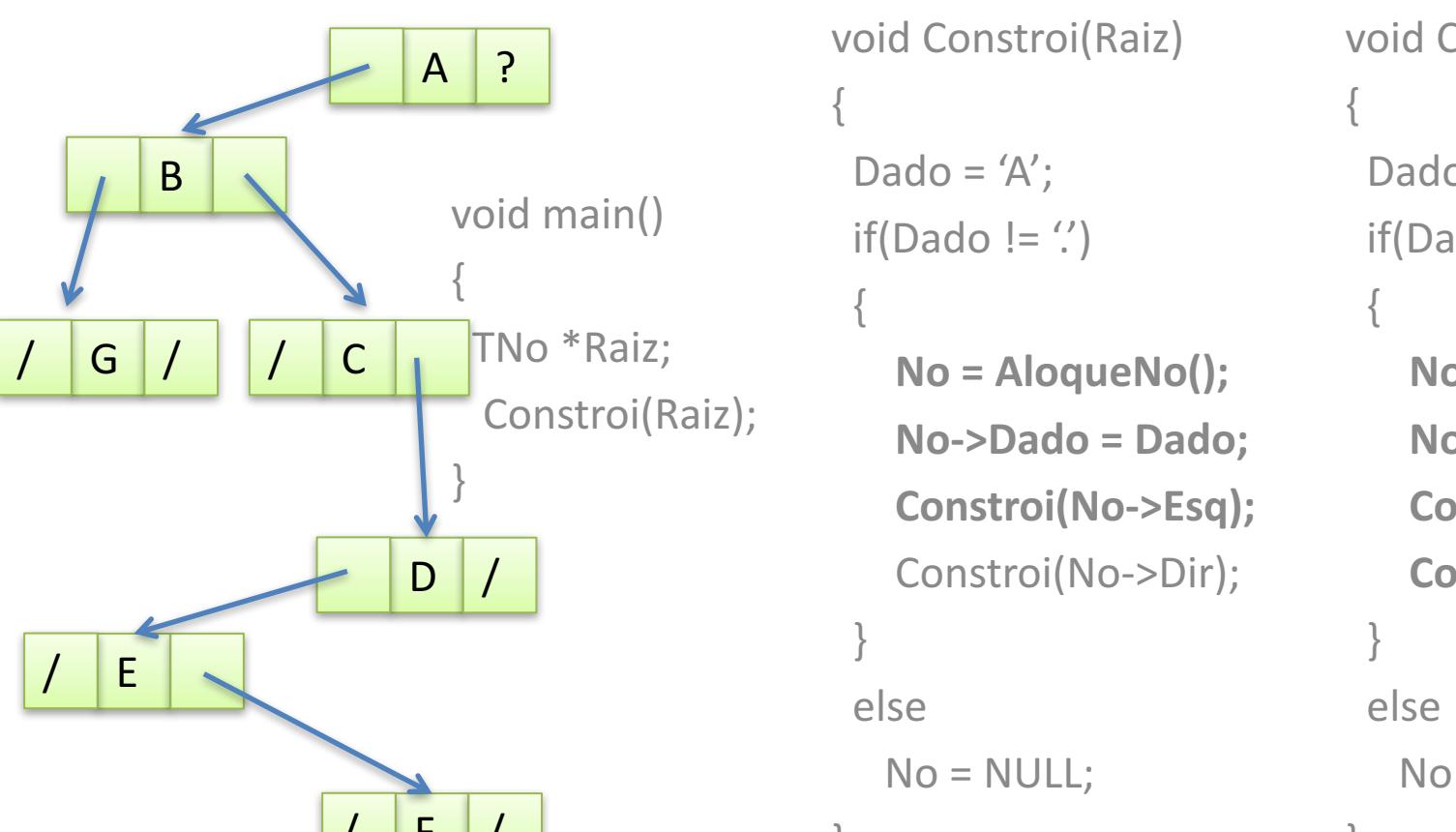
```
}
```

```
}
```

```
}
```

```
}
```

A B G . . C D E . F . . .



```
void main()
{
    TNo *Raiz;
    Constroi(Raiz);
}
```

```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(B->Dir)
{
    Dado = 'C';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(C->Esq)
{
    Dado = 'D';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(D->Dir)
{
    Dado = 'E';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(E->Esq)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(F->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(G->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

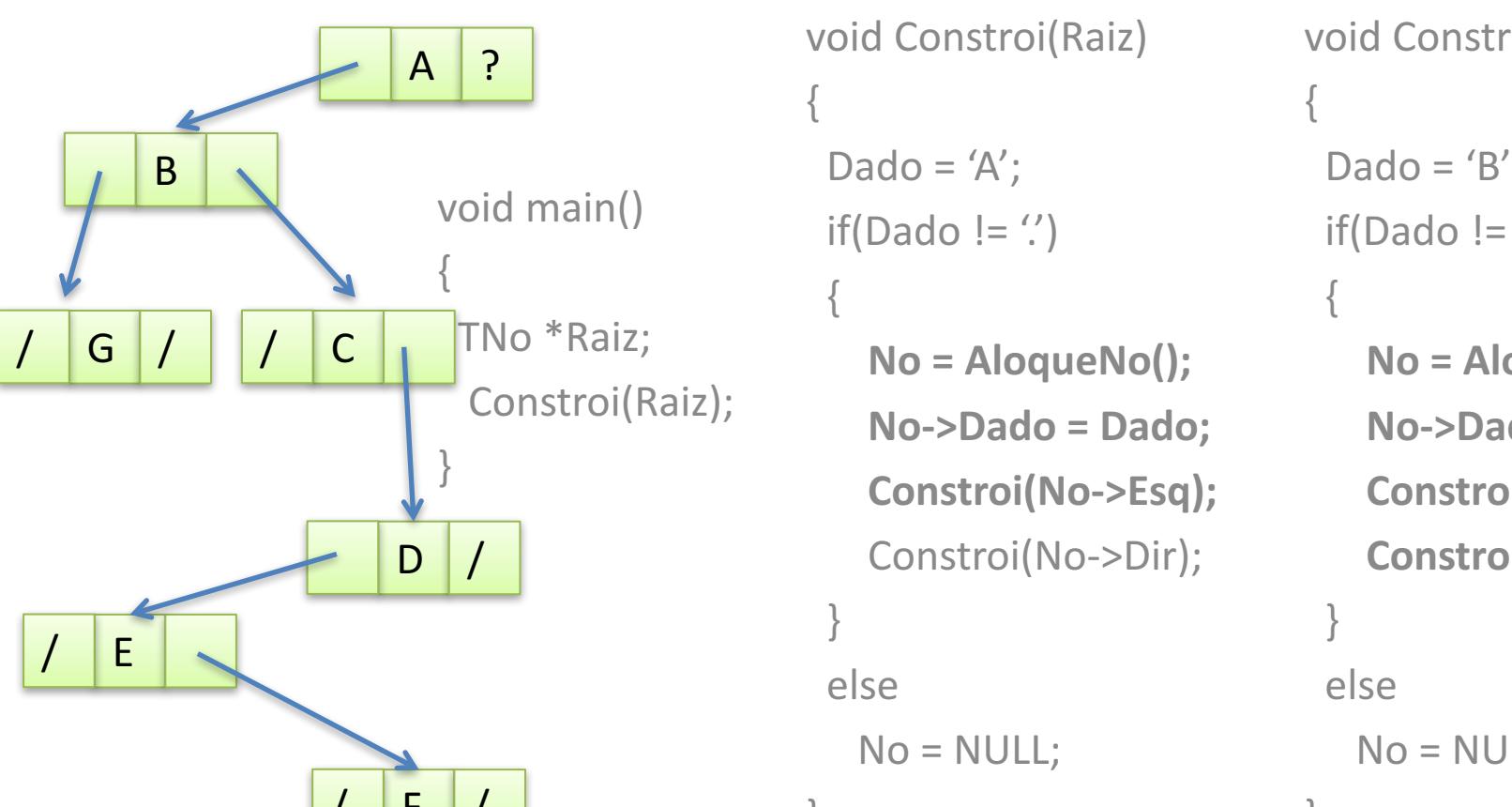
```
void Constroi(H->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(I->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(J->Dir)
{
    Dado = 'F';
    if(Dado != '?')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
}
```

# ABG...C.DE.F....



```
void main()
{
    TNo *Raiz;
    Constroi(Raiz);
}
```

```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(A->Esq)
{
    Dado = 'B';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

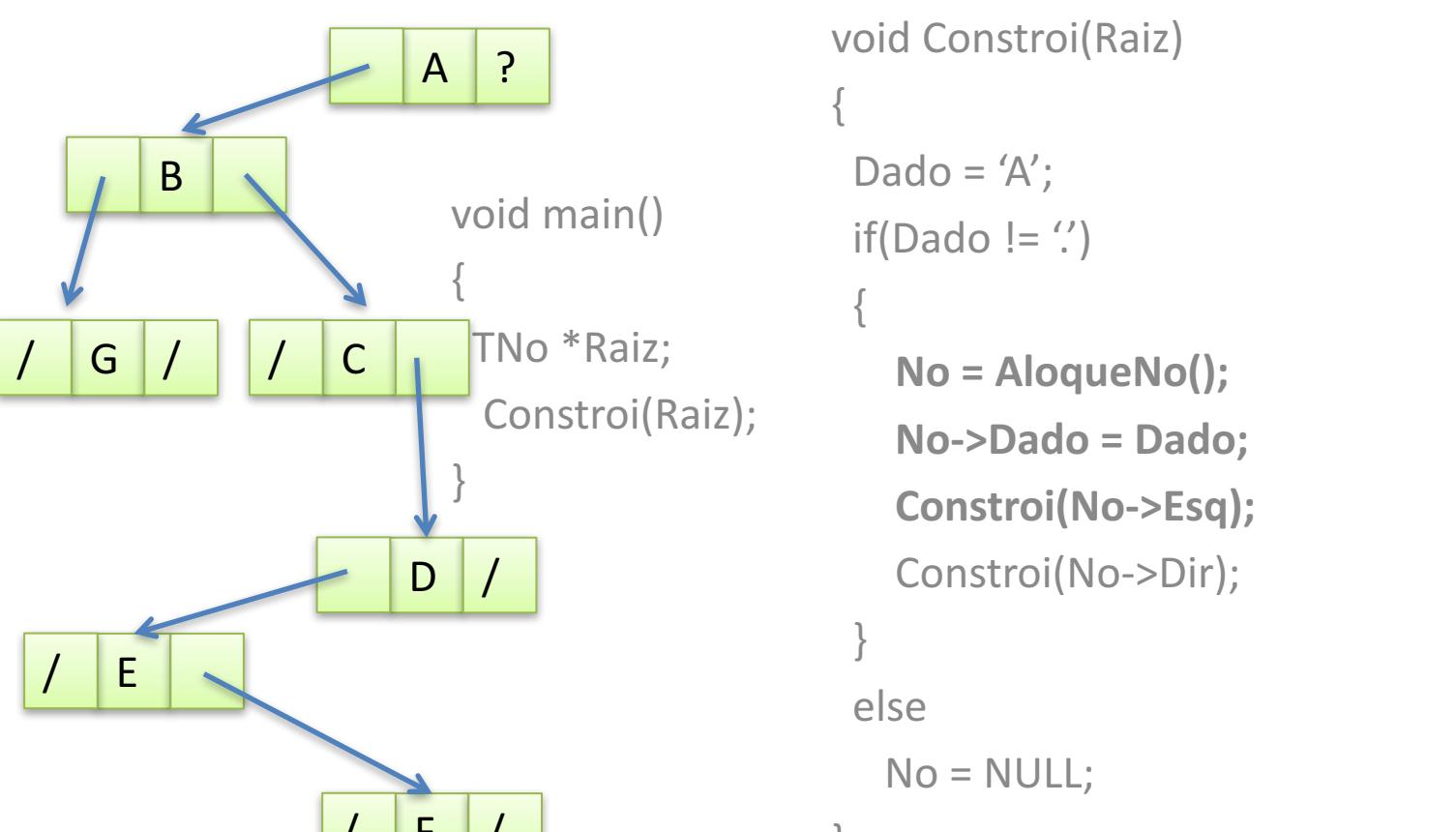
```
void Constroi(A->Dir)
{
    Dado = 'C';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(B->Esq)
{
    Dado = 'D';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(B->Dir)
{
    Dado = 'E';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

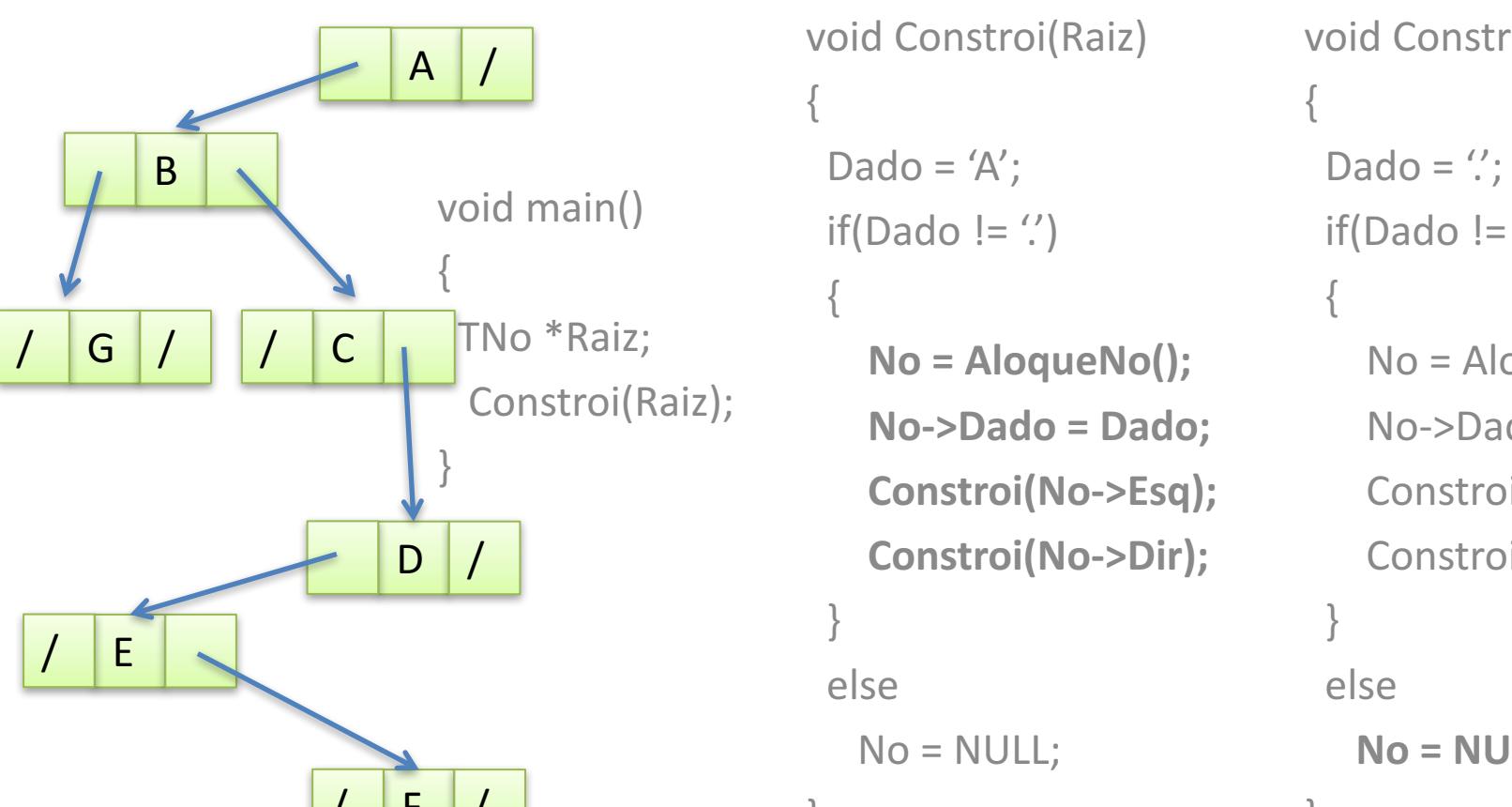
```
void Constroi(C->Dir)
{
    Dado = 'F';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

A B G . . C D E . F . . .



```
void Constrói(Raiz)
{
    Dado = 'A';
    if(Dado != ':')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constrói(No->Esq);
        Constrói(No->Dir);
    }
    else
        No = NULL;
}
```

# ABG..C.DE.F....



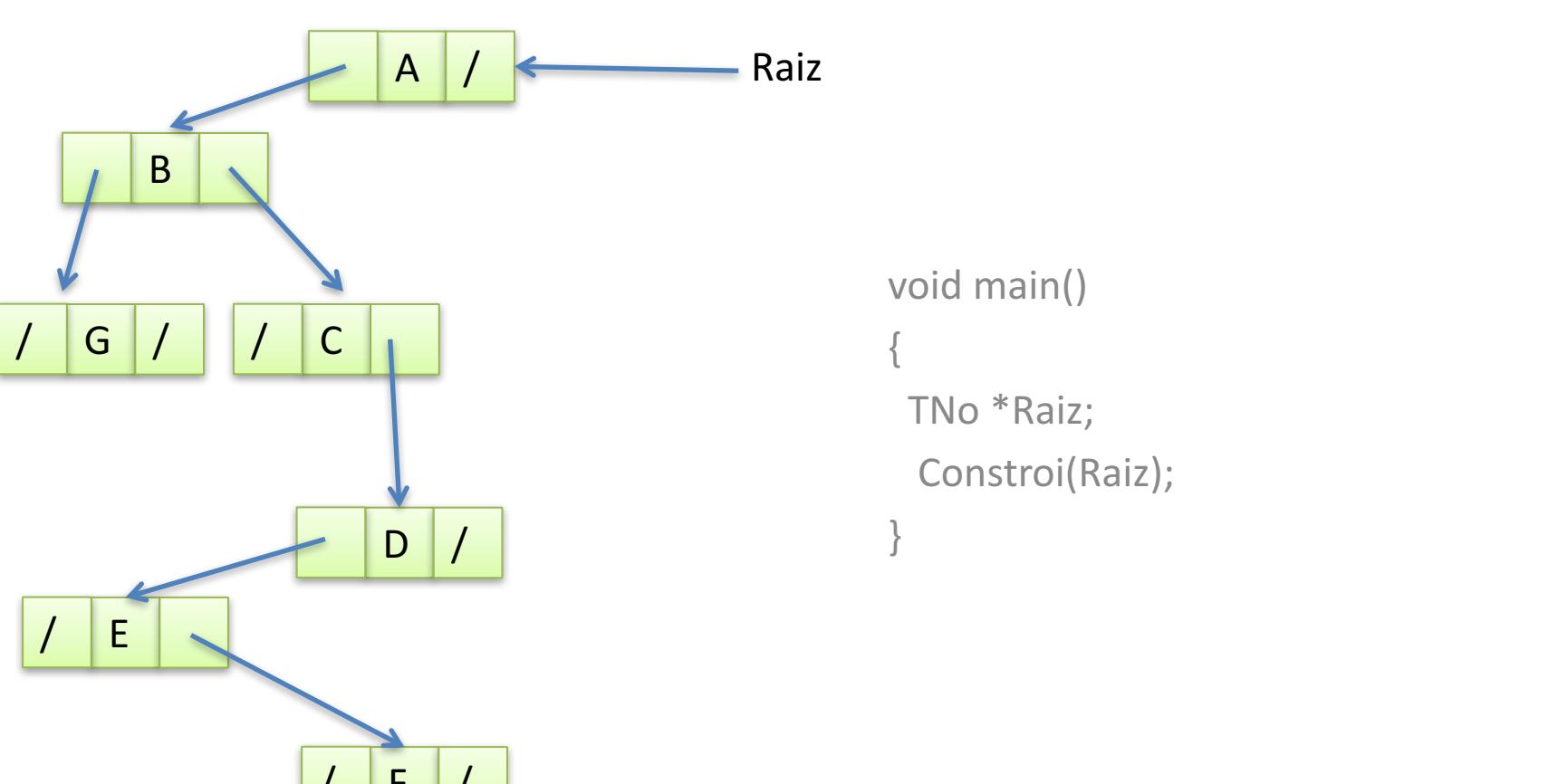
```
void main()
{
    TNo *Raiz;
    Constroi(Raiz);
}
```

```
void Constroi(Raiz)
{
    Dado = 'A';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

```
void Constroi(A->Dir)
{
    Dado = '';
    if(Dado != '/')
    {
        No = AloqueNo();
        No->Dado = Dado;
        Constroi(No->Esq);
        Constroi(No->Dir);
    }
    else
        No = NULL;
}
```

The code illustrates the construction of a binary tree. The `main()` function initializes the root node and calls the `Constroi(Raiz)` function. Inside `Constroi(Raiz)`, it checks if the current node's data is a backslash (/). If not, it creates a new node (No) for the character 'A', sets its data, and recursively calls `Constroi(No->Esq)` and `Constroi(No->Dir)`. If the data is a backslash (/), it sets the node to NULL. The `Constroi(A->Dir)` function follows a similar logic for creating nodes with character '' (empty string) and recursive calls.

A B G . . C D E . F . . .



Construção de Árvores Binárias

```
void main()
{
    TNo *Raiz;
    Constroi(Raiz);
}
```

Estrutura de Dados e Algoritmos

Prof. Nilton Correia da Silva