Fundamentos Teóricos da Computação

CIÊNCIA DA COMPUTAÇÃO

Prof. Dr. João Paulo Aramuni

UNIVERSIDADE FUMEC

50 Anos

Manipulação de Gramáticas

* **Manipulação de Gramáticas**

Sumário

* **Técnicas para:**
  * Eliminar Variáveis inúteis
    * Variáveis que não produzem sentenças
    * Variáveis não alcançáveis
  * Eliminar Regras
  * Eliminar Regras λ
    * Determinar Variáveis Anuláveis
  * Eliminar Regras Unitárias
    * Determinar Variáveis Encadeadas

Variáveis Inúteis

* Detectar e Eliminar variáveis inúteis é importante em gramáticas grandes (como as de linguagens de programação)
  * Caso se esqueça de definir regras relativas a determinadas variáveis
  * Caso existam regras para uma variável, mas esta não foi utilizada na formação de novas regras
* Sempre existe uma $\mathrm{GLC}\ G'$ equivalente a uma $\mathrm{GLC}\ G$, mas sem variáveis inúteis
  * Se $L(G) \neq \emptyset$

Variáveis Úteis

* Seja uma GLC $G = (V, \Sigma, R, P)$.
* Uma variável $X \in V$ é dita ser uma variável útil se, e somente se, existem $u, v \in (V, \Sigma)^*$ e $w \in \Sigma^*$ tais que:

$$P \overset{*}{\Rightarrow} uXv \overset{*}{\Rightarrow} w$$

* Ou seja, qualquer variável útil tem que fazer parte de alguma derivação de uma sentença a partir da variável de partida

Exemplo 1

* Seja a gramática $G = (\{P, A, B, C\}, \{a,b,c\}, R, P)$, em que R contém as seguintes regras:

* $P \rightarrow AB \mid a$
* $B \rightarrow b$
* $C \rightarrow c$

* Quais as variáveis úteis?

Exemplo 1

* C é inútil
  * Não existem u e v tais que $P \overset{*}{\Rightarrow} uCv$
* A é inútil
  * Não existe $w \in \Sigma^*$ tal que $A \overset{*}{\Rightarrow} w$
* B é inútil
  * $P \overset{*}{\Rightarrow} uBv$ para $u = A$ e $v = \lambda$
  * Não existe $w \in \Sigma^*$ tal que $AB \overset{*}{\Rightarrow} w$

Exemplo 1

* Pode-se eliminar também os terminais b e c.

* A gramática fica então $G = (\{P\}, \{a\}, \{P \rightarrow a\}, P)$

Determinando Variáveis que Produzem Sentenças

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

* Seja a gramática $G = (\{A, B, C, D, E, F\}, \{0,1\}, R, A)$, em que R contém as seguintes regras:

* $A \rightarrow ABC \mid AEF \mid BD$
* $B \rightarrow B0 \mid 0$
* $C \rightarrow 0C \mid EB$
* $D \rightarrow 1D \mid 1$
* $E \rightarrow BE$
* $F \rightarrow 1F1 \mid 1$

* Aplicar o algoritmo PRODUZ-SENTENÇA na gramática para determinar variáveis que produzem sentenças

Exemplo 2

* $V' = \{\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:   V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

* $V' = \{\}$
  * $T = \{B,D,F\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:   V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

* $V' = \{\}$
  * $T = \{B,D,F\}$
* $V' = \{B,D,F\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

- $V' = \{\}$
  - $T = \{B,D,F\}$
- $V' = \{B,D,F\}$
  - $T = \{A\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

* $V' = \{\}$
  * $T = \{B,D,F\}$
* $V' = \{B,D,F\}$
  * $T = \{A\}$
* $V' = \{A,B,D,F\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

* $V' = \{\}$
  * $T = \{B,D,F\}$
* $V' = \{B,D,F\}$
  * $T = \{A\}$
* $V' = \{A,B,D,F\}$
  * $T = \{\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

- $V' = \{\}$
  - $T = \{B,D,F\}$
- $V' = \{B,D,F\}$
  - $T = \{A\}$
- $V' = \{A,B,D,F\}$
  - $T = \{\}$
- $V' = \{A,B,D,F\}$

função **PRODUZ-SENTENÇA**(G) **retorna** V'

entradas: G, Uma GLC na forma (V, Σ, R, P)

saídas: V', Conjunto $\{X \in V \mid X \text{ produz uma sentença}\}$

$V' \leftarrow \{\}$;

repita

$T \leftarrow \{X \notin V' \mid X \rightarrow z \in R \text{ e } z \in (V' \cup \Sigma)^*\}$;

$V' \leftarrow V' \cup T$

até $T = \{\}$

retorne V'

Exemplo 2

* $V' = \{\}$
  * $T = \{B,D,F\}$
* $V' = \{B,D,F\}$
  * $T = \{A\}$
* $V' = \{A,B,D,F\}$
  * $T = \{\}$
* $V' = \{A,B,D,F\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

* $V' = \{\}$
  * $T = \{B,D,F\}$
* $V' = \{B,D,F\}$
  * $T = \{A\}$
* $V' = \{A,B,D,F\}$
  * $T = \{\}$
* $V' = \{A,B,D,F\}$

```
função PRODUZ-SENTENÇA(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X produz uma sentença}

  V' ← {};
  repita
    T ← {X ∉ V' | X → z ∈ R e z ∈ (V' ∪ Σ)*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 2

* A gramática G' equivalente à G é então:
* $G' = (\{A, B, D, F\}, \{0,1\}, R, A)$, em que R contém as seguintes regras:

* $A \rightarrow BD$
* $B \rightarrow B0 \mid 0$
* $D \rightarrow 1D \mid 1$
* $F \rightarrow 1F1 \mid 1$

* Permanecem as regras que não incluem variáveis que não produzem sentenças

Determinando Variáveis que são Alcançáveis a partir de P

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

* Determine a gramática G'' equivalente à G' eliminando as variáveis não alcançáveis.

  * $G' = (\{A, B, D, F\}, \{0{,}1\}, R, A)$, em que R contém as seguintes regras:

    * $A \longrightarrow BD$
    * $B \longrightarrow B0 \mid 0$
    * $D \longrightarrow 1D \mid 1$
    * $F \longrightarrow 1F1 \mid 1$

Exemplo 3

* $V' = \{\}$

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

* $V' = \{\}$
  * $T = \{A\}$

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

* $V' = \{\}$
  * $T = \{A\}$
* $V'' = \{A\}$

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

- $V' = \{\}$
  - $T = \{A\}$
- $V'' = \{A\}$
  - $T = \{B,D\}$

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

* $V' = \{\}$
  * $T = \{A\}$
* $V'' = \{A\}$
  * $T = \{B,D\}$
* $V'' = \{A,B,D\}$

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

- $V' = \{\}$
  - $T = \{A\}$
- $V'' = \{A\}$
  - $T = \{B,D\}$
- $V'' = \{A,B,D\}$
  - $T = \{\}$

função ALCANÇÁVEL(G) **retorna** V''

entradas: G, Uma GLC na forma (V,Σ,R,P)

saídas: V'', Conjunto $\{X \in V \mid X$ *é alcançável a partir de* $P\}$

$V'' \leftarrow \{\}$;

$T \leftarrow \{P\}$;

repita

$\quad V'' \leftarrow V'' \cup T$

$\quad T \leftarrow \{Y \notin V'' \mid X \rightarrow uYv$ para algum $X \in T$ e $u,v \in (V \cup \Sigma)^*\}$;

até $T = \{\}$

retorne V''

Exemplo 3

* $V' = \{\}$
  * $T = \{A\}$
* $V'' = \{A\}$
  * $T = \{B,D\}$
* $V'' = \{A,B,D\}$
  * $T = \{\}$

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

* $V' = \{\}$
  * $T = \{A\}$
* $V'' = \{A\}$
  * $T = \{B,D\}$
* $V'' = \{A,B,D\}$
  * $T = \{\}$

```
função ALCANÇÁVEL(G) retorna V''
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V'', Conjunto {X ∈ V | X é alcançável a partir de P}

  V'' ← {};
  T ← {P};
  repita
    V'' ← V'' ∪ T
    T ← {Y ∉ V'' | X → uYv para algum X ∈ T e u,v ∈ (V ∪ Σ)*};
  até T = {}
  retorne V''
```

Exemplo 3

- A gramática G'' equivalente à G' é então:
- $G'' = (\{A, B, D\}, \{0,1\}, R, A)$, em que R contém as seguintes regras:

- $A \rightarrow BD$
- $B \rightarrow B0 \mid 0$
- $D \rightarrow 1D \mid 1$

- Permanecem as regras que <u>não</u> incluem variáveis não alcançáveis a partir de A

Atenção!

* Para eliminar variáveis inúteis, deve-se seguir esta ordem:
  * 1) Determina-se as variáveis que produzem sentenças
    * Elimina-se da gramática as variáveis que não produzam sentença, bem como as regras que as utilizem
  * 2) Determina-se as variáveis alcançáveis
    * Elimina-se da gramática as variáveis que não são alcançáveis e as regras que as utilizem

Eliminação de Regras

- Muitas vezes é necessário eliminar uma regra da gramática, sem modificar a linguagem gerada
  - Pode-se eliminar regras da forma:
  - $X \rightarrow w$
    - Onde X não é a variável de partida
  - Eliminando-se regras reduz-se o número de derivações necessárias
    - Mas aumenta-se o número de regras da gramática

Eliminação de Regras

* Para eliminar uma regra, simula-se a aplicação da mesma em todos os contextos:

* Cada regra com n ocorrências de X do lado direito, dá origem a até 2^n regras
  * Casos em que X é substituído por w
  * Casos em que não é substituído
    * Para que outras regras de X sejam utilizadas

Exemplo 4

* Seja a GLC $G = (\{P, A, B\}, \{a,b,c\}, R, P)$, em que R contém as seguintes regras:
  * $P \rightarrow ABA$
  * $A \rightarrow aA \mid a$
  * $B \rightarrow bBc \mid \lambda$
* Eliminando-se a regra $A \rightarrow a$ de G, obtém-se a gramática G′, com as seguintes regras:
  * $P \rightarrow ABA \mid ABa \mid aBA \mid aBa$
  * $A \rightarrow aA \mid aa$
  * $B \rightarrow bBc \mid \lambda$

Eliminação de Regras λ

* Qualquer regra λ pode ser eliminada de uma gramática sem alterar a linguagem gerada
  * Exceto $P \rightarrow \lambda$, onde P é a variável de partida
    * Neste caso, $\lambda \in L(G)$
* Para eliminar regras λ, o primeiro passo é determinar as variáveis anuláveis de G
  * Uma variável X é anulável se existe uma sequência finita de derivações usando as regras de G que transforme X em λ

$$X \overset{*}{\Rightarrow}_G \lambda$$

Determinando Variáveis Anuláveis

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

- Seja a GLC $G = (\{P, A, B, C\}, \{a,b,c\}, R, P)$, em que R contém as seguintes regras:
  - $P \rightarrow APB \mid C$
  - $A \rightarrow AaaA \mid \lambda$
  - $B \rightarrow BBb \mid b$
  - $C \rightarrow cC \mid \lambda$

  - Determine as variáveis anuláveis de G.

Exemplo 5

* $V' = \{\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

* $V' = \{\}$
  * $T = \{A,C\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

- $V' = \{\}$
  - $T = \{A,C\}$
- $V' = \{A,C\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

* $V' = \{\}$
  * $T = \{A,C\}$
* $V' = \{A,C\}$
  * $T = \{P\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

* $V' = \{\}$
  * $T = \{A,C\}$
* $V' = \{A,C\}$
  * $T = \{P\}$
* $V' = \{A,C,P\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

* $V' = \{\}$
  * $T = \{A,C\}$
* $V' = \{A,C\}$
  * $T = \{P\}$
* $V' = \{A,C,P\}$
  * $T = \{\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:   V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

* $V' = \{\}$
  * $T = \{A,C\}$
* $V' = \{A,C\}$
  * $T = \{P\}$
* $V' = \{A,C,P\}$
  * $T = \{\}$
* $V' = \{A,C,P\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

* $V' = \{\}$
  * $T = \{A,C\}$
* $V' = \{A,C\}$
  * $T = \{P\}$
* $V' = \{A,C,P\}$
  * $T = \{\}$
* $V' = \{A,C,P\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:   V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Exemplo 5

- $V' = \{\}$
  - $T = \{A,C\}$
- $V' = \{A,C\}$
  - $T = \{P\}$
- $V' = \{A,C,P\}$
  - $T = \{\}$
- $V' = \{A,C,P\}$

```
função ANULÁVEIS(G) retorna V'
  entradas: G, Uma GLC na forma (V,Σ,R,P)
  saídas:  V', Conjunto {X ∈ V | X é anulável}

  V' ← {};
  repita
    T ← {Y ∉ V' | Y → z ∈ R e z ∈ V'*};
    V' ← V' ∪ T
  até T = {}
  retorne V'
```

Eliminação de Regras λ

- Seja a GLC $G = (\{V, \Sigma, R, P\}$. Uma gramática $G' = \{V, \Sigma, R', P\}$ equivalente à G, mas sem regras λ é obtida com os seguintes passos:
  - Cada regra de R cujo lado direito não possua variáveis anuláveis é inserida em R'
  - Cada regra de R cujo lado direito possua variáveis anuláveis deve ser inserida em R' para todas as combinações das variáveis anuláveis presentes ou não
  - Se P for anulável, adicione a regra $P \rightarrow \lambda$ em R'

Exemplo 6

- Seja a GLC $G = (\{P, A, B, C\}, \{a,b,c\}, R, P)$, em que R contém as seguintes regras:
  - $P \rightarrow APB \mid C$
  - $A \rightarrow AaaA \mid \lambda$
  - $B \rightarrow BBb \mid b$
  - $C \rightarrow cC \mid \lambda$

- Determine uma gramática G' equivalente que não contenha regras λ.

Exemplo 6

* Pelo exemplo 5, as variáveis anuláveis de G são $\{P, A, C\}$.

* Então, o resultado é a GLC $G' = (\{P, A, B, C\}, \{a,b,c\}, R', P)$, em que R' contém as seguintes regras:
  * $P \rightarrow APB \mid PB \mid AB \mid B \mid C \mid \lambda$
  * $A \rightarrow AaaA \mid aa\mathrm{A} \mid \mathrm{A}aa \mid aa$
  * $B \rightarrow BBb \mid b$
  * $C \rightarrow cC \mid c$

Eliminação de Regras Unitárias

* Para o trabalho com formas normais, que será visto em sequência, é preciso eliminar as regras unitárias da GLC

* Para eliminar regras unitárias, o primeiro passo é determinar as variáveis encadeadas de G
  * Diz-se que uma variável Z de G é encadeada a uma variável X se $Z = X$ ou existe uma sequência de regras $X \rightarrow Y_1$, $Y_1 \rightarrow Y_2$, ..., $Y_n \rightarrow Z$
    * Se $n = 0$ então a regra é $X \rightarrow Z$
    * Ao conjunto de variáveis encadeadas a X é dado o nome $enc(X)$

Determinando Variáveis Encadeadas

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:  enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

* Seja a GLC $G = (\{E, T, F\}, \{+, *, (,), \mathrm{t}\}, R, P)$, em que R contém as seguintes regras:
  * $E \rightarrow E + T \mid T$
  * $T \rightarrow T * F \mid F$
  * $F \rightarrow (E) \mid \mathrm{t}$

* Determine os conjuntos:
  * $enc(E)$
  * $enc(T)$
  * $enc(F)$

Exemplo 7

* $\text{enc}(E) = \{\}$

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

* $enc(E) = \{\}$
  * $T = \{E\}$

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

- enc(E) = {}
  - T = {E}
- enc(E) = {E}

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

- $\text{enc}(E) = \{\}$
  - $\text{T} = \{E\}$
- $\text{enc}(E) = \{E\}$
  - $\text{T} = \{T\}$

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:  enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

* enc(E) = {}
  * T = {E}
* enc(E) = {E}
  * T = {T}
* enc(E) = {E,T}

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

- $enc(E) = \{\}$
  - $T = \{E\}$
- $enc(E) = \{E\}$
  - $T = \{T\}$
- $enc(E) = \{E,T\}$
  - $T = \{F\}$

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

* enc(E) = {}
  * T = {E}
* enc(E) = {E}
  * T = {T}
* enc(E) = {E,T}
  * T = {F}
* enc(E) = {E,T,F}

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

* $\mathrm{enc}(E) = \{\}$
  * $\mathrm{T} = \{E\}$
* $\mathrm{enc}(E) = \{E\}$
  * $\mathrm{T} = \{T\}$
* $\mathrm{enc}(E) = \{E,T\}$
  * $\mathrm{T} = \{F\}$
* $\mathrm{enc}(E) = \{E,T,F\}$
  * $\mathrm{T} = \{\}$

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

- $enc(E) = \{\}$
  - $T = \{E\}$
- $enc(E) = \{E\}$
  - $T = \{T\}$
- $enc(E) = \{E,T\}$
  - $T = \{F\}$
- $enc(E) = \{E,T,F\}$
  - $T = \{\}$

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

* $enc(E) = \{\}$
  * $T = \{E\}$
* $enc(E) = \{E\}$
  * $T = \{T\}$
* $enc(E) = \{E,T\}$
  * $T = \{F\}$
* $enc(E) = \{E,T,F\}$
  * $T = \{\}$

```
função ENCADEADAS(G, X) retorna enc(X)
  entradas: G, Uma GLC na forma (V,Σ,R,P)
            X, Uma variável de V
  saídas:   enc(X), o conjunto das variáveis encadeadas a X

  enc(X) ← {};
  T ← {X};
  repita
    enc(X) ← enc(X) ∪ T
    T ← {Y ∉ enc(X) | Z → Y ∈ R para algum Z ∈ enc(X)};
  até T = {}
  retorne enc(X)
```

Exemplo 7

- Usando o mesmo procedimento para os demais conjuntos, obtém-se

  - $enc(E) = \{E,T,F\}$
  - $enc(T) = \{T,F\}$
  - $enc(F) = \{F\}$

Eliminação de Regras Unitárias

* Seja uma GLC $G = (\{V, \Sigma, R, P\}$. Uma gramática $G' = \{V, \Sigma, R', P\}$ equivalente à G, mas sem regras unitárias é obtida inserindo a regra $X \rightarrow w$ quando:

  * $\mathrm{Y} \in enc(X)$; e
  * $\mathrm{Y} \rightarrow w \in R$; e
  * $w \notin V$;

Exemplo 8

- Seja a GLC G = ({E, T, F}, {+, *, (,), t}, R, E), em que R contém as seguintes regras:
  - $E \rightarrow E + T \mid T$
  - $T \rightarrow T * F \mid F$
  - $F \rightarrow (E) \mid \text{t}$

- Obtenha uma gramática G' equivalente a G, mas sem regras unitárias

Exemplo 8

* Construindo a GLC $G' = (\{E, T, F\}, \{+, *, (,), \text{t}\}, R', E)$, em que R' contém as seguintes regras:
  * $E \rightarrow E + T \mid T$
  * $T \rightarrow T * F \mid F$
  * $F \rightarrow (E) \mid \text{t}$

Exemplo 8

- Construindo a GLC $G' = (\{E, T, F\}, \{+, *, (,), \mathrm{t}\}, R', E)$, em que R' contém as seguintes regras:
  - $E \rightarrow E + T \mid T * F \mid F$
  - $T \rightarrow T * F \mid F$
  - $F \rightarrow (E) \mid \mathrm{t}$

Exemplo 8

* Construindo a GLC $G' = (\{E, T, F\}, \{+, *, (,), \mathrm{t}\}, R', E)$, em que R' contém as seguintes regras:
  * $E \rightarrow E + T \mid T * F \mid (E) \mid \mathrm{t}$
  * $T \rightarrow T * F \mid F$
  * $F \rightarrow (E) \mid \mathrm{t}$

Exemplo 8

- Construindo a GLC $G' = (\{E, T, F\}, \{+, *, (,), \text{t}\}, R', E)$, em que R' contém as seguintes regras:
  - $E \rightarrow E + T \mid T * F \mid (E) \mid \text{t}$
  - $T \rightarrow T * F \mid (E) \mid \text{t}$
  - $F \rightarrow (E) \mid \text{t}$

Exemplo 8

- O resultado é a GLC $G' = (\{E, T, F\}, \{+, *, (,), \mathrm{t}\}, R', E)$, em que R' contém as seguintes regras:
  - $E \to E + T \mid T * F \mid (E) \mid \mathrm{t}$
  - $T \to T * F \mid (E) \mid \mathrm{t}$
  - $F \to (E) \mid \mathrm{t}$

Manipulações em Sequência

* Aplicar as técnicas de eliminação em sequência pode fazer com que certos tipos de regra já eliminados reapareçam
  * Eliminando regras λ, podem reaparecer regras unitárias
    * $A \rightarrow BC$
    * $B \rightarrow \lambda$
  * Eliminando regras unitárias, podem reaparecer regras λ
    * $P \rightarrow \lambda$
    * $A \rightarrow P$
  * Eliminando regras λ, podem reaparecer variáveis inúteis
    * $B \rightarrow \lambda$, se esta for a única regra para B
  * Eliminando regras unitárias, podem reaparecer variáveis inúteis
    * $A \rightarrow B$ e B não aparece do lado direito de nenhuma outra regra

Ordem de Aplicação de Manipulações

1) Acrescenta-se uma regra da forma:

* $P' \to P$, onde P é a variável de partida da gramática e P' é uma variável nova, que passa a ser a nova variável de partida

2) Elimina-se as regras λ

3) Elimina-se as regras unitárias

4) Elimina-se os símbolos inúteis

* Variáveis e terminais

Obrigado.

joaopauloaramuni@gmail.com
joaopauloaramuni@fumec.br

UNIVERSIDADE FUMEC