

## PERTEMUAN 4

### TATA BAHASA BEBAS KONTEKS

#### A. TUJUAN PEMBELAJARAN

Setelah mempelajari materi ini, mahasiswa mampu menganalisis suatu bahasa berdasarkan tata bahasa bebas konteks.

#### B. URAIAN MATERI

##### 1. Tata bahasa Bebas Konteks

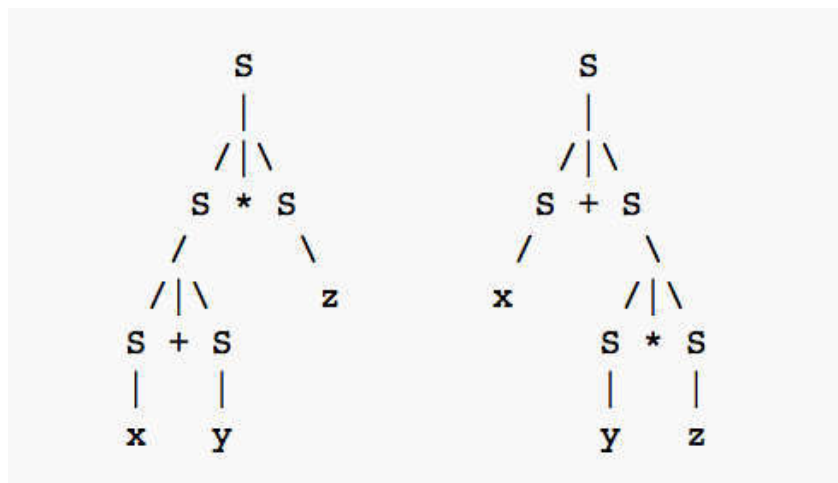
Tata bahasa bebas konteks adalah struktur matematika yang mengklasifikasikan string (urutan simbol) "valid" atau "tidak valid", dengan menentukan sekumpulan "aturan produksi" yang menentukan bagaimana string yang valid dapat dibentuk. Sekumpulan String yang dihasilkan oleh tata bahasa bebas konteks dikenal sebagai bahasa bebas konteks.

Menurut Sipser (1997) hal. 99 : “Tata bahasa tanpa konteks pertama kali digunakan dalam studi bahasa manusia. Salah satu cara untuk memahami hubungan istilah-istilah seperti kata benda, kata kerja, preposisi dan frasa masing-masing yang mengarah pada rekursi alami karena frasa kata benda dapat muncul di dalam frasa kata kerja ataupun sebaliknya”. Tata bahasa tanpa konteks dapat menangkap aspek penting dari hubungan ini. Meskipun tata bahasa bebas konteks pada awalnya digunakan untuk mempelajari bahasa manusia, kami menyelidiki bahasa bebas konteks dalam konteks yang lebih teoritis. Secara khusus, kami tertarik pada “urutan penghitungan”: mengingat suatu bahasa, urutan penghitungannya adalah urutan bilangan asli yang menyatakan berapa banyak string dari setiap panjang yang merupakan elemen bahasa.

Dalam pengertiannya setiap bahasa bebas konteks dihasilkan oleh tata bahasa bebas konteks. Selain tata bahasa bebas konteks yang bersifat sewenang-wenang, seseorang dapat mempertimbangkan juga tata bahasa bebas konteks yang sesuai dengan beberapa persyaratan, misalnya, produksi atau turunannya dari tata bahasa. Maka wajarlah untuk menanyakan apakah

setiap bahasa tanpa konteks memiliki sebuah tata bahasa bebas konteks, jika demikian bagaimana mengubah tata bahasa menjadi normal.

Salah satu alasan untuk mempertimbangkan bentuk normal mungkin karena matematika yang melekat, ketertarikan pada cara menghasilkan bahasa bebas konteks dengan tata bahasa deskripsi yang sesederhana mungkin. Selain itu, formulir normal dapat menyederhanakan pembuktian dan deskripsi di bidang bahasa formal dan parsing.



**Gambar 4.1** Dua pohon parse yang mendeskripsikan CFG yang menghasilkan string "x + y \* z".

Bahasa tanpa konteks dapat dihasilkan dari tata bahasa tanpa konteks. Mereka melakukan ini dengan mengambil satu set variabel yang didefinisikan secara rekursif, satu sama lain, oleh satu set aturan produksi. Tata bahasa bebas konteks dinamai demikian karena salah satu aturan produksi dalam tata bahasa dapat diterapkan apa pun konteksnya dan tidak bergantung pada simbol lain yang mungkin atau tidak mungkin ada di sekitar simbol tertentu yang menerapkan aturan padanya.

Ada empat komponen penting dalam deskripsi gramatikal suatu bahasa:

- Ada seperangkat simbol terbatas yang membentuk string bahasa yang sedang didefinisikan. Set ini adalah  $\{0,1\}$  dalam contoh palindrome yang baru saja kita lihat. Kami menyebut alfabet ini sebagai terminal atau simbol terminal.
- Ada satu set variabel, kadang-kadang juga disebut kategori non terminals atau sintaksis. Setiap variabel mewakili bahasa; yaitu, satu set string. Dalam

contoh di atas, hanya ada satu variabel  $P$ , yang digunakan untuk mewakili kelas palindrom di atas alfabet  $\{0,1\}$ .

- c. Salah satu variabel mewakili bahasa yang didefinisikan itu disebut simbol awal. Variabel lain mewakili kelas tambahan dari string yang digunakan untuk membantu mendefinisikan bahasa simbol awal. Pada contoh  $P$ , satu-satunya variabel, adalah simbol awal.
- d. Ada sekumpulan produksi atau aturan terbatas yang mewakili pengertian rekursif dari suatu bahasa. Setiap produksi terdiri dari:
  - 1) Variabel yang (sebagian) ditentukan oleh produksi. Variabel inilah yang sering disebut dengan kepala produksi.
  - 2) Simbol produksi  $\rightarrow$ .
  - 3) Sebuah string dari nol atau lebih terminal dan variabel. String ini, disebut body of the production, mewakili satu cara untuk membentuk string dalam bahasa variabel head. Dengan demikian, kami membiarkan terminal tidak berubah dan mengganti setiap variabel tubuh string apa pun yang diketahui dalam bahasa variabel itu.

## 2. Definisi Formal

Tata bahasa bebas konteks pada dasarnya terdiri dari seperangkat aturan tata bahasa yang terbatas. Untuk mendefinisikan aturan-aturan frame, kita asumsikan bahwa kita memiliki dua jenis simbol: terminal, yang merupakan simbol dari alfabet yang mendasari bahasa yang sedang dipertimbangkan, dan nonterminals, yang berperilaku seperti variabel yang berkisar di string terminal. Sebuah aturan dalam bentuk  $A \rightarrow \alpha$ , dimana  $A$  adalah satu nonterminal, dan sisi kanan  $\alpha$  adalah string simbol terminal dan / atau nonterminal. Seperti biasa, pertama-tama kita perlu mendefinisikan apa objeknya (konteks-free grammar), dan kemudian kita perlu menjelaskan cara penggunaannya. Tidak seperti automata, tata bahasa digunakan untuk menghasilkan pegas, daripada mengenali string.

Tata bahasa bebas konteks dapat dijelaskan dengan tupel empat elemen  $(V, \Sigma, R, S)$ , dimana

- a.  $V$ . adalah satu set variabel yang terbatas (yang non-terminal);
- b.  $\Sigma$  adalah himpunan yang terbatas ((terputus dari  $V$ )  $V$ ) simbol terminal;

- c. R adalah sekumpulan aturan produksi di mana setiap aturan produksi memetakan variabel ke string  $S \in (V \cup \Sigma)^*$ ;
- d. S (yang ada di V) yang merupakan simbol awal.

Contoh :

Munculkan tata bahasa yang akan menghasilkan bahasa bebas konteks (dan juga reguler) yang berisi semua string dengan tanda kurung yang cocok.

Variabel non-terminal =  $\{ (, ) \}$

Aturan produksi:

$$S \rightarrow ()$$

$$S \rightarrow AA$$

$$S \rightarrow (A)$$

Cara untuk memadatkan aturan produksi adalah sebagai berikut:

Kita bisa mengambil

$$S \rightarrow ()$$

$$S \rightarrow AA$$

$$S \rightarrow (A)$$

Dan menerjemahkannya menjadi satu baris:  $S \rightarrow () | AA | (A) | \epsilon$ , dimana  $\epsilon$  adalah string kosong.

### 3. Penyederhanaan tata bahasa bebas konteks

Seringkali kita jumpai sebuah tata bahasa bebas konteks terdapat beberapa aturan produksi yang tidak berperan dalam penurunan string, atau aturan produksi yang terlalu panjang pada pohon penurunan, yang menyebabkan percabangan terlalu lebar dan sulit dikendalikan. Perhatikan tata bahasa berikut :

$$S \rightarrow abcdefS \mid abcdef$$

Akan menyebabkan pohon penurun yang melebar kesebelah kanan.

Sedangkan tata bahasa berikut akan menghasilkan pohon yang tinggi dan sempit:

$$S \rightarrow A$$

$$A \rightarrow B$$

$$B \rightarrow C$$

$$C \rightarrow D$$

$$D \rightarrow a|A$$

Jika dicermati dari tata bahasa diatas terlihat bahwa sebenarnya aturan produksi yang sangat panjang, namun aturan produksi tersebut dapat disederhanakan menjadi:

$$S \rightarrow a$$

dengan membuang  $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$  yang merupakan proses penurunan yang “melingkar” dan dapat dihilangkan tanpa mengurangi bahasa yang dihasilkan oleh tata bahasa tersebut.

Secara garis besar tata bahasa bebas konteks dapat disederhanakan dengan:

- a. Membuang aturan produksi yang tidak berguna (useless)
- b. Menghilangkan produksi unit
- c. Menghilangkan produksi epsilon ( $\epsilon$ )

Berikut contoh Penyederhanaan CFG:

$$S \rightarrow AB | a$$

$$A \rightarrow a$$

\* Aturan produksi  $S \rightarrow AB$  tidak berarti karena B tidak memiliki penurunan.

#### 1) Penghilangan Produksi yang tidak berguna (useless)

Di sini produksi useless didefinisikan sebagai :

- a) Produksi yang memuat simbol variabel yang tidak memiliki penurunan yang akan menghasilkan terminal-terminal seluruhnya.
- b) Produksi yang tidak akan pernah dicapai dengan penurunan apapun dari simbol awal, sehingga produksi itu berlebih.

Contoh soal 1 :

$$S \rightarrow tU | V$$

$$U \rightarrow x | Tu$$

$$V \rightarrow uVu \mid twY \mid tu$$

$$Y \rightarrow vYU$$

Cara penyelesaian :

$$U \rightarrow \mid Tu \text{ ( T tidak punya penurunan , sehingga harus dihilangkan)}$$

$$Y \rightarrow vYU \text{ ( Y tidak ada turunan keterminal , sehingga harus dihilangkan)}$$

$$V \rightarrow twY \text{ ( Karena Y sudah di hilangkan , maka } V \rightarrow twY \text{ harus dihilangkan karena useless)}$$

Jadi hasilnya adalah :

$$S \rightarrow tU \mid V$$

$$U \rightarrow x$$

$$V \rightarrow uVu \mid tu$$

Contoh soal 2 :

$$S \rightarrow Tt \mid U$$

$$T \rightarrow tu \mid W$$

$$U \rightarrow u \mid X$$

$$V \rightarrow uu$$

$$x \rightarrow tXt$$

Cara penyelesaian :

$$T \rightarrow W \text{ tidak memiliki penurunan, jadi W useless harus dihapus}$$

$$V \rightarrow uu \text{ (V tidak berguna, karena ini Redudant)}$$

$$X \rightarrow tXt \text{ tidak memiliki penurunan ke terminal jadi harus dihapus}$$

$$U \rightarrow X \text{ tidak memiliki penurunan, jadinya useless}$$

Jadi hasilnya adalah :

$$S \rightarrow Tt \mid U$$

$$T \rightarrow tu$$

$$U \rightarrow u$$

## 2) Penghilangan Produksi Unit

Aturan produksi unit jika ruas kiri dan ruas kanan dalam aturan produksi tersebut berupa satu buah non terminal : A B, dengan A dan B adalah non terminal. Aturan produksi unit dapat dihilangkan sehingga menjadi bentuk tata bahasa yang lebih sederhana.

Contoh soal 1:

$$S \rightarrow Tt \mid U$$

$$U \rightarrow T \mid uu$$

$$T \rightarrow t \mid uv \mid U$$

Cara penyelesaian :

$$S \rightarrow U \text{ diganti menjadi } T \rightarrow uu.$$

$$T \rightarrow t \mid uv \mid uu$$

$$\text{Lalu } U \rightarrow T \mid uu \text{ bisa diubah menjadi } U \rightarrow t \mid uv \mid uu$$

$$\text{Lalu } S \rightarrow Tt \mid U \text{ pun bisa diubah menjadi } S \rightarrow Tt \mid t \mid uv \mid uu$$

Jadi hasilnya adalah :

$$S \rightarrow Tt \mid t \mid uv \mid uu$$

$$U \rightarrow t \mid uv \mid uu$$

$$T \rightarrow t \mid uv \mid uu$$

Contoh soal 2 :

$$S \rightarrow T \mid Tt$$

$$T \rightarrow U$$

$$U \rightarrow V \mid u$$

$$V \rightarrow W \mid tu$$

$$W \rightarrow u$$

Cara Penyelesaian :

$$V \rightarrow W \mid tu, W \text{ diganti } u. \text{ Menjadi } V \rightarrow u \mid tu$$

$$U \rightarrow V \mid u, \text{ Menjadi } U \rightarrow u \mid tu$$

$$T \rightarrow U, U \text{ diganti jadi } T \rightarrow uv \mid u$$

$$S \rightarrow T \mid Tt, \text{ menjadi } S \rightarrow u \mid t \mid Tt$$

Jadi hasilnya adalah :

$$S \rightarrow U \mid t \mid Tt$$

$$T \rightarrow tu \mid u$$

$$U \rightarrow tu \mid u$$

$$V \rightarrow u \mid tu$$

$$W \rightarrow u$$

### 3) Penghilangan Produksi $\epsilon$

Produksi epsilon ( $\epsilon$ ) adalah aturan produksi yang berbentuk :

$$\alpha \rightarrow \epsilon$$

Dalam semua aturan produksi yang ada dalam tata bahasa, aturan produksi epsilon dapat dihilangkan.

Perhatikan tata bahasa berikut :

$$S \rightarrow bcAaa \mid a$$

$$A \rightarrow \epsilon$$

Tata bahasa tersebut jika aturan  $A \rightarrow \epsilon$  disubstitusikan pada aturan pertama akan dihasilkan suatu tata bahasa :

$$S \rightarrow bcaa \mid a$$

Perhatikan tata bahasa berikut :

$$S \rightarrow bcAaa \mid a$$

$$A \rightarrow dd \mid \epsilon$$

Pada tata bahasa terakhir ini nonterminal A tidak dapat dihilangkan sama sekali karena ada dua kemungkinan penurunan dari A. Akan tetapi aturan produksi

$$A \rightarrow \epsilon$$

dapat dihilangkan dengan cara disubstitusikan pada aturan yang memuat A.

Jika  $A \rightarrow \epsilon$  disubstitusikan menghasilkan :  $S \rightarrow bcaa$

Dengan demikian hasil penyederhanaan adalah :

$$S \rightarrow bcAaa \mid a \mid bcaa$$

$$A \rightarrow dd$$

Contoh soal 1 :

$$S \rightarrow AB$$

$$A \rightarrow abB \mid aCa \mid \epsilon$$

$$B \rightarrow bA \mid BB \mid \epsilon$$

$$C \rightarrow \epsilon$$

Cara Penyelesaian :

a) Hilangkan  $\epsilon$  pada C

$$S \rightarrow AB$$



$$A \rightarrow abB \mid aa \mid \varepsilon$$

$$B \rightarrow bA \mid BB \mid \varepsilon$$

b) Hilangkan  $\varepsilon$  pada B

$$S \rightarrow AB \mid A$$

$$A \rightarrow abB \mid aa \mid \varepsilon \mid ab$$

$$B \rightarrow bA \mid BB$$

c) Hilangkan  $\varepsilon$  pada A

$$S \rightarrow AB \mid B$$

$$A \rightarrow abB \mid aa \mid ab$$

$$B \rightarrow bA \mid BB \mid b$$

Ketika sudah tidak ada  $\varepsilon$  maka selesai.

Jadi hasilnya adalah :

$$S \rightarrow AB \mid B$$

$$A \rightarrow abB \mid aa \mid ab$$

$$B \rightarrow bA \mid BB \mid b$$

Contoh soal 2 :

$$S \rightarrow aBCD \mid bb \mid A \mid \varepsilon$$

$$A \rightarrow CDa \mid ef$$

$$B \rightarrow b \mid Af \mid \varepsilon$$

$$C \rightarrow BbC \mid ea$$

$$D \rightarrow \varepsilon$$

Cara Penyelesaian :

a) Hilangkan  $\varepsilon$  pada D

$$S \rightarrow aBC \mid bb \mid A \mid \varepsilon$$

$$A \rightarrow Ca \mid ef$$

$$B \rightarrow b \mid Af \mid \varepsilon$$

$$C \rightarrow BbC \mid ea$$

b) Hilangkan  $\varepsilon$  pada B

$$S \rightarrow aBC \mid bb \mid A \mid \varepsilon \mid aC$$

$$A \rightarrow Ca \mid ef$$

$$B \rightarrow b \mid Af$$

$$C \rightarrow BbC \mid ea \mid bC$$

c) Hilangkan  $\epsilon$  pada S

$$S \rightarrow aBC \mid bb \mid A \mid aC$$

$$A \rightarrow Ca \mid ef$$

$$B \rightarrow b \mid Af$$

d)  $C \rightarrow BbC \mid ea \mid bC$

Jadi hasilnya adalah :

$$S \rightarrow aBC \mid bb \mid A \mid aC$$

$$A \rightarrow Ca \mid ef$$

$$B \rightarrow b \mid Af$$

$$C \rightarrow BbC \mid ea \mid bC$$

#### 4. Penerapan Tata Bahasa Bebas Konteks

Tata bahasa bebas konteks awalnya dirancang oleh N. Chomsky sebagai cara untuk mendeskripsikan bahasa alami. Janji itu belum terpenuhi. Namun, karena penggunaan untuk konsep yang didefinisikan secara rekursif dalam Ilmu Komputer telah berlipat ganda, begitu juga kebutuhan CFG sebagai cara untuk menggambarkan contoh konsep ini. Kami akan membuat sketsa dua kegunaan ini, satu lama dan satu baru.

Tata bahasa digunakan untuk mendeskripsikan bahasa pemrograman. Lebih penting lagi, ada cara mekanis untuk mengubah deskripsi bahasa sebagai CFG menjadi parser, komponen penyusun yang menemukan struktur program sumber dan merepresentasikan struktur tersebut dengan pohon parse. Aplikasi ini adalah salah satu penggunaan CFG yang paling awal, sebenarnya ini adalah salah satu cara pertama di mana ide-ide teoretis dalam Ilmu Komputer menemukan jalannya ke dalam praktik.

#### 5. Metode Parsing

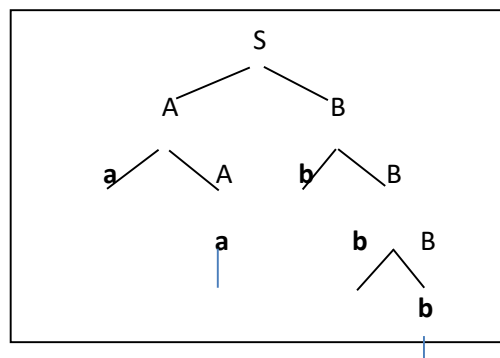
Tatabahasa bebas konteks merupakan dasar dalam pembentukan suatu parser/proses analisis sintaksis. Bagian sintaks dalam suatu kompilator kebanyakan di definisikan dalam tata bahasa bebas konteks. **Pohon penurunan** (*derivation tree/parse tree*) berguna untuk menggambarkan simbol-

simbol variabel menjadi simbol-simbol terminal. Setiap simbol variabel akan di turunkan menjadi terminal sampai tidak ada yang belum tergantikan.

Contoh, terdapat CFG dengan aturan produksi sebagai berikut dengan simbol awal S :

- a.  $S \rightarrow AB$
- b.  $A \rightarrow aA \mid a$
- c.  $B \rightarrow bB \mid b$

Maka jika ingin dicari gambar *pohon penurunan* dengan string : 'aabbb' hasilnya adalah seperti di bawah :



**Gambar 4.2** Context Free Grammar (CFG) - Parse Tree

Proses penurunan / parsing bisa dilakukan dengan cara sebagai berikut :

- a. Penurunan terkiri (*leftmost derivation*): simbol variabel terkiri yang di perluas terlebih dahulu.
- b. Penurunan terkanan ( *rightmost derivation* ) : simbol variabel terkanan yang diperluas terlebih dahulu.

Misal : Tata bahasa sbb :

- a.  $S \rightarrow aAS \mid a$
- b.  $A \rightarrow SbA \mid ba$

Untuk memperoleh string 'aabbaa' dari tata bahasa dilakukan dengan cara :

- a. Penurunan terkiri:  $S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$
- b. Penurunan terkanan :  $S \Rightarrow aAS \Rightarrow aAa \Rightarrow aSbAa \Rightarrow aAbbaa \Rightarrow aabbaa$

### C. SOAL LATIHAN/TUGAS

1. Jelaskan perbedaan tata bahasa regular dengan tata bahasa bebas konteks!
2. Tentukan bahasa-bahasa yang dibangkitkan dari CFG-CFG berikut:
  - a.  $S \rightarrow bSb ; S \rightarrow \epsilon$
  - b.  $S \rightarrow aSb ; S \rightarrow \epsilon$
  - c.  $S \rightarrow bSa ; S \rightarrow \epsilon$
  - d.  $S \rightarrow bSb \mid aSa ; S \rightarrow \epsilon$
3. Sederhanakan tata bahasa berikut dengan membuang produksi unit :  
 $G = \{S \rightarrow AC \mid ABab ; A \rightarrow B ; B \rightarrow C ;$   
 $C \rightarrow abC \mid aa \mid \epsilon \}$

### D. REFERENSI

- Nijhoit Anton. (1980). Context-Free Grammars: Covers, Normal Forms, and Parsing. Berlin, Heidelberg, New York.
- Flajolet Philippe. (1987). Analytic models and ambiguity of context-free languages. In Theoretical Computer Science. North-Holland
- Hamzah Amir. (2009). Teori Bahasa dan Otomata. Yogyakarta: Akprind
- Arlindo L. Oliveira. (2000). Grammatical Inference: Algorithms and Applications. Lisbon: 5th International Colloquium

## GLOSARIUM

**Tata bahasa bebas konteks** adalah sebuah tata bahasa dimana tidak terdapat pembatasan pada hasil produksinya Contoh pada aturan produksi :  $\alpha \rightarrow \beta$  batasannya hanyalah ruas kiri ( $\alpha$ ) adalah sebuah simbol variable non terminal.

**Pohon parse** : suatu pohon dimana akarnya (root) adalah simbol awal grammar (starting symbol), setiap node dalam (inner node) adalah simbol nonterminal, dan daunnya (leaf) dibaca dari kiri ke kanan adalah deretan token masukan.