Anggota kelompok:

* Marco Dominicus (2019077002)
* Gregorius Rizcy Orlando Pradana (2022071012)
* Dika Arnanda Putra (2022071014)
* Fayed Yamani (2022071017)
* Muhammad Fathansyah Wijaya (2022071038)

1. **Tentukan bahasa-bahasa yang dibangkitkan dari CFG berikut:**

A group of black letters

Description automatically generated

Untuk soal ini, kita perlu menentukan bahasa (string) yang bisa dibangkitkan dari masing-masing CFG.

1. S 🡪 bSb ; S 🡪 ε

CFG ini bisa menghasilkan string yang memiliki karakter 'b' di setiap ujungnya dengan string yang simetris di dalamnya.

Bahasa yang dibangkitkan: semua string dengan urutan b simetris seperti "", "bb", "bbbb", "bbbbbb", dst.

1. S 🡪 aSb ; S 🡪 ε

CFG ini menghasilkan string yang memiliki 'a' di depan dan 'b' di belakang, dengan simetri tertentu.

Bahasa yang dibangkitkan: semua string dengan jumlah 'a' sama dengan jumlah 'b' dalam bentuk simetris.

1. S 🡪 bSa ; S 🡪 ε

CFG ini menghasilkan string dengan 'b' di depan dan 'a' di belakang.

Bahasa yang dibangkitkan: string dengan jumlah 'b' sama dengan jumlah 'a', seperti "", "ba", "bbaa", dst.

1. S 🡪 bSb | aSa ; S 🡪 ε

CFG ini menghasilkan string simetris, baik dengan 'b' atau 'a'.

Bahasa yang dibangkitkan: string simetris dengan kombinasi 'a' dan 'b', seperti "", "aa", "bb", "aba", "bab", dst.

1. **Sederhanakanlah tata bahasa berikut dengan membuang produksi yang tidak berguna:**

Tata bahasa pertama:

A group of black text

Description automatically generated

Jawaban:

Bahasa yang dibangkitkan oleh grammar G tersebut dapat dilacak dari:

Aturan penurunan: S 🡪 AaB 🡪 abaB 🡪 abaBa 🡪 abaBaa 🡪 abaBa…a 🡪 abaBb…ba…a

Atau penurunan: S 🡪 Aaa 🡪 abaa

Atau penurunan: S 🡪 ABab 🡪 εBbbab 🡪 Bb…bab 🡪 Bbb…bbab

Atau penurunan: S 🡪 Baa 🡪 Baaa 🡪 Baa…aa

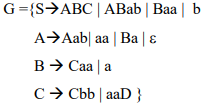
Jika diperhatikan, aturan produksi S 🡪 Baa, A 🡪 Ba, B 🡪 Ba, dan B 🡪 Bbb tidak akan pernah menghasilkan string terminal karena tidak adanya aturan non terminal 🡪 terminal. Terlihat juga bahwa aturan produksi yang melibatkan non terminal C dan D tidak akan pernah digunakan untuk memproduksi string terminal karena tidak dapat dilacak mundur sedemikian sehingga ia merupakan aturan produksi yang berasal dari S. Dengan demikian atruan produksi yang melibatkan non terminal B, C, dan D adalah aturan produksi yang tidak berguna dalam grammar tersebut dan oleh karenanya ia dapat dibuang.

Hasil penyederhanaannya adalah:

G = {S 🡪 Aaa | b

A 🡪 ab | aa | ε}

Tata bahasa kedua:



Jawaban:

Bahasa yang dibangkitkan oleh grammar G tersebut dapat dilacak dari:

Aturan penurunan: S 🡪 ABC 🡪 AabaCbb 🡪 aaabaCbbbb 🡪 aaabaCbb…bb

Atau penurunan: S 🡪 ABab 🡪 εa 🡪 a

Atau penurunan: S 🡪 Baa 🡪 Caa 🡪 Cbbaa 🡪 Cbbbbaa 🡪 Cbb…bbaa

Atau penurunan: S 🡪 b

Jika diperhatikan, terlihat bahwa aturan produksi yang melibatkan non terminal C tidak akan pernah menghasilkan string terminal tidak adanya aturan non terminal 🡪 terminal. Dengan demikian atruan produksi yang melibatkan non terminal C adalah aturan produksi yang tidak berguna dalam grammar tersebut dan oleh karenanya ia dapat dibuang.

Hasil penyederhanaannya adalah:

G = {S 🡪 ABab 🡪 Baa 🡪 b

A 🡪 Aab | aa | Ba | ε

B 🡪 a}

1. **Sederhanakan tata bahasa berikut dengan membuang produksi unit:**

G = { S 🡪 AC | ABab ;

A 🡪 B ;

B 🡪 C ;

C 🡪 abC | aa | ε }

Jawaban:

Hasil yang disederhanakan adalah:

S 🡪 CC | CCab

C 🡪 abC | aa | ε

1. **Sederhanakan tata bahasa berikut dengan membuang produksi ε:**

A group of black text

Description automatically generated

Jawaban:

Tata bahasa awal:

1. S 🡪 bB | AA | aA
2. B 🡪 aaB | ε
3. A 🡪 bA | ε

Langkah 1: Hapus produksi ε dari B dan A:

* Dari B 🡪 aaB | ε, kita bisa menyimpulkan bahwa B bisa menjadi string kosong atau aaB. Oleh karena itu, B bisa menjadi string aa atau lebih panjang (misalnya aaaa).
* Dari A 🡪 bA | ε, kita simpulkan bahwa A bisa menjadi string kosong atau satu atau lebih b (misalnya b, bb, dll.).

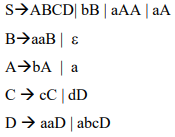
Langkah 2: Substitusi produksi yang melibatkan A dan B:

Dari S 🡪 bB | AA | aA, kita gantikan B dan A dengan hasil dari langkah sebelumnya:

* S 🡪 bB: Karena B bisa menjadi string kosong atau aa atau lebih panjang, maka bB menghasilkan string b atau baa, baaaa, dan seterusnya.
* S 🡪 AA: Karena A bisa menjadi string kosong atau beberapa b, maka AA bisa menjadi string kosong (ε) atau b, bb, bbb, dan seterusnya.
* S 🡪 aA: Ini bisa menghasilkan a, ab, abb, abbb, dan seterusnya.

Hasil tata bahasa yang disederhanakan:

1. S 🡪 b | baa | baaaa | AA | a | ab | abb | abbb.
2. A 🡪 bA | ε (bisa menghasilkan b, bb, bbb, dll.).
3. B 🡪 aaB | ε (bisa menghasilkan aa, aaaa, dll.).
4. **Sederhanakan CFG berikut sesederhana mungkin:**



Jawaban:

Produksi dengan nilai ε ada di B.

Jadi untuk setiap produksi yang menggunakan B, kita bikin alternatif tanpa B.

**CFG yang telah disederhanakan:**

* S 🡪 ABCD | b | aAA | aA.

**bB 🡪 b (karena B bisa menghasilkan ε)**

* B → aaB | ε
* A → bA | a (tidak perlu diubah)
* C → cC | dD (tidak perlu diubah)
* D → aaD | abcD (tidak perlu diubah)