



Pemrosesan Bahasa Alami Lanjut

Perkuliahahan Pekan Keenam
Constituency parsing

Oleh: Ade Romadhony



Referensi

- Speech and Language Processing, Dan Jurafsky,
<https://web.stanford.edu/~jurafsky/slp3/12.pdf>
<https://web.stanford.edu/~jurafsky/slp3/13.pdf>
- Slide dari Natalie Parde,
http://www.natalieparde.com/teaching/cs_421_fall2019/Syntactic%20and%20Dependency%20Parsing.pdf



Kerangka Bahasan

- Pendahuluan *syntactic parsing*
- Apa itu *constituency parsing*
- Review *context free grammar* (CFG)
- Metode *parsing*: *top-down* dan *bottom-up parsing*
- Algoritma CKY

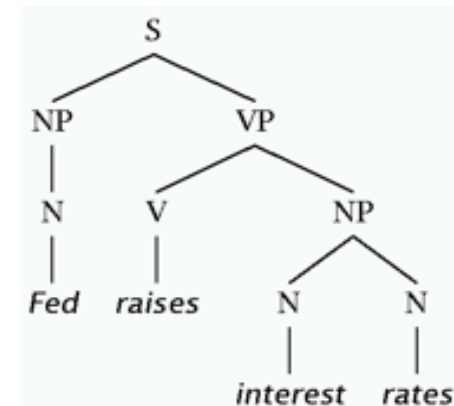


Apa itu *syntactic parsing*?

Proses otomatis untuk mengenali dan memberikan struktur sintaktik (tata Bahasa/*grammar*) kalimat

Dua sudut pandang dalam struktur linguistik: *constituency* dan *dependency parser* (1)

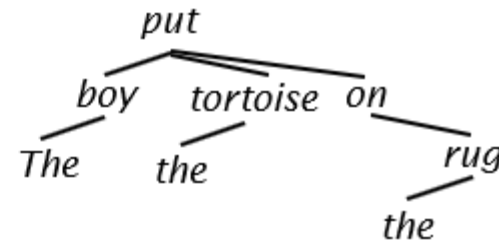
- *Constituency* (struktur frase) yang akan kita bahas kali ini
 - Struktur frase terdiri atas beberapa kata dalam bentuk *constituent* bersarang
 - Seperti apa *constituent* itu? Kita bisa lihat dari beberapa karakteristik, antara lain:
 - *Distribution*: sebuah constituent dapat muncul di posisi yang berbeda, contoh:
John talked [about drugs] [to the children]
John talked [to the children] [about drugs]
 - *Substitution*, contoh:
I sat [on the box/right on the top of the box/there]



Dua sudut pandang dalam struktur linguistik: *constituency* dan *dependency parser* (2)

- Struktur *dependency*

Struktur *dependency* menunjukkan keterkaitan antara satu kata dengan kata lainnya



Review *Context Free Grammar*

- Apa itu *Grammar*?
 - Ingat *definisi Grammar* di kuliah *Teori Bahasa dan Automata*? 😊
 - *Grammar* : aturan-aturan yang menjelaskan struktur suatu fenomena.
 - Dalam bahasa, *grammar* atau tata bahasa: aturan yang menjelaskan bagaimana merangkai kata-kata menjadi kalimat.
 - Istilah lain: syntax. Secara umum, berarti hal-hal yang berkaitan dengan tata-bahasa.
 - Dengan *grammar*, kita bisa memahami struktur kalimat -> model lebih kaya dan ekspresif.

Context Free Grammar (CFG): sebuah grammar formalism

- Cara yang paling umum untuk memodelkan *constituency*
- CFG = *Context-Free Grammar* = *Phrase Structure Grammar* = BNF = *Backus-Naur Form*
- Definisi formal CFG
 - $G = (T, N, S, R)$
 - T: himpunan simbol terminal (lexicon)
 - N: himpunan simbol non-terminal
 - S: symbol *start* (salah satu simbol nonterminal)
 - R: aturan produksi dalam bentuk $X \rightarrow \gamma$, di mana X adalah sebuah nonterminal dan γ adalah sekuens yang terdiri atas symbol terminal dan nonterminal (boleh kosong).
 - Sebuah *grammar* G akan membangkitkan sebuah Bahasa/*language* L.

Contoh CFG

- $G = (T, N, S, R)$
- $T = \{\text{that, this, a, the, man, book, flight, meal, include, read, does}\}$
- $N = \{S, NP, NOM, VP, Det, Noun, Verb, Aux\}$
- $S = S$
- $R = \{$

| | |
|----------------------------|---|
| $S \rightarrow NP VP$ | $Det \rightarrow \text{that} \mid \text{this} \mid \text{a} \mid \text{the} \text{ (i)}$ |
| $S \rightarrow Aux NP VP$ | $Noun \rightarrow \text{book} \mid \text{flight} \mid \text{meal} \mid \text{man} \text{ (ii)}$ |
| $S \rightarrow VP$ | $Verb \rightarrow \text{book} \mid \text{include} \mid \text{read} \text{ (iii)}$ |
| $NP \rightarrow Det NOM$ | $Aux \rightarrow \text{does}$ |
| $NOM \rightarrow Noun$ | |
| $NOM \rightarrow Noun NOM$ | |
| $VP \rightarrow Verb$ | $\text{this book read include man ? This book [det NOM] read [verb] man [noun] } \rightarrow \text{kurang det sebelum man}$ |
| $VP \rightarrow Verb NP\}$ | $\text{that man read book (tidak bisa di-parse)}$ |

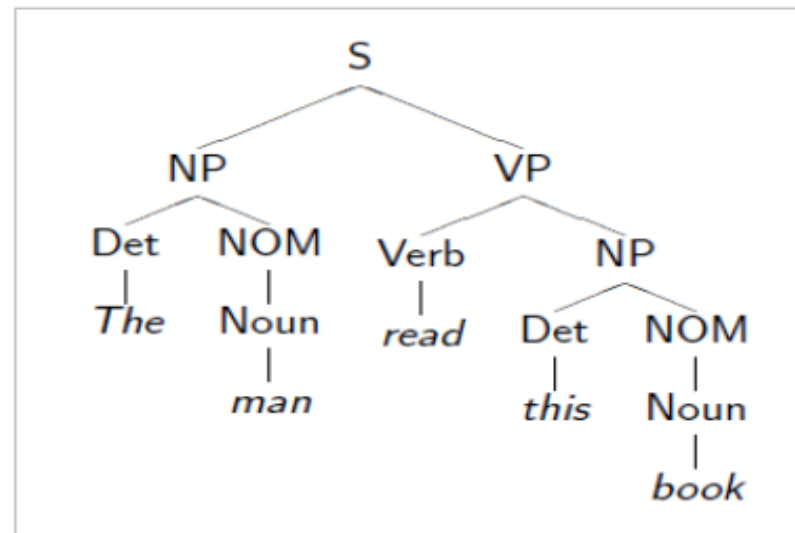
does the man read

$S \rightarrow Aux NP VP$
 $S \rightarrow \text{does NP VP}$
 $S \rightarrow \text{does Det NOM VP}$
 $S \rightarrow \text{does the NOM VP}$
 $S \rightarrow \text{does the Noun VP}$
 $S \rightarrow \text{does the man VP}$
 $S \rightarrow \text{does the man Verb}$
 $S \rightarrow \text{does the man read}$

Contoh pembangkitan kalimat berdasar aturan produksi pada *Grammar*

- S -> NP VP
 - > Det NOM VP
 - > The NOM VP
 - > The Noun VP
 - > The man VP
 - > The man Verb NP
 - > The man read NP
 - > The man read Det NOM
 - > The man read this NOM
 - > The man read this Noun
 - > The man read this book

Pohon *parse*





Aturan tata bahasa

- CFG mendefinisikan Bahasa formal: himpunan semua kalimat yang dapat diturunkan berdasarkan Grammar
 - Kalimat yang berada pada himpunan tersebut, disebut sebagai *grammatical*
 - Sementara yang berada di luar himpunan, disebut sebagai *ungrammatical*

Apa itu *parsing*?

- Menjalankan *grammar* untuk menemukan **struktur kalimat**
- *Parsing* dapat dipandang sebagai sebuah persoalan pencarian (*search problem*)
- Pencarian dilakukan secara menyeluruh, untuk mencari struktur yang sesuai dengan kalimat
- *Parsing* dapat dilakukan secara *top-down* atau *bottom-up*

Cobalah melakukan *parsing* sebuah kalimat dalam bahasa Inggris menggunakan perangkat berikut: <https://demo.allennlp.org/constituency-parsing/MTA4OTkwMA==>



Recognizer dan parser

- Ingat kembali materi kuliah TBA 😊
- *Recognizer*: sebuah program yang menerima masukan berupa *grammar* dan kalimat, dan akan mengembalikan YES jika kalimat tersebut **diterima** oleh *grammar*, dan sebaliknya NO
- *Parser*: selain berfungsi sebagai *recognizer*, juga akan mengembalikan himpunan pohon *parse* yang strukturnya sesuai dengan kalimat masukan



Kriteria parser: *soundness* dan *completeness*

- Kriteria *soundness* terpenuhi jika: sebuah *parser* selalu mengembalikan hasil *parsing* yang benar
- *Terminates*: kriteria yang dipenuhi parser jika ada jaminan bahwa tidak terjadi *infinite loop*
- *Complete*: sebuah parser dikatakan *complete* jika untuk tiap grammar dan kalimat kriteria *sound* dipenuhi, dan memenuhi kriteria *terminates*.
- Pada banyak kasus, kriteria *sound* tanpa harus *complete* dianggap cukup (misal: *probabilistic parser*)

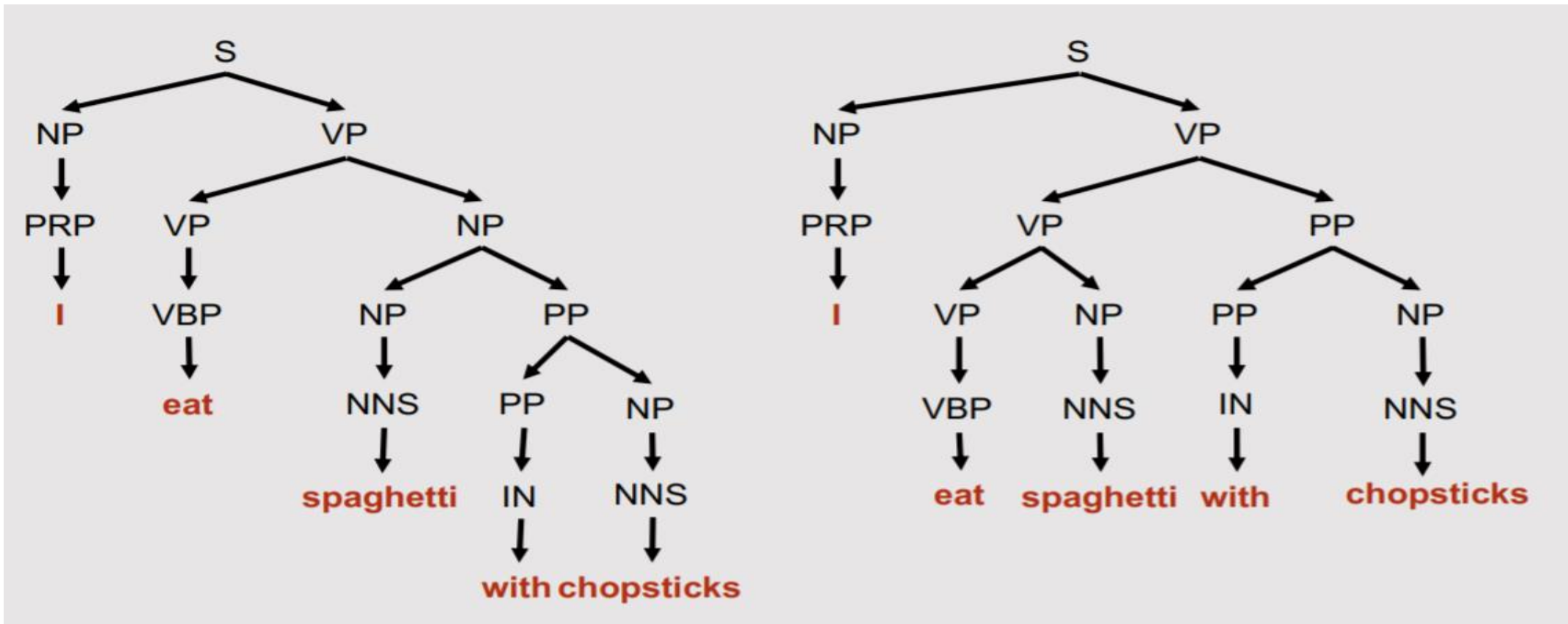
Mengapa *syntactic parsing* diperlukan?

- Pemeriksaan tata-bahasa
 - Kalimat yang tidak bisa di-*parse* kemungkinan besar tata bahasanya tidak benar (paling tidak sulit untuk dibaca)
- Analisis semantik
- Pemanfaatan di aplikasi *downstream*
 - Ekstraksi informasi
 - Sistem tanya jawab

Contoh aplikasi *parsing* (dengan metode berbasis statistika)

- *High precision question answering* [Pasca and Harabagiu SIGIR 2001]
- *Improving biological named entity finding* [Finkel et al. JNLPBA 2004]
- *Syntactically based sentence compression* [Lin and Wilbur 2007]
- *Extracting opinions about products* [Bloom et al. NAACL 2007]
- *Improved interaction in computer games* [Gorniak and Roy 2005]
- *Helping linguists find data* [Resnik et al. BLS 2005]
- *Source sentence analysis for machine translation* [Xu et al. 2009]
- *Relation extraction systems* [Fundel et al. *Bioinformatics* 2006]

Tantangan dalam pemrosesan bahasa: ambiguitas





Metode *parsing*

- *Top-down*
 - *goal-driven*
 - Pohon *parse* dibangun berawal dari simbol *Start*, hingga dicapai simbol terminal
- *Bottom-up*
 - *Data-driven*
 - Pohon *parse* dibangun berawal dari simbol terminal, hingga dicapai simbol *Start*
- Strategi penyelesaian: naïve (enumerasi semua solusi yang mungkin), atau menggunakan pendekatan pemrograman dinamis



Top-down parsing

- Mulai dari symbol Start (S), cek aturan produksi *Grammar* dengan sisi kiri berupa symbol S
- Lakukan proses rekursif untuk menelusuri semua jalur yang mungkin dari simbol S hingga mencapai level POS, dan terakhir kata (simbol terminal)
- Pohon diperoleh jika terdapat jalur dari S hingga semua simbol terminal yang sesuai dengan rangkaian kata pada kalimat masukan

Contoh *top-down parsing* (1)

- Contoh *grammar*

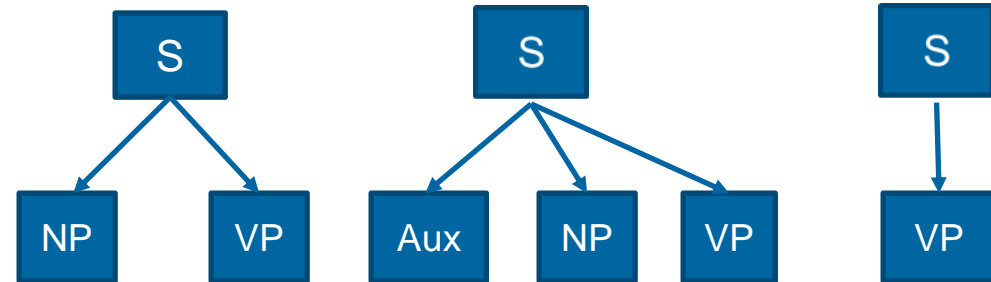
| Grammar | Lexicon |
|------------------------------------|---|
| $S \rightarrow NP VP$ | $Det \rightarrow that \mid this \mid the \mid a$ |
| $S \rightarrow Aux NP VP$ | $Noun \rightarrow book \mid flight \mid meal \mid money$ |
| $S \rightarrow VP$ | $Verb \rightarrow book \mid include \mid prefer$ |
| $NP \rightarrow Pronoun$ | $Pronoun \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper-Noun$ | $Proper-Noun \rightarrow Houston \mid NWA$ |
| $NP \rightarrow Det Nominal$ | $Aux \rightarrow does$ |
| $Nominal \rightarrow Noun$ | $Preposition \rightarrow from \mid to \mid on \mid near \mid through$ |
| $Nominal \rightarrow Nominal Noun$ | |
| $Nominal \rightarrow Nominal PP$ | |
| $VP \rightarrow Verb$ | |
| $VP \rightarrow Verb NP$ | |
| $VP \rightarrow Verb NP PP$ | |
| $VP \rightarrow Verb PP$ | |
| $VP \rightarrow VP PP$ | |
| $PP \rightarrow Preposition NP$ | |

Kalimat masukan yang akan di-*parse*:
Book that flight

Contoh *top-down parsing* (2)

Kalimat masukan yang akan di-parse:
 Book that flight

| Grammar | Lexicon |
|------------------------------------|---|
| $S \rightarrow NP VP$ | <i>Det</i> \rightarrow <i>that this the a</i> |
| $S \rightarrow Aux NP VP$ | <i>Noun</i> \rightarrow <i>book flight meal money</i> |
| $S \rightarrow VP$ | <i>Verb</i> \rightarrow <i>book include prefer</i> |
| $NP \rightarrow Pronoun$ | <i>Pronoun</i> \rightarrow <i>I she me</i> |
| $NP \rightarrow Proper-Noun$ | <i>Proper-Noun</i> \rightarrow <i>Houston NWA</i> |
| $NP \rightarrow Det Nominal$ | <i>Aux</i> \rightarrow <i>does</i> |
| $Nominal \rightarrow Noun$ | <i>Preposition</i> \rightarrow <i>from to on near through</i> |
| $Nominal \rightarrow Nominal Noun$ | |
| $Nominal \rightarrow Nominal PP$ | |
| $VP \rightarrow Verb$ | |
| $VP \rightarrow Verb NP$ | |
| $VP \rightarrow Verb NP PP$ | |
| $VP \rightarrow Verb PP$ | |
| $VP \rightarrow VP PP$ | |
| $PP \rightarrow Preposition NP$ | |

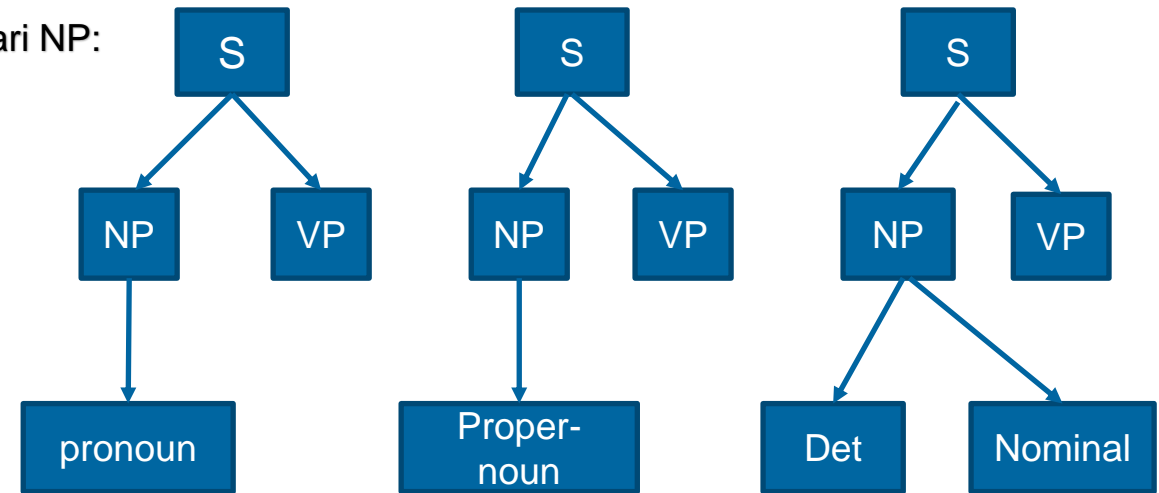


Contoh *top-down parsing* (3)

Kalimat masukan yang akan di-*parse*:
 Book that flight

Ekspan dari NP:

| Grammar | Lexicon |
|------------------------------------|---|
| $S \rightarrow NP VP$ | <i>Det</i> \rightarrow <i>that</i> <i>this</i> <i>the</i> <i>a</i> |
| $S \rightarrow Aux NP VP$ | <i>Noun</i> \rightarrow <i>book</i> <i>flight</i> <i>meal</i> <i>money</i> |
| $S \rightarrow VP$ | <i>Verb</i> \rightarrow <i>book</i> <i>include</i> <i>prefer</i> |
| $NP \rightarrow Pronoun$ | <i>Pronoun</i> \rightarrow <i>I</i> <i>she</i> <i>me</i> |
| $NP \rightarrow Proper-Noun$ | <i>Proper-Noun</i> \rightarrow <i>Houston</i> <i>NWA</i> |
| $NP \rightarrow Det Nominal$ | <i>Aux</i> \rightarrow <i>does</i> |
| $Nominal \rightarrow Noun$ | <i>Preposition</i> \rightarrow <i>from</i> <i>to</i> <i>on</i> <i>near</i> <i>through</i> |
| $Nominal \rightarrow Nominal Noun$ | |
| $Nominal \rightarrow Nominal PP$ | |
| $VP \rightarrow Verb$ | |
| $VP \rightarrow Verb NP$ | |
| $VP \rightarrow Verb NP PP$ | |
| $VP \rightarrow Verb PP$ | |
| $VP \rightarrow VP PP$ | |
| $PP \rightarrow Preposition NP$ | |

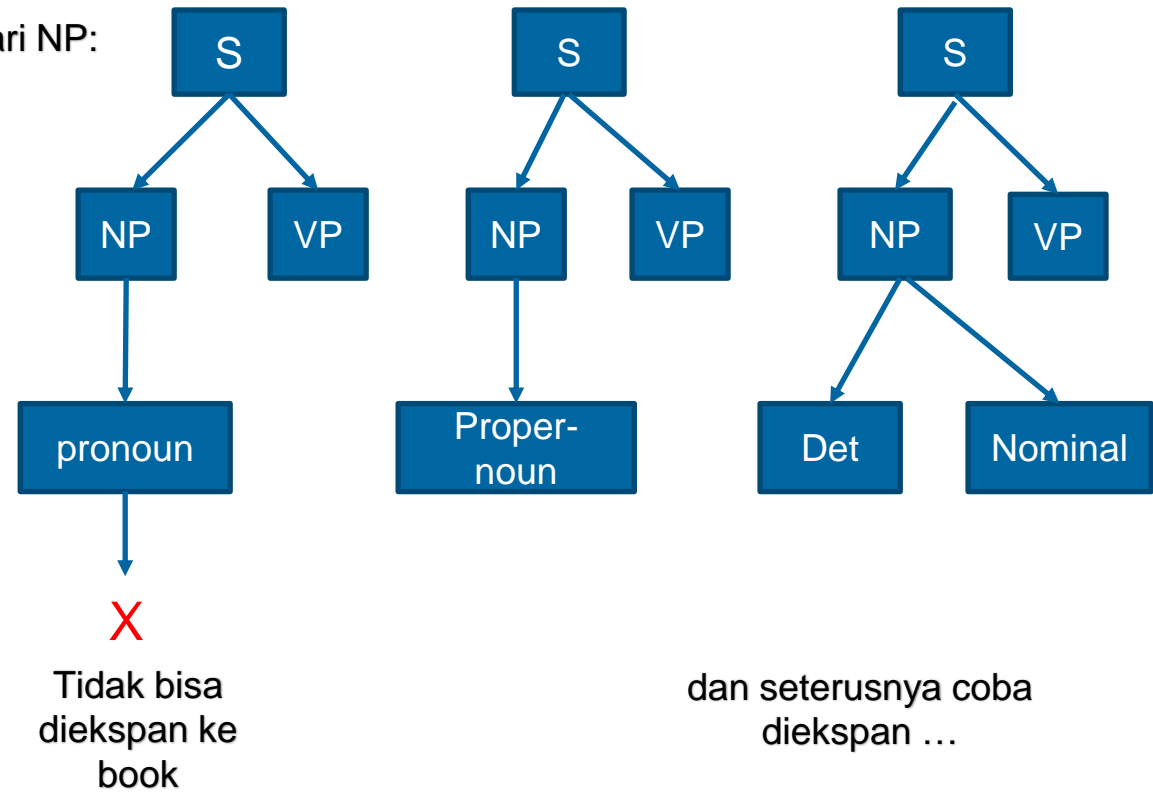


Contoh *top-down parsing* (3)

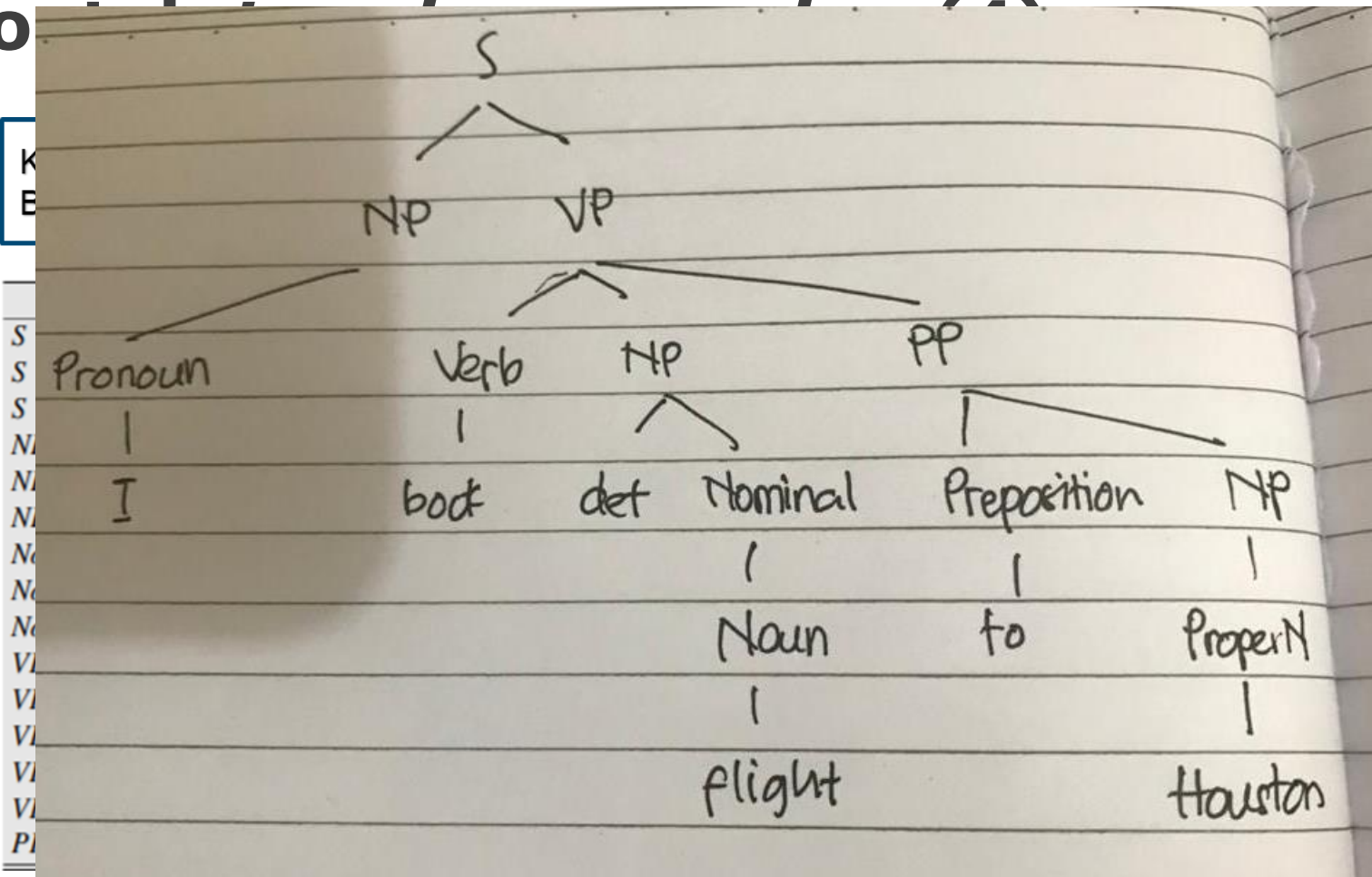
Kalimat masukan yang akan di-*parse*:
 Book that flight

| Grammar | Lexicon |
|------------------------------------|---|
| $S \rightarrow NP VP$ | <i>Det</i> \rightarrow <i>that</i> <i>this</i> <i>the</i> <i>a</i> |
| $S \rightarrow Aux NP VP$ | <i>Noun</i> \rightarrow <i>book</i> <i>flight</i> <i>meal</i> <i>money</i> |
| $S \rightarrow VP$ | <i>Verb</i> \rightarrow <i>book</i> <i>include</i> <i>prefer</i> |
| $NP \rightarrow Pronoun$ | <i>Pronoun</i> \rightarrow <i>I</i> <i>she</i> <i>me</i> |
| $NP \rightarrow Proper-Noun$ | <i>Proper-Noun</i> \rightarrow <i>Houston</i> <i>NWA</i> |
| $NP \rightarrow Det Nominal$ | <i>Aux</i> \rightarrow <i>does</i> |
| $Nominal \rightarrow Noun$ | <i>Preposition</i> \rightarrow <i>from</i> <i>to</i> <i>on</i> <i>near</i> <i>through</i> |
| $Nominal \rightarrow Nominal Noun$ | |
| $Nominal \rightarrow Nominal PP$ | |
| $VP \rightarrow Verb$ | |
| $VP \rightarrow Verb NP$ | |
| $VP \rightarrow Verb NP PP$ | |
| $VP \rightarrow Verb PP$ | |
| $VP \rightarrow VP PP$ | |
| $PP \rightarrow Preposition NP$ | |

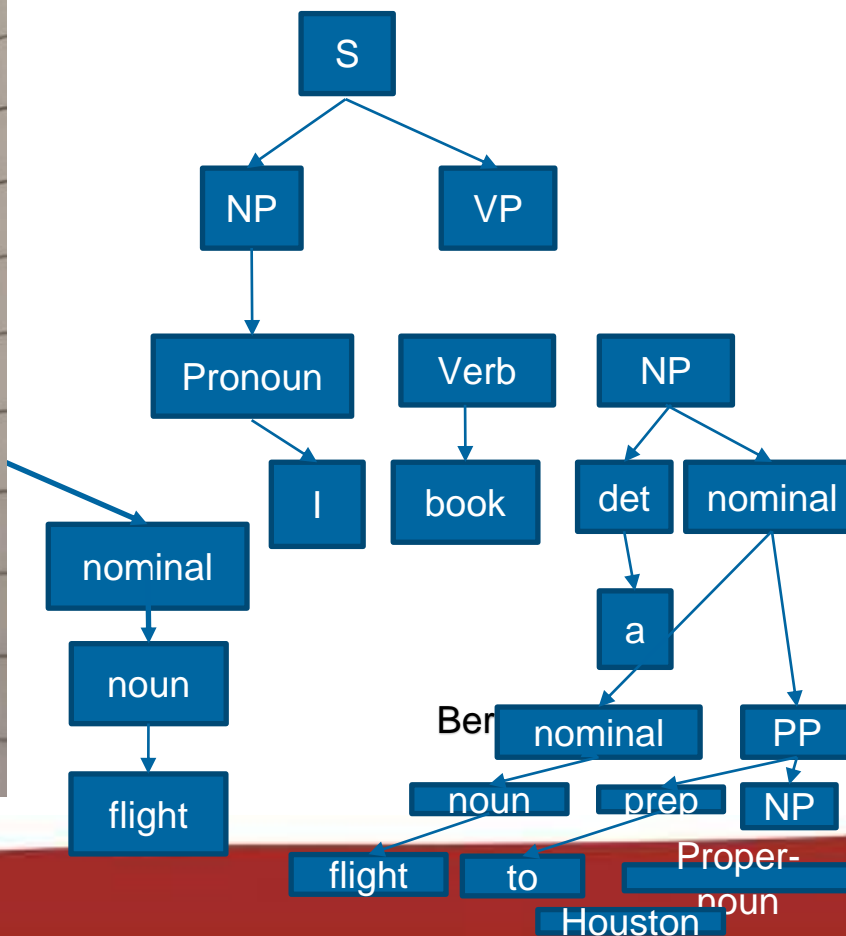
Ekspan dari NP:



Co



Coba parse kalimat:
 I book a flight to Houston





Bottom-up (LR) parsing (1)

- Algoritma *parsing* yang ditemukan paling awal
- Berdasarkan data (*data-driven*)
- Proses dimulai dari daftar kata pada kalimat yang di-*parse* (menjadi daftar awal *goal*)
- Lakukan proses pencocokan *goal list* ke aturan produksi. Jika ada bagian dari *goal list* yang muncul di sisi kanan (*right-hand side/RHS*), gantikan dengan aturan di sisi kiri (*LHS*)
- Proses *parsing* selesai saat *goal list* hanya berisi simbol *start/S*



Bottom-up parsing (2)

- Jika terdapat beberapa RHS aturan yang sesuai dengan goal list, dipilih salah satunya (persoalan pencarian)
- Proses pencarian dapat berupa DFS atau BFS (*depth* atau *breadth*)
- Metode standar yang umum diterapkan adalah *shift-reduce parsing*



Ide dasar *bottom-up parsing: shift-reduce*

- *Bottom-up parsing* melakukan *reduce* sebuah string ke simbol Start berdasarkan inversi aturan produksi

str <- input string of terminals

repeat

Identify β in str such that $A \rightarrow \beta$ is a production

(i.e., str = $\alpha\beta\gamma$)

Replace β by A in str (i.e., str becomes $\alpha A \gamma$)

until str = S

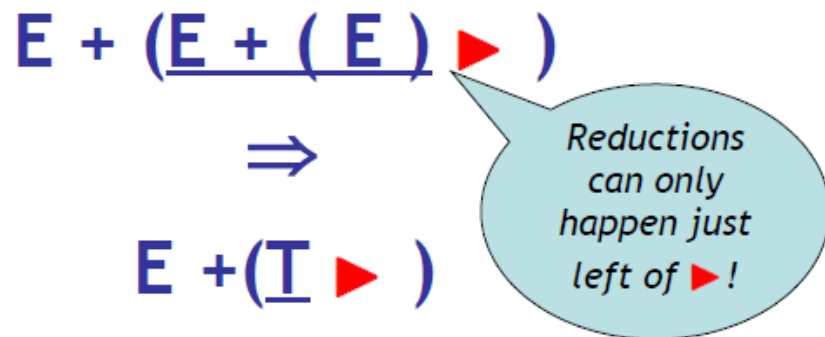


Bottom-up parsing: shift

- Shift: memindahkan pointer ► satu posisi ke arah kanan
- Shift/geser sebuah symbol terminal ke arah kiri string. Contoh:
 - dari : E + (int ►)
 - menjadi: E + (int) ►

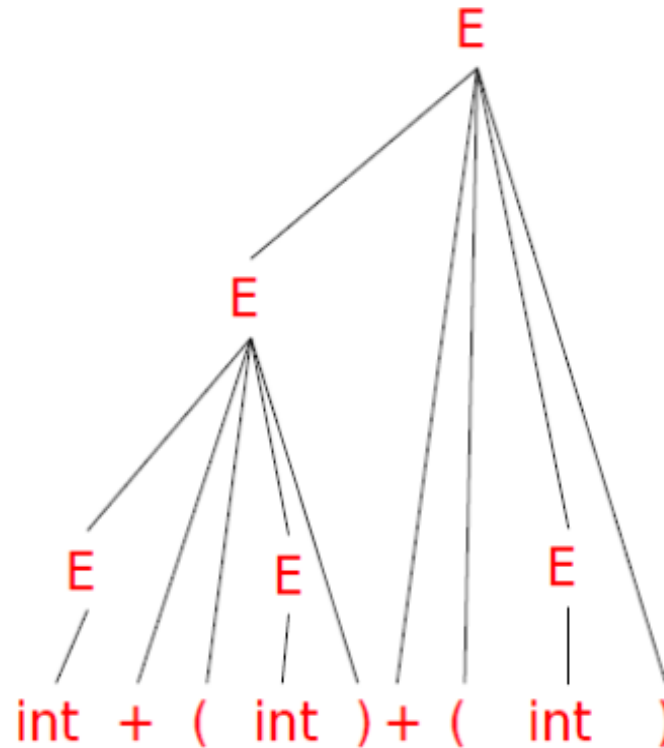
Bottom-up parsing: reduce

- Reduce: menerapkan invers aturan produksi di bagian paling kanan sisi kiri pointer
 - Contoh, jika diketahui sebuah aturan produksi $T \rightarrow E + (E)$, contoh *reduce* adalah sebagai berikut:



Contoh *parsing* dengan *shift-reduce*

▶ int + (int) + (int)\$ shift
 int ▶ + (int) + (int)\$ red. E → int
 E ▶ + (int) + (int)\$ shift 3 times
 E + (int ▶) + (int)\$ red. E → int
 E + (E ▶) + (int)\$ shift
 E + (E) ▶ + (int)\$ red. E → E + (E)
 E ▶ + (int)\$ shift 3 times
 E + (int ▶)\$ red. E → int
 E + (E ▶)\$ shift
 E + (E) ▶ \$ red. E → E + (E)
 E ▶ \$ accept



Grammar:
 $E \rightarrow E + (E) \mid \text{int}$

$E \rightarrow E + (E)$
 $E \rightarrow \text{int}$

sumber: <http://user.it.uu.se/~kostis/Teaching/KT1-11/Slides/handout06.pdf>

Latihan *bottom-up parsing*

- Berdasarkan *Grammar* berikut, lakukan *parsing* untuk kalimat: *book that flight*

| Grammar | Lexicon |
|------------------------------------|---|
| $S \rightarrow NP VP$ | $Det \rightarrow that \mid this \mid the \mid a$ |
| $S \rightarrow Aux NP VP$ | $Noun \rightarrow book \mid flight \mid meal \mid money$ |
| $S \rightarrow VP$ | $Verb \rightarrow book \mid include \mid prefer$ |
| $NP \rightarrow Pronoun$ | $Pronoun \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper-Noun$ | $Proper-Noun \rightarrow Houston \mid NWA$ |
| $NP \rightarrow Det Nominal$ | $Aux \rightarrow does$ |
| $Nominal \rightarrow Noun$ | $Preposition \rightarrow from \mid to \mid on \mid near \mid through$ |
| $Nominal \rightarrow Nominal Noun$ | |
| $Nominal \rightarrow Nominal PP$ | |
| $VP \rightarrow Verb$ | |
| $VP \rightarrow Verb NP$ | |
| $VP \rightarrow Verb NP PP$ | |
| $VP \rightarrow Verb PP$ | |
| $VP \rightarrow VP PP$ | |
| $PP \rightarrow Preposition NP$ | |

-> book that flight shift 1x
 book -> that flight reduce
 verb -> that flight shift
 verb that -> flight reduce
 verb Det -> flight shift
 verb Det flight -> reduce
 verb Det noun -> reduce
 verb Det Nominal -> reduce
 verb NP -> reduce
 VP -> reduce
 S



Top-down parsing vs bottom-up parsing

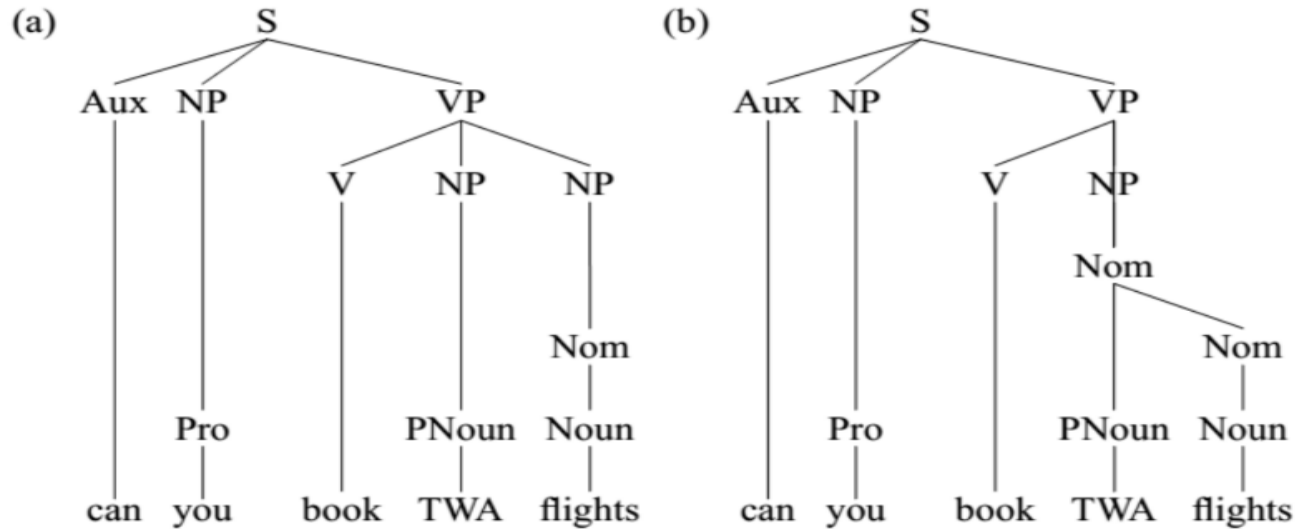
- Kelebihan dan kekurangan masing-masing pendekatan?

Probabilistic CFG: pengembangan CFG untuk mengatasi ambiguitas (1)

- Menambahkan keterangan probability pada aturan produksi, contoh:

| | | | | | | | | | |
|-----------------------------------|-------|----------------------------------|-------|--|-------|-------|--|-----|-------|
| $S \rightarrow NP VP$ | [.80] | $Det \rightarrow that$ | [.05] | | the | [.80] | | a | [.15] |
| $S \rightarrow Aux NP VP$ | [.15] | $Noun \rightarrow book$ | | | | | | | [.10] |
| $S \rightarrow VP$ | [.05] | $Noun \rightarrow flights$ | | | | | | | [.50] |
| $NP \rightarrow Det Nom$ | [.20] | $Noun \rightarrow meal$ | | | | | | | [.40] |
| $NP \rightarrow Proper-Noun$ | [.35] | $Verb \rightarrow book$ | | | | | | | [.30] |
| $NP \rightarrow Nom$ | [.05] | $Verb \rightarrow include$ | | | | | | | [.30] |
| $NP \rightarrow Pronoun$ | [.40] | $Verb \rightarrow want$ | | | | | | | [.40] |
| $Nom \rightarrow Noun$ | [.75] | $Aux \rightarrow can$ | | | | | | | [.40] |
| $Nom \rightarrow Noun Nom$ | [.20] | $Aux \rightarrow does$ | | | | | | | [.30] |
| $Nom \rightarrow Proper-Noun Nom$ | [.05] | $Aux \rightarrow do$ | | | | | | | [.30] |
| $VP \rightarrow Verb$ | [.55] | $Proper-Noun \rightarrow TWA$ | | | | | | | [.40] |
| $VP \rightarrow Verb NP$ | [.40] | $Proper-Noun \rightarrow Denver$ | | | | | | | [.40] |
| $VP \rightarrow Verb NP NP$ | [.05] | $Pronoun \rightarrow you$ | [.40] | | I | [.60] | | | |

Probabilistic CFG: pengembangan CFG untuk mengatasi ambiguitas (2)



| | Rules | P | | Rules | P |
|-------|-------------|-----|-------|-------------|-----|
| S | → Aux NP VP | .15 | S | → Aux NP VP | .15 |
| NP | → Pro | .40 | NP | → Pro | .40 |
| VP | → V NP NP | .05 | VP | → V NP | .40 |
| NP | → Nom | .05 | NP | → Nom | .05 |
| NP | → PNoun | .35 | Nom | → PNoun Nom | .05 |
| Nom | → Noun | .75 | Nom | → Noun | .75 |
| Aux | → Can | .40 | Aux | → Can | .40 |
| NP | → Pro | .40 | NP | → Pro | .40 |
| Pro | → you | .40 | Pro | → you | .40 |
| Verb | → book | .30 | Verb | → book | .30 |
| PNoun | → TWA | .40 | Pnoun | → TWA | .40 |
| Noun | → flights | .50 | Noun | → flights | .50 |

Pohon *parse* mana yang nilai probability-nya lebih tinggi?

Pohon a:
 $P(S \rightarrow \text{Aux NP VP}) * P(\text{aux-can}) * \dots$

Lebih lanjut tentang ambiguitas

- Ambiguitas struktural: terjadi saat sebuah *grammar* memungkinkan ada > 1 hasil *parse* dari 1 kalimat masukan
- Dua jenis bentuk ambiguitas:
 - *Attachment ambiguity*: terjadi saat sebuah *constituent* dapat dipasangkan pada > 1 posisi di pohon *parse*. Contoh: *I eat spaghetti with chopstick*
 - *Coordination ambiguity*: terjadi saat himpunan frase yang berbeda bisa digabungkan dengan sebuah konjungsi. Contoh: *I grabbed a muffin from the table marked "nut-free scones and muffins," hoping I'd parsed the sign correctly.*

Ambiguitas lokal

- Ingat contoh parsing kalimat *book that flight?*

POSTag *book* dapat berupa *verb* atau *noun*

| Grammar | Lexicon |
|------------------------------------|---|
| $S \rightarrow NP VP$ | $Det \rightarrow that \mid this \mid the \mid a$ |
| $S \rightarrow Aux NP VP$ | $Noun \rightarrow book \mid flight \mid meal \mid money$ |
| $S \rightarrow VP$ | $Verb \rightarrow book \mid include \mid prefer$ |
| $NP \rightarrow Pronoun$ | $Pronoun \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper-Noun$ | $Proper-Noun \rightarrow Houston \mid NWA$ |
| $NP \rightarrow Det Nominal$ | $Aux \rightarrow does$ |
| $Nominal \rightarrow Noun$ | $Preposition \rightarrow from \mid to \mid on \mid near \mid through$ |
| $Nominal \rightarrow Nominal Noun$ | |
| $Nominal \rightarrow Nominal PP$ | |
| $VP \rightarrow Verb$ | |
| $VP \rightarrow Verb NP$ | |
| $VP \rightarrow Verb NP PP$ | |
| $VP \rightarrow Verb PP$ | |
| $VP \rightarrow VP PP$ | |
| $PP \rightarrow Preposition NP$ | |

$S \rightarrow Aux NP VP$

CNF:

$S \rightarrow Aux A$

$A \rightarrow NP VP$

Efek dari ambiguitas antara lain juga membuat pencarian semakin rumit

- Solusi:

- *Backtracking*

Mengekspan *search space* secara incremental, mengeskplor 1 *state* pada 1 waktu. Saat menemui titik yang tidak sesuai dengan masukan, proses pencarian dilanjutkan dengan kembali ke solusi yang belum dieksplor/dikunjungi. Kelemahan: subpohon yang valid mungkin diabaikan (tidak diekplor lebih lanjut).

- Alternatif lain: pemrograman dinamis, di mana digunakan tabel untuk menyimpan *constituent* saat proses eksplorasi. Algoritma yang banyak digunakan: [CKY \(Cocke-Kasami-Younger\)](#), *Early*, *chart-parsing*

Algoritma CKY

- Algoritma parsing bottom-up dynamic programming
- Versi standar hanya dapat dijalankan pada CFG dalam bentuk Chomsky Normal Form (CNF). **Ingat tentang CNF?**
 - Syarat grammar yang memenuhi CNF adalah aturan produksinya berada dalam salah satu bentuk berikut:
 - $A \rightarrow B C$ (ruas kanan terdiri atas 2 simbol non terminal)
 - $A \rightarrow w$ (ruas kanan terdiri atas 1 simbol terminal)
 - Setiap CFG dapat diubah menjadi CNF yang akan menerima himpunan string yang sama persis dengan yang diterima oleh CFG awal.
 - **Bagaimana mengubah CFG ke CNF? (coba cari tahu)**

Algoritma CKY

- Menggunakan matriks dua dimensi (tabel) untuk representasi struktur pohon [alasan mengapa grammar harus CNF]
- Diberikan 1 kalimat dengan jumlah kata n , kita akan mengisi bagian segitiga bawah/atas tabel dengan ukuran $(n+1) \times (n+1)$
- Setiap sel tabel $[i,j]$ akan berisi symbol non-terminal yang merepresentasikan semua *constituent* pada posisi $i-j$
 - Sel yang merepresentasikan rangkaian kata berada di baris paling bawah/paling atas
 - Pengisian sel tabel dilakukan secara *bottom-up*



Contoh *parsing* dengan algoritma CKY (1)

- Diketahui sebuah *grammar* sebagai berikut:
 - $S \rightarrow AB \mid BC$
 - $A \rightarrow BA \mid a$
 - $B \rightarrow CC \mid b$
 - $C \rightarrow AB \mid a$
- Periksa apakah string 'baaba' merupakan bahasa yang diterima *grammar* tersebut!

sumber: <https://web.cs.ucdavis.edu/~rogaway/classes/120/winter12/CYK.pdf>

Contoh *parsing* dengan algoritma CKY (2)

- Membuat tabel, dan mengisi baris paling bawah

| | | | | |
|------------|---------------|---------------|------------|---------------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| {B} | {A, C} | {A, C} | {B} | {A, C} |
| b | a | a | b | a |

$S \rightarrow AB \mid BC$
 $A \rightarrow BA \mid a$
 $B \rightarrow CC \mid b$
 $C \rightarrow AB \mid a$

Contoh *parsing* dengan algoritma CKY (3)

- Mengisi baris kedua dari bawah, dari kombinasi sel di baris paling bawah: $X_{1,2} = (X_{i,i}, X_{i+1,j}) = (X_{1,1}, X_{2,2})$
 - $\{B\}\{A,C\} = \{BA, BC\}$
- Periksa di aturan produksi, mana yang RHS nya BA atau BC. Ada 2, yaitu $S \rightarrow AB \mid BC$ dan $A \rightarrow BA \mid a$, maka $X_{1,2} = \{S, A\}$

| | | | | |
|--------|--------|--------|-----|--------|
| | | | | |
| | | | | |
| | | | | |
| {S, A} | | | | |
| {B} | {A, C} | {A, C} | {B} | {A, C} |
| b | a | a | b | a |

$S \rightarrow AB \mid BC$
 $A \rightarrow BA \mid a$
 $B \rightarrow CC \mid b$
 $C \rightarrow AB \mid a$

ba, aa, ab, ba

Contoh *parsing* dengan algoritma CKY (4)

- Lanjutkan mengisi sel lain di baris kedua dengan cara yang sama

| | | | | | |
|--------|--------|--------|--------|--------|--|
| | | | | | |
| | | | | | |
| | | | | | |
| {S, A} | {B} | {S, C} | {S, A} | | |
| {B} | {A, C} | {A, C} | {B} | {A, C} | |
| b | a | a | b | a | |

$\{A, C\}\{A, C\} = \{AA, CC, AC, CA\}$

S → AB | BC
 A → BA | a
 B → CC | b
 C → AB | a

Contoh *parsing* dengan algoritma CKY (5)

- Mengisi baris ketiga dari bawah, dari kombinasi sel di baris paling bawah: $X_{1,3} = (X_{i,i}, X_{i+1,j}), (X_{i,i+1}, X_{i+2,j}) = (X_{1,1}, X_{2,3}), (X_{1,2}, X_{3,3})$
 $\rightarrow \{B\}\{B\} \cup \{S, A\}\{A, C\} = \{BB, SA, SC, AA, AC\} = Y$
- Periksa di aturan produksi, mana yang RHS nya Y. Tidak ada, maka $X_{1,3} = \emptyset$

| | | | | | |
|-------------|--------|--------|--------|--------|--|
| | | | | | |
| | | | | | |
| \emptyset | | | | | |
| {S, A} | {B} | {S, C} | {S, A} | | |
| {B} | {A, C} | {A, C} | {B} | {A, C} | |
| b | a | a | b | a | |

baa: b – aa, ba - a

S → AB | BC

A → BA | a

B → CC | b

C → AB | a

Contoh *parsing* dengan algoritma CKY (5)

- Lanjutkan mengisi sel lain di baris ketiga dengan cara yang sama

| | | | | | |
|-------------|--------|--------|--------|--------|--|
| | | | | | |
| | | | | | |
| \emptyset | {B} | {B} | | | |
| {S, A} | {B} | {S, C} | {S, A} | | |
| {B} | {A, C} | {A, C} | {B} | {A, C} | |
| b | a | a | b | a | |

$S \rightarrow AB \mid BC$
 $A \rightarrow BA \mid a$
 $B \rightarrow CC \mid b$
 $C \rightarrow AB \mid a$

Contoh *parsing* dengan algoritma CKY (6)

- Teruskan proses mengisi sel pada baris keempat dan kelima dengan cara yang sama
- Hasil akhir tabel:

Sel paling akhir mengandung S, artinya, baaba adalah hahasa yang dikenali *grammar*

- S → AB | BC
- A → BA | a
- B → CC | b
- C → AB | a

| | | | | |
|-----------|-------------|--------|--------|--------|
| {S, A, C} | ← $X_{1,5}$ | | | |
| ∅ | {S, A, C} | | | |
| ∅ | {B} | {B} | | |
| {S, A} | {B} | {S, C} | {S, A} | |
| {B} | {A, C} | {A, C} | {B} | {A, C} |
| b | a | b | a | b |
| | a | | b | a |



Kompleksitas algoritma CKY

- Kompleksitas waktu: $O(n^3)$
- Kompleksitas ruang: $O(n^2)$



Diskusi

- Dari mana informasi *grammar* diperoleh?
 - Didefinisikan oleh ahli Bahasa
 - Diambil dari korpus berlabel



THANK YOU