Computación y Estructuras Discretas III

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co
Ángela Villota
apvillota@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes
UNIVERSIDAD ICESI

2024-2

Agenda del día

1. **Decision Algorithms and Grammars**
   - The decision problem
   - The CYK Algorithm
2. **The Chomsky's Normal Form**
   - Ejercicios

Agenda del día

1. **Decision Algorithms and Grammars**
   - The decision problem
   - The CYK Algorithm

2. **The Chomsky's Normal Form**
   - Ejercicios

The decision problem

What is the decision problem?

The decision problem

What is the decision problem?

Given two (1) a property $\mathcal{P}$, (or a predicate) and (2) a study subject *s*, (let's say a grammar), the decision problem is to find an algorithm able to answer the **YES**/**NO** question: does s complies with the property $\mathcal{P}$,?

The decision problem

What is the decision problem?

Given two (1) a property $\mathcal{P}$, (or a predicate) and (2) a study subject *s*, (let's say a grammar), the decision problem is to find an algorithm able to answer the **YES**/**NO** question: does s complies with the property $\mathcal{P}$,?

What about grammars?

A decision problem in this context is to find an algorithm able to answer the following: Given a context-free grammar *G* and a string *s*, determine whether *s* can be generated by *G*.

Grammars & the decision Problem

Is there an algorithm able to solve the previous problem?

Grammars & the decision Problem

Is there an algorithm able to solve the previous problem?

- Indeed! there exists an efficient algorithm (polynomial) to solve the question about the belonging of a string to the language generated by a grammar.

Grammars & the decision Problem

Is there an algorithm able to solve the previous problem?

- Indeed! there exists an efficient algorithm (polynomial) to solve the question about the belonging of a string to the language generated by a grammar.
- The **CYK** algorithm (named in honor of the researchers Cocke, Younger, and Kasami) uses dynamic programming to determine whether a given string can be generated by a given CFG. It’s a fundamental algorithm in computational linguistics and parsing.

Agenda del día

1. **Decision Algorithms and Grammars**
   - The decision problem
   - The CYK Algorithm
2. **The Chomsky's Normal Form**
   - Ejercicios

The CYK Algorithm

More formally, the decision problem can be stated as

The CYK Algorithm

More formally, the decision problem can be stated as

Input

- A context-free grammar *G* in Chomsky Normal Form (CNF), represented as a set of production rules.
- A string *s* over the alphabet of terminals of *G*

Output

- Yes if there exists a derivation of *s* from the start symbol of *G*
- No otherwise.

So, when we apply the *CYK* algorithm to a real-world problem, the decision problem is whether a given string can be derived from a given *CFG* according to the rules of the grammar. If the CYK algorithm successfully finds a valid derivation for the string then the answer is**Yes**; otherwise, it's **No**.

The CYK Algorithm

Then, we have to learn how to transform a Grammar in the CNF first!!

The Chomsky's Normal Form

What is the CNF again??

The Chomsky's Normal Form

What is the CNF again??

A way to shape Context free grammars a grammars complies with the CNF iff:

1. *G* does not have unreachable symbols and unproductive symbols

The Chomsky's Normal Form

What is the CNF again??

A way to shape Context free grammars a grammars complies with the CNF iff:

1. G does not have unreachable symbols and unproductive symbols
2. G does not have λ productions (except maybe $S \to \lambda$).

The Chomsky's Normal Form

What is the CNF again??

A way to shape Context free grammars a grammars complies with the CNF iff:

1. G does not have unreachable symbols and unproductive symbols
2. G does not have λ productions (except maybe $S \rightarrow \lambda$).
3. All productions are in the form: $A \rightarrow a$ ó $A \rightarrow BC$ (binary productions).

The Chomsky's Normal Form

What is the CNF again??

A way to shape Context free grammars a grammars complies with the CNF iff:

1. G does not have unreachable symbols and unproductive symbols
2. G does not have λ productions (except maybe $S \to \lambda$).
3. All productions are in the form: $A \to a$ ó $A \to BC$ (binary productions).

Teorema

All CFG G has an equivalent in the CNF

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)
2. Eliminate unreachable symbols. (Remove any non-terminals that cannot be reached from the start symbol *S*.)

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)
2. Eliminate unreachable symbols. (Remove any non-terminals that cannot be reached from the start symbol S.)
3. Remove λ productions.

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)
2. Eliminate unreachable symbols. (Remove any non-terminals that cannot be reached from the start symbol S.)
3. Remove λ productions.
4. Eliminate unit productions (productions of the form A→B)

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)
2. Eliminate unreachable symbols. (Remove any non-terminals that cannot be reached from the start symbol S.)
3. Remove λ productions.
4. Eliminate unit productions (productions of the form A$\rightarrow$B)
5. The resulting productions (other than $S \rightarrow \lambda$) must be in the form $A \rightarrow a$ or in the form $A \rightarrow w$, where $a \in \Sigma$, $w \in V^*$, and $|w| \geq 2$.

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)
2. Eliminate unreachable symbols. (Remove any non-terminals that cannot be reached from the start symbol S.)
3. Remove λ productions.
4. Eliminate unit productions (productions of the form A$\rightarrow$B)
5. The resulting productions (other than $S \rightarrow \lambda$) must be in the form $A \rightarrow a$ or in the form $A \rightarrow w$, where $a \in \Sigma$, $w \in V^*$, and $|w| \geq 2$. The latter can be simulated with productions of the form $A \rightarrow BC$ or $A \rightarrow a$.

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)
2. Eliminate unreachable symbols. (Remove any non-terminals that cannot be reached from the start symbol S.)
3. Remove λ productions.
4. Eliminate unit productions (productions of the form A$\rightarrow$B)
5. The resulting productions (other than $S \rightarrow \lambda$) must be in the form $A \rightarrow a$ or in the form $A \rightarrow w$, where $a \in \Sigma$, $w \in V^*$, and $|w| \geq 2$. The latter can be simulated with productions of the form $A \rightarrow BC$ or $A \rightarrow a$. First, for each $a \in \Sigma$, a new variable T_a is introduced, with its only production being $T_a \rightarrow a$.

The Chomsky's Normal Form

Transforming a grammar in the CNF

1. Eliminate unproductive symbols. (Remove any non-terminals that cannot derive any terminal string)
2. Eliminate unreachable symbols. (Remove any non-terminals that cannot be reached from the start symbol S.)
3. Remove λ productions.
4. Eliminate unit productions (productions of the form A$\rightarrow$B)
5. The resulting productions (other than $S \rightarrow \lambda$) must be in the form $A \rightarrow a$ or in the form $A \rightarrow w$, where $a \in \Sigma$, $w \in V^*$, and $|w| \geq 2$. The latter can be simulated with productions of the form $A \rightarrow BC$ or $A \rightarrow a$. First, for each $a \in \Sigma$, a new variable T_a is introduced, with its only production being $T_a \rightarrow a$. Next, new variables are introduced, with binary productions, to simulate the desired productions.

Agenda del día

1. **Decision Algorithms and Grammars**
   - The decision problem
   - The CYK Algorithm
2. **The Chomsky's Normal Form**
   - Ejercicios