

EMPTY CATEGORIES IN TRANSFORMATIONAL RULES

Ruly Adha

IAIN Langsa

adharuly@yahoo.com

Abstract

There are three theories that are always developed in any study of language, namely theory of language structure, theory of language acquisition, and theory of language use. Among those three theories, theory of language structure is regarded as the most important one. It is assumed that if someone knows the structure of language, he/she can develop theories about how language is acquired and used. It makes Chomsky interested in developing the theory of language structure. Chomsky introduced a theory of grammar called Transformational Generative Grammar or Transformational Syntax. Transformational Syntax is a method of sentence formation which applies some syntactic rules (or also called transformational rules). Transformational rules consist of three types, namely movement transformation, deletion transformation, and substitution transformation. When those transformational rules are applied in a sentence, they will leave empty categories. Empty categories can be in the form of Complementizer (Comp), Trace, and PRO. The objectives of this article are to elaborate those empty categories; to show their appearance in the transformational rules; and to describe the characteristics of each empty category. Comp, Trace, and PRO can be found in movement and deletion transformation. In this article, tree diagram and bracketing are used as the methods of sentence analysis.

Keywords

Empty Categories, Transformational Rules, Comp, Trace, PRO

INTRODUCTION

Chomsky (in Radford, 1988) states that there are three inter-related theories which any detailed study of language ultimately seeks to develop, namely:

1. Theory of Language Structure
2. Theory of Language Acquisition
3. Theory of Language Use

The Theory of Language Structure is concerned with the defining structural properties of natural (human) language; the Theory of Language Acquisition is concerned with the question of how children acquire their native language(s); and the Theory of Language Use is concerned with the question of how linguistic and non-linguistic knowledge interact in speech comprehension and production.

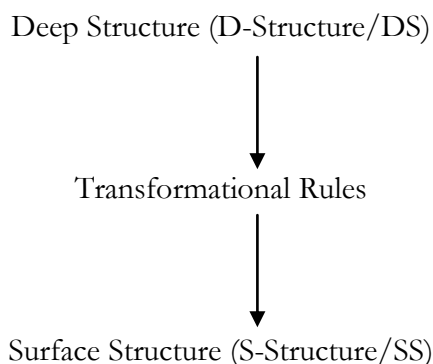
From the theories above, the theory of language structure is the most important one. The reason is that if someone knows the structure of language, he/she can develop theories about how it is acquired and used. Therefore, most of Chomsky's works have been devoted to the attempt to develop a Theory of Language Structure.

Theory of Language Structure can be developed in two steps. The first step is to formulate detailed descriptions (known technically as grammars) of particular languages for example English, French, etc. This is known as the study of Particular Grammar. A grammar of particular language will take the familiar form of a set of rules or principles which tell us how to speak and understand the language. More precisely, a grammar will comprise a set of rules or principles which specify how to form, pronounce, and interpret phrases and sentences in the language concerned. The second

step is to abstract from particular grammars common. This is known as the study of Universal Grammar (Radford, 1988).

Chomsky introduced a theory of grammar called Transformational Generative Grammar (TGG) or Transformational Syntax. TGG or Transformational Syntax is a method of sentence formation. In sentence formation, a sentence derives from Deep Structure (DS) which exists in the mind of speakers. Deep Structure will show the meaning intended by the speakers. Syntactic rules will be applied in the Deep Structure; then, Surface Structure (SS) will be obtained. Deep Structure can be defined as the structure of a sentence which is represented in the tree diagram and phrase markers which becomes the input of the application of (a) syntactic rule(s), which underlies the meaning of the sentence; meanwhile, Surface Structure is the structure which is obtained from the application of (a) syntactic rule(s) to the deep structure of a sentence, which is generally used in communication.

The syntactic rules (transformational rules) are applied to the Deep Structure and produce, as their output, the surface structure. In other words, the Deep Structure accounts for the meaning of the sentence; meanwhile, the surface structure accounts for the form of the sentence (Lester, 1971). Ouhalla (1999) prefers to say that Deep Structure as the underlying representation and Surface Structure as the derived representation. Schematically, the relation between Deep Structure and Surface Structure can be seen in the following.



Rules play an important role in sentence formation. In Transformational Generative Grammar, there are some rules which are involved in sentence formation. Before the sentence is uttered, it has experienced some steps of rules in the mind of speakers and the rules can be formulated. These rules can determine whether the sentence is grammatical or not. So, the rules are very significant in sentence formation. There is a number of transformational rules which are responsible for the derivation of various kinds of sentence construction such as movement transformation, deletion transformation, and substitution transformation.

In movement transformation, the constituents can only move into empty positions that have similar type. For instance, a moved NP can only be moved into an empty NP-position, and not into an empty VP-position. One example of movement transformation is Wh-Movement that can be seen in the following.

I know what she said.

DS : I know [CP [IP she said what]]

Wh-M : ↑

SS : I know what she said

In deletion transformation, the constituents are not moved, but deleted. The reason for this deletion is that the two similar lexical categories (words) have similar position. So, it is not necessary to mention the word twice. Therefore, one of the words must be deleted. One example of deletion transformation is the deletion of coreferential VP or gapping that can be seen in the following.

Andre likes tea and Tony coffee.

DS : Andre likes tea and Tony likes coffee

Gap. : Andre likes tea and Tony \emptyset coffee

SS : Andre likes tea and Tony coffee

Substitution transformation is the process of substitution of a certain constituent. The example of substitution transformation can be seen in interrogative sentence. See the following example.

Who wrote the statement?

DS [+Q] : someone wrote the statement

Wh-q : who wrote the statement

SS : who wrote the statement?

The changing Deep Structure into Surface Structure by applying transformational rules will leave empty categories. The empty categories can be in the form of Complementizer (Comp), Trace, and PRO. In the next session, those empty categories will be elaborated along with their characteristics.

DISCUSSION

Empty category is an important theory in Generative Syntax. It was introduced first by Chomsky in his book 'Aspects of the Theory of Syntax' in 1965. In his first book 'Syntactic Structures', Chomsky did not mention this theory, but he found this theory later and he discussed it in his next books as Jumino (2004) says:

In the first phase of the theory of Generative Linguistics (henceforth GL) which was marked by the publication of the legendary monograph *Syntactic Structures* in 1957, Chomsky as the pioneer had not proposed the so-called empty category.....In further development of the theory in the publication of *Aspects of the Theory of Syntax* in 1965, Chomsky made some revisions to the theory posited earlier, and thus known as a *Standard Theory*. At this phase empty category was introduced in the form of a dummy node symbolized as Δ . In 1975, Chomsky introduced the Trace Theory of Movement Rules in the publication of *Reflection on Language*.(p. 93-94)

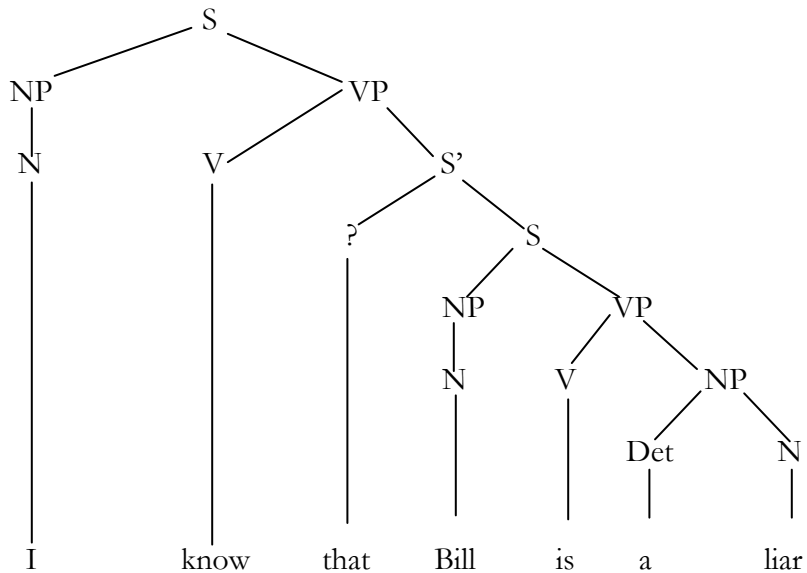
In the theory of Generative Linguistics, empty category consists of three types. They are Complementizer (Comp), Trace, and PRO.

1. Comp

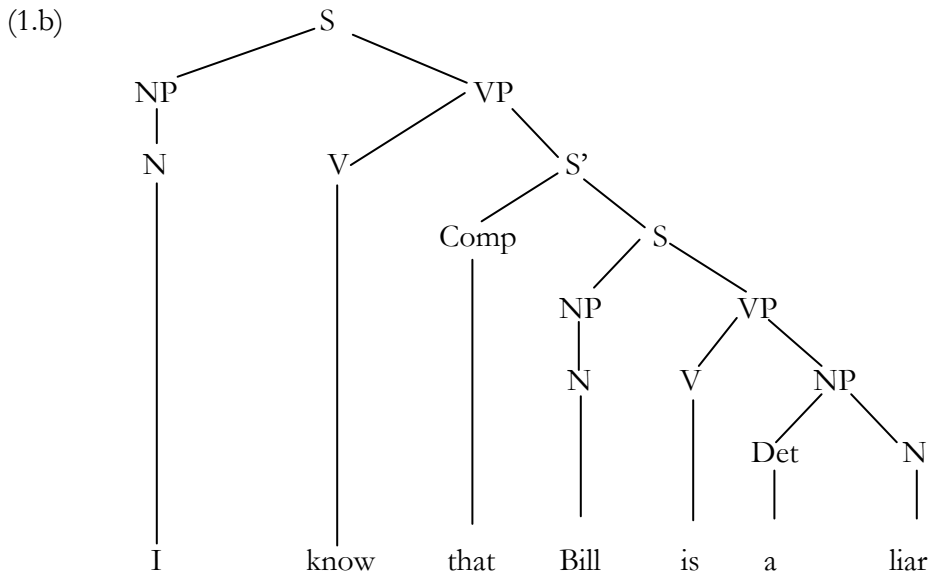
According to Jumino (2004), Complementizer is derived from its name in which the complement of a transitive verb in the form of a clause generally begins with a subordinator. He also adds that in the phrase markers, the subordinator cannot be treated properly without an additional

node the so-called Complementizer, now known as Comp. Without this node such a sentence as (1.a) cannot be represented in the proper way.

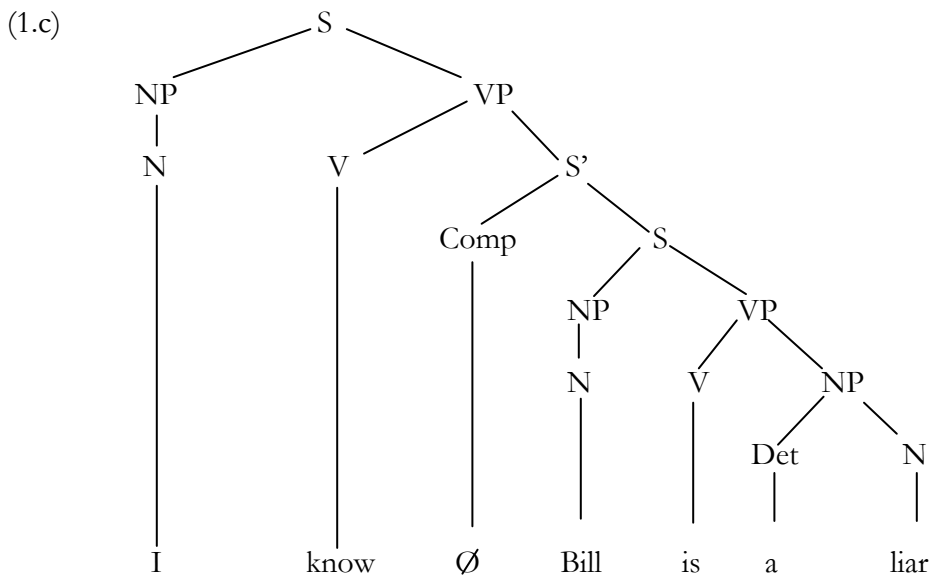
(1.a) I know that Bill is a liar.



In sentence (1.a) above, the label of the node under S' (S-bar) to treat the complementizer 'that' is still in a question mark (?). So, the proper name for this label is Comp as in (1.b).

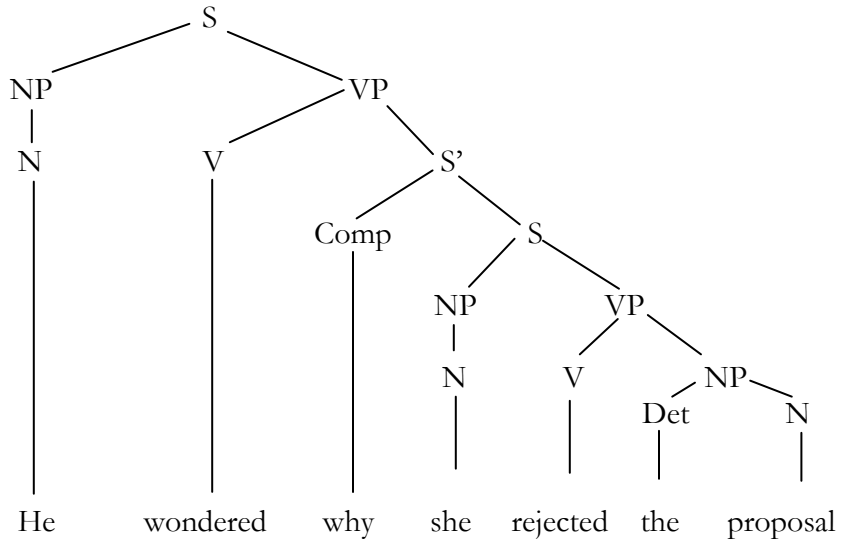


The presence of the subordinator ‘that’ in (1.b) is optional in English. Then, when such a complementation is expressed without a subordinator, the lexical category in the Comp becomes empty so that Comp in this sense is called an empty category as in (1.c).



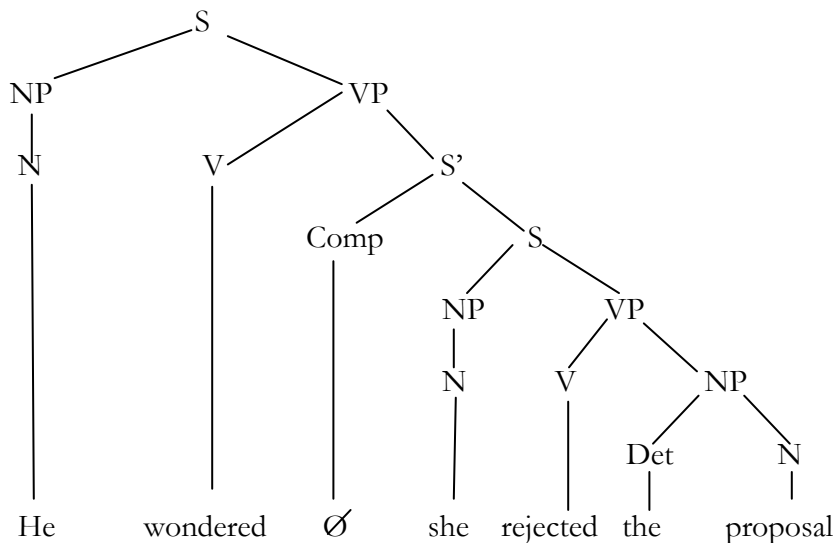
Instead of the complementizer ‘that’, Comp may also be occupied by other lexical categories such as wh-words as in (2.a).

(2.a) He wondered why she rejected the proposal (Jumino, 2004, p.100).



The wh-question ‘why’ in (2.a) is also under Comp but, again, it may also be empty as (2.b).

(2.b)

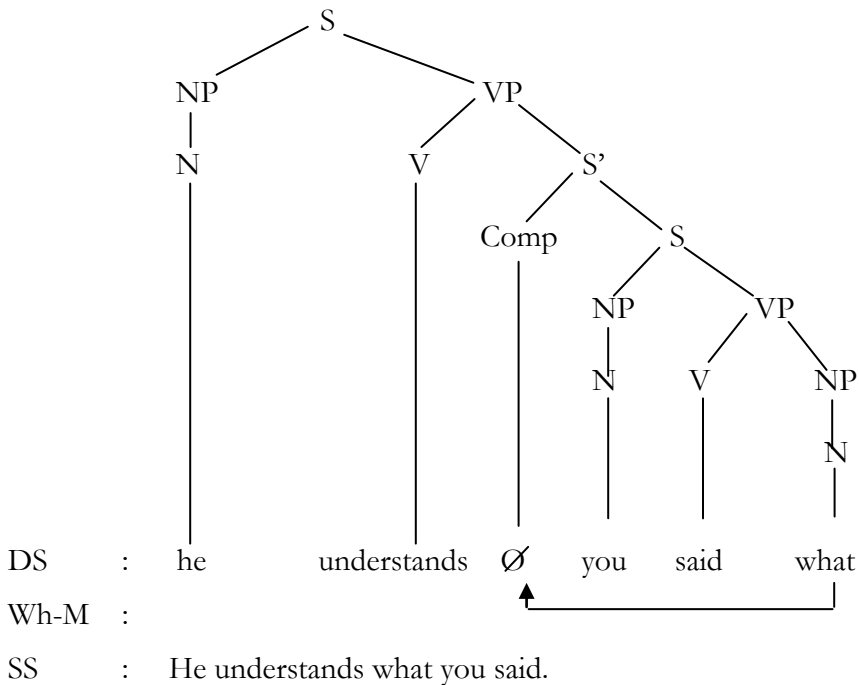


The provision of Comp as an empty category is felt necessary in the derivation of wh-movement. The wh-movement transformation is impossible unless the node Comp is provided in the D-Structure so that ‘what’ is obligatorily moved to the position of the empty Comp. This is postulated as a **Wh-Movement** rule as in Radford (1981:169) and Ouhalla (1999:70). The rule says:

*Move wh-XP to Comp or
Adjoin a wh-phrase immediately to the left of Comp*

For example:

(3) He understands what you said.



In the latest version of the theory of Generative Linguistics, the position of Comp principle in syntax becomes more important and gets its new name Complementizer Phrase (CP) which carries the same function as formulated earlier.

2. Trace

Trace is an empty category which is considered the most significant one in the theory of Generative Linguistics. It is the result of a movement transformation as postulated in the following rule called **Trace Convention** (Ouhalla, 1999,p.66). The rule says:

Movement transformations leave a trace behind

For a clear description, let us see the example below.

(4) I know what you mean (Jumino, 2004: 104).

DS : I know [CP [IP you mean what]]

Wh-M : I know [CP what_x [IP you mean t_x]]

SS : I know what you mean.

Trace in (4) was occupied by the wh-phrase ‘what’ and the movement transformation of ‘what’ from the object position of the embedded clause to the position of Comp in the S’ (S-bar) leaves an empty node called trace (t).

Trace principle gives rise to a number of deductions of syntactic evidence. First, the provision of empty category maintains the principle of the features attributed to the VP which is [+transitive], which implies that it must have an object. Thus, trace is the object of the VP which is represented in a null element. Secondly, trace in this respect satisfies proper government principle known as **Empty Category Principle** (ECP) as in Chomsky (1981:250) (Jumino, 2004:104). The rule says:

[Xe] must be properly governed

Thirdly, trace also has an antecedent which is derived from the moved element. The trace in (4) is properly governed by the verb ‘mean’ and it is coreferential to the wh-phrase ‘what’ as its antecedent, in which they have the same index.

Syntactically, trace may account for the various syntactic properties such as passive, topicalization, wh-movement, and cyclic rule (Jumino, 2004). For example:

a. in passive construction

- (5) [IP John I [VP wrote the book]]
 [IP The book_x I [VP was [VP written t_x (by John)]]]

b. in topicalization

- (6) [IP She I [VP hates [NP horror movies]]]
 [[NP Horror movies]_x, IP she I [VP hates [NP t_x]]]

c. in wh-movement

- (7) I know what she likes.

DS : I know [CP [IP she [VP likes [NP what]]]]

Wh-M : I know [CP what_x [IP she [VP likes [NP t_x]]]]

SS : I know what she likes.

d. in cyclic rule

- (8) Which girl do you know John will marry?

DS : [CP [IP you know [CP [IP John will marry which girl]]]]

SS : [CP which girl_x [IP do you know [CP t_x [IP John will marry t_x]]]]

3. PRO

PRO known as a big PRO is an empty category which typically occurs in the subject position of non-finite clauses. For example:

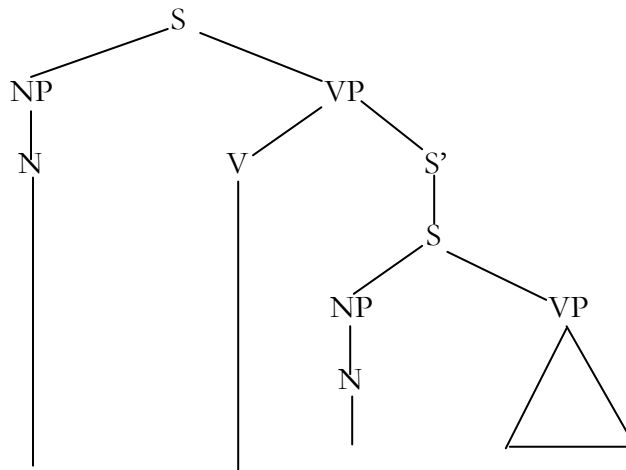
(9.a) Santo wanted to leave (Jumino, 2004).

DS : Santo_x wanted [*S'* [*S* Santo_x to leave]]

SS : Santo_x wanted [*S'* [*S* PRO_x to leave]]

In the tree diagram, the sentence (9.a) can be analyzed as follows:

(9.b)



DS : Santo_x wanted Santo_x to leave

CNPD : Santo_x wanted PRO_x to leave

SS : Santo wanted to leave.

The NP subject of the embedded non-finite clause ‘Santo’ is deleted because it is coreferential to the subject of the matrix clause. However, an NP in the embedded clause is not necessarily deleted if it is not coreferential to the NP subject of the matrix clause. For example:

(9.c) Santo_x wanted [*S'* [*S* Susy_y to leave]]

PRO may be identified from its features (Jumino, 2004). The first feature is that PRO is the result of a deletion transformation as can be seen in (9.a). In (9.a), the deletion transformation is called Coreferential NP Deletion (CNPD). However, a constituent is also deleted if it is unspecified

as in (10). The transformational rule in (10) is called Unspecified NP Deletion (UNPD).

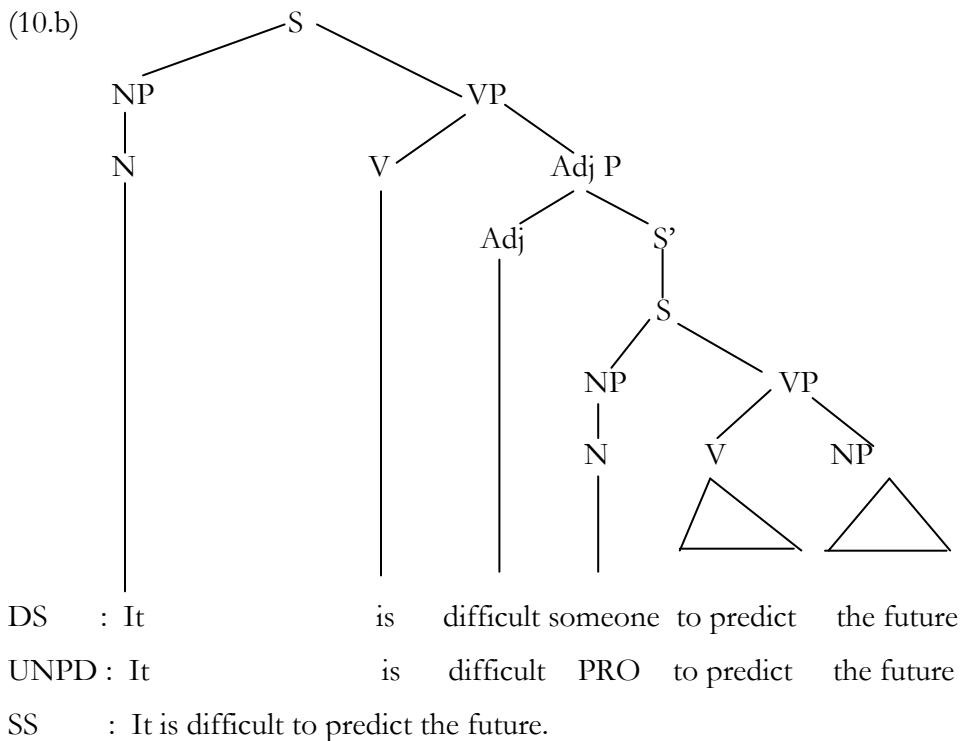
(10.a) It is difficult to predict the future.

DS : It is difficult [*S'* [*S* someone to predict the future]]

UNPD : It is difficult [*S'* [*S* PRO to predict the future]]

SS : It is difficult to predict the future.

In tree diagram, the sentence (10.a) can be analyzed as follows:



The subject NP in the non-finite embedded clause ‘someone’ is unspecified, so it can be deleted.

The second feature of PRO is that it is ungoverned as adopted from Ouhalla (1999:247) and Haegeman (1994:272) called **PRO Theorem** which says that:

PRO must be ungoverned

For example, let see the example of sentence (10) above in which the subject of the lower clause ‘someone’ is not governed by any category so that PRO in (10) is ungoverned. However, see another example below.

(11) It is difficult for someone to predict the future.

*It is difficult [*IP* for [*IP* PRO to predict the future]]

The sentence (11) is excluded as PRO in this respect is governed by the preposition ‘for’, thus violating PRO Theorem.

The third feature of PRO is that it may or may not have an antecedent. For example, PRO in (9.a) has the same index as the subject of the matrix clause ‘Santo’ which constitutes its antecedent and this kind of PRO is called a controlled PRO, and the antecedent is the controller. However, PRO may sometimes have no antecedent as in (10) in which the empty category PRO does not refer to any logical argument in the sentence and this kind of PRO is called an arbitrary PRO (Jumino, 2004:109).

The fourth feature of PRO is that it is pronominal as (12).

(12) To err is human.

[*CP* [*IP* PRO to err]] is human

PRO in (12) is pronominal as it may be taken to refer to a specific referent such as he, she, you, etc. or interpreted as equivalent to the arbitrary pronoun ‘one’. However, PRO may also be anaphoric as (13).

(13) Jim_x wanted [*CP* [*IP* he_x to leave]]

Jim_x wanted [*CP* [*IP* PRO_x to leave]]

PRO in (13) is anaphoric as it is dependent on another NP for its interpretation i.e. 'Jim' which can be observed from the index.

PRO is significant to account for not only infinitive clauses as given earlier but also other types of non-finite clauses in various distributions (Jumino, 2004:109). The case of gerund clause may also support the evidence of PRO Theorem as (14).

(14) Bill hates sitting on the front row.

DS : Bill_x hates [CP [IP Bill_x sitting on the front row]]

CNPD : Bill_x hates [CP [IP PRO_x sitting on the front row]]

SS : Bill hates sitting on the front row.

CONCLUSION

Sentence is derived from Deep Structure (DS) which exists in the mind of speakers. Deep Structure shows the meaning intended by the speakers. Syntactic rules will be applied in the Deep Structure; then, Surface Structure (SS) will be obtained. Deep Structure can be defined as the structure of a sentence which is represented in the tree diagram and phrase markers which becomes the input of the application of (a) syntactic rule(s), which underlies the meaning of the sentence; meanwhile, Surface Structure is the structure which is obtained from the application of (a) syntactic rule(s) to the deep structure of a sentence, which is generally used in communication. The syntactic rules (transformational rules) that are applied in the Deep Structure can be in the form of movement transformation, deletion transformation, or substitution transformation. In the process of changing Deep Structure into Surface Structure by using transformational rules, we can find empty categories. The empty categories that can be found are Complementizer (Comp), Trace, and PRO. Complementizer (Comp) is the

complement of a transitive verb in the form of a clause generally begins with a subordinator; Trace is an empty category which occurs because of the result of a movement transformation; and PRO is an empty category which occurs because of the result of a deletion transformation.

REFERENCE

- Bach, E. (1964). *An introduction to transformational grammars*. New York: Holt, Rinehart and Winston, Ltd.
- Brown, E.K. and Miller, J.E. (1982). *Syntax: generative grammar*. London: Hutchin Son & Co. Ltd.
- Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton Publishers.
- Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, Massachusetts: M.I.T. Press.
- Chomsky, N. (1977). On Wh-Movement. In Peter Culicover , *Formal Syntax*. (pp.73-132). New York: Academic Press.
- Chomsky, N. (1981). *Lectures on Government & Binding*. Dordrecht: Foris Publications.
- Chomsky, N. (1995). *The minimalist program*. Cambridge: MA: M.I.T. Press.
- Culicover, P. W. (1997). *Principles and parameters* : An introduction to syntactic theory. Oxford: Oxford University Press.
- Haegeman, L. (1994). *Introduction to Government & Binding theory*. Oxford: Blackwell.
- Lester, M. (1971). *Introductory transformational grammar of English*. New York: Holt, Rinehart, and Winston, Inc.
- Ouhalla, J. (1999). *Introducing transformational grammar (2nd ed)*. New York: Oxford University Press.

Radford, A. (1981). *Transformational syntax*. London: Cambridge University Press.

Radford, A. (1988). *Transformational grammar : A first course*. London: Cambridge University Press.

Radford, A. (1997). *Syntax : A minimalist introduction*. New York: Cambridge University Press.

Suhadi, J. (2004). The adequacy of empty category in the theory of syntax. *Julisa*, 4(2). Medan: Faculty of Literature UISU Medan.