Using this pattern, the following steps are carried out to extract simple sentences from the compound sentences.

1. Production (4a) can be written in terms of non-terminals and coordinating conjunction as such -

C -> S + <CC> + S (5a)

(1) To identify the pattern of compound sentence we need to first drop the conjunction gives the equation 5a, 5b, 5c and 5d from the corresponding equations 4b, 4c, 4d and 4e.

C -> S1 + S2 (5a)

C -> S1 + S2 - <subject2> (5b)

C -> S1 − < verb1 > − < object 1 > + S2 (5c)

C -> S1 + S2 − < subject 2 > − < verb2 > (5d)

Explanation: From the production (4a) we have already defined the non-terminal S and C for a compound sentence having particularly 2 sentences. S1 is simply a non-terminal with the same production as S but it is different from that of S2 (which again has the same production as S). We shall prove later that this works for more than 2 sentences too.

The productions 5b, 5c, 5d denote the different cases in which such declarative sentences fall under as explained in the corresponding equations 4b, 4c, 4d, 4e.

5a denotes the ideal case where the 2 sentences joined were simply, both independent clauses. This production is the definition of the first type. In such a type, simply removing the coordinating conjunction suffices to form the 2 separate simple sentences, while reserving both the sentences meaning.

5b denotes the case where the second sentence may not have a subject (subject2 denotes the subject of the second sentence. Such numbering is going to be useful in the further derivations). This indirectly means that the second sentence shares the subject with the first and is hence a dependent clause. This is type 2. For such a case (as we shall see later) we need only provide it with a subject (from the previous sentence or the clause it is dependent to) and complete the sentence to make it an independent clause. The two separated sentences would then be independent clauses and hence they would be decomposed to simple sentences from a compound sentence.

5c denotes the case where the first sentence is the dependent clause and the second sentence is the independent clause. Hence falling into type 3. 5d is an extension of this type, it only differs in the fact that it is the exact reverse of 5c making them both fall under the same type because the approach to decompose these compound sentences remain the same. We use the subject or object from the dependent clause along with the verb to complete the independent clause.

(2) The possibilities of the productions of the 2 sentences after splitting and detection of the type the sentences falls into are given in 6a,6b,6c

Type 1:

S1 → < subject 1 > + < verb1 > + < object1>

&

S2 → < subject2 > + < verb2 > + < object2 >

Type 2:

S1 → < subject 1 > + < verb1 > + < object1>

&

S2 → < verb2 > + < object2 >

Type 3:

S1 → < subject 1>

&

S2 → < subject2 > + < verb2 > + < object2 >

OR

S1 → < subject 1 > + < verb1 > + < object1>

&

S2 → < object2 >

(3) Once the type is known, the sentences are then simply joined with their missing pair to form the complete sentences. Hence since type 1 is already a form of 2 simple sentences, it has already been transformed. The other 2 types are then completed.

Type 2:

S1 → < subject 1 > + < verb1 > + < object1>

&

S2 → <subject1> + < verb2 > + < object2 >

Type 3:

S1 → < subject 1> + <verb2> + <object2>

&

S2 → < subject2 > + < verb2 > + < object2 >

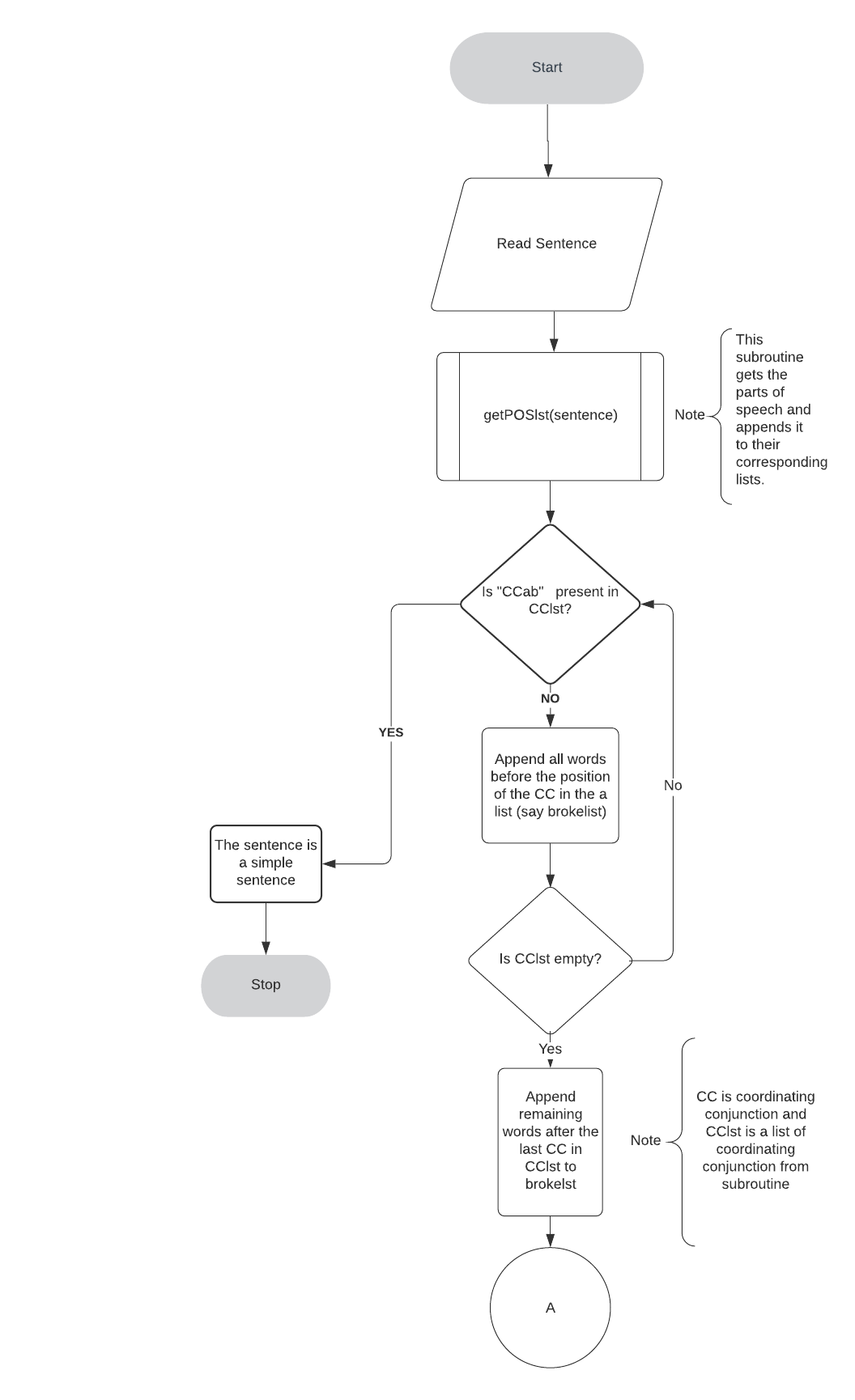
OR

S1 → < subject 1 > + < verb1 > + < object1>

&

S2 → <subject1> + <verb1>+< object2 >

To extend these sentences to work for more than one conjunction, we need only do the same and feed the split sentences back into the algorithm until the algorithm provides us with independent clauses. Once we detect that a clause is independent, we say that the algorithm has completed and the sentence cannot be decomposed further.

FLOWCHART

