

TBTA Tutorial

Lesson 9: Transfer Rules Continued

1.0 Introduction

In the previous lesson you were introduced to the transfer rules. You began by modifying one of the prebuilt case frame adjustment rules. Then you wrote several rather simple rules that collapsed noun Number and verb Time. You also wrote a rule that generated Grammatical Relations from Semantic Roles. In this lesson we're going to continue studying transfer rules. They're capable of doing far more than was seen in the previous lesson. In this lesson we're going to develop several transfer rules that radically restructure the semantic representations. Then we'll look at some of the more advanced capabilities of transfer rules.

2.0 Transfer Rules Continued

2.1 Transfer Rules – Clause Restructuring

Let's begin by writing a transfer rule that will significantly alter the structure of a proposition. Go to English Grammar Introduction, Clauses 1:55. This proposition yields *John prevented Mary from reading a book*. PREVENT is a complex concept and your target language may not have a good lexical match. Suppose the best equivalent in your target language is *Because of John, Mary did not read a book*. Let's modify the case frame adjustment rule for PREVENT so that it will perform these changes. Begin by resting the cursor on PREVENT in the large window, do a right click, and then select the menu option 'Edit this Verb's Case Frame Adjustment Rule'. You'll then see the case frame adjustment rule for PREVENT. Before we do any work on this rule, let's consider for a moment the tasks we want this rule to accomplish. We want this rule to change something like *John prevented Mary from reading a book* to *Mary did not read a book because of John*. We can represent *John prevented Mary from reading a book* as "X PREVENT [Y V]" where X represents *John*, Y represents *Mary*, and V represents whatever verb is in the object complement of PREVENT, in this case *read*. The object complement clause is represented by the brackets '[' and ']'. We could enter an optional noun phrase Z into the object complement to represent *book*, but our rule won't affect that NP so let's not include it. Let's represent the desired output of this rule as "Y not V because of X". So we want this rule to change "X PREVENT [Y V]" to "Y not V because of X". Comparing the input representation with the desired output representation, we see that this rule must perform four tasks: 1) delete the VP that contains PREVENT, 2) delete the complement clause's boundaries so that Y V is no longer embedded in a subordinate clause but instead is part of the main clause, 3) change the noun phrase X from having a Semantic Role of Participant to an Oblique that contains the words "because of", and 4) set the Polarity of the verb V which was in the

object complement to Negative in order to generate *not*. Since we need to change the polarity of the verb in the complement clause, let's begin by inserting a VP and a verb into the complement clause in the input structure of the transfer rule. Do this in the normal way. You'll see this VP and the verb appear in the output structure as well; you can always modify the input structure and see those modifications appear in the output structure. The input structure now has everything we need, so we're ready to start modifying the output structure.

In the output structure let's first delete the VP containing PREVENT. Make sure that no buttons are depressed and then click on the opening VP boundary in the output structure. You'll see it turn blue; then push the Delete key on your keyboard. You'll see that the phrase and everything in it is then tagged with "Delete." We also want to delete the object complement's clause boundaries so click on the opening subordinate clause boundary and push the Delete key. You'll see that the clause boundaries and everything in the subordinate clause become tagged with "Delete." In this case we don't want everything in the clause deleted; we only want the subordinate clause boundaries deleted. So click on the VP's opening boundary in the object complement clause and push the Delete key again. You'll see that this removes the Delete tag for the entire VP. The program now knows that you only want to delete the object complement's boundaries rather than the entire object complement.¹ Now change the Polarity of the verb that was in the object complement to Negative. Then change the Semantic Role of the first NP from Participant to Not Applicable and set its Grammatical Relation to Oblique. We now need to insert the target word *because* into that first NP. Click on the button labeled Insert Word and then click on the closing NP boundary. Choose the Target Words option in the upper left corner and then select Adpositions from the dropdown. You probably haven't yet entered the word *because* into your target lexicon, so click on the button labeled New Word and enter *because*. Click on the OK button, and you'll see the word *because* inserted into that first NP. Your complete rule should now look like the one shown below in figure 1.

¹ You may wonder why tagging the clause boundaries with Delete causes TBTA to delete the entire clause rather than just the clause boundaries. The same principle applies to phrases: if you tag an opening phrase boundary with Delete, the entire phrase will be deleted rather than just the phrase boundaries. You may think that if the phrase or clause boundaries are tagged with Delete, then only the phrase or clause boundaries should be deleted and everything else in the phrase or clause should be left alone. The problem with that approach is that if you then wanted an entire phrase or clause to be deleted, you would have to specify every possible constituent that might be included in that phrase or clause. Since that's impractical, you just need to remember the following guideline: If you want to delete an entire phrase or clause, just tag the opening phrase or clause boundary with Delete. If you want to delete just the phrase or clause boundaries and leave everything else alone, then put some constituent into the phrase or clause and make sure that it doesn't have the Delete tag. Then TBTA will know that you only want to delete the phrase or clause boundaries.

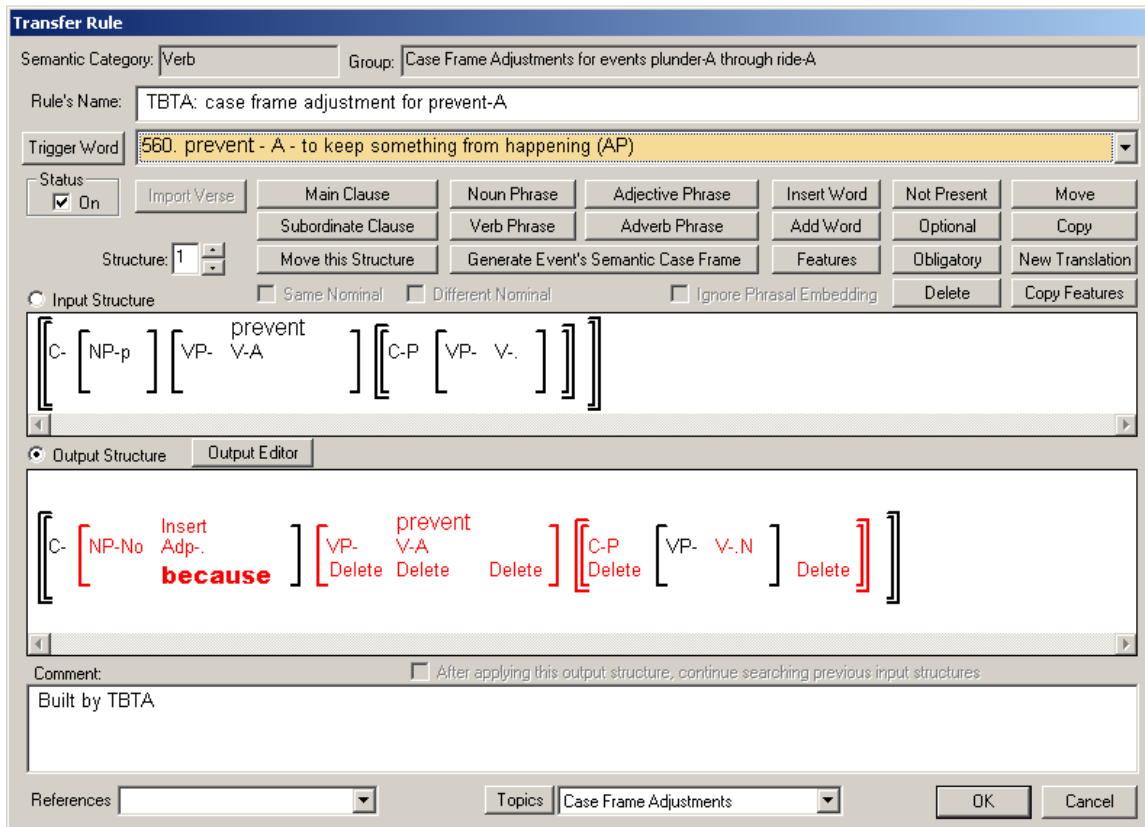


Figure 1. Transfer Rule for PREVENT

You'll see that all of the constituents affected by this rule are displayed in red in the output structure, and the constituents not affected by this rule are still in black. Save the rule and map any unmapped concepts in Clauses 1:55 to target words except for PREVENT. Since we're working in a language that doesn't have a lexical equivalent for PREVENT, map it to a space (' ') just to get rid of the question marks '???' below PREVENT. When you write your real grammar, you'll probably want to enter an empty space as a target word into each syntactic category and then link all of the concepts that don't have target equivalents to that space. Generate the verse and make any necessary adjustments in your descriptive grammar to produce the desired results. You should end up with something like *Mary not read a book because John*. That's close enough for now.

2.2 Transfer Rules – Copying Constituents

Now let's write a transfer rule that will illustrate the ability to copy constituents. In English, aspect is often indicated by the addition of another verb. For example, one way of indicating inceptive aspect is to add the verb *start* as in *John started walking*. Completive aspect may be signaled by adding the verb *finish* as in *John finished walking*. So let's write a transfer rule that will add these aspectual verbs *start* and *finish*. Begin by entering a new group into the transfer rules called **Aspect**. Then double click on the group name and select Verbs for the rule's syntactic category. Build the input structure so that it contains a verb in

a VP, and set the verb's aspect to Inceptive. Then switch to the output structure and click on the button labeled Copy. Click on the verb in the output structure, and then click on the closing VP boundary. You'll see another verb inserted immediately before the closing VP boundary with the word "Copy" directly above it. This copy of the original verb will have all of the features of the original verb and it will be mapped to the same target word as the original verb. We want this new verb to be mapped to the target word *start*, so click on the button labeled New Translation, and then click on the copied verb. Enter the new verb *start* into the dialog, and select it as the translation for this copied verb. Let's test the rule as it is now, so enter a name such as **Inceptive and Completive Aspects**, assign it to the grammar topic Aspect, and then save the rule. Your finished rule should look like the one shown below in figure 2.

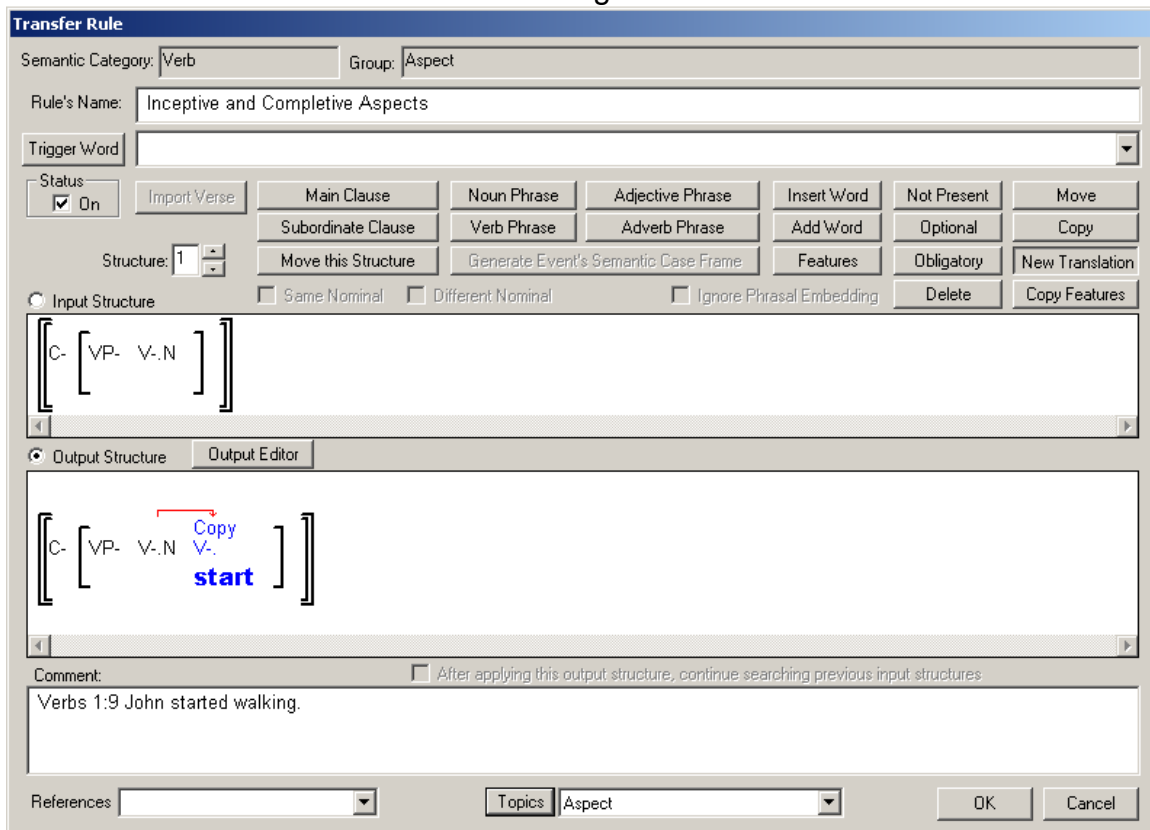


Figure 2. Transfer Rule for Inceptive Aspect

Go to Verbs 1:8 and map any unmapped concepts to target words. Make sure that your descriptive grammar has whatever rules are necessary to generate the past tense forms for your verbs. Then push the Generate button to test this new transfer rule. You should see that the VP now contains the verbs *started* and *walked*. Linguists often distinguish these two verbs from one another by calling *walk* the semantic verb and *start* the grammatical verb. We want the grammatical verb to be marked for tense, and we want the semantic verb to be a participle. Let's add a new feature for verbs called Type, and it will have two values: Grammatical and Semantic. But note that this feature doesn't depend on the target words in way; this feature depends on the context. Therefore this new

feature won't go into the lexicon. Instead we'll add this feature to the contextual features in the semantic representations just like we added the feature Grammatical Relation for noun phrases. To add this feature to verbs, open the transfer rule you just wrote for inceptive aspect. Click the Features button, and click on one of the verbs in the output structure. When you see the features dialog, click on the Feature Set button at the bottom. You'll see the same dialog you saw before when you added Grammatical Relation to NPs. Select Verbs in the dropdown on the left, and in the dropdown on the right select Spare 10². Change this feature's name to Type and enter two values: Grammatical and Semantic. Close the Feature Set dialog and you'll see that the features dialog now has a new column called Type with your two new values. In the output structure set the Type feature on the original verb to Semantic and set the Type feature on the copied verb to Grammatical. The output structure should now look like that shown in figure 3.

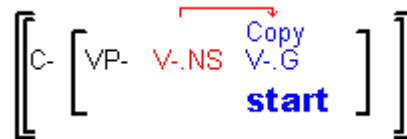


Figure 3. Output Structure for Semantic and Grammatical Verbs

Then make whatever changes are necessary in your descriptive grammar so that the verb tagged as Semantic will use the Gerund form from the lexicon. Also make sure that the past tense form is selected only if the verb's Type feature is Grammatical. In your VP phrase structure rule, insert another verb and specify that the first verb must have a Type value of Grammatical, and the second verb must have a Type value of Semantic. You should then be able to generate *John started walking*.

Now go back to your new transfer rule and use the second structure to handle completive aspect. Verify that it works properly with Verbs 1:7 which produces *John finished walking*.

2.3 Transfer Rules – Moving Constituents

Transfer rules are also able to move constituents. Moving a constituent is very similar to copying a constituent. For example, go to Verbs 1:27. In the semantic representations Patient NPs have not been placed inside their VPs because OSV and VSO languages require their object NPs to be outside the VP. If you disagree with this analysis and want the Patient NPs to be inside the VPs, you can write a transfer rule that will move them. Add a new group to the transfer rules called **Constituent Movements**. Rather than building the input structure for this rule in the usual way, we're going to use a shortcut. Do a right click somewhere in the semantic representation window and choose the menu

² When you add a new feature to a syntactic category such as we're doing here, you should always start with the last spare feature. This is because the spare features have two purposes: 1) so that you can add features relevant to your target language, and 2) so that TBTA's feature system can be expanded in the future. TBTA's expansion will start with Spare 1 and move toward the end. Therefore you should start at the end and work your way toward the beginning.

option 'Build Input Structure for Transfer Rule'. When you select this option, you're able to start highlighting the constituents in the semantic representation that you want to copy and paste into a rule's input structure. If you click on the features under READ, you'll see that the verb's features and its VP are highlighted in purple. If we had wanted to actually include the event READ in this rule, we would have clicked on READ. But since we want this rule to apply to all events, we won't include READ. Now click on the features under BOOK. You'll again see the noun's features and the NP highlighted in purple. The order in which you select these constituents is never important; you may select them in any order. If you mistakenly select a constituent, just click on it again and it will be deselected. Your semantic representation for Verbs 1:27 should now look like that shown in figure 4.

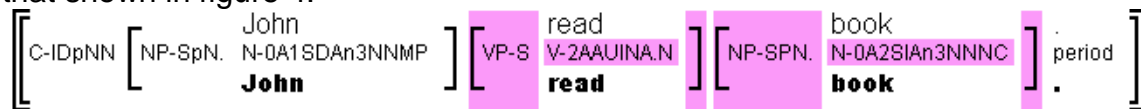


Figure 4. Copying Constituents from Semantic Representation into Transfer Rule

After you've highlighted the relevant constituents, double click on the new group name in the transfer rules. You'll then see a new rule, and the selected constituents will have been pasted into the input and output structures. Your input and output structures should look like figure 5.

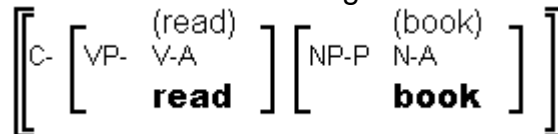


Figure 5. Input and Output Structures after Pasting Constituents

Notice in the input and output structures that READ and BOOK are in parentheses. This indicates that they are just examples; the rule won't require READ and BOOK to be present when executing this rule. You should also notice that when you copy and paste constituents from the semantic representation into a transfer rule, no features will be copied into the rule except for the Semantic Roles of the NPs. If any features other than the semantic roles are relevant, you'll need to specify them yourself. We're now ready to move the Patient NP into the VP.

Click on the button labeled Move. You'll see that it remains depressed indicating that you're ready to move constituents. Click on the opening NP boundary in the output structure. Then click on the opening VP boundary in the output structure. You'll see a red arrow going from the NP to the VP. When you move a constituent, you first click on the constituent that you want to move. You then click on the opening phrase or clause boundary where you want the constituent to be inserted. Notice that the destination of the move must always be an opening phrase or clause boundary. Don't try to move a constituent to the end of a phrase or to the middle of a phrase. It will be the phrase structure rules that ultimately determine where this constituent will be placed within the phrase or clause. The transfer rules just move the constituents into the desired phrases

and clauses. Give this rule a name and assign it to one or more grammar topics. Your rule should now look something like that shown in figure 6.

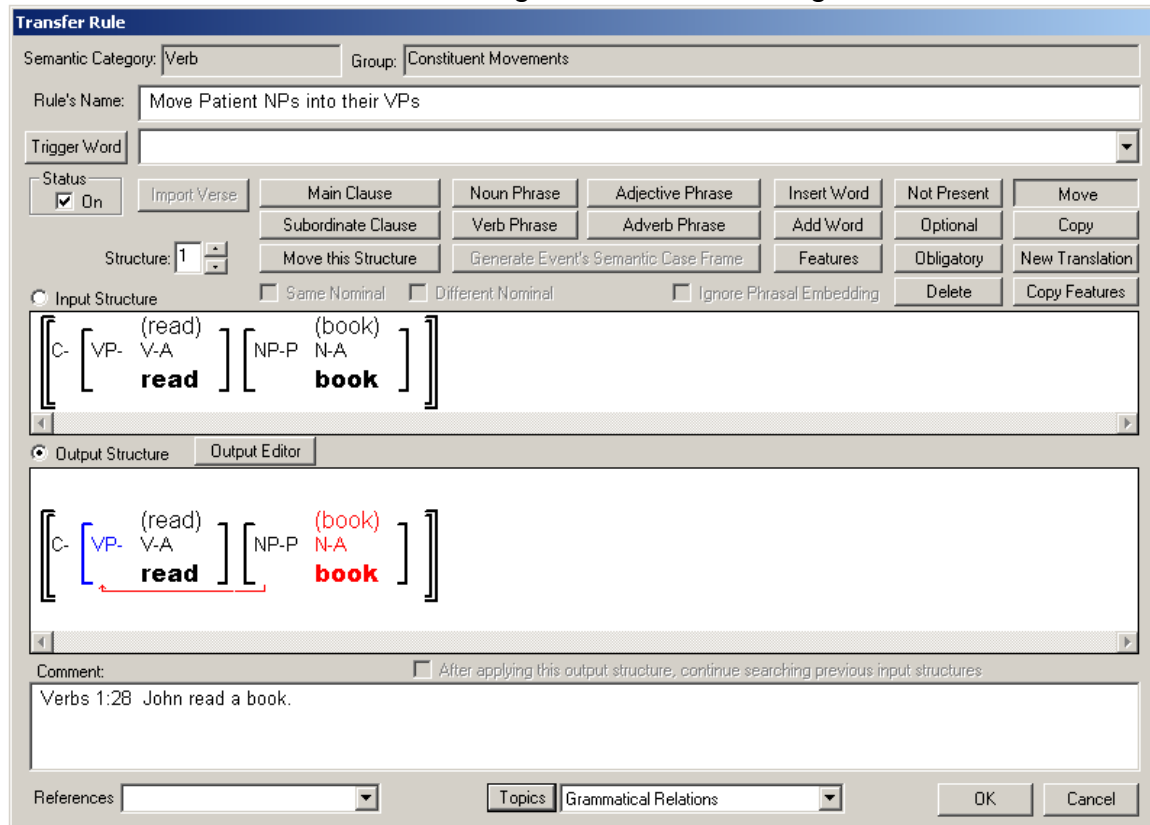


Figure 6. Transfer Rule that Moves Patient NPs into their VPs

Confirm that this rule moves the Patient NP into the VP by generating Verbs 1:27. Notice that now it is the VP phrase structure rule rather than the clause phrase structure rule that is responsible for the position of the Patient NP. After you've specified a move in a transfer rule, if you ever want to undo that move, all you need to do is click on the Move button, click on the constituent that is being moved, and then click on that same constituent again. Essentially this is moving a constituent to itself. After you've done this, you'll see that the arrow representing the old move will be erased.

Go to Clauses 1:27 and confirm that the Patient NP is moved into the VP. Just for the sake of illustration, let's suppose that we only want to move 'light' Patient NPs into their VPs. In other words, if a Patient NP doesn't contain a relative clause, then we want to move it into the VP; if a Patient NP does contain a relative clause, then it will be considered 'heavy' and we won't move it into the VP. The Patient NP in Clauses 1:27 contains a relative clause. So let's modify our rule so that it won't move Patient NPs with relative clauses into their VPs. Open your rule and insert a subordinate clause into the patient NP in the input structure. You'll see the subordinate clause inserted into the output structure simultaneously. Now click on the Not Present button and then click on the opening subordinate clause boundary in the input structure. You'll see that the subordinate clause will be covered with a dark gray crosshatch. This gray

crosshatch indicates that this constituent must not be present in order for this rule to apply. Save your rule and generate Clauses 1:27 again. This time the Patient NP will not be moved into the VP because it is considered heavy. Go back to Verbs 1:27 and confirm that the Patient NP in that clause is still moved into the VP.

As was mentioned in the previous lesson, there are many possible combinations of Optional and Not Present. You may apply these options to particular concepts, words, phrases and clauses. For a complete description of all the possible combinations, you should read the help text. To read the help text for the transfer rules, open any transfer rule, click on the help button in the main toolbar (arrow with question mark), and then click anywhere on the rule's dialog. You'll see a complete discussion of the capabilities of that type of rule.

2.4 Transfer Rules – Same Nominal / Different Nominal

Another very important function of the transfer rules is to identify Same Nominals and Different Nominals. Using the Same Nominal and Different Nominal options is useful when working with relative clauses, object complements, adverbial clauses, clause chains, etc. In the first lesson you saw that one of the features associated with every noun is a Noun List Index. For example, go to Clauses 1:25 and rest the cursor on the features under the two occurrences of MAN. You'll see that they both have a Noun List Index value of 2 indicating that both instances of MAN are referring to the same entity. This verse generates *The man that saw John saw Mary*. In every relative clause there will be a noun with the same Noun List Index as the head noun. TBTA is able to identify which noun in the relative clause is coreferential with the head noun by examining the Noun List Index. English requires a rule that deletes the noun in a relative clause that is coreferential with the head noun. So we need a method to identify the nouns in relative clauses that are coreferential with the head nouns. Then the spellout rules will handle those nouns appropriately. Let's write a transfer rule that will tag the NP in the relative clause that contains the noun that is coreferential with the head noun. Begin by entering a new group into your transfer rules called **Relative Clauses**. Enter a new rule into that group by double clicking on the group name; select Nouns as the syntactic category. Immediately above the input structure you'll see one check box labeled Same Nominal and another check box labeled Different Nominal. Check the box labeled Same Nominal. You'll immediately see two NPs inserted into the input structure, each containing a noun. TBTA knows that you want these two nouns to have the same Noun List Index. You're now free to modify the input structure in any way that you want. In order to form a relative clause, the second NP must be in a subordinate clause that is embedded in the first NP. Let's begin by moving the closing phrase boundary of the first NP so that it follows the second NP. Click on the Move³ button and then click on the first NP's closing phrase

³ Notice that the Move button has a different purpose when you're working on the input structure. When working on the input structure, the Move button lets you move individual constituents to different locations within the input structure. As you saw earlier in this lesson, when you're working on the output structure, the Move button lets you specify that TBTA must move particular constituents during the execution of the rule.

boundary. You'll see the closing phrase boundary disappear. We want to reinsert this closing phrase boundary right before the closing clause boundary, so click on the closing clause boundary. You'll see the NP's boundary be reinserted immediately before the closing clause boundary. The second NP is now embedded in the first NP. Now insert a subordinate clause around the embedded NP in the regular way. Set the Type feature on this subordinate clause to Restrictive Thing Modifier (restrictive relative clause). Your input structure should now look like that shown in figure 7.

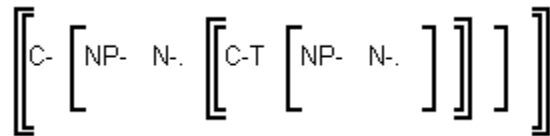


Figure 7. Input Structure for Relative Clause

There's no overt indication in this input structure that the two nouns in this rule have the same Noun List Index. However, since you checked the Same Nominal option, TBTA knows that the two nouns must have the same Noun List Index. If you had checked the Different Nominal option, you would have seen the exact same structure, but TBTA would know that those two nouns must have different Noun List Index values.

Now go to the output structure. We want this rule to set the coreferential NP's Type value to a new value called Relative. Click on the Features button and click on the NP in the relative clause. Right now there isn't a value called Relative under Type, so click on the button labeled Feature Set and add this value to the list of Type values for Noun Phrases. You'll need to select Noun Phrase as the syntactic category, and then add a new value called Relative to the list of values under Type. After you've done this, select Relative in the Type column for the NP. Your final output structure should look like that shown in figure 8.

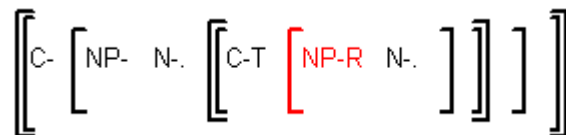


Figure 8. Output Structure for Relative Clauses

Enter a name for this rule such as **Identify Relativized NPs**. Save your rule and test it with Clauses 1:25 – 36. In each verse you should see that the NP in the relative clause that contains the noun that is coreferential with the head noun has a Type value of Relativized. Later your spellout rules will deal with nouns in Relativized NPs appropriately for your target language. Note: to see this rule work properly in Clauses 1:26, you'll need to turn off the transfer rule that moves object NPs into their VPs.

2.5 Transfer Rules – Collocation Correction

Collocation correction is another common task performed by transfer rules. For example, the semantic representations include two senses of the event

OPEN. OPEN-A refers to opening something such as a container. OPEN-B refers to opening something such as a door or gate. However, OPEN-B is also used in the semantic representations when a person opens his eyes. Suppose your language has a good lexical match for OPEN-B whenever the object being opened is a door, window, gate, etc. But suppose your language requires a different verb when someone opens his eyes. Let's write a transfer rule that will remap OPEN-B from the first target verb to the second target verb whenever the object being opened is an eye. Begin by going to Infected Eye 1:6 which is in English Documents. That verse contains an instance of OPEN-B and the patient NP contains the noun EYE. Map OPEN-B to the English word *open*. Then enter a new group for transfer rules called **Collocation Correction**. We could modify the case frame adjustment rule for OPEN-B, but instead we'll write our own rule just to practice copying and pasting from the semantic representation. Do a right click on any word in the large window and select the menu option 'Build Input Structure for Transfer Rule'. You're now ready to begin highlighting constituents in the current verse that you want to copy and paste into the input structure of a transfer rule. We want this rule to be triggered by OPEN-B so click on it first; be sure to click on OPEN, not on the features. You'll see that OPEN, the features and the verb phrase are now highlighted in purple. Then click on EYE and you'll see that EYE, its features and its phrase are highlighted in purple. These two phrases are all that's relevant to this rule, so now double click on your group called Collocation Correction in the grammar tree. You'll see that the items you highlighted in the semantic representation have now been copied and pasted into the input and output structures of your new rule. Right now both structures should look like figure 9.

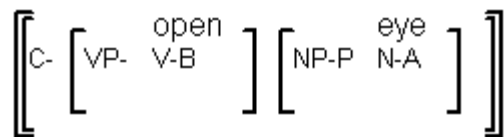


Figure 9. Input and Output Structures for Collocation Correction Rule

Notice this time that OPEN and EYE are not in parentheses. This means that the rule will require both OPEN-B and EYE-A to be present before this rule will apply.

Now go to the output structure. Click on the button labeled New Translation and then click on the verb in the output structure. Enter and select another target verb such as *raise*. You'll then see that target verb displayed below OPEN in the output structure. Your final output structure should look like that shown in figure 10.

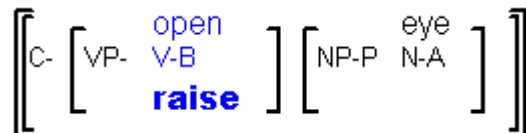


Figure 10. Output Structure for Collocation Correction Rule

Test this rule at Infected Eye 1:6. You should see that the rule changes *open* to *raise* whenever the patient is *eye*. You'll undoubtedly need many collocation correction rules for your target language.

2.6 Transfer Rules – Combining Clauses

Transfer rules may also be used to combine clauses. If you have two simple clauses in a sequence and they both have the same Participant noun, you may want to combine them into a single clause. For example, look at Ruth 1:7. This verse produces *Then these three women left Moab. The women started traveling to Judah.* If you rest the cursor under the two occurrences of WOMAN, you'll see that they both have the same Noun List Indexes. Since neither of these clauses contain an adverbial clause, an object complement, or a subject complement, let's write a transfer rule that will combine these two simple clauses into a single clause. Add a new group to your transfer rules called **Clause Combining**. Then add a new rule and choose Nouns as the syntactic category; whenever you use the Same Nominal or Different Nominal option, you should always select Noun as the syntactic category. Select the Same Nominal option and you'll see the same structure inserted as you did before. Then click on the Main Clause button. You'll see another main clause appended at the end of the input structure. Now move the second NP from the first clause to the second clause using the Move button. Then tag both NPs as Participants. Insert a subordinate clause into each main clause, and tag the subordinate clauses as Event Modifier, Participant and Patient. Then tag both subordinate clauses as Not Present. Your input structure should now look like that shown in figure 11.

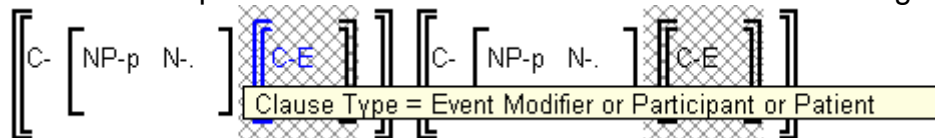


Figure 11. Input Structure for Combining Clauses

So this input structure specifies that we're looking for a main clause that doesn't have a subordinate clause, and it is immediately followed by another main clause that doesn't have a subordinate clause. Also, both of the main clauses have the same participant noun. We're now ready to write the output structure so click on the Output Structure option.

Click on the Move button and then click on the second clause's opening boundary in the output structure. Then click on the first clause's opening boundary. You'll see a red arrow indicating that the second clause will be moved into the first clause. Then click on the Features button and set the second clause's Type to Coordinate Independent. Then click on the Insert Word button and insert the source conjunction AND at the beginning of the second main clause. Your output structure should now look like that shown in figure 12.

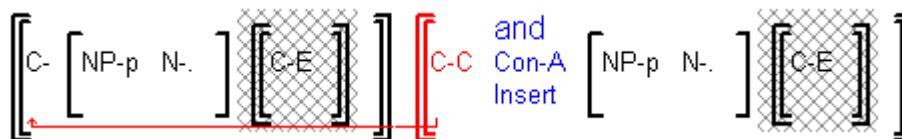


Figure 12. Output Structure for Combining Clauses

Modify your clause phrase structure rule so that conjunctions will be placed at the beginning of clauses and Coordinate Independent clauses will be placed at the end of the clause but before periods. Then test your transfer rule at Ruth 1:7. You don't need to map any concepts to target words; just verify that your transfer rule is moving the second clause into the first clause and making it a subordinate Coordinate Independent clause. Use Grammar Wizard to write another Find/Replace rule that will change a period followed by another period to a single period.

Now let's make a couple of improvements to this rule. The rule currently inserts the conjunction AND into the second clause even if there is already a conjunction present. This would result in two conjunctions and we certainly don't want that. So let's revise this rule so that it will only insert AND if there isn't already a conjunction. The second improvement we'll make is to delete the participant noun in the second clause. With these improvements the verse will generate *Then these three women left Moab and started traveling to Judah*. Open your rule and delete AND from the output structure. Then insert AND into the input structure. You'll see it appear in the output structure also. Then click Add Word, click on AND in the input structure, and select BUT. You won't see BUT appear in the input or output structures, but if you rest the cursor on AND, you'll see a popup indicating that both AND and BUT are possible options in that position. You can always add as many source concepts as you want to any position. Now click on the button labeled Optional and click on AND in the input structure. You'll see AND be covered with a light purple crosshatch indicating that the conjunction is optional, but if it is present, it must be either AND or BUT. Now click on the button labeled Obligatory and click on AND in the output structure. You'll see the crosshatch change to light green in the output structure. The green crosshatch indicates that if a conjunction isn't present in the semantic representation, this rule will insert the conjunction AND. So we've now finished the first improvement.

The second improvement will be much simpler. Click on the Obligatory button again so that it is no longer depressed. Now there shouldn't be any buttons depressed. In the output structure click on the noun in the second clause and push the Delete key on your keyboard. You'll see the noun be tagged with Delete. Now push the Delete key a second time. You'll see the label change to Delete Target Word. This means that the noun and its features will remain in the underlying representation, but the target word will be deleted. This is the equivalent of PRO (big pro) - syntactically active but phonologically null. There will be times when you'll want to keep a word with its features in the underlying representation, perhaps for something like verbal subject agreement, but you won't want the actual target word in the text. You can accomplish this by pushing the Delete key twice. Now push the Delete key again and you'll see the Delete Target Word label disappear and the noun will be back to its original state. You may then start the process over again by pushing the Delete key as many times as you like. Continue pressing the Delete key until you see Delete Target Word under the noun. Your final rule should appear as shown below in figure 13.

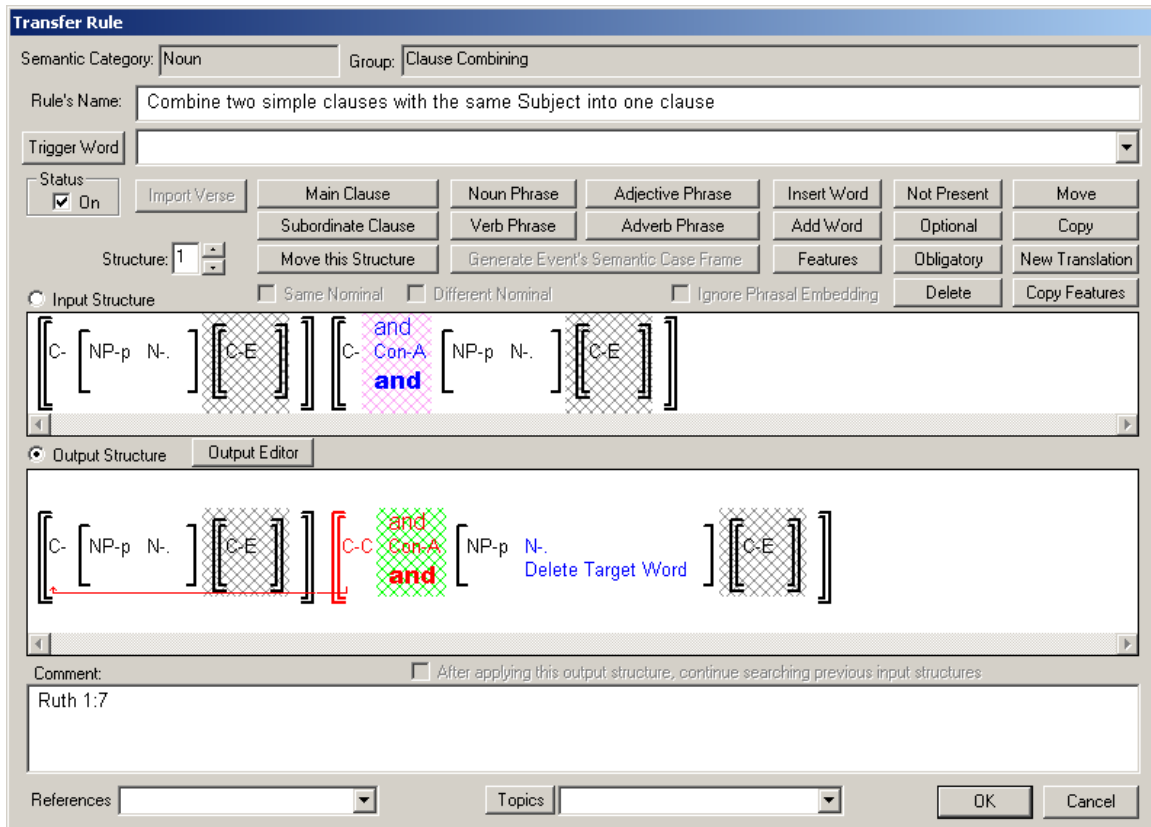


Figure 13. Revised Transfer Rule for Combining two Propositions

Save your rule and test it at Ruth 1:7 again. Confirm that the participant noun in the newly embedded clause is still present, but the target word under WOMAN has been eliminated. When you're writing a real grammar, you'll probably want to write transfer rules that will combine propositions under certain circumstances because the propositions in the semantic representations are generally very short and simple. The text generated by your grammar will be much more readable if you write rules that combine clauses whenever possible.

3.0 Conclusion

This tutorial has introduced you to several of the more advanced capabilities of the transfer rules. You modified the case frame adjustment rule for PREVENT so that the original proposition was significantly altered. You've also written several transfer rules that copied and moved constituents. You were also introduced to the Same Nominal and Different Nominal options. By combining all of these capabilities, you'll be able to restructure the semantic representations into a new underlying representation that is appropriate for your target language.

4.0 Exercises

Exercise 1: Modify whatever rules necessary to properly generate Verbs 1:9 which illustrates cessative aspect. Verbs 1:9 should generate *John stopped*

walking. Begin by modifying your transfer rule that handles inceptive and completive aspect. Use the third structure of that rule to handle cessative aspect. Note that you'll also need to set the Type value for all verbs initially to Grammatical. Use the fourth structure to accomplish this. But in the fourth structure be sure to check the box labeled 'After applying this output structure, continue searching previous input structures.'

Exercise 2: Clauses 1:24 has a descriptive relative clause rather than a restrictive relative clause. You can easily modify your transfer rule for relative clauses so that it also applies to Clauses 1:24. Just add the feature value Descriptive Thing Modifier to the subordinate clause's Type feature in the input structure. Verify that your modified transfer rule works correctly in Clauses 1:24.

Exercise 3: Pronouns are best handled in TBTA by using spellout rules for nouns that are triggered whenever a noun's Surface Realization⁴ feature is Unambiguous Pronoun. However, the semantic representations only have the Unambiguous Pronoun value selected for third person pronouns. Nouns that are tagged as First Person, Second Person, etc., have the Surface Realization feature set to Not Applicable. The reason for this is that when a noun's Person feature is set to anything other than Third, that noun should always be realized by a pronoun, so it would be redundant to also tag that noun as Unambiguous Pronoun. Therefore, if you want your spellout rules that generate your pronouns to be triggered whenever the Surface Realization feature is Unambiguous Pronoun, you must set the Surface Realization to Unambiguous Pronoun whenever the noun Person is any value other than Third. So write a transfer rule for Nouns that will look at the Person feature. If the Person is any value other than Third, the output structure should set the Surface Realization feature to Unambiguous Pronoun. Test your rule with Pronouns 1:1-2.

Exercise 4: Sometimes you'll want a single transfer rule to be triggered by multiple concepts. Earlier in this lesson you wrote a transfer rule for the concept PREVENT. That transfer rule changed the case frame for PREVENT so that Clauses 1:55 produced *Mary did not read the book because of John*. Suppose you want that same rule to be triggered by the concept FORBID. You may want to convert a sentence such as *John forbid Mary from reading the book* to *Mary did not read the book because of John*,. You would generally do this by modifying the case frame adjustment rule for FORBID. But to illustrate a particular feature, let's add the concept FORBID to the rule that you wrote for PREVENT. Open that rule and push the Add Word button. Click on PREVENT

⁴ The Surface Realization feature on nouns has three possible values: 1) Always a Noun, 2) Unambiguous Pronoun, and 3) Not Applicable. Some nouns in the semantic representations are tagged as Always a Noun because we never want a pronoun used there. For example, John 1:1 "In the beginning was the Word, and the Word was with God, and the Word was God." We want all three occurrences of "Word" to be represented with nouns rather than pronouns so those three nouns are tagged as Always a Noun. Occasionally in the semantic representations you'll see a noun tagged with Unambiguous Pronoun. This value is used only when two occurrences of the same noun occur in close proximity and there are no other relevant nouns in the context. For example, "John walked to the store. John bought some bread." The second occurrence of John would be tagged as Unambiguous Pronoun so that you can realize it with a third person pronoun, if your language prefers that. Nouns that have a Person value of anything other than Third Person should always be realized with a pronoun. Those nouns haven't been tagged as Unambiguous Pronoun because that would be redundant.

in the input structure, and then select the word FORBID. Now when you rest the cursor on PREVENT in either the input structure or the output structure, the popup will show you that both PREVENT and FORBID are included in this rule. Test this modified rule in Ruth 2:15. The last sentence includes the verb FORBID, so you should see that its case frame is modified according to your rule. (Note: The case frame for FORBID actually includes a Destination NP which PREVENT doesn't have. Therefore, when you're writing a real grammar, you probably would not want the same transfer rule to handle the case frame adjustments for both PREVENT and FORBID.)