Chapter 11: Automated Proof Systems (2)

RS: DECOMPOSITION TREES

The process of searching for the proof of a formula A in **RS** consists of building a certain tree, called a **decomposition tree** whose root is the formula A, nodes correspond to sequences which are conclusions of certain rules (and those rules are well defined at each step by the way the node is built), and leafs are axioms or are sequences of a non- axiom literals.

We prove that each formula A (sequence Γ) generates its *unique and finite* decomposition tree, $\mathbf{T}_A$ ($\mathbf{T}_\Gamma$).

The tree constitutes the proof of A (Γ) in **RS** if all its leafs are axioms.

If there is a leaf of $\mathbf{T}_A$ ($\mathbf{T}_\Gamma$) that *is not an axiom*, the tree is not a proof, moreover, the proof of A **does not exist**.

Before we give a proper definition of the proof search procedure by building a decomposition tree we list few important observations about the structure of the rules of the system **RS**.

Introduction of Connectives

The rules of **RS** are defined in such a way that each of them *introduces* a new logical connective, or a negation of a connective to a sequence in its domain (rules $(\cup), (\Rightarrow), (\cap)$) or a negation of a new logical connective (rules $(\neg\cup), (\neg\cap), (\neg \Rightarrow), (\neg\neg)$).

The rule $(\cup)$ introduces a new connective $\cup$ to a sequence $\Gamma^{'}, A, B, \Delta$ and it becomes, after the application of the rule, a sequence $\Gamma^{'}, (A \cup B), \Delta$.

Hence a name for this rule is $(\cup)$.

The rule $(\neg\cup)$ introduces a negation of a connective, $\neg\cup$ by combining sequences $\Gamma^{'}, \neg A, \Delta$ and $\Gamma^{'}, \neg B, \Delta$ into one sequence (conclusion of the rule) $\Gamma^{'}, \neg(A \cup B), \Delta$.

Hence a name for this rule is $(\neg\cup)$.

The same applies to all remaining rules of **RS**, hence their names say which connective, or the negation of which connective has been introduced by the particular rule.

Decomposition Rules

Building decomposition tree (a proof search tree) consists of using the inference rules in an inverse order; we transform them into rules that transform a conclusion into its premisses.

We call such rules the **decomposition rules**.

Here are all of **RS** decomposition rules.

Disjunction decomposition rules

$$(\cup)\ \frac{\Gamma^{'}, (A \cup B), \Delta}{\Gamma', A, B, \Delta}, \qquad (\neg\cup)\ \frac{\Gamma^{'}, \neg(A \cup B), \Delta}{\Gamma', \neg A, \Delta \ : \ \Gamma', \neg B, \Delta}$$

Conjunction decomposition rules

$$(\cap)\ \frac{\Gamma^{'}, (A \cap B), \Delta}{\Gamma', A, \Delta \ ; \ \Gamma', B, \Delta}, \qquad (\neg\cap)\ \frac{\Gamma^{'}, \neg(A \cap B), \Delta}{\Gamma', \neg A, \neg B, \Delta}$$

Implication decomposition rules

$$(\Rightarrow)\ \frac{\Gamma^{'}, (A \Rightarrow B), \Delta}{\Gamma', \neg A, B, \Delta}, \qquad (\neg \Rightarrow)\ \frac{\Gamma^{'}, \neg(A \Rightarrow B), \Delta}{\Gamma', A, \Delta \ : \ \Gamma', \neg B, \Delta}$$

Negation decomposition rule

$$(\neg\neg)\ \frac{\Gamma^{'}, \neg\neg A, \Delta}{\Gamma', A, \Delta}$$

where $\Gamma^{'} \in \mathcal{F}'^{*}, \Delta \in \mathcal{F}^{*}, A, B \in \mathcal{F}$.

We write the decomposition rules in **a visual tree form** as follows.

Tree Decomposition Rules

(∪) **rule:**

$$\Gamma^{'}, (A \cup B), \Delta$$

$$\mid (\cup)$$

$$\Gamma^{'}, A, B, \Delta$$

$(\neg\cup)$ rule:

$$\Gamma', \neg(A \cup B), \Delta$$

$$\bigwedge(\neg\cup)$$

$$\Gamma', \neg A, \Delta \quad \Gamma', \neg B, \Delta$$

$(\cap)$ rule:

$$\Gamma', (A \cap B), \Delta$$

$$\bigwedge(\cap)$$

$$\Gamma', A, \Delta \qquad \Gamma', B, \Delta$$

$(\neg\cup)$ **rule:**

$$\Gamma^{'}, \neg(A \cap B), \Delta$$

$$\mid (\neg\cap)$$

$$\Gamma^{'}, \neg A, \neg B, \Delta$$

$(\Rightarrow)$ **rule:**

$$\Gamma^{'}, (A \Rightarrow B), \Delta$$

$$\mid (\cup)$$

$$\Gamma^{'}, \neg A, B, \Delta$$

$(\neg \Rightarrow)$ **rule:**

$$\Gamma^{'}, \neg(A \Rightarrow B), \Delta$$

$$\bigwedge (\neg \Rightarrow)$$

$$\Gamma^{'}, A, \Delta \qquad \Gamma^{'}, \neg B, \Delta$$

$(\neg\neg)$ **rule:**

$$\Gamma^{'}, \neg\neg A, \Delta$$

$$\mid (\neg\neg)$$

$$\Gamma^{'}, A, \Delta$$

Observe that we use the same names for the inference and decomposition rules, as once the we have built the decomposition tree (with use of the decomposition rules) with all leaves being axioms, it constitutes a proof of A in **RS** with branches labeled by the proper inference rules.

Now we still need to introduce few useful definitions and observations.

Indecomposable Sequence

A sequence $\Gamma^{'}$ built only out of literals, i.e. $\Gamma \in \mathcal{F}'^{*}$ is called **an indecomposable sequence**.

Decomposable Formula

A formula that is not a literal is called **a decomposable formula**.

Decomposable Sequence

A sequence Γ that contains a decomposable formula is called **a decomposable sequence**.

Observation 1

For any **decomposable** sequence, i.e. for any $\Gamma \notin \mathcal{F}'^{*}$ there is **exactly one** decomposition rule that can be applied to it.

This rule is determined by the first decomposable formula in Γ, and by the main connective of that formula.

Observation 2

If the main connective of the first decomposable formula is $\cup, \cap$, or $\Rightarrow$, then the decomposition rule determined by it is $(\cup), (\cap)$, or $(\Rightarrow)$, respectively.

Observation 3

If the main connective of the first decomposable formula is $\neg$, then the decomposition rule determined by it is determined by the *second connective* of the formula. If the second connective is $\cup, \cap, \neg$, or $\Rightarrow$, then corresponding decomposition rule is $(\neg\cup), (\neg\cap), (\neg\neg)$ and $(\neg \Rightarrow)$.

Because of the importance of the above observations we write them in a form of the following

Unique Decomposition Lemma

For any sequence $\Gamma \in \mathcal{F}^*$,

$\Gamma \in \mathcal{F}'^*$ or Γ is in the domain of only one of the **RS** Decomposition Rules.

Decomposition Tree $\mathbf{T}_A$

For each $A \in \mathcal{F}$, a decomposition tree $\mathbf{T}_A$ is a tree build as follows.

Step 1. The formula A is the **root** of $\mathbf{T}_A$ and for any node Γ of the tree we follow the steps below.

Step 2. If Γ is **indecomposable,** then Γ becomes a **leaf** of the tree.

Step 3. If Γ is decomposable, then we traverse Γ from left to right to identify the first **decomposable formula** B and identify the unique (**Unique Decomposition Lemma**) decomposition rule determined by the main connective of B.

We put its left and right premisses as the left and right leaves, respectively.

Step 4. We repeat steps 2 and 3 until we obtain only leaves.

Decomposition Tree $\mathbf{T}_{\Gamma}$

For each $\Gamma \in \mathcal{F}^*$, a decomposition tree $\mathbf{T}_{\Gamma}$ is a tree build as follows.

Step 1. The sequence Γ is the **root** of $\mathbf{T}_{\Gamma}$ and for any node Δ of the tree we follow the steps bellow.

Step 2. If Δ in indecomposable, then Δ becomes a **leaf** of the tree.

Step 3. If Δ is decomposable, then we traverse Δ from left to right to identify the first **decomposable formula** B and identify the unique (**Unique Decomposition Lemma**) decomposition rule determined by the main connective of B.

We put its left and right premisses as the left and right leaves, respectively.

Step 4. We repeat steps 2 and 3 until we obtain only leaves.

Now we prove the following theorem.

Decomposition Tree Theorem

For any sequence $\Gamma \in \mathcal{F}^*$ the following conditions hold.

1. $\mathbf{T}_\Gamma$ is finite and unique.

2. $\mathbf{T}_\Gamma$ is a proof of Γ in **RS** if and only if all its leafs are axioms.

3. $\nvdash_{\mathbf{RS}}$ if and only if $\mathbf{T}_\Gamma$ has a non- axiom leaf.

Proof: The tree $\mathbf{T}_\Gamma$ is unique by the **Unique Decomposition Lemma**. It is finite because there is a finite number of logical connectives in Γ and all decomposition rules diminish the number of connectives. If the tree has a non- axiom leaf it is not a proof by definition. By the its uniqueness it also means that the proof does not exist.

Example

Let's construct, as an example a decomposition tree $\mathbf{T}_A$ of the following formula A.

$$A = ((a \cup b) \Rightarrow \neg a) \cup (\neg a \Rightarrow \neg c))$$

The formula A forms a one element decomposable sequence. The first decomposition rule used is determined by its main connective.

We put a box around it, to make it more visible.

$$((a \cup b) \Rightarrow \neg a)\boxed{\cup}(\neg a \Rightarrow \neg c))$$

The first and only rule applied is (∪) and we can write the first segment of our *decomposition tree* $\mathbf{T}_A$:

$$((a \cup b) \Rightarrow \neg a)\boxed{\cup}(\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \Rightarrow \neg a), (\neg a \Rightarrow \neg c)$$

Now we decompose the sequence

$$((a \cup b) \Rightarrow \neg a), (\neg a \Rightarrow \neg c).$$

It is a decomposable sequence with the first, decomposable formula

$$((a \cup b) \Rightarrow \neg a).$$

The next step of the construction of our decomposition tree is determined by its main connective $\Rightarrow$ (we put the box around it). item[The only rule] determined by the sequence is $(\Rightarrow)$ applied (as decomposition rule) to the sequence

$$((a \cup b)\boxed{\Rightarrow}\neg a), (\neg a \Rightarrow \neg c).$$

The second stage of the decomposition tree is now as follows.

$$((a \cup b) \Rightarrow \neg a)\boxed{\cup}(\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b)\boxed{\Rightarrow}\neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\neg(a \cup b), \neg a, (\neg a \Rightarrow \neg c)$$

The next sequence to decompose is the sequence

$$\neg(a \cup b), \neg a, (\neg a \Rightarrow \neg c)$$

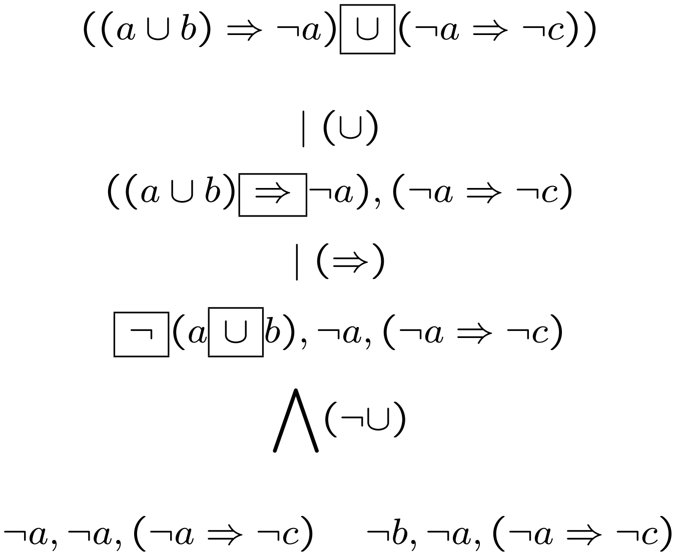
with the first decomposable formula

$$\neg(a \cup b).$$

Its main connective is $\neg$, so to find the appropriate rule we have to examine **next connective**, which is $\cup$.

The decomposition rule determine by this stage of decomposition is $(\neg\cup)$.

Next stage of the construction of the decomposition tree $\mathbf{T}_A$ is as follows.



$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\boxed{\neg} (a \boxed{\cup} b), \neg a, (\neg a \Rightarrow \neg c)$$

$$\bigwedge (\neg \cup)$$

$$\neg a, \neg a, (\neg a \Rightarrow \neg c) \qquad \neg b, \neg a, (\neg a \Rightarrow \neg c)$$

Now we have two decomposable sequences: $\neg a, \neg a, (\neg a \Rightarrow \neg c)$ and $\neg b, \neg a, (\neg a \Rightarrow \neg c)$.

They both happen to have the same first decomposable formula $(\neg a \Rightarrow \neg c)$. We decompose it and obtain the following:

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\boxed{\neg} (a \boxed{\cup} b), \neg a, (\neg a \Rightarrow \neg c)$$

$$\bigwedge (\neg \cup)$$

$$\neg a, \neg a, (\neg a \boxed{\Rightarrow} \neg c) \qquad \neg b, \neg a, (\neg a \boxed{\Rightarrow} \neg c)$$

$$\mid (\Rightarrow) \qquad\qquad\qquad \mid (\Rightarrow)$$

$$\neg a, \neg a, \neg\neg a, \neg c \qquad \neg b, \neg a, \neg\neg a, \neg c$$

It is easy to see that we need only one more step to complete the process of constructing the unique decomposition tree of $\mathbf{T}_A$, namely, by decomposing the sequences:

$$\neg a, \neg a, \neg\neg a, \neg c$$

and

$$\neg b, \neg a, \neg\neg a, \neg c.$$

The complete decomposition tree $\mathbf{T}_A$ is:

$$\mathbf{T}_A$$

$$((a \cup b) \Rightarrow \neg a) \boxed{\cup} (\neg a \Rightarrow \neg c))$$

$$\mid (\cup)$$

$$((a \cup b) \boxed{\Rightarrow} \neg a), (\neg a \Rightarrow \neg c)$$

$$\mid (\Rightarrow)$$

$$\boxed{\neg} (a \boxed{\cup} b), \neg a, (\neg a \Rightarrow \neg c)$$

$$\bigwedge (\neg \cup)$$

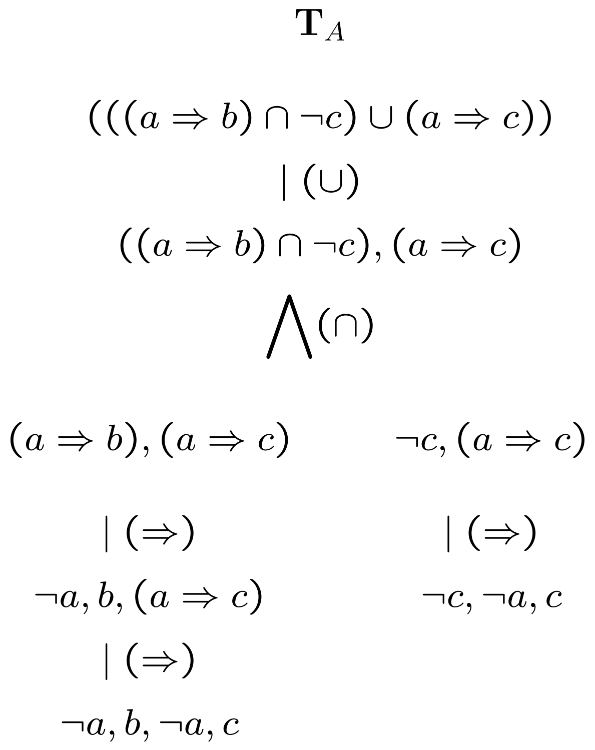
$$\begin{array}{cc}
\neg a, \neg a, (\neg a \boxed{\Rightarrow} \neg c) & \neg b, \neg a, (\neg a \boxed{\Rightarrow} \neg c) \\
\mid (\Rightarrow) & \mid (\Rightarrow) \\
\neg a, \neg a, \boxed{\neg\neg} a, \neg c & \neg b, \neg a, \boxed{\neg\neg} a, \neg c \\
\mid (\neg\neg) & \mid (\neg\neg) \\
\neg a, \neg a, a, \neg c & \neg b, \neg a, a, \neg c
\end{array}$$

All leafs are axioms, the tree represents a proof of A in **RS**

Example Consider now the formula

$$A = (((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$$

and its decomposition tree:



$\mathbf{T}_A$

$(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$

$\mid (\cup)$

$((a \Rightarrow b) \cap \neg c), (a \Rightarrow c)$

$\bigwedge (\cap)$

$(a \Rightarrow b), (a \Rightarrow c)$ $\qquad$ $\neg c, (a \Rightarrow c)$

$\mid (\Rightarrow)$ $\qquad$ $\mid (\Rightarrow)$

$\neg a, b, (a \Rightarrow c)$ $\qquad$ $\neg c, \neg a, c$

$\mid (\Rightarrow)$

$\neg a, b, \neg a, c$

The above tree $\mathbf{T}_A$ is **unique** by the Decision Tree Theorem and represents the only possible search for proof of the formula

$$A = ((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$$

in **RS**. It has a non-axiom leaf, hence the proof of A in **RS does not exists**; i. e.

$$\nvdash \ A.$$

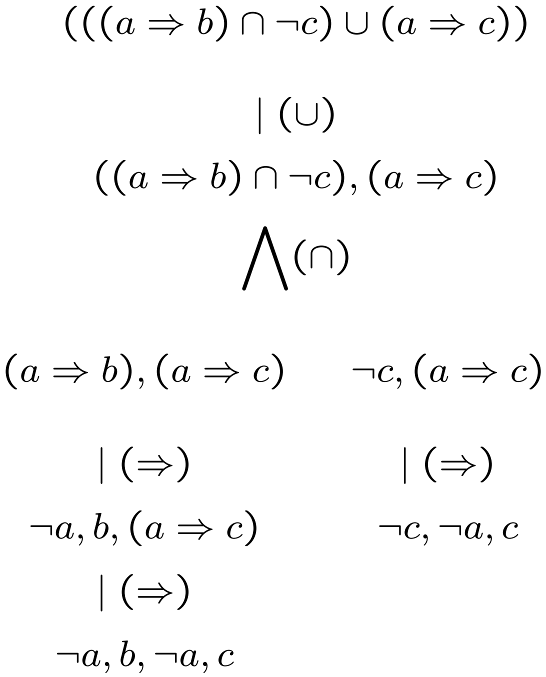
We use this information to construct a truth assignment that would falsify the formula A. Such a variable assignment is called a **counter-model generated by the decomposition tree.**

Counter-model generated by the decomposition tree.

Example: Given a formula A:

$$((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$$

and its decomposition tree $\mathbf{T}_A$.



$(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c))$

$\mid (\cup)$

$((a \Rightarrow b) \cap \neg c), (a \Rightarrow c)$

$\bigwedge(\cap)$

$(a \Rightarrow b), (a \Rightarrow c)$ $\quad$ $\neg c, (a \Rightarrow c)$

$\mid (\Rightarrow)$ $\quad$ $\mid (\Rightarrow)$

$\neg a, b, (a \Rightarrow c)$ $\quad$ $\neg c, \neg a, c$

$\mid (\Rightarrow)$

$\neg a, b, \neg a, c$

Consider a non-axiom leaf:

$$\neg a, b, \neg a, c$$

Let v be any variable assignment

$$v:\ VAR \longrightarrow \{T, F\}$$

such that it makes this non-axiom leaf False, i.e. we put

$$v(a) = T, v(b) = F, v(c) = F.$$

Obviously, we have that

$$v^*(\neg a, b, \neg a, c) = F.$$

Moreover, all our rules of inference are sound (to be proven formally in the next section).

Rules soundness means that if one of premisses of a rule is FALSE, so is the conclusion.

Hence, the soundness of the rules proves (by induction on the degree of sequences $\Gamma \in \mathbf{T}_A$) that v, as defined above falsifies all sequences on the branch of $\mathbf{T}_A$ that ends with the non-axiom leaf $\neg a, b, \neg a, c$.

In particular, the formula A is on this branch, hence

$$v^*(((a \Rightarrow b) \cap \neg c) \cup (a \Rightarrow c)) = F$$

and v is *a counter-model* for A.

The truth assignments defined by a non- axiom leaves are called **counter-models generated** by the decomposition tree.

The construction of the counter-models generated by the decomposition trees are crucial to the proof of the Completeness Theorem for **RS**.