1 Craig Interpolation Theorem

Remark 33.1.1 The Craig Interpolation Theorem was the last deep theorem for first-order logic (Completeness, Compactness, Löwenheim-Skolem), first proven in 1957 by William Craig. It has important applications, two of which we will consider later: the Beth Definability Theorem and the Robinson Consistency Theorem. (These were first proven earlier.)

Lecture 17 (which I did not cover in lecture) gave a proof of the Craig Interpolation Theorem for propositional logic. I will redo that here. In the last homework assignment you will provide another proof of the theorem for propositional logic.

Definition 33.1.2 (Interpolant) A sentence γ is an *interpolant* for the implication $\alpha \to \beta$ if every nonlogical symbol (predicate, constant, parameter) in γ occurs in both α and β and both $\models \alpha \to \gamma$ and $\models \gamma \to \beta$.

Example 33.1.3 Here are some examples of implications with interpolants for propositional sentences.

- $(A \to (B \to C)) \to (B \to (A \to C))$ has $\neg A \vee \neg B \vee C$ as an interpolant. This is shown in example below.
- $(A \vee (B \wedge C)) \to (A \vee \neg\neg B))$ has $A \vee B$ as an interpolant. I leave this for you to show.
- The implication $(A \wedge \neg A) \to B$ has $\perp$ as an interpolant.
- The implication $B \to (A \vee \neg A)$ has $\top$ as an interpolant.
- The first-order implication $(P(a) \wedge (\forall x)Q(x)) \to ((\forall y)S(y) \vee Q(b))$ has $(\forall x)Q(x)$ as an interpolant. This is shown in example below.

An implication which is not a tautology need not have an interpolant. For example $P \to Q$ has no interpolant. In this case the only propositions without propositional symbols are equivalent to $\perp$ and $\top$. Neither of these could be an interpolant.

Remark 33.1.4 We will show that every valid implication $\alpha \to \beta$ has an interpolant. The proof of this fact isolates a first-order consistency property and uses the Model Existence Theorem. But the goal is to do more than prove the mere *existence* of an interpolant, it is to produce a procedure for *constructing* the interpolant. In fact we will show how to turn a contradictory tableau for $\alpha \to \beta$ (it is valid, so has a contradictory tableau) into the construction of an interpolant. It is with this end in mind that we introduce a bookkeeping device, a *biased sentence*. In a tableau proof of a conditional $\alpha \to \beta$, we start with $\neg(\alpha \to \beta)$, then decompose this into α and $\neg\beta$. From this point forward we apply tableau rule to produce new sentences, but we do not keep track whether the ultimate source of the new sentence came from the antecedent α or the consequent β. Keeping track of this will be important in turning a tableau proof of $\alpha \to \beta$ into an interpolant for the conditional.

We enhance the tableau machinery to keep track of the ancestor, α or $\neg\beta$, a sentence was derived from. Think of α as the "left" and $\neg\beta$ as the "right", the respective positions of α and β in $\alpha \leftrightarrow \beta$. We will symbolize this by writing $L(\alpha)$ and $R(\beta)$ and use the notation"L" and "R" throughout each application of a tableau rule, passing on the same symbol "L" or "R" onto the components of the sentence. This is the only role of "L" and "R".

Definition 33.1.5 (Biased Sentence) A *biased sentence* is an expression of the form $L(\phi)$ or $R(\phi)$, where ϕ is a sentence.

An *interpolant* for a finite set of biased sentences, $\{L(\phi_1), \ldots, L(\phi_n), R(\psi_1), \ldots, R(\psi_n)\}$ is an interpolant, in the sense of definition 33.1.2, for the conditional

$$\left(\phi_1 \wedge \ldots \wedge \phi_n\right) \to \left(\neg\psi_1 \vee \ldots \vee \neg\psi_n\right)$$

Take the empty conjunction to be $\top$ and the empty disjunction to be $\perp$.

We use the notation $S \overset{\text{int}}{\Longrightarrow} \phi$ to mean that ϕ is an interpolant for the finite biased set of sentences S.

Remark 33.1.6 Note that $\alpha \to \beta$ has an interpolant if and only if the biased set $\{L(\alpha), R(\neg\beta)\}$ has an interpolant. To see this, note that the latter set has an interpolant if and only if $\alpha \to \neg\neg\beta$ has one. Since β is logically equivalent to $\neg\neg\beta$, it follows that this last sentence has an interpolant if and only if $\alpha \leftrightarrow \beta$ does.

Definition 33.1.7 (Craig Consistency) A finite set S of sentences is *Craig-consistent* if there is a way of assigning "L" and "R" to each sentence so that the biased set of sentences S^* lacks an interpolant. A *Craig-inconsistent* set is one which is not Craig consistent. A finite set S is Craig-inconsistent if for any way of assigning "L" and "R" to each sentence, the biased set produced S^* has an interpolant.

Lemma 33.1.8 Let $\mathcal{C}$ be the collection of Craig-consistent sets. $\mathcal{C}$ is a first-order consistency property.

The Craig interpolation theorem follows from the lemma.

Theorem 33.1.9 (Craig Interpolation Theorem) If $\alpha \to \beta$ is valid, then it has an interpolant.

Proof. The proof is by contraposition: if $\alpha \to \beta$ lacks an interpolant, then it is not valid. Suppose $\alpha \to \beta$ lacks an interpolant. This is equivalent to the biased set $\{L(\alpha), R(\neg\beta)\}$ lacking an interpolant. So, the finite set $\{\alpha, \neg\beta\}$ is Craig-consistent. By the Model Completeness Theorem this set is satisfiable. So $\alpha \to \beta$ is not valid. $\square$

Proof of Lemma 33.1.8. We will actually show that Craig-inconsistency is a first-order inconsistency property. Every infinite set of sentences is Craig-inconsistent, so it is sufficient to show that Craig-consistency over finite sets satisfies conditions (I1) to (I5) of Definition 31.1.5.

Let T be a finite set of (unbiased) sentences. Then a finite set of biased sentences S is *generated* from T if it is obtained from T by choosing a biasing for each of the sentences from T. Let $S = \{L(X_1), \dots, L(X_n), R(Y_1), \dots, R(Y_m)\}$. We will use the following abbreviations:

$$S_L = X_1 \wedge \ldots \wedge X_n \qquad S_R = \neg Y_1 \vee \ldots \vee \neg Y_m.$$

(I1). Let S be any set of biased sentences. It is easy to verify that each of the following are true (where A is any sentence, not just an atomic sentence)

$$\begin{aligned}
S \cup \{L(A), L(\neg A)\} &\overset{\text{int}}{\Longrightarrow} \perp \\
S \cup \{R(A), R(\neg A)\} &\overset{\text{int}}{\Longrightarrow} \top \\
S \cup \{L(A), R(\neg A)\} &\overset{\text{int}}{\Longrightarrow} A \\
S \cup \{R(A), L(\neg A)\} &\overset{\text{int}}{\Longrightarrow} \neg A \\
S \cup \{L(\perp)\} &\overset{\text{int}}{\Longrightarrow} \perp \\
S \cup \{R(\perp)\} &\overset{\text{int}}{\Longrightarrow} \top
\end{aligned}$$

Let S_L be the conjunction of all left-biased sentences of S (or $\top$, if there are none) and S_R be the disjunction of all negated right-biased sentences of S (or $\perp$, if there are none). An interpolant for $S \cup \{L(A), R(\neg A)\}$ is A, since

$$\models S_L \wedge A \to A \quad \text{and} \quad \models A \to \neg\neg A \vee S_R$$

and $\neg A$ is an interpolant for $S \cup \{R(A), L(\neg A)\}$ since

$$\models S_L \wedge \neg A \to \neg A \quad \text{and} \quad \models \neg A \to \neg A \vee S_R$$

The rest are verified similarly.

(I2). Let α be a type-A sentence with components α_1 and α_2. Suppose that $T \cup \{\alpha_1, \alpha_2\}$ is Craig-inconsistent, where T is a finite set of sentences. We must show that $T \cup \{\alpha\}$ is Craig-inconsistent. Consider any biased set S generated from T, so that we must produce an interpolant for both $S \cup \{L(\alpha)\}$ and $S \cup \{R(\alpha)\}$. Let A be an interpolant for $S \cup \{L(\alpha_1), L(\alpha_2)\}$. Then

$$\models S_L \wedge \alpha_1 \wedge \alpha_2 \rightarrow A \quad \text{and} \quad \models A \rightarrow S_R.$$

But since $\alpha \leftrightarrow \alpha_1 \wedge \alpha_2$ is valid,

$$\models S_L \wedge \alpha \rightarrow A \quad \text{and} \quad \models A \rightarrow S_R,$$

so that A is an interpolant for $S \cup \{L(\alpha)\}$. Let B be an interpolant for $S \cup \{R(\alpha_1), R(\alpha_2)\}$. Then

$$\models S_L \rightarrow B \quad \text{and} \quad \models B \rightarrow S_R \vee \neg\alpha_1 \vee \neg\alpha_2.$$

But since $\neg\alpha \leftrightarrow \neg\alpha_1 \vee \neg\alpha_2$ is valid,

$$\models S_L \rightarrow B \quad \text{and} \quad \models B \rightarrow S_R \vee \neg\alpha$$

so that A is an interpolant for $S \cup \{R(\alpha)\}$.

(I3). Let β be a type-B sentence with components β_1 and β_2 and suppose that both $T \cup \{\beta_1\}$ and $T \cup \{\beta_2\}$ are Craig inconsistent, where T is any finite set of (unbiased) sentences. We must show that $T \cup \{\beta\}$ is Craig-inconsistent. Consider any biased set S generated from T, so that we must produce an interpolant for both $S \cup \{L(\beta)\}$ and $S \cup \{R(\beta)\}$. Let A_1 be an interpolant for $S \cup \{L(\beta_1)\}$ and A_2 be an interpolant for $S \cup \{L(\beta_2)\}$. Each of the following are true by hypothesis

$$\begin{array}{lcl} \models S_L \wedge \beta_1 \rightarrow A_1 & \text{and} & \models A_1 \rightarrow S_R \\ \models S_L \wedge \beta_2 \rightarrow A_2 & \text{and} & \models A_2 \rightarrow S_R \end{array}$$

Let $A = A_1 \vee A_2$, so it follows from the above that

$$\begin{array}{l} \models S_L \wedge (\beta_1 \vee \beta_2) \rightarrow A \\ \models A \rightarrow S_R \end{array}$$

Since $\beta \leftrightarrow (\beta_1 \vee \beta_2)$, it follows that A is an interpolant for $S \cup \{L(\beta)\}$. Now let B_1 be an interpolant for $S \cup \{R(\beta_1)\}$ and B_2 be an interpolant for $S \cup \{R(\beta_2)\}$. Each of the following are true by hypothesis

$$\begin{array}{lcl} \models S_L \rightarrow B_1 & \text{and} & \models B_1 \rightarrow \neg\beta_1 \vee S_R \\ \models S_L \rightarrow B_2 & \text{and} & \models B_2 \rightarrow \neg\beta_2 \vee S_R \end{array}$$

Let $B = B_1 \wedge B_2$, so it follows from the above that

$$\begin{array}{l} \models S_L \rightarrow B \\ \models B \rightarrow (\neg\beta_1 \wedge \neg\beta_2) \vee S_R \end{array}$$

Since $\neg\beta \leftrightarrow (\neg\beta_1 \wedge \neg\beta_2)$ is valid, it follows that B is an interpolant for $S \cup \{R(\beta)\}$.

(I4). This is the most complicated of the cases. Let γ be a type-C sentence and T a finite set of (unbiased) sentences. Suppose that $T \cup \{\gamma(t)\}$ is Craig-inconsistent for some variable-free term t. We must show that $T \cup \{\gamma\}$ is Craig-inconsistent. Let S be any biased set generated from T, and let $A(t)$ be an interpolant for $S \cup \{L(\gamma(t)\}$, where t may occur in $A(t)$. Then

$$\models S_L \wedge \gamma(t) \rightarrow A(t) \quad \text{and} \quad \models A(t) \rightarrow S_R.$$

The term t may or may not occur in $S_L \wedge \gamma$. If t occurs in $S \wedge \gamma$, then we can take the interpolant to be $A(t)$ itself, since $(S_L \wedge \gamma) \rightarrow (S_L \wedge \gamma(t))$ is valid, so that

$$\models S_L \wedge \gamma \rightarrow A(t) \quad \text{and} \quad \models A(t) \rightarrow S_R.$$

If t does not occur in $S_L \wedge \gamma$, then we cannot take $A(t)$ to be the interpolant. In this case, however, $(\forall x)A$ is an interpolant (where $A = A^t_x$ and x is freely substitutable for t in $A(t)$.) It is certainly true that

$$\models (\forall x)A \rightarrow S_R,$$

since $(\forall x)A^t_x \rightarrow A(t)$ is valid. The tricky part is showing that

$$\models S_L \wedge \gamma \rightarrow (\forall x)A.$$

Suppose there were a structure $\mathcal{B}$ such that $S_L \wedge \gamma$ were true, but $(\forall x)A$ was false. Let b be in the domain of $\mathcal{B}$ with $A(b) = A^t_b$ false. Since t is variable-free, it is either a constant or a parameter. Let $\mathcal{B}^*$ be the structure which is exactly like $\mathcal{B}$, except that $t^{\mathcal{B}^*} = b$. Since t did not occur in $S_L \wedge \gamma$, this is still true, but now $A(t)$ is false. However γ is true and universal, so $\gamma(t)$ is also true in $\mathcal{B}^*$. But now $S_L \wedge \gamma(t)$ is true and $A(t)$ is false, which contradicts our assumption that $A(t)$ was an interpolant for $S \cup \{L(\gamma(t)\}$. So, $(\forall x)A$ is an interpolant for $S \cup \{L(\gamma\}$.

Let $B(t)$ be an interpolant for $S \cup \{R(\gamma(t)\}$. Then

$$\models S_L \rightarrow B(t) \quad \text{and} \quad \models B(t) \rightarrow \neg\gamma(t) \vee S_R.$$

If t occurs in $\neg\gamma \vee S_R$, then we can take $B(t)$ to be an interpolant for $S \cup \{R(\gamma)\}$, since $\neg\gamma(t) \rightarrow \neg\gamma$ is valid, so that

$$\models S_L \rightarrow B(t) \quad \text{and} \quad \models B(t) \rightarrow \neg(\forall x)\gamma \vee S_R.$$

If t does not occur in $\neg\gamma \vee S_R$, then $(\exists x)B$ will be an interpolant (where $B = B^t_x$ and x is freely substitutable for t in $B(t)$). It is certainly true that

$$\models S_L \rightarrow (\exists x)B,$$

since $B(t) \rightarrow (\exists x)B$ is valid. Suppose it is not true that

$$\models (\exists x)B \rightarrow \neg(\forall x)\gamma \vee S_R.$$

Let $\mathcal{A}$ be a structure in which $(\exists x)B$ is true, but $\neg(\forall x)\gamma \vee S_R$ is false. Let a be an element in the domain of $\mathcal{A}$ with $B(a) = B^t_a$ true. Let $\mathcal{A}^*$ be a structure just like $\mathcal{A}$, except that $t^{\mathcal{A}^*} = a$. Then $\neg(\forall x)\gamma \vee S_R$ is still false in $\mathcal{A}^*$, but not $B(t)$ is true in $\mathcal{A}^*$. This contradicts our assumption that $B(t)$ was an interpolant for $S \cup \{\gamma(t)\}$. So, $(\exists x)B$ is an interpolant for $S \cup \{R(\gamma\}$.

(I5). Let δ be a type-C sentence and T a finite set of (unbiased) sentences. Suppose that $T \cup \{\delta(c)\}$ is Craig-inconsistent for some parameter c not occurring in T or δ. We must show that $T \cup \{\delta\}$ is Craig-inconsistent. Let S be any biased set generated from T. Let $A(c)$ be an interpolant for $S \cup \{L(\delta(c)\}$ (where c may or may not occur in $A(c)$). Then

$$\models S_L \wedge \delta(c) \rightarrow A(c) \quad \text{and} \quad \models A(c) \rightarrow S_R.$$

Since c does not occur in S_R or δ (it does not occur in S), it follows that c does not occur in $A(c)$. It follows by existential generalization that

$$\models S_L \wedge \delta \rightarrow A$$

where $A = A(c)$. So, A is an interpolant for $S \cup \{L(\delta\}$.

Let $B(c)$ be an interpolant for $S \cup \{R(\delta(c)\}$. Then

$$\models S_L \rightarrow B(c) \quad \text{and} \quad \models B(c) \rightarrow \neg\delta(c) \vee S_R.$$

Again, c cannot occur in $B(c)$. It follows by universal generalization that

$$\models B \rightarrow \neg\delta \vee S_R$$

So, B is an interpolant for $S \cup \{R(\delta\}$. $\square$

2 Craig Interpolation Theorem–Constructively

Remark 33.2.1 It is nice to know that every valid implication $\alpha \to \beta$ has an interpolant, but how to we find one? The proof we gave above suggests a way to actually construct an interpolant out of a *proof* that $\alpha \to \beta$ is valid! The key insight is to stand a tableau proof of the validity of $\alpha \to \beta$ "on its head".

In Lecture 30 we introduced the idea of a first-order consistency property and showed in the Model Completeness Theorem that if $\mathcal{C}$ is a first-order consistency property and $S \in \mathcal{C}$, then S is satisfiable. The argument that this was so is not essentially any different then constructing a tableau for S (ignoring all paths that lead to contradictions). A tableau is constructive, while the argument for the Model Completeness Theorem is not. For example, in a tableau when faced with a type-B sentence β we split into two branches with β_1 on one branch and β_2 on the other branch. In the Model Completeness Theorem we merely noted that one branch must be noncontradictory, if it was noncontradictory up to β, and chose the right path. In fact the conditions (C2) to (C5) of first-order consistency property were simply the rules of construction for semantic tableaux.

The conditions for a first-order inconsistency property are the rules of tableau run in reverse. We start with a contradiction and work backward from the *components* of a sentence to the sentence itself. Since our proof of the Craig Interpolation Theorem was via a first-order inconsistency property, it makes sense to take a contradictory tableau, stand it on its head, and work from the contradictions at the leaves to the root containing $\neg(\alpha \to \beta)$.

We motivated semantic tableaux as an attempt to produce a counterexample; if our attempt succeeds, we know the original sentence cannot be valid and if our attempt fails, we have a proof of validity. Perhaps Logicians are optimists: they believe every sentence may be valid and try to prove validity starting with a valid starting point (the axioms) and use valid rules of inference. It if for this reason that first-order *inconsistency* properties are more natural when we think of proof systems. In fact semantic tableau is somewhat neglected in the current logical literature by those who study proof, while its inverted twin (Gentzen systems) is much more popular. A Gentzen system is a semantic tableau stood on its head: these systems literally start at the contradictions (the leaves of the tableau, called *axioms*) and work back up the tableau by applying the rules in reverse!! That is precisely what we will do to produce an interpolant.

Definition 33.2.2 (Biased Tableau) A *biased tableau* is a semantic tableau extended to biased sentences in a straightforward way, with the bias preserved from a sentence to its components. For instance the standard type-A rule gives rise to two biased type-A rules

$$\begin{array}{c} L(\alpha) \\ | \\ L(\alpha_1) \\ L(\alpha_2) \end{array} \qquad\qquad \begin{array}{c} R(\alpha) \\ | \\ R(\alpha_1) \\ R(\alpha_2) \end{array}$$

The other rules are similar. A tableau constructed using these biased rules is a *biased tableau.* A branch of a tableau is closed if it contains biased sentences $\pi(X)$ and $\rho(\neg X)$ regardless of the bias π and ρ.

Remark 33.2.3 If a sentence $X \to Y$ has a tableau proof, it can be converted to a closed bias tableau for $\{L(X), R(\neg Y)\}$. Simply take the closed tableau beginning with $\neg(X \to Y)$, drop the first line, getting a tableau beginning with X and $\neg Y$, replace X with the biased $L(X)$ and $\neg Y$ with the biased $R(\neg Y)$, then continue applying the biased rules to determine the bias of each succeeding sentence in the tableau. Our procedure for producing an interpolant will extract one from the *biased proof.*

The idea is essentially this. Begin with each closed branch, assign an interpolant (to be defined shortly) to it (based on the nature of the contradiction), then, one by one, undo each tableau rule application, calculating the interpolants for the resulting shortened branches from those for the original longer ones. For example, suppose the last rule applied on a branch is one of the biased type-A rules, from $L(\alpha)$ to constituents $L(\alpha_1)$ and $L(\alpha_2)$, and suppose we have an interpolant for the branch $S \cup \{L(\alpha), L(\alpha_1), L(\alpha_2)\}$, say A. Then our procedure will compute a new interpolant for the set of biased sentences $S \cup \{L(\alpha)\}$, corresponding to the branch before the biased type-A rule was applied. In this case, we will have a rule of the form

$$\frac{S \cup \{L(\alpha_1), L(\alpha_2)\} \stackrel{\text{int}}{\Longrightarrow} A}{S \cup \{L(\alpha)\} \stackrel{\text{int}}{\Longrightarrow} A}$$

which means that if Z is an interpolant for the biased set $S \cup \{L(\alpha_1), L(\alpha_2)\}$, then it is also an interpolant for the biased set $S \cup \{L(\alpha)\}$, corresponding to the branch before the type-A rule was applied. Continuing in this way, we work our way back to the beginning $\{L(X), R(\neg Y)\}$, producing an interpolant for $X \to Y$.

Definition 33.2.4 (Rules for Interpolants) We give the calculation rules for producing interpolants based on the proof of Theorem 33.1.9. The first set of rules is for closed branches. A is any sentence.

$$\begin{aligned}
&S \cup \{L(A), L(\neg A)\} \stackrel{\text{int}}{\Longrightarrow} \bot \\
&S \cup \{R(A), R(\neg A)\} \stackrel{\text{int}}{\Longrightarrow} \top \\
&S \cup \{L(A), R(\neg A)\} \stackrel{\text{int}}{\Longrightarrow} A \\
&S \cup \{R(A), L(\neg A)\} \stackrel{\text{int}}{\Longrightarrow} \neg A \\
&S \cup \{L(\bot)\} \stackrel{\text{int}}{\Longrightarrow} \bot \\
&S \cup \{R(\bot)\} \stackrel{\text{int}}{\Longrightarrow} \top \\
&S \cup \{L(\neg \top)\} \stackrel{\text{int}}{\Longrightarrow} \bot \\
&S \cup \{R(\neg \top)\} \stackrel{\text{int}}{\Longrightarrow} \top
\end{aligned}$$

I have split the type-A rules to separate out negation, where Z is any sentence:

$$\frac{S \cup \{L(\neg\neg Z\} \stackrel{\text{int}}{\Longrightarrow} A}{S \cup \{L(Z\} \stackrel{\text{int}}{\Longrightarrow} A} \qquad \frac{S \cup \{R(\neg\neg Z\} \stackrel{\text{int}}{\Longrightarrow} A}{S \cup \{R(Z\} \stackrel{\text{int}}{\Longrightarrow} A}$$

The general rule for type-A sentences with two components:

$$\frac{S \cup \{L(\alpha_1), L(\alpha_2)\} \stackrel{\text{int}}{\Longrightarrow} A}{S \cup \{L(\alpha)\} \stackrel{\text{int}}{\Longrightarrow} A} \qquad \frac{S \cup \{R(\alpha_1), R(\alpha_2)\} \stackrel{\text{int}}{\Longrightarrow} A}{S \cup \{R(\alpha)\} \stackrel{\text{int}}{\Longrightarrow} A}$$

Type-B sentences:

$$\frac{S \cup \{L(\beta_1)\} \stackrel{\text{int}}{\Longrightarrow} A \qquad S \cup \{L(\beta_2)\} \stackrel{\text{int}}{\Longrightarrow} B}{S \cup \{L(\beta)\} \stackrel{\text{int}}{\Longrightarrow} A \vee B} \qquad \frac{S \cup \{R(\beta_1)\} \stackrel{\text{int}}{\Longrightarrow} A \qquad S \cup \{R(\beta_2)\} \stackrel{\text{int}}{\Longrightarrow} B}{S \cup \{R(\beta)\} \stackrel{\text{int}}{\Longrightarrow} A \wedge B}$$

This takes care of the propositional cases. The quantifier cases are as follows. First, if $S = \{L(X_1), \ldots, L(X_n), R(Y_1), \ldots, R$ then let $S_L = \{L(X_1), \ldots, L(X_n)\}$ and $S_R = \{R(Y_1), \ldots, R(Y_m)\}$.

For type-C sentences, there are four rules. The nonlogical symbol c is a constant or parameter, and the cases split on whether or not c occurs among the side sentences S. In each case we assume that the variable x is substitutable for c.

$$\frac{S \cup \{L(\gamma(c))\} \stackrel{\text{int}}{\Longrightarrow} A}{S \cup \{L(\gamma)\} \stackrel{\text{int}}{\Longrightarrow} A} \text{ if } c \text{ occurs in } S_L \qquad \frac{S \cup \{R(\gamma(c))\} \stackrel{\text{int}}{\Longrightarrow} A}{S \cup \{R(\gamma)\} \stackrel{\text{int}}{\Longrightarrow} A} \text{ if } c \text{ occurs in } S_R$$

If c does not occur among the side formula in S, we must quantify it out:

$$\frac{S \cup \{L(\gamma(c))\} \overset{\text{int}}{\Longrightarrow} A}{S \cup \{L(\gamma)\} \overset{\text{int}}{\Longrightarrow} (\forall x)A^c_x}\ \text{otherwise} \qquad \frac{S \cup \{R(\gamma(c))\} \overset{\text{int}}{\Longrightarrow} A}{S \cup \{R(\gamma)\} \overset{\text{int}}{\Longrightarrow} (\exists x)A^c_x}\ \text{otherwise}$$

Finally, the type-D rules. Here, the nonlogical symbol a is a parameter that does not occur in either δ or the side sentences S.

$$\frac{S \cup \{L(\delta(a))\} \overset{\text{int}}{\Longrightarrow} A}{S \cup \{L(\delta)\} \overset{\text{int}}{\Longrightarrow} A} \qquad \frac{S \cup \{R(\delta(a))\} \overset{\text{int}}{\Longrightarrow} A}{S \cup \{R(\delta)\} \overset{\text{int}}{\Longrightarrow} A}$$

Example 33.2.5 We compute an interpolant for the tautology $(A \to (B \to C)) \to (B \to (A \to C))$. We begin with a closed biased tableau. At the end of each branch, instead of a $\otimes$, we give in square brackets an interpolant for the set of biased sentences in the branch according to the rules just given.

$$\begin{array}{c}
L\big(A \to (B \to C)\big)\\
R\big(\neg (B \to (A \to C))\big)\\
R\big(B\big)\\
R\big(\neg (A \to C)\big)\\
R\big(A\big)\\
R\big(\neg C\big)\\
\begin{array}{cc}
\begin{array}{c} L\big(\neg A\big)\\ [\neg A] \end{array} &
\begin{array}{c} L\big(B \to C\big)\\
\begin{array}{cc} \begin{array}{c} L\big(\neg B\big)\\ [\neg B] \end{array} & \begin{array}{c} \mathrm{L}\big(\mathrm{C}\big)\\ [C] \end{array} \end{array}
\end{array}
\end{array}
\end{array}$$

Now we work our way back up the tree through the leaves. We start on the rightmost branch:

$$\begin{array}{c}
L\big(A \to (B \to C)\big)\\
R\big(\neg (B \to (A \to C))\big)\\
R\big(B\big)\\
R\big(\neg (A \to C)\big)\\
R\big(A\big)\\
R\big(\neg C\big)\\
\begin{array}{cc}
\begin{array}{c} L\big(\neg A\big)\\ [\neg A] \end{array} &
\begin{array}{c} L\big(B \to C\big)\\
\begin{array}{cc} \begin{array}{c} L\big(\neg B\big)\\ [\neg B] \end{array} & \begin{array}{c} \mathrm{L}\big(\mathrm{C}\big)\\ [C] \end{array} \end{array}
\end{array}
\end{array}
\end{array}
\qquad
\begin{array}{c}
L\big(A \to (B \to C)\big)\\
R\big(\neg (B \to (A \to C))\big)\\
R\big(B\big)\\
R\big(\neg (A \to C)\big)\\
R\big(A\big)\\
R\big(\neg C\big)\\
\begin{array}{cc}
\begin{array}{c} L\big(\neg A\big)\\ [\neg A] \end{array} &
\begin{array}{c} L\big(B \to C\big)\\ [\neg B \vee C] \end{array}
\end{array}
\end{array}
\qquad
\begin{array}{c}
L\big(A \to (B \to C)\big)\\
R\big(\neg (B \to (A \to C))\big)\\
R\big(B\big)\\
R\big(\neg (A \to C)\big)\\
R\big(A\big)\\
R\big(\neg C\big)\\
[\neg A \vee (\neg B \vee C)]
\end{array}$$

The rest of the rules are type-A, which do not modify the interpolant, so the interpolant is $\neg A \vee (\neg B \vee C)$.

Example 33.2.6 $(P(a) \wedge (\forall x)Q(x)) \to ((\forall y)S(y) \vee Q(b))$

$$\begin{array}{c}
L\big(P(a) \wedge (\forall x)Q(x)\big)\\
R\big(\neg ((\forall y)S(y) \vee Q(b))\big)\\
L\big(P(a)\big)\\
L\big((\forall x)Q(x)\big)\\
R\big(\neg (\forall y)S(y)\big)\\
R\big(\neg Q(b)\big)\\
L\big(Q(b)\big)\\
[Q(b)]
\end{array}$$

We work our way back up the tree starting with the last line, an application of a type-C rule, where b does not occur among the left biased side sentences before $Q(b)$ is introduced.

$$L\big(P(a) \wedge (\forall x)Q(x)\big)$$
$$R\big(\neg((\forall y)S(y) \vee Q(b))\big)$$
$$L\big(P(a)\big)$$
$$L\big((\forall x)Q(x)\big)$$
$$R\big(\neg(\forall y)S(y)\big)$$
$$R\big(\neg Q(b)\big)$$
$$[(\forall x)Q(x)]$$

From this point forward, all rule applications are type-A, which do not change the interpolant for the set. So, $(\forall x)Q(x)$ is an interpolant.