Craig-Interpolation

Theorem 3.41 (Craig 1957) *Let F and G be two propositional formulas such that $F \models G$. Then there exists a formula H (called the* interpolant *for $F \models G$), such that H contains only prop. variables occurring both in F and in G, and such that $F \models H$ and $H \models G$.*

Proof. Translate F and $\neg G$ into CNF. let N and M, resp., denote the resulting clause set. Choose an atom ordering $\succ$ for which the prop. variables that occur in F but not in G are maximal. Saturate N into N' w. r. t. $\mathit{Res}_{\mathrm{sel}}^{\succ}$ with an empty selection function sel. Then saturate $N' \cup M$ w. r. t. $\mathit{Res}_{\mathrm{sel}}^{\succ}$ to derive $\bot$. As N' is already saturated, due to the ordering restrictions only inferences need to be considered where premises, if they are from N', only contain symbols that also occur in G. The conjunction of these premises is an interpolant H.

The theorem also holds for first-order formulas, but in the general case, a proof based on resolution technology is complicated because of Skolemization. □

3.13 Redundancy

So far: local restrictions of the resolution inference rules using orderings and selection functions.

Is it also possible to delete clauses altogether? Under which circumstances are clauses unnecessary? (e. g., if they are tautologies)

Intuition: If a clause is guaranteed to be neither a minimal counterexample nor productive, then we do not need it.

A Formal Notion of Redundancy

Let N be a set of ground clauses and C a ground clause (not necessarily in N). C is called *redundant* w. r. t. N, if there exist $C_1, \dots, C_n \in N$, $n \geq 0$, such that $C_i \prec C$ and $C_1, \dots, C_n \models C$.

Redundancy for general clauses: C is called *redundant* w. r. t. N, if all ground instances $C\sigma$ of C are redundant w. r. t. $G_\Sigma(N)$.

Intuition: If a ground clause C is redundant, then $I_C \models C$.

Note: The same ordering $\succ$ is used for ordering restrictions and for redundancy (and for the completeness proof).

Examples of Redundancy

In general, redundancy is undecidable. Decidable approximations are sufficient for us, however.

Proposition 3.42 *Some redundancy criteria:*

- *C tautology (i. e., $\models C$) $\Rightarrow$ C redundant w. r. t. any set N.*
- *$C\sigma \subset D$ $\Rightarrow$ D redundant w. r. t. $N \cup \{C\}$.*

(Under certain conditions one may also use non-strict subsumption, but this requires a slightly more complicated definition of redundancy.)

Saturation up to Redundancy

N is called *saturated up to redundancy* (w. r. t. $\mathit{Res}^{\succ}_{\text{sel}}$) if

$$\mathit{Res}^{\succ}_{\text{sel}}(N \setminus \mathit{Red}(N)) \subseteq N \cup \mathit{Red}(N)$$

Theorem 3.43 *Let N be saturated up to redundancy. Then*

$$N \models \bot \Leftrightarrow \bot \in N$$

Proof (Sketch).
(i) Ground case: Consider the construction of the candidate interpretation $I^{\succ}_{N}$ for $\mathit{Res}^{\succ}_{\text{sel}}$.

If a clause $C \in N$ is redundant, then there exist $C_1, \dots, C_n \in N$, $n \geq 0$, such that $C_i \prec C$ and $C_1, \dots, C_n \models C$.

By minimality, $I_C \models C_i$, therefore $I_C \models C$.

In particular, C is not productive.

$\Rightarrow$ Redundant clauses are not used as premises for "essential" inferences, so the rest of the proof works as before.

(ii) Lifting: no additional problems over the proof of Theorem 3.40. □

Monotonicity Properties of Redundancy

When we want to delete redundant clauses during a derivation, we have to ensure that redundant clauses *remain redundant* in the rest of the derivation.

Theorem 3.44

(i) $N \subseteq M \Rightarrow \mathit{Red}(N) \subseteq \mathit{Red}(M)$

(ii) $M \subseteq \mathit{Red}(N) \Rightarrow \mathit{Red}(N) \subseteq \mathit{Red}(N \setminus M)$

Proof. (i) Obvious.

(ii) Follows from the compactness of first-order logic and the well-foundedness of the multiset extension of the clause ordering. □

Recall that $\mathit{Red}(N)$ may include clauses that are not in N.

Computing Saturated Sets

Redundancy is preserved when, during a theorem proving derivation one adds new clauses or deletes redundant clauses. This motivates the following definitions:

A *run* of the resolution calculus is a sequence $N_0 \vdash N_1 \vdash N_2 \vdash \ldots$, such that
(i) $N_i \models N_{i+1}$, and
(ii) all clauses in $N_i \setminus N_{i+1}$ are redundant w. r. t. N_{i+1}.

In other words, during a run we may add a new clause if it follows from the old ones, and we may delete a clause, if it is redundant w. r. t. the remaining ones.

For a run, we define $N_\infty = \bigcup_{i\geq 0} N_i$ and $N_* = \bigcup_{i\geq 0} \bigcap_{j\geq i} N_j$.The set N_* of all *persistent* clauses is called the *limit* of the run.

Lemma 3.45 *Let* $N_0 \vdash N_1 \vdash N_2 \vdash \ldots$ *be a run. Then* $\mathit{Red}(N_i) \subseteq \mathit{Red}(N_\infty)$ *and* $\mathit{Red}(N_i) \subseteq \mathit{Red}(N_*)$ *for every* i.

Proof. Exercise. □

Corollary 3.46 $N_i \subseteq N_* \cup \mathit{Red}(N_*)$ *for every* i.

Proof. If $C \in N_i \setminus N_*$, then there is a $k \geq i$ such that $C \in N_k \setminus N_{k+1}$, so C must be redundant w. r. t. N_{k+1}. Consequently, C is redundant w. r. t. N_*. □

Even if a set N is inconsistent, it could happen that $\bot$ is never derived, because some required inference is never computed.

The following definition rules out such runs:

A run is called *fair*, if the conclusion of every inference from clauses in $N_* \setminus \mathit{Red}(N_*)$ is contained in some $N_i \cup \mathit{Red}(N_i)$.

Lemma 3.47 *If a run is fair, then its limit is saturated up to redundancy.*

Proof. If the run is fair, then the conclusion of every inference from non-redundant clauses in N_* is contained in some $N_i \cup \mathit{Red}(N_i)$, and therefore contained in $N_* \cup \mathit{Red}(N_*)$. Hence N_* is saturated up to redundancy. □

Theorem 3.48 (Refutational Completeness: Dynamic View) *Let $N_0 \vdash N_1 \vdash N_2 \vdash \dots$ be a fair run, let N_* be its limit. Then N_0 has a model if and only if $\bot \notin N_*$.*

Proof. ($\Leftarrow$): By fairness, N_* is saturated up to redundancy. If $\bot \notin N_*$, then it has a Herbrand model. Since every clause in N_0 is contained in N_* or redundant w. r. t. N_*, this model is also a model of $G_\Sigma(N_0)$ and therefore a model of N_0.

($\Rightarrow$): Obvious, since $N_0 \models N_*$. □

Simplifications

In theory, the definition of a run permits to add arbitrary clauses that are entailed by the current ones.

In practice, we restrict to two cases:

- We add conclusions of $\mathit{Res}^{\succ}_{\text{sel}}$-inferences from non-redundant premises.
  $\leadsto$ necessary to guarantee fairness
- We add clauses that are entailed by the current ones if this *makes* other clauses redundant:

  $$N \cup \{C\} \;\vdash\; N \cup \{C, D\} \;\vdash\; N \cup \{D\}$$
  $$\text{if } N \cup \{C\} \models D \text{ and } C \in \mathit{Red}(N \cup \{D\}).$$

  Net effect: C is *simplified* to D
  $\leadsto$ useful to get easier/smaller clause sets

Examples of simplification techniques:

- Deletion of duplicated literals:

  $$N \cup \{C \vee L \vee L\} \;\vdash\; N \cup \{C \vee L\}$$

- Subsumption resolution:

  $$N \cup \{D \vee L,\; C \vee D\sigma \vee \overline{L}\sigma\} \;\vdash\; N \cup \{D \vee L,\; C \vee D\sigma\}$$