Computational Logic Lab V

Camilo Thorne   Sergio Tessaris

25/4/2011

Stalmarck's Method. Stalmarck's method constitutes yet another forward-chaining (i.e., saturation-based) negative calculus for propositional logic, that is both sound and complete for *unsatisfiability*. It is defined by combining a *dilemma* rule with any set of (satisfiability preserving) deduction rules for propositional formulas. In what follows, we will deal with its standard version, based on the Boolean algebra laws of propositional logic, combined the replacement theorem.

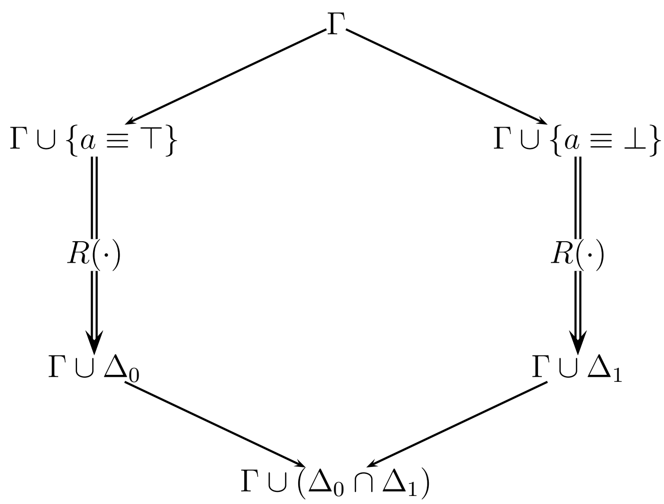
The inference system $R(\cdot)$ of rules is defined by repeated application of the following deduction rules. On the one hand, the *Boolean laws*

$$
\begin{array}{rlrl}
\Gamma \cup \{\varphi \vee \top\} \Longrightarrow \Gamma \cup \{\top\} & & \Gamma \cup \{\varphi \wedge \top\} \Longrightarrow \Gamma \cup \{\varphi\} \\
\Gamma \cup \{\varphi \vee \bot\} \Longrightarrow \Gamma \cup \{\varphi\} & & \Gamma \cup \{\varphi \wedge \bot\} \Longrightarrow \Gamma \cup \{\bot\} \\
\Gamma \cup \{\varphi \vee \neg\bot\} \Longrightarrow \Gamma \cup \{\varphi \vee \top\} & & \Gamma \cup \{\varphi \wedge \neg\bot\} \Longrightarrow \Gamma \cup \{\varphi \wedge \top\} \\
\Gamma \cup \{\varphi \vee \neg\top\} \Longrightarrow \Gamma \cup \{\varphi \vee \bot\} & & \Gamma \cup \{\varphi \wedge \neg\top\} \Longrightarrow \Gamma \cup \{\varphi \wedge \bot\} \\
\Gamma \cup \{\varphi \wedge \varphi\} \Longrightarrow \Gamma \cup \{\varphi\} & & \Gamma \cup \{\varphi \vee \varphi\} \Longrightarrow \Gamma \cup \{\varphi\}
\end{array}
$$

and, on the other hand, the *replacement law*

$$\Gamma \cup \{\varphi \equiv \psi\} \Longrightarrow \Gamma \cup \Gamma[\psi/\varphi].$$

The *dilemma* $S(\cdot)$ rule is defined by repeatedly applying to a set Γ of clauses, for every atom $a \in Atom(\Gamma)$, the following case-based split/merge rule



where Δ_0 and Δ_1 are the clauses derived from, resp., $R(\Gamma \cup \{a \equiv \top\})$ and $R(\Gamma \cup \{a \equiv \bot\})$.

The system $R(\cdot)$ is trivially sound and complete for unsatisfiability, since *(i)* its rules correspond to the classical Boolean laws of the propositional calculus, and *(ii)* by the replacement theorem, replacing equivalent formulas preserves truth. Soundness and completeness also holds for $S(\cdot)$:

Proposition 1. *The dilemma rule $S(\cdot)$ is sound and complete for unsatisfiability.*

The rules give way to so-called saturations, in which $S(\cdot)$ and $R(\cdot)$ are applied iteratively until a contradiction or a satisfiable set of clauses is derived. More precisely, a Stalmarck *saturation* Γ_∞ is defined by induction by putting:

$$\begin{array}{rcl} \Gamma_0 & := & R(\Gamma) \\ \Gamma_{i+1} & := & S(R(\Gamma_i)) \\ \Gamma_\infty & := & \bigcup_{i\geq 0}\Gamma_i \end{array}$$

where the integer i is the saturation *level* and each set Γ_i is called a saturation *state* A formula φ is said to be *i-easy* if there exists a level i s.t. $\varphi \in \Gamma_i$, otherwise it is said to be *i-hard*. This calculus is sound and complete in the following sense:

Proposition 2. *A set Γ of clauses is unsatifiable iff there exists a state $\Gamma_i \subseteq \Gamma_\infty$ s.t. $\top \equiv \bot \in \Gamma_i$, i.e., if it i-easy for some $i \geq 0$.*

The calculus is a decision procedure for propopositional logic: saturations finitely converge and, hence, any algorithm implementing the calculus trivially terminates:

Proposition 3. *Let Γ be a set of clauses. Then the Stalmarck saturation of Γ is finite (i.e., $\Gamma_{i+1} = \Gamma_i$, for some finite $i \geq 0$).*

Exercises.

1. Prove Proposition 1.
2. Prove Proposition 2.
3. Provide a pseudo-code specification of $\textsc{Stal}(\cdot)$, i.e., the algorithm that computes the Stalmarck saturation Γ_∞. Prove that is is sound and complete (termination follows from Proposition 2). Conclude that $\textsc{Stal}(\cdot)$ is a decision procedure for the propositional calculus (why?).
4. Recall the exercise sheet from 18/3/2011 on Urquhart formulas and Harrison's `stalmarck` function.
   - Why is Stålmarck's method more efficient than, say, $\textsc{Dpll}(\cdot)$, on these formulas?
   - Show that, in general, a propositional formula φ can be shown to be i-easy according to Harrison's procedure in time $\mathbf{O}(|\varphi|^{2i+1})$ (since i is bounded above by $|\varphi|$, this yields a worst-case exponential behavior!).