MESSAGE PASSING IN LOGICAL INFERENCE

RESEARCH NOTES IN THE ENEXA AND QROM PROJECTS

November 4, 2025

1 Unit Clause Propagation (UCP)

Unit clauses are assignments to single variables. They are propagated between clusters of formulas

UCP is Knowledge Propagation (see Algorithm 1 in the report) in case of

- Domain edges by single variable nodes, i.e.

$$\mathcal{E}^k = \left\{\{v\} \,:\, v \in \mathcal{V}\right\}$$

- Initial queue by those edges containing single nodes

$$\mathcal{Q} = \left\{\{v\} \,:\, \{v\} \in \mathcal{E}\right\}$$

After termination of the Knowledge Propagation Algorithm we return "$UNSAT$", if any knowledge core vanishes. In that case we have an unsatisfiable CSP. If no knowledge core vanishes, we define A as the subset of variables with nontrivial knowledge cores and output x_A where for $v \in A$

$$x_v = \epsilon^{-1}(\kappa\left[\{v\}\right]) \,.$$

We always have, that UCP is sound (since KP is sound).

Definition 1. *We say that UCP is complete for a CSP, if it outputs "$UNSAT$" or for no node $v \in A$ we have that at all solutions of the CSP have the same index at v.*

1.1 Forward Chaining for Horn-SAT

Definition 2 (Definite Horn-SAT)**.** *Let $\{\tau^e\left[X_e\right] \,:\, e \in \mathcal{E}\}$ be a CSP. We say it is a definite Horn-SAT problem, if each τ^e is a clause, which has exactly one positive literal.*

Forward chaining is a linear time and complete satisfiability checker on Horn Logic (a subset of Propositional Logic where each clause has at most one positive literal).

- Messages passed are the one-hot encodings of assignment to single variables (unit clauses).
- Clusters are the Horn clauses, which receive messages for their negative literals. A Cluster sends a nontrivial message to a variable, if the variable is the only unassigned literal in the clause and the clause has not been satisfied yet.

Lemma 1. *Let $\tau^{\mathcal{G}} = \{\tau^e\left[X_e\right] \,:\, e \in \mathcal{E}\}$ be a Definite Horn-SAT problem, then UCP does never output "$UNSAT$". Let further x_A be the output of UCP. Then $x_v = 1$ for each $v \in A$ and*

$$1_A \times 0_{\mathcal{V}/A} \quad \textit{and} \quad 1_A \times 1_{\mathcal{V}/A}$$

are solutions of $\tau^{\mathcal{G}}$.

Proof. First of all, when all knowledge cores are trivial, vanishing or ϵ_1, then the nonzero transformation of any contraction is trivial, vanishing or ϵ_1. Thus, since this assumption is met at the start all knowledge cores remain such during the algorithm.

At the termination of UCP we simplify the definite clauses by erasing the literals in A. This erasure results either in empty clauses or in definite clauses with at least 2 literals, since otherwise the algorithm would not have terminated. Since at least one positive and one negative literal remain, these are satisfied if all atoms are 1 and if all atoms are 0. □

Theorem 1. *UCP for Definite Horn-SAT is complete.*

Proof. From the above lemma we know that the remaining variables are in at least one solution 0 and in at least one 1. Thus they are neither entailed nor contradicted. □

1.2 UCP for 2-SAT

Definition 3 (2-SAT)**.** *Let $\{\tau^e[X_e] : e \in \mathcal{E}\}$ be a CSP. We say it is an 2-SAT problem, if each τ^e has order at most 2 and is a clause.*

2-SAT is in P and can be solved by the message passing algorithm Unit Clause Propagation (UCP).

- At each connected component of the problems factor graph, choose on variable an do the message passing scheme below with initialization by the variable on 0 and on 1.
- Messages passed are the one-hot encodings of assignment to single variables (unit clauses).
- Any clause that receives a message has only a single literal left and either gets directly trivial (if the message coincides with the literal) or assigns the remaining variable and passes further.

Lemma 2. *Let $\{\tau^e[X_e] : e \in \mathcal{E}\}$ be a 2-SAT instance. Given the outputs $x_c^{s_c}$, $s_c \in [n_c]$ and $n_c \in \{0,1,2\}$ of UCP for each connected component $c \subset \mathcal{V}$ we have*

$$\langle \{\tau^e[X_e] : e \in \mathcal{E}\} \rangle [X_{\mathcal{V}}] = \bigotimes_c \left(\sum_{s_c \in [n_c]} \epsilon_{x_c^{s_c}}[X_c] \right) .$$

Proof. For each component c of $\mathcal{G}$ we choose a start variable and choose a value $x_v \in [2]$. We then have

$$\langle \tau^c \cup \{\epsilon_{x_v}[X_v]\} \rangle [X_c] = \begin{cases} 0[X_c] & \text{if UCP returns "UNSAT"} \\ \epsilon_{x_c^{s_c}}[X_c] & \text{if UCP returns } x_c^{s_c} \end{cases} .$$

□

We use UCP for entailment/contradiction decision by checking whether for each $k \in c$ $x_k^{s_c}$ is constant. Exception: When one component is not sat, the whole 2-SAT instance is unsatisfiable and all entailment and contradiction properties hold.

Theorem 2. *UCP for 2-SAT is complete.*

Proof. We assume that 2-SAT at hand is satisfiable. Exactly when $x_v^{s_c}$ is constant for $s_c \in [n_c]$ we can write, using the above Lemma

$$\langle \{\tau^e[X_e] : e \in \mathcal{E}\} \rangle [X_{\mathcal{V}}] = \epsilon_{x_k^{s_c}}[X_k] \otimes \langle \{\tau^e[X_e] : e \in \mathcal{E}\} \rangle \left[X_{\mathcal{V}/\{v\}}\right] .$$

In case of $x_v^{s_c} = 1$ this is an equivalent criterion for entailment (respectively contradiction in case of $x_v^{s_c} = 0$). □

1.3 UCP for Tree-SAT

Definition 4 (Tree-SAT)**.** *Let $\{\tau^e[X_e] : e \in \mathcal{E}\}$ be a CSP. We say it is an Tree-SAT problem, if the factor graph is minimal connected.*

Note that we do not demand the constraint cores to be clauses in the Tree-SAT definition.

We modify UCP slightly: If any constraint tensor is decomposed into a tensor product of a basis vector of one variable and an arbitrary rest tensor, the constraint tensor is added to the queue at initialization.

Theorem 3. *The modified UCP for Tree-SAT is complete.*

Proof. Since the message-passing provides exact contractions and the messages in UCP communicate the support. □

2 With backtracking: DPLL for generic SAT

DPLL combines backtracking search with unit clause propagation (UCP). A form of message passing is applied to reduce the clauses given the current partial assignment: When guessed an assignment to a variable, the variable is removed from all clauses, either making the clause trivial (coinciding assignment) or smaller. If only one literal remains in a clause, the variable would be assigned accordingly (unit propagation). This can be directly done in the intermediate message passing scheme, or understood as the next backtracking step ("Find-Unit-Clause" in Figure 7.17 in **?**).

3 With randomization: WalkSAT

WalkSAT is a stochastic local search algorithm for SAT. It starts with a random assignment and iteratively flips variables to reduce the number of unsatisfied clauses. This can be understood as a (modified) Gibbs sampling algorithm, where the number of unsatisfied clauses is the energy function to be minimized. The modifications are:

- Selection of variable to be resampled: Typically chosen by looking at unsatisfied clauses and picking a variable that minimizes the number of newly unsatisfied clauses (whereas in Gibbs sampling one follows a fixed variable order).
- Marginal probability: Typically fixed by a mixing parameter, whereas in Gibbs sampling would be sensitive to the energy differences.