$\mathrm{tnreason}$ FOR CONSTRAINT SATISFACTION PROBLEMS

RESEARCH NOTES IN THE ENEXA PROJECT

March 26, 2025

We show, how Constraint Satisfaction Problems can be formalized by boolean tensor networks in the $\mathrm{tnreason}$ notation, and how their solution can be approached by tree search, possible in combination with monte carlo approaches.

1 Formalization

Definition 1. *Let there be a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{T}^{\mathcal{G}}$ be a tensor network of boolean constraint tensors $T^e[X_e]$ to each $e \in \mathcal{E}$, that is*

$$\mathcal{T}^{\mathcal{G}} = \{T^e[X_e] \,:\, e \in \mathcal{E}\}\,.$$

The Constraint Satisfaction Problem CSP to $\mathcal{T}^{\mathcal{G}}$ is the decision whether there is a state $x_{\mathcal{V}}$ such that

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [X_{\mathcal{V}} = x_{\mathcal{V}}] = 1\,.$$

We say the CSP is satisfiable, when there is such a state, and unsatisfiable if not.

2 Solution

2.1 Global contraction

The most obvious solution is to contract the tensor network and decide the CSP based on

$$\left\langle \mathcal{T}^{\mathcal{G}} \right\rangle [\varnothing] > 0\,.$$

This can be demanding, when having a large and densely connected tensor network.

2.2 Message passing

Local contractions instead of global provide a tradeoff between generality and efficiency. Their results can be passed to further contractions by message passing. Whenever a local contraction vanishes, the CSP is unsatisfiable. However, the converse is not true: In general, we cannot conclude that the CSP is satisfiable, when a message-passing scheme does converge without vanishing.

2.3 Guessing

To decide a CSP it is enough to construct a state $x_{\mathcal{V}}$ which satisfies all constraint tensors. One procedure is to consecutively guess the state of some variables $v \in \mathcal{V}$ and check, whether the guess can satisfy the constraint tensor.

2.4 Orchestration in a Backtracking Search Tree

We build a search tree for a constructive solution of the CSP, where each node represents a guess of a variables state. After a guess, a message passing scheme of varying complexity is performed to calculate the imediate consequences of the guess. Then an intelligent agent decides whether to

- stop the message passing scheme

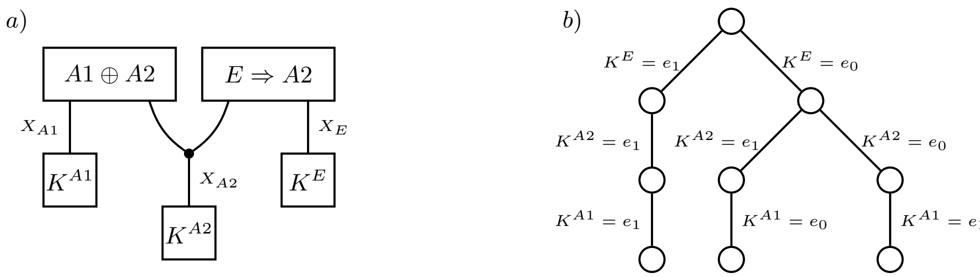

Figure 1: a) Tensor Network Representation of the Toy CSP, where two formulas $A1 \oplus A2$ and $E \Rightarrow A2$ have to be satisfied. The Knowledge Cores K contain the possible choices after a constraint propagation and are taken from $e_0, e_1, \underline{1}, \underline{0}$. b) The corresponding search tree, where the assignments to variables are guessed in the order $E, A2, A1$. At each parent with a single child, the other choice has been ruled out by constraint propagation. In our toy example with minimally connected constraints, single-core constraint propagation ensures that the guessed reasoning path remains consistent.

- increase the complexity of the scheme by extending the set of communicated variables, or the size of the contraction
- continue with a further guess of another variable
- undo the previous guess (e.g. necessary when reached a vanishing message indicating inconsistency)

3 Examples

3.1 Temporal Clue Game

- Nodes $\mathcal{V}$: To each characteristic of a murder (who, when, where, how etc) there is a categorical variable X_v with finite possibilities
- Edges $\mathcal{E}$: To each clue and accusation, there is a constraint tensor storing the possible states consistent with the clue

3.2 Graph Coloring

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we design

- a variable to each nodes $v \in \mathcal{V}$ carrying the color
- a constraint to each edge $e \in \mathcal{E}$ by differing colors, i.e. $T^{(v_1,v_2)}\left[X_{v_1}, X_{v_2}\right] = \mathbb{I}^{x_{v_1} \neq x_{v_2}}\left[X_{v_1}, X_{v_2}\right]$

3.3 Knowledge Base Satisfiability

As described in the main report.

3.4 Sudoku

As described in the main report.

4 Toy Example

As a toy example, let there be a knowledge base of atoms $A1$ and $A2$ representing two accounts to be used in an accounting proposal, and E represents incoming invoices. The constraints are that exactly one of $A1$ or $A2$ is booked, that is $X_{A1} \oplus X_{A2}$.