Qualitative Constraint Network: Reasoning

Travaux dirigés du cours *Contraintes* (HAI910I)

Michael Sioutis

27 Nov 2023

1 Graph Concepts

As a reminder from the course, we use $G(\mathcal{N})$ to denote the *constraint graph* (V, E) of a QCN $\mathcal{N} = (V, C)$, where $\{v, v'\} \in E$ iff $C(v, v') \neq \mathsf{B}$. In other words, the constraint graph of a QCN corresponds to the part of the QCN that involves constraints, i.e., non-universal relations.

Let us now recall the definition of a chordal graph:

Definition 1. *Let $G = (V, E)$ be an undirected graph, then G is* chordal *(or* triangulated*) if every cycle of length greater than* 3 *has a* chord*, which is an edge connecting two non-adjacent nodes of the cycle*

Next, we view the relationship between a chordal graph and a tree decomposition:

Theorem 1. *A graph G is chordal if and only if it has a tree decomposition $(T, \{X_1, \ldots, X_n\})$ where cluster X_i is a clique of G for every $i \in \{1, \ldots, n\}$.*

A tree decomposition is formally defined as follows:

Definition 2. *A* tree decomposition *of a graph $G = (V, E)$ is a tuple (T, X) where $T = (I, F)$ is a tree and $X = \{X_i \subseteq V \mid i \in I\}$ is a collection of clusters (subsets of V) that satisfies the following conditions:*

1. *For every $v \in V$ there is at least one node $i \in I$ such that $v \in X_i$.*
2. *For every $(u, v) \in E$ there exists a node $i \in I$ such that both $u, v \in X_i$.*
3. *Let i_1, i_2, i_3 be three nodes in I such that i_2 lies on the (unique) path between i_1 and i_3 in T. Then, if $v \in V$ belongs to both X_{i_1} and X_{i_3}, v must also belong to X_{i_2}.*

Let us view the example presented in Figure 1. In the upper part of the figure we can view a graph $G = (V, E)$ (which can correspond to the constraint graph $G(\mathcal{N}) = (V, E)$ of a QCN $\mathcal{N}$). For the moment, we consider only the solid edges to be part of G and we disregard the dashed edges $\{3, 4\}$ and $\{4, 5\}$.

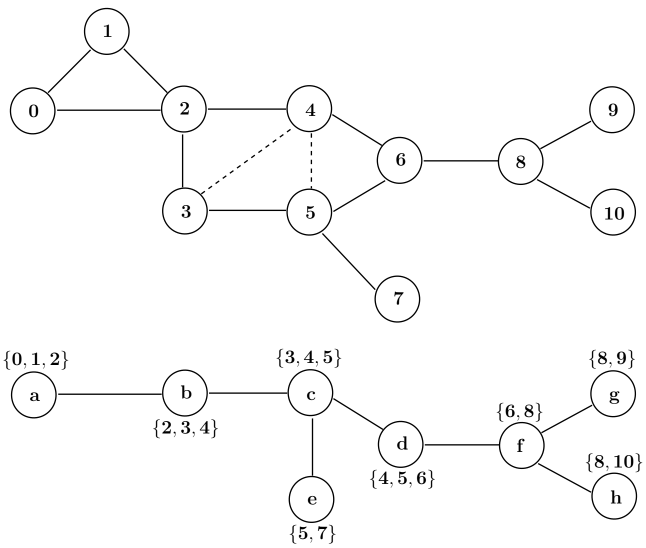

Figure 1: A graph (upper part) and its tree decomposition (lower part)

A tree decomposition of G comprises a tree $T = (I, F)$ with the set of nodes $I = \{a,b,c,d,e,f,g,h,i\}$ and a cluster X_i for every node $i \in I$ of that tree, as shown in the lower part of the figure, e.g., $X_a = \{0, 1, 2\}$. The first two conditions in our definition state that G is the union of the subgraphs induced by X_i, for every $i \in I$. The third condition implies that these subgraphs are organized roughly like a tree. Now, if we include the dashed edges $\{3, 4\}$ and $\{4, 5\}$ in G, then the clusters of the tree decomposition become *cliques*, namely, sets of vertices such that every two vertices in a set are connected by an edge; it is easy to verify that G is a chordal graph (as ensured by Theorem 1).

2 Exercices

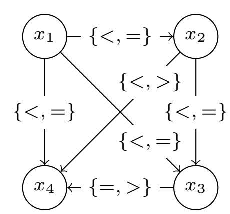

Figure 2: A QCN of Point Algebra

Exercice 1. *Consider the* QCN *of Point Algebra in Figure 2, and answer the following questions:*

1. *Check whether the* QCN *is $\diamond$-consistent*[1] *and detail your steps; if it is not $\diamond$-consistent, apply algebraic closure under $\diamond$-consistency to make it $\diamond$-consistent and detail your steps.*
2. *Check whether the* QCN *is satisfiable and detail your steps.*
3. *Check whether the* QCN *is minimal (recall the definition of a minimal* QCN *from the course and the TD of last week) and detail your steps.*

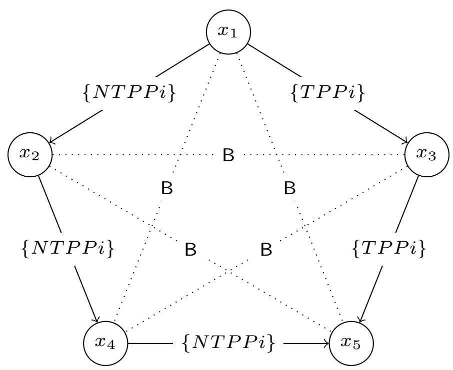

Figure 3: A QCN of RCC8

Exercice 2. *Consider the* QCN *of* RCC8 *in Figure 3, and answer the following questions:*

1. *Provide the constraint graph of the* QCN.
2. *Check whether the constraint graph of the* QCN *is chordal and detail your steps; if it is not chordal, introduce a set of chords to make it chordal.*
3. *Check whether the* QCN *is $\underset{G}{\diamond}$-consistent w.r.t the chordal graph G of the previous step and detail your steps; if it is not $\underset{G}{\diamond}$-consistent, apply algebraic closure under $\underset{G}{\diamond}$-consistency to make it $\underset{G}{\diamond}$-consistent and detail your steps.*
4. *Check whether the* QCN *is satisfiable and detail your steps; specifically, detail the conditions and the procedures that allow you to decide if the* QCN *is satisfiable.*
5. *Produce a tree decomposition of the chordal graph G that you obtained in step 2, where every cluster is a clique of G.*[2]

1. Based on our definition of the constraint graph of a QCN in the beginning of Section 1, the constraint graph here is the graph (V, E) where:

[1] A a reminder, $\diamond$-consistency is $\underset{G}{\diamond}$-consistency where G is the complete graph on the set of variables of a given QCN.

[2] This task serves to help you visualize how the different parts of the QCNs are patched together.

- $V = \{x_1, x_2, x_3, x_4, x_5\}$;
- and $E = \{\{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_4\}, \{x_3, x_5\}, \{x_4, x_5\}\}$.

Please note that a constraint graph is a undirected graph, i.e., $\{x_i, x_j\}$ is the same edge as $\{x_j, x_i\}$.

2. Based on the definition of a chordal graph in Section 1, the constraint graph of the QCN, viz., the graph (V, E) in the previous step, is not chordal, as variables $\{x_1, x_2, x_3, x_4, x_5\}$ form a cycle of length 5. There are many ways to make (V, E) chordal. In every case, we add a chord, i.e., an edge connecting two non-adjacent nodes (variables) of the cycle, and we check whether there still exists a cycle of length greater than 3. Here, we can start by introducing the chord $\{x_1, x_4\}$. Our graph will still not be chordal, as variables $\{x_1, x_3, x_4, x_5\}$ will be forming a cycle of length 4 in $(V, E \cup \{\{x_1, x_4\}\})$. If we also introduce the chord $\{x_1, x_5\}$, the graph will become chordal. We call this graph G. Specifically, $G = (V, E \cup \{\{x_1, x_4\}, \{x_1, x_5\}\})$. Please note that, by definition, a complete graph is also a chordal graph; so, one (bad) way of making a graph chordal is to make it complete by introducing all possible edges (i.e., by saturating the graph).

3. We have already seen how to check for and apply $\overset{\diamond}{G}$-consistency in the previous TD, so we will not detail this step. It suffices to say that in the end of the algebraic closure procedure, which will produce the unique largest[3] $\overset{\diamond}{G}$-consistent sub-QCN of our QCN $\mathcal{N} = (V, C)$, the following updates will happen:

- $C(x_1, x_4)$ will be refined to $\{NTPPi\}$;
- and $C(x_1, x_5)$ will be refined to $\{TPPi, NTPPi\}$.

Clearly, these refinements are particular to the chordal graph that we produced in the previous step, and if we had produced a different chordal graph, we would have refined different constraints, possibly.

4. With respect to RCC8, and from our course notes, we can verify all of the following properties:

- our QCN is defined over a distributive and, hence, tractable subclass of relations of RCC8;
- all of the (maximal) tractable subclasses of relations of RCC8 have *patchwork*;
- our QCN is $\overset{\diamond}{G}$-consistent and not trivially inconsistent (i.e., it does not contain the empty set relation, viz., $\emptyset$);
- and the chordal graph G is a supergraph of the constraint graph of our QCN, i.e., $G \supseteq G(\mathcal{N})$.

From the proposition in slide 84 of our course notes, we can deduce that the QCN is satisfiable.

[3] w.r.t $\subseteq$

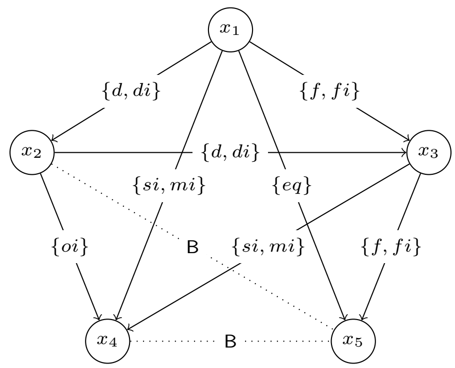

Figure 4: A QCN of Interval Algebra

Exercice 3. *Consider the* QCN *of Interval Algebra in Figure 4, and answer the following questions:*

1. *Check whether the constraint graph of the* QCN *is chordal and detail your steps; if it is not chordal, introduce a set of chords to make it chordal.*

2. *Check whether the* QCN *is* $\diamond_G$*-consistent, where* G *is the chordal graph that you obtained in the previous step, and detail your steps; if it is not* $\diamond_G$*-consistent, apply algebraic closure under* $\diamond_G$*-consistency to make it* $\diamond_G$*-consistent and detail your steps.*

3. *Check whether the* QCN *is satisfiable and detail your steps; specifically, detail the conditions and the procedures that allow you to decide if the* QCN *is satisfiable.*

3 Project (Optional)

Extend your implementation from the previous TD with methods to enforce $\diamond_G$-consistency and perform backtracking search. Basically, you will implement the two algorithms that were presented in the course. Note that you will have to make provision for a graph G to be provided as additional input to your solver.