Scribe Notes: 2/27/2013
Presenter: Dr. Berthe Choueiry
Scribe: Tony Schneider
Reading: Chapter 8 of Dechter's Textbook

Chapter 9 Wrap-up

Slide 28

Time complexity of CTE: $O(deg\cdot(n + N)\cdot d^{w^*+1})$
The number of variables in a cluster in the tree decomposition is bounded by w^*+1. (In the largest cluster, the deepest variable *x* has at most w^* parents. Thus, the size of the largest cluster is the values of the induced width, w^*, plus 1, for *x* itself). Therefore, the size of a problem in a given cluster is bounded by d^{w^*+1}.

According to the slides, $n+N$ is the number of clusters (*N*) plus the number of variables / conditional probability tables (*n*). This is an error, as a conditional probability table describes a relation, not a variable. Indeed, Robert pointed out that the book reframes the time complexity as $O(deg\cdot(r+N)\cdot k^{w^*})$, where *r* is the number of constraints. Note also that the book uses w* to denote the treewidth, which is equal to the induced width+1. The (r+N) term is then the number of constraints that are processed by a given node, summed over every node in the decomposition.

The time complexity relies on the degree because it assumes every time a cluster receives a message, the cluster re-computes the join of all its relations. Note that this fact conflicts with the algorithm, which states that the cluster computes the join only once, as soon as all but one of its neighbors have sent their messages.

Space complexity of CTE: $O(N\cdot d^{sep})$
Here, *N* is the number of clusters, *d* the size of the largest domain, and *sep* the size of the largest separator (i.e., number of variables common to two clusters). d^{sep} gives the largest message that needs to be stored and passed. By construction of the algorithm (i.e., it requires a tree of clusters), each cluster will have at most 1 parent, so only *N* separators need to be stored.

Time and space complexity by hyperwidth: $O(N\cdot t^{2hw})$ time and $O(N\cdot t^{hw})$
Note: the slides give the space complexity as $O(N\cdot t^{tw})$, but here t^{tw} should be t^{hw}

The size of a constraint can be bound by the number of tuples (*t*) rather than by d^{arity} (which is a pessimistic bound). The hyperwidth (*hw*) is the largest number of constraints in a cluster. In the worst case, all the constraints in the cluster will be in the separator. Each constraint will be size *t*, so in the worst case a separator's message will be of size be t^{hw}.

Additionally, each constraint will have at most t tuples, so a join operation within a given cluster will require O(t^{hw}) operations, after which the resulting relation of size O(t^{hw}) will be recorded. The resulting relation must then be joined with each neighbor (also of size O(t^{hw})) , requiring O(t^{2hw}) operations per neighbor. If there are N clusters in the decomposition, the overall time complexity will be O($N \cdot t^{2hw}$).

Slide 29

The main idea here is that the process taken to create a join-tree produces a special, restricted form of a tree decomposition (i.e., one that is derived by triangulation, which is a simple process). Other, more complex, methods can derive better tree decompositions. The topic will be discussed in more depth in the presentation of Daniel Dobos. In the slide, figure (a) can be created by JTC while figure (b) cannot. Thus, some tree-decompositions are more restrictive than others (i.e., "there are tree decompositions that cannot be derived by triangulation" [Dechter's book, page 262]).

Slide 30

This slide frames adaptive consistency in terms of message passing (look to slide 33 for an example).

- Each bucket is a cluster of the tree decomposition
- Each cluster is connected to the bucket that receives its message.
- The relations in the bucket are those placed in the bucket by the construction process (i.e., they are over the bucket's variable and have not appeared at deeper levels in the ordering).
- The set of variables in the bucket is the union of the scopes of the relations in the bucket.

Questions over Chapter 8

Tony's Questions

Question: *Why is the upper bound for enforcing i-consistency with relational m-consistency 2^i (depending on the number of constraints in the network)?*

Answer: Reminders:

- *i-consistency*: every consistent assignment of i-1 variables can be extended to any i^{th} variable. So, we may have to generate constraints of arity i-1.
- *Relational m-consistency* (RmC): Look at every combination of *m* relations and consider S (the union of their scopes). Every consistent assignment of |S|-1 variables can be extended to the S^{th} variable. So, we have to generate constraints of arity |S|-1.

Why 2^i? We need to consider the m relations such that the union of their scopes covers i variables. Because each variable may or may not be in one of the m

relations, and we want to cover every combination of i variables, we need to generate 2^i constraints. Thus, m may have to go up to 2^i in the worst case. [*Can somebody provide a better explanation? Please contribute*.]
Note: it's too expensive to use RmC in practice. In fact, to the best of our knowledge, none of the relational consistencies in the book have been implemented because they require generating new relations. Yet, the approach is invaluable because it provides us with the terminology and the vision to discuss the topic and contribute to its advancement.

Question: *Is the only difference between relational m-consistency and variable-based i-consistency on the dual-graph that is mentioned in the exercise the filtering involved (in reference to exercise 8.9.3 on page 242)? If so, are there other differences?*

Answer: Exercise 8.9.3 asks to compare m-consistency on the dual graph and RmC.

- m-consistency *on the dual graph* generates a *single* constraint for every combination of m-1 dual variables. If the problem has e constraints, we will generate $\binom{e}{m-1}$ new constraints. The scope of each new constraint will be the union of the scopes of the m-1 constraints.
- RmC generates, for a given combination of m constraints whose scopes' union is a set S of *variables*, $\binom{|S|}{|S|-1} = |s| - 1$ new constraints whose scope is 'S but one variable.' It also needs to generate that many constraints for each combination of m constraints, that is $\binom{e}{m}$ combinations.

It is quite likely (for a large e and small m) that:

- The number of constraints generated by m-consistency on the dual graph be much smaller than that generated by RmC.
- Further, the arity of the constraints generated by m-consistency on the dual graph is smaller than that generated by RmC.

[*Can somebody provide a better explanation? Please contribute*.]

Follow up question: *Will one filter more than the other?*
Answer: It's likely that RmC is stronger (the constraint generated by i-consistency on the dual graph could be a subset of the constraint generated over the S-1 variables in RmC).

Robert's Questions

Question: *Why does the introduction to chapter 8 seemingly discard all the algorithms in chapter 3 (calls them "irrelevant") for non-binary constraints when GAC is among them?*

Answer: The text makes a strong statement here to emphasize the introduction of relation-based consistency as an 'independent' concept from variable-based consistency, but not saying that GAC is irrelevant. Robert noted in his Piazza question (as a follow up) that she later frames GAC as a relational consistency (i.e., R(1,1)C and GAC are the same), so the intent of the book's statement was likely misinterpreted.

Question: *In section 8.4.1, it says "Since relational 2-consistent networks with bivalued domains are globally consistent (Theorem 8.2), the resolution algorithm, enforcing relational 2-consistency, generates a globally consistent, and therefore backtrack-tree, representation." (Last paragraph of page 226.) Do you know if anyone solves SAT problems using relational 2-consistency? Or is that a relatively unexplored technique (as SAT people do not typically apply consistency techniques)?*

("Indeed, Unit-Propagation, the algorithm presented in Figure 3.16, applies relational arc-consistency for SAT." Therefore, at least relational 1-consistency is frequently used in the SAT community.)

Answer: The problem with relational 2-consistency is that for every 2 constraints, to guarantee that every partial solution can be extended, every constraint of scope S-1 needs to be generated. Then that new constraint also needs to be considered, and more constraints generated, etc., which results in an explosion of added relations. It is unlikely to be used in SAT. *Addition: one may wonder whether it would correspond to no-good learning in SAT.*

This seems to be more of an a posteriori statement: if there's a problem that is already relational 2-consistent, and it happens to have bi-valued domains, then it is known to be globally consistent. The bottom line: it is largely just a theoretical result, not something used in practice.

Daniel's Questions

Question: *If intersection of the scopes of m relations is empty, are they considered to be relational m-consistent?*

Answer: Yes.

Question: *Why is it that relational m-consistency does not guarantee all relational consistencies less than m?*

Answer: Consider two tuples t1 and t2 in a relation of arity |s|-1 generated by m-consistency. The combination of a portion of t1 and a portion of t2 that are not over the same constraints is not necessarily consistent. Although those portions are allowed by RmC they are not necessarily allowed by other relational consistencies less than m.

Question: *Should relations be projected over all subsets of the scope in order to avoid situations like example 8.5?*

Answer: No. Because projection is not enough: it loses information. We need to generate new constraints that prevent inconsistent combinations.