Explorations into Belief State Compression

Ali Elhalawaty[1] and Haythem Ismail[2,1]

[1] German University in Cairo, Cairo, Egypt
[2] Cairo University, Cairo, Egypt

Abstract. We explore variants of belief state compression which is an intuitive, weak form of knowledge base size reduction. It differs from previous approaches in at least three aspects. First, it takes its objects to be support-structured sets of unconstrained, rather than flat sets of syntactically-constrained, logical formulas. Second, classical notions of minimality and redundancy are replaced by weaker, resource-bounded alternatives based on the support structure. Third, in "lossy" variants of compression, the compressed knowledge base logically implies only a practically-relevant subset of the original knowledge base.

1 Introduction

Several factors affect the complexity of automated reasoning and knowledge base maintenance. One obvious, and long-recognized, such factor is the sheer size of the knowledge base [6,16,18,1]. The size of the knowledge base is particularly crucial for logic-based *acting* agents which roam the world for extended periods of time with their knowledge bases continually expanding by sensory inputs. Such inputs are often not simply added to the knowledge base; they also result in expanding the knowledge base by conclusions about the environment that follow from them and that may be crucial for directing future timely actions of the agent [17,9]. Many approaches to knowledge base size reduction, such as knowledge base minimization and redundancy elimination, have been discussed in the literature [8,11,7,1]. Such approaches typically either impose some restrictions on the structure of the knowledge base or demand possible drastic changes to it, which may impede their applicability to many practical reasoning systems. Herein, we investigate variants of an approach to knowledge base size reduction which we dub "belief state compression." We take a belief state to be a set of logical formulas structured by a "support" relation, akin to classical (assumption-based) reason maintenance systems [4,2].[3] Loosely speaking, compression only requires the information in the compressed belief state to be (a possibly improper) part of the information in the original one. The notion of "information" here is, generally, *not* the classical notion of the set of logical consequences. Rather, we adopt a resource-bounded notion of information, limiting it to the set of formulas which were actually derived.

[3] While it may be true that many knowledge-based systems lack a reason maintenance component, knowledge-based acting agents mandate some way of handling inconsistency (cf. [9]). Moreover, interest in reason maintenance never faltered and has recently been strengthened [10,13, for example].

2 Belief States

We assume a language L with a classical consequence operator Cn. It will help, as is common, to construe formulas of L as representing (possible) beliefs of a cognitive agent. Henceforth, we refer to this agent as $\mathcal{A}$.

Definition 1. *A belief state is a triple* $\mathbb{S} = \langle K, B, \sigma \rangle$ *where* $K \subset L$ *is finite,* $B \subseteq K$ *and* $\sigma : K \rightarrow 2^{2^{B}-\emptyset}$*, such that*

1. $\phi \in \mathrm{Cn}(s)$*, for every* $s \in \sigma(\phi)$*;*
2. $\phi \notin s$*, for every* $s \in \sigma(\phi)$*; and*
3. $\sigma(\phi) \neq \emptyset$*, whenever* $\phi \in K - B$*.*
   The size $|\mathbb{S}|$ *of* $\mathbb{S}$ *is defined as* $|K| + \sum_{\phi \in K} \sum_{s \in \sigma(\phi)} |s|$*.*

Intuitively, K is the set of all (explicit) beliefs of $\mathcal{A}$ and B is a distinguished subset of *base* beliefs. Members of B are those formulas which $\mathcal{A}$ came to believe by some means other than logical reasoning. Typically, they are formulas which result from sensory input or from communication with other agents. We often refer to members of B as *hypotheses*. Formulas in $K - B$ are believed only as a result of $\mathcal{A}$'s doing some reasoning. Hence, σ provides, to each member of K a set of *supports*. Each support of a formula ϕ is a set of hypotheses that were used to derive ϕ.

Intuitively, belief state compression reduces the size of the belief state while not adding any new information to it.

Definition 2. *Let* $\mathbb{S} = \langle K, B, \sigma \rangle$ *be a belief state. A compression* $\mathbb{S}'$ *of* $\mathbb{S}$ *is a belief state* $\langle K', B', \sigma' \rangle$ *with*

1. $K' \subseteq K$*;*
2. *for every* $\phi \in K'$*,* $\sigma'(\phi) \subseteq \sigma(\phi)$*; and*
3. $|\mathbb{S}'| < |\mathbb{S}|$*.*

Instead of the classical, strong notions of irredundancy and minimality [8,11], we adopt weaker, more practical ones based on the support relation.

Definition 3. *If* $\mathbb{S}$ *is a belief state and* $H \subseteq L$*, we say that* $\phi \in K$ *is supported by* H *in* $\mathbb{S}$ *if (i)* $\phi \in H$ *or (ii) there is some* $s \in \sigma(\phi)$ *with every* $\psi \in s$ *supported by* H *in* $\mathbb{S}$*. Hence, a compression* $\mathbb{S}'$ *of* $\mathbb{S}$ *is* lossless *if every* $\phi \in B$ *is supported by* B' *in* $\mathbb{S}$*, otherwise it is* lossy*. If* $\mathbb{S}'$ *is lossless, then it is* irredundant *(respectively,* minimal*) if there is no lossless compression* $\mathbb{S}''$ *of* $\mathbb{S}$ *with* $B'' \subset B'$ *(respectively,* $|\mathbb{S}''| < |\mathbb{S}'|$*).*

In what follows, we explore some variants of compression. We start with a brief discussion of a simple, obvious variant.

3 *K*-Only Compression

A K-only compression of belief state $\mathbb{S}$ is a compression $\mathbb{S}'$ with $B' = B$ and $\sigma' = \sigma|_{K'}$. Such compressions are clearly lossless. An extreme case of K-only compression is the compression $\mathbb{S}_{K_0} = \langle B, B, \sigma_B \rangle$ which only retains hypotheses and gives up all derived formulas. This compression operator can be efficiently implemented (in time $O(|\mathbb{S}|)$) and may result in significant size reduction if agent $\mathcal{A}$ is an introspective agent whose set of beliefs is dominated by inferred beliefs rather than beliefs with independent standing. This, of course, may come with a heavy price. For it is possible that beliefs in B are mostly useless and the given up derived beliefs are the interesting ones. Although, since $\mathbb{S}_{K_0}$ is lossless, all retracted beliefs may be re-derived, this will still be on the expense of $\mathcal{A}$'s (possibly valuable) time.

4 *B*-Compression

A B-compression is a compression which necessarily changes B. The exploration to follow of B-compression relies heavily on the notion of a *base-support graph*. Henceforth, let $\Sigma(\mathbb{S}) = \cup_{\varphi \in K} \sigma(\varphi)$, where $\mathbb{S}$ is a belief state.

Definition 4. *For a K_0-compressed belief state $\mathbb{S} = \langle B, B, \sigma \rangle$, a base-support graph is a bipartite graph $\mathcal{G}(\mathbb{S}) = (V_1, V_2, E)$, where*

- $V_1 = B$
- $V_2 = \Sigma(\mathbb{S})$
- *for $n_1 \in V_1$ and $n_2 \in V_2$,*
  1. $(n_1, n_2) \in E$ *if and only if* $n_1 \in n_2$
  2. $(n_2, n_1) \in E$ *if and only if* $n_2 \in \sigma(n_1)$

In the sequel, we refer to nodes in V_1 as *hyp* nodes (or just *hyps*) and to nodes in V_2 as *support* nodes (or just *supports*). Figure 1 is an example of the base-support graph $\mathcal{G}(\mathbb{S})$, with $\mathbb{S} = \langle \{1,2,3\}, \{1,2,3\}, \sigma \rangle$ where $\sigma(1) = \{\{2\}\}$, $\sigma(2) = \{\{1\}\}$, and $\sigma(3) = \{\{1,2\}\}$. (Hyp nodes are circular and generically denoted by numerals while support nodes are rectangular and denoted by letters.)

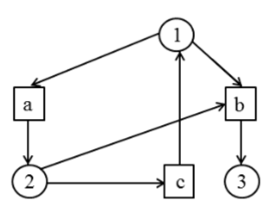

Fig. 1. A base-support graph

Given the construction of the base-support graph, the following properties immediately follow.

Observation 1 *Let $\mathcal{G}(\mathbb{S}) = (V_1, V_2, E)$ be a base-support graph.*

1. *For every $n \in V_2$, $deg^-(n) > 0$ and $deg^+(n) > 0$.*
2. *For every $n_1, n_2 \in V_2$, if $\{n|(n, n_1) \in E\} = \{n|(n, n_2) \in E\}$, then $n_1 = n_2$.*
3. *E is asymmetric. (That is, if $(n_1, n_2) \in E$, then $(n_2, n_1) \notin E$.)*

4.1 Trimming

To conveniently discuss B-compression operators, we introduce the following piece of notation. If $\mathcal{G} = (V_1, V_2, E)$ is a base-support graph with $N \subseteq V_1$, then $\mathcal{G} - N$ is the sub-graph of $\mathcal{G}$ induced by $(V_1 - N) \cup (V_2 - \{n \in V_2|$ if $(n', n) \in E$ or $(n, n') \in E$, then $n' \in N\})$. Simply, $\mathcal{G} - N$ is the sub-graph of $\mathcal{G}$ with N removed from V_1 and with every support node that has edges only to nodes in N or edges only from nodes in N removed from V_2 (to save the first clause of Observation 1). Our first B-compression operator, the B *trimming* operator B_{Trim}, is based on algorithm `GTrim` (see Algorithm 1) which operates on a directed bipartite graph (V_1, V_2, E) with $deg^-(n) \neq 0$, for every $n \in V_2$. (Possibly a base-support graph.)

Algorithm 1: Base trimming algorithm

Function `GTrim`(G):
if *there is* $n \in V_1$ *with* $deg^+(n) = 0$ *and* $deg^-(n) \neq 0$ **then**
| return `GTrim` $(G - \{n\})$
end
return G

Figure 2 shows the result of trimming the graph in Figure 1. Node 3 is the only node satisfying the condition in Algorithm `GTrim` and is, hence, removed together with node b. This results in no nodes satisfying the condition and the algorithm terminates.

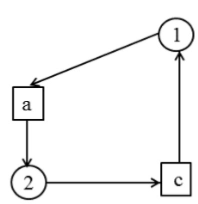

Fig. 2. The graph resulting from applying Algorithm `GTrim` on the graph of Figure 1

It is easy to show that trimming is an instance of B-compression. If G is the base-support graph of some K_0-compressed belief state $\mathbb{S}$ and $\texttt{GTrim}(G) = (V_1, V_2, E)$, then the *trimming* of $\mathbb{S}$, denoted $Tr(\mathbb{S})$ is the structure $\langle V_1, V_2, \sigma\rangle$, where, for every $n_1 \in V_1$ and every $n_2 \in V_2$ with $(n_2, n_1) \in E$, $n_2 \in \sigma(n_1)$.

Corollary 1 *If $\mathbb{S}$ is a K_0-compressed belief state, then $Tr(\mathbb{S})$ is a belief state. Moreover, if $\mathbb{S} \neq Tr(\mathbb{S})$, then $Tr(\mathbb{S})$ is a lossless compression of $\mathbb{S}$.*

Trimmed base-support graphs have an interesting property which we prove next. First, a piece of notation. For a graph G, we say that a path P in G ends (respectively, begins) with a cycle if $P = n_1, \ldots, n_k$ with $n_i = n_k$ (respectively, $n_i = n_1$) for some $1 \leq i < k$ (respectively, $1 < i \leq k$). In the sequel, a node n in graph G is G^+-trapped (respectively, G^--trapped) if (i) $deg^-(n) = 0$ (respectively, $deg^+(n) = 0$) or (ii) n is on a path in G that ends (respectively, begins) with a cycle. If n is G^+-trapped and G^--trapped, then it is G-trapped.

Proposition 1 *If $G = (V_1, V_2, E)$ is a directed bipartite graph with $deg^-(n) \neq 0$, for every $n \in V_2$, then $\texttt{GTrim}(G)$ is a graph $G' = (V_1', V_2', E')$ where every $n \in V_1'$ is G'^+-trapped.*

Proof. We prove the proposition by induction on $|V_1|$.

Base Case. If $|V_1| = 0$, then, trivially, no nodes satisfy the condition of the if-statement and $\texttt{GTrim}(G) = G$. Hence, $|V_1'| = 0$ and the proposition trivially holds.

Induction Hypothesis. If $G = (V_1, V_2, E)$ is a directed bipartite graph with $|V_1| = k$, for some $k \in \mathbb{N}$, and $deg^-(n) \neq 0$, for every $n \in V_2$, then $\texttt{GTrim}(G)$ is a graph $G' = (V_1', V_2', E')$ where every $n \in V_1'$ is G'^+-trapped.

Induction Step. Suppose that $G = (V_1, V_2, E)$ is a directed bipartite graph with $|V_1| = k + 1$ and $deg^-(n) \neq 0$, for every $n \in V_2$. We consider two cases.

1. There is a node $n \in V_1$ with $deg^- \neq 0$ and $deg^+ = 0$. Thus, $\texttt{GTrim}(G) = \texttt{GTrim}(G - \{n\}) = G'$. By construction, $G - \{n\}$ contains exactly k hyp nodes. Hence, by the induction hypothesis, every node in V_1' is G'^+-trapped.
2. There is no node $n \in V_1$ with $deg^- \neq 0$ and $deg^+ = 0$. Hence, $\texttt{GTrim}(G) = G$. Consider an arbitrary $n \in V_1$. If $deg^-(n) = 0$, then n is G^+-trapped. Otherwise, $deg^-(n) \neq 0$ and $deg^+(n) \neq 0$. Thus, there must be a path P that starts with n and on which every node n' has $deg^-(n') \neq 0$ and $deg^+(n') \neq 0$. Since G is finite, it follows by the pigeonhole principle that there is at least one node r which appears on P at least twice. Hence, the sub-path of P which ends with the second occurrence of r ends with a cycle. Consequently, n is G'^+-trapped.

As an immediate corollary, trimming sometimes gives rise to a minimal compression. In what follows, for a belief state $\mathbb{S} = \langle B, K, \sigma\rangle$, we refer to the set $\{\phi \in B | \sigma(\phi) = \{\}\}$ as $max(B)$.

Corollary 2 *If $\mathbb{S} = \langle B, B, \sigma\rangle$ is a K_0-compressed belief state such that $\mathcal{G}(\mathbb{S})$ is acyclic, then $Tr(\mathbb{S}) = \langle max(B), max(B), \sigma|_{max(B)}\rangle$ which is a minimal compression of $\mathbb{S}$.*

Another variant of trimming is what we call *reverse trimming*. Again, we start with a transformation of the base-support graph. In particular, if G is a base-support graph, then the reverse-trimming of G is given by $\texttt{RGTrim}(G) = (\texttt{GTrim}(G^R))^R$, where G^R is the reverse of G (i.e., the graph resulting from reversing all edges of G). Similar to trimmed graphs, a reverse-trimmed graph has a distinctive structure: Every node in a reverse-trimmed graph is G'^--trapped. The similarity ends right here, however. In particular, a reverse-trimmed base-support graph is not itself necessarily a base-support graph. Nevertheless, one can construct a belief state from the reverse-trimmed graph.

Definition 5. *Let G be the base-support graph of a K_0-compressed and trimmed belief state $\mathbb{S} = \langle B, B, \sigma\rangle$ and let $\texttt{RGTrim}(G) = (V_1, V_2, E)$. The reverse-trimming of $\mathbb{S}$, denoted $RTr(\mathbb{S})$, is a triple $\langle B', B', \sigma'\rangle$ where*

1. *$B' = V_1 \cup max(B)$;*
2. *$\sigma' : B' \to 2^{2^{B'}-\emptyset}$ is such that, for every $\phi \in B'$, $\sigma'(\phi) = \{H | H$ is a smallest subset of B' such that ϕ is supported by H in $\mathbb{S}\}$.*[4]

Corollary 3 *If $\mathbb{S}$ is a K_0-compressed and trimmed belief state, then $RTr(\mathbb{S})$ is a belief state. Further, if $\mathbb{S} \neq RTr(\mathbb{S})$, then $RTr(\mathbb{S})$ is a lossless compression of $\mathbb{S}$.*

It should be clear that, if $\mathbb{S}$ is a K_0-compressed belief state, then every node in $G = \texttt{RGTrim}(\texttt{GTrim}(\mathcal{G}(\mathbb{S})))$ is G-trapped.

4.2 Towards Minimal B-Compression

Trimming, reverse-trimming, and their composition do not secure a minimal (or even an irredundant) compression if the base-support graph is cyclic. It is to the problem of constructing minimal, irredundant, or tighter-than-trimmed compressions that we now turn.

Definition 6. *Let $\mathcal{G}(\mathbb{S}) = (V_1, V_2, E)$ be a base-support graph of some K_0-compressed, trimmed, and reverse-trimmed belief state $\mathbb{S}$ and let $N \subseteq V_1$. The closure of N in $\mathcal{G}(\mathbb{S})$, $N^*_{\mathcal{G}(\mathbb{S})}$,*[5] *is the largest set such that there is a sequence (H_i, S_i), $i = 1 \ldots n$ for some $n \in \mathbb{N}$, where*

1. *$H_0 = N$;*

[4] Instead of constructing σ' this way, a less expensive, but less tight, construction may be given thus:

$$\sigma'(\phi) = \{n_2 \cup max(B) | (n_2, \phi) \in E\}$$

[5] We drop the subscript if it is clear from the context.

2. $S_i = \{s \in V_2 | s \subseteq H_{i-1}\}$
3. $H_i = H_{i-1} \cup \{h \in V_1 | (s,h) \in E \text{ for some } s \in S_i\}$
4. $N^*_{\mathcal{G}(\mathbb{S})} = H_n$

The problem of *minimal B-compression* can be described as a graph problem.

- **Instance:** A base-support graph $\mathcal{G}(\mathbb{S}) = (V_1, V_2, E)$ of some K_0-compressed, trimmed, and reverse-trimmed belief state $\mathbb{S}$.
- **Question:** What is a minimum set $N \subseteq V_1$ such that $N^* = V_1$?

This problem is formally identical to the optimization variants of the minimum axiom-set problem [5], the candidate-key problem in database [12], and the inverse scope problem in bio-informatics [14]. These problems are **NP**-complete, which means that finding efficient, exact algorithms for minimal B-compression is unlikely. Candidate exact algorithms, mostly based on a breadth-first-search step, are discussed in the database literature in the context of candidate keys [12,15,20]. Herein, we introduce an approximation algorithm, which provides promising results. In a nutshell, the algorithm uncovers the smallest cycles of dependency in the base-support graph and selects an arbitrary node from each. We show below that the selection thus constructed is guaranteed to have the entire set of hyp nodes as its closure. Graphically, the "smallest cycles of dependency" are what are known as *chordless cycles* [21,3,19]. A chordless cycle in a graph G is an induced sub-graph of G which is a cycle. Equivalently, it is a cycle C in G such that no edge connecting two nodes of C is not itself in C. (If it exists, such an edge is called a *chord*.) To illustrate how the algorithm works, consider Figure 3 which displays a possible (perhaps trimmed and reverse-trimmed) base-support graph. The graph contains two chordless cycles $(1, a, 2, b)$ and $(2, c, 3, d)$. Considering the set of hyps on each chordless cycle, we have the two sets $\{1, 2\}$ and $\{2, 3\}$. We now solve a hitting-set problem and select a node from each set while minimizing the size of the selection. Thus, we select $\{2\}$, and, given Definition 6, we have $\{2\}^* = \{1, 2, 3\}$. Although, in this example, we get a minimal compression of the belief state, the approach often yields redundant compressions. For example, in a belief state with three hyps where each hyp individually supports the other two, the base-support graph will have three chordless-cycles, and our algorithm will select no less than two hyps. Clearly, however, a single hyp is sufficient.

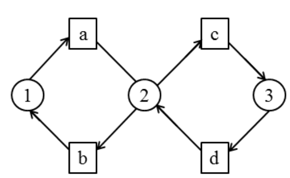

Fig. 3. A base-support graph with two chordless cycles

Though not every hyp selected by the algorithm is necessary, the selected set is sufficient. To reach that conclusion, we first make the following simple observation.

Observation 2 *Any cycle of a graph G has a chordless sub-cycle.*

In the sequel, if G is a base-support graph, let $\mathcal{C}(G) = \{N | N$ is the set of hyp nodes on a cycle in $G\}$; similarly, let $\mathcal{CH}(G) = \{N | N$ is the set of hyp nodes on a chordless cycle in $G\}$. A hitting set of a family of sets $\mathcal{F}$ is a set H such that $H \cap S \neq \{\}$ for every $S \in \mathcal{F}$.

Lemma 1. *Let $G = (V_1, V_2, E)$ be a trimmed and reverse-trimmed base-support graph. If H is a hitting set of $\mathcal{C}(G)$, then $H^* = V_1$.*

Proof. By induction on the number of cycles in G.

Base Case. Since G has a single cycle and since every node in G is G-trapped, then $deg^-(n) = deg^+(n) = 1$, for every node n in G. Thus, by Definition 6, $\{n\}^* = V_1$ for any node $n \in V_1$.

Induction Hypothesis. For G with k or fewer cycles, if H is a hitting set of $\mathcal{C}(G)$, then $H^* = V_1$.

Induction Step. Let G have $k + 1$ cycles. Pick a cycle C and a hyp $n \in C$. Let G_1 be the graph resulting from removing all edges ending at n in G. Given G-trapping, n is the only node in G_1 with a zero in-degree. Now, let $G_2 = \texttt{RGTrim}(\texttt{GTrim}(G_1)) = (V_1^2, V_2^2, E_2)$. By Corollary 3, $(V_1^2 \cup \{n\})^* = V_1$. Since G_2 has at least one cycle, namely C, fewer than G, then by the induction hypothesis, $H_2^* = V_1^2$, for any hitting set H_2 of $\mathcal{C}(G_2)$. Thus, from Definition 6, it follows that $(H_2 \cup \{n\})^* = V_1$. Since, $\mathcal{C}(G_2) \subset \mathcal{C}(G)$, then, for every hitting set H of $\mathcal{C}(G)$, there is some hitting set H_2 of $\mathcal{C}(G_2)$ such that $(H_2 \cup \{n\}) \subseteq H$. Thus, $H^* = V_1$.

Theorem 1. *Let $G = (V_1, V_2, E)$ be a trimmed and reverse-trimmed base-support graph. If H is a hitting set of $\mathcal{CH}(G)$, then $H^* = V_1$.*

Proof. By Observation 2, a hitting set of $\mathcal{CH}(G)$ is also a hitting set of $\mathcal{C}(G)$. Hence, by Lemma 1, the theorem follows.

Now, the algorithm to construct a compression which is either minimal (if the base-support graph is acyclic) or strictly smaller in size than trimmed and reverse-trimmed compressions comprises the following steps, given a belief state $\mathbb{S} = \langle K, B, \sigma \rangle$.

1. Construct the base-support graph G of the K_0-compression of $\mathbb{S}$.
2. Construct $G_{tr} = \texttt{RGTrim}(\texttt{GTrim}(G))$.
3. Construct $\mathcal{CH}(G_{tr})$. (This is done using the algorithm of [3] appropriately modified for directed, bipartite graphs.)
4. Construct a smallish hitting set H of $\mathcal{CH}(G_{tr})$. (This is done using a greedy approximation hitting set algorithm.)

5. Construct the belief state $\langle H \cup max(B), H \cup max(B), \sigma \rangle$, where $\sigma(\phi) = \{\}$, for every ϕ in $H \cup max(B)$.

Note that at no point does the algorithm check if some subset N of V_1 is such that $N^* = V_1$. Instead, the algorithm selects nodes from V_1 which are known a priori to satisfy the closure check. Of the above four steps, only the third takes exponential time in the worst-case, which occurs if the graph has exponentially-many chordless cycles. Fortunately, such cases are not very common (cf. [19]). The redundancy of the resulting compression is rooted both in the third step and the approximation made in the fourth step.

Tables 1 and 2 display some results, comparing three B-compression algorithms. The first is our approximation, chordless-cycles-based algorithm, the second is a brute-force breadth-first search (BFS), and the third is a composition of both in which we run BFS-compression on the result of the approximation algorithm. Note that the first algorithm results in compressions which, in general are neither minimal nor irredundant; the second is guaranteed to yield minimal compressions; and the third is guaranteed to result in irredundant, but possibly not minimal, compressions. Table 1 displays the average time taken by each algorithm on randomly generated base-support graphs with varying numbers of nodes. (Around fifty graphs for each size range.) For relatively small graphs with fewer than 20 nodes, BFS has the best run time; but for larger graphs the other two algorithms (especially the first, naturally) dominate. (MLE is a shorthand for "memory limit exceeded.") Table 2 displays the average sizes of the compressions resulting from each algorithm (on the same set of graphs used in Table 1). As expected, BFS gives the smallest compressions (as long as it produces *some* compression) and the approximation algorithm is the worst in this regard. The composition algorithm, however, provides what could be an interesting compromise since the compression sizes it produces are comparable to those provided by BFS.

Nodes	Approximation algorithm	BFS	Composition
1 to 10	12.7 ms	1.3 ms	29.4 ms
11 to 20	103.2 ms	17 ms	115.6 ms
21 to 40	362.9 ms	2833.2 ms	563.7 ms
41 to 50	679.9 ms	20999.2 & MLE	1163.1 ms
51 to 60	1099.3 ms	MLE	24783.2

Table 1. Average run times for the approximation algorithm, a breadth-first-search algorithm, and a composition of both

Nodes	Approximation algorithm	BFS	Composition
1 to 10	1	1	1
11 to 20	4	2	2
21 to 40	12	4	5
41 to 50	13	5	5
51 to 60	18	MLE	9

Table 2. Average compression sizes for the approximation algorithm, a breadth-first-search algorithm, and a composition of both

5 A Use-Based Lossy Compression

The final compression operator we consider generally yields lossy compressions; it contracts members of B which are not very useful in a sense defined below. To define such an operator, we need to revise our notion of support and, hence, belief state; we enrich the support structure with information about the exact chain of reasoning used to derive a formula, not just the hypotheses underlying the derivation. For example, consider a belief state $\mathbb{S} = \langle K, B, \sigma \rangle$ with $B = \{Q, Q \rightarrow P, P \rightarrow R, P \rightarrow T\}$, $K = B \cup \{P, R, T\}$, and

- $\sigma(P) = \{\{Q, Q \rightarrow P\}\}$
- $\sigma(R) = \{\{Q, Q \rightarrow P, P \rightarrow R\}\}$
- $\sigma(T) = \{\{Q, Q \rightarrow P, P \rightarrow T\}\}$
- $\sigma(\phi) = \{\}$, for every $\phi \in B$

Assuming some standard proof theory, Q was used to derive P and was never used again in any derivations. Intuitively, P seems to be the most important formula in this example since it is used to derive both R and T. The presence of Q, and the absence of P, from the supports of R and T is merely a side-effect of Q's being a hypothesis; but it is P, together with the respective material implication, that was used in actually deriving R and T. It, thus, may be fine to *forget* Q in this belief state and to *lift* P to become a hypothesis. The definition of support in a belief state (σ) does not, however, provide sufficient information to allow us to simulate such useful forgetfulness.

Definition 7. *Let $B \subseteq K \subseteq L$. A layered support for $\phi \in K$ is a finite set $s = \{C_1, C_2, ..., C_n\}$ where:*

- *for $1 \leq i \leq n$, $C_i = \langle \psi_{i1}, \ldots, \psi_{i|C_i|} \rangle$ with $\psi_{ij} (\neq \phi) \in K$ and $\psi_{i1} \in B$; and*
- *$\phi \in Cn(s_i(\phi))$, for $1 \leq i \leq m$ with $m = max_i\{|C_i|\}$ and $s_i(\phi) = \{\psi|$ there are C_l, C_r such that $C_l \circ \langle \psi \rangle \circ C_r \in s$ and either $|C_r| = i - 1$ or $|C_r| < i - 1$ and $|C_l| = 0\}$.*[6]

[6] Henceforth, we use $\circ$ to denote sequence concatenation.

The definition of belief state is now revised with a layered support replacing the original, flat support and with two "integrity" conditions.[7]

Definition 8. *A layered belief state is a triple* $\mathbb{S} = \langle B, K, \sigma \rangle$ *with* $B \subseteq K \subseteq L$ *and* σ *a function mapping each* $\phi \in K$ *to a set of layered supports for* ϕ *such that for every* $\psi \in K$ *if there is* $s \in \sigma(\psi)$ *with a chain* C *such that* $C = C_l \circ \langle \phi \rangle \circ C_r \in s$ *and* $|C_l| \neq 0$*, then there is* $s^{'} \in \sigma(\phi)$ *with* $C_l \in s^{'}$ *and for every* $C' \in s^{'}$ $C' \circ \langle \phi \rangle \circ C_r \in s$*. The size* $|\mathbb{S}|$ *of* $\mathbb{S}$ *is defined as* $|K| + \sum_{\phi \in K} \sum_{s \in \sigma(\phi)} \sum_{C \in s} |C|$*.*

Given this new notion of support, we introduce a suitably-tailored definition of compression.

Definition 9. *Let* $\mathbb{S} = \langle K, B, \sigma \rangle$ *be a layered belief state. A compression* $\mathbb{S}'$ *of* $\mathbb{S}$ *is a layered belief state* $\langle K', B', \sigma' \rangle$ *with*

1. $K' \subseteq K$*;*
2. *for every* $\phi \in K'$ *and* s' *in* $\sigma'(\phi)$*, there is some* s *in* $\sigma(\phi)$ *such that every chain* C' *in* s' *is a suffix of some chain* C *in* s*; and*
3. $|\mathbb{S}'| < |\mathbb{S}|$.

Using layered supports we can determine if a hypothesis is useful in its own right or if it is interesting only because it was once used to derive some formulas that are the ones frequently involved in reasoning processes. This way we, intuitively, replace members of B by members of $K - B$ to retain the possibility of deriving interesting propositions. Of course, this is, generally, lossy since we retract unsupported, albeit insignificant, hypotheses.

A layered belief state representing the situation discussed in the above example will have the following support function.

- $\sigma(P) = \{\{\langle Q \rangle, \langle Q \rightarrow P \rangle\}\}$
- $\sigma(R) = \{\{\langle Q, P \rangle, \langle Q \rightarrow P, P \rangle, \langle P \rightarrow R \rangle\}\}$
- $\sigma(T) = \{\{\langle Q, P \rangle, \langle Q \rightarrow P, P \rangle, \langle P \rightarrow T \rangle\}\}$
- $\sigma(\phi) = \{\}$, for every $\phi \in B$

Intuitively, Q may be forgotten since the number of chains in which it appears on its own is less than the number of chains in which it is followed by at least one other proposition.

Definition 10. *Let* $\mathbb{S} = \langle K, B, \sigma \rangle$ *be a layered belief state and let* $\alpha \in \mathbb{Q}$*. A formula* $\phi \in B$ *is* α*-disposable if* $|\{s \in \Sigma(\mathbb{S}) | \langle \phi \rangle \circ C \in s\}| = 0$ *or*

$$\frac{|\{s \in \Sigma(\mathbb{S}) | \langle \phi \rangle \in s\}|}{|\{s \in \Sigma(\mathbb{S}) | \langle \phi \rangle \circ C \in s\}|} < \alpha$$

Towards capturing our intuitions about the proposed lossy compression, we introduce the following terminology.

[7] The layered support may be constructed from the flat support together with information about the formulas directly used in derivations (akin to *justification-based* reason maintenance systems [4].

Definition 11. *Let s be a layered support, $\mathbb{S} = \langle K, B, \sigma \rangle$ a layered belief state, and $\alpha \in \mathbb{Q}$.*

- *A tree in s is a set $s' \subseteq s$ such that there is a chain $r(s')$ such that $s' = \{C | C = C_l \circ r(s'), \text{ for some } C_l\}$. A tree s' in s is an α-disposable tree if there is some $C_l \circ r(s') \in s$ such that C_l ends with an α-disposable formula.*
- *$\mathcal{D}(\mathbb{S}, \alpha) = \{\phi \in B | \phi \text{ is } \alpha\text{-disposable}\}$.*
- *$Base(\mathbb{S}, \alpha) = \{\phi \in K - B | \sigma(\phi) \text{ contains an } \alpha\text{-disposable tree}\}$.*
- *$\mathcal{R}(\mathbb{S}, \alpha, s) = \{r(s') | s' \text{ is a maximal } \alpha\text{-disposable tree of } s\} \cup \{C | C \in s \text{ is not in any } \alpha\text{-disposable tree}\}$.*

Definition 12. *For a layered belief state $\mathbb{S}$ $\langle K, B, \sigma \rangle$, the α-cut of $\mathbb{S}$, denoted $cut(\mathbb{S}, \alpha)$, is a triple $\langle K^{'}, B^{'}, \sigma^{'} \rangle$, where*

- *$K^{'} = K - \mathcal{D}$.*
- *$B^{'} = B - \mathcal{D} \cup Base(\mathbb{S}, \alpha)$.*
- *$\sigma^{'}(\phi) = \{\mathcal{R}(\mathbb{S}, \alpha, s) | s \in \sigma(\phi) \text{ and } \langle\rangle \notin \mathcal{R}(\mathbb{S}, \alpha, s)\}$.*

Thus, insignificant hypotheses, deemed as such by the parameter α, are forgotten and significant non-hypotheses are promoted to hypothesis status.

Corollary 4 *If $\mathbb{S}$ is a layered belief state and $\alpha \in \mathbb{Q}$, then $cut(\mathbb{S}, \alpha)$ is a layered belief state. Further, if $\mathcal{D}(\mathbb{S}, \alpha) \neq \{\}$, then $cut(\mathbb{S}, \alpha)$ is a compression of $\mathbb{S}$.*

6 Conclusions and Future Work

Belief state compression is a variant of knowledge base size reduction which does not put any syntactic constraints on the form of the knowledge base, but assumes a support-structured set of formulas akin to classical reason maintenance systems. We have explored various approaches to belief state compression. In particular, varieties of base compression which can sometimes be minimal have been pointed out, and an efficient, tight, but not necessarily minimal nor redundant, compression technique has been proposed. An example of a lossy compression which disposes of insignificant pieces of information has also been described.

We have experimented with variants of our chordless-cycles-based approximation algorithm. In particular, we have some preliminary results with iteratively applying the algorithm on its own output until a fixed-point is reached. The results are promising, and we conjecture that the method yields irredundant compressions. We save the proof and further experiments to future work.

References

1. Anton Belov, Mikolas Janota, Ines Lynce, and Joao Marques-Silva. Algorithms for computing minimal equivalent subformulas. *Artificial Intelligence*, 216:309–326, 2014.
2. Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

3. Elisângela Silva Dias, Diane Castonguay, Humberto Longo, and Walid Abdala Rfaei Jradi. Efficient enumeration of chordless cycles. *arXiv preprint arXiv:1309.1051*, 2013.
4. John Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.
5. Michael Garey and David Johnson. *Computers and Intractability.* W. H. Freeman and Company, 1979.
6. Allen Ginsberg. Knowledge base reduction: A new approach to checking knowledge bases for inconsistency and redundancy. In *Proceedings of AAAI-88*, pages 585–589, 1988.
7. Judy Goldsmith, Matthias Hagen, and Martin Mundhenk. Complexity of DNF minimization and isomorphism testing for monotone formulas. *Information and Computation*, 206:760–775, 2008.
8. Edith Hemaspaandra and Gerd Wechsung. The minimization problem for boolean formulas. *SIAM Journal on Computing*, 31:1948–1958, 2002.
9. Haythem O. Ismail and Nasr Kasrin. Focused belief revision as a model of fallible relevance-sensitive perception. In *Proceedings of the 33rd Annual German Conference on Artificial Intelligence (KI-2010)*, pages 126–134. Springer, 2010.
10. Jakub Kotowski, François Bry, and Simon Brodt. Reasoning as axiom change: Incremental view maintenance reconsidered. In *Proceedings of the 5th International Conference on Web Reasoning and Rule Systems (RR-2011)*, pages 139–154. Springer, 2011.
11. Paulo Liberatore. Redundancy in logic I: CNF propositional formulae. *Artificial Intelligence*, 163:203–232, 2005.
12. Cláudio L. Lucchesi and Sylvia L. Osborn. Candidate keys for relations. *Journal of Computer and System Sciences*, 17(2):270–279, 1978.
13. Hai H. Nguyen, Natasha Alechina, and Brian Logan. Axiom pinpointing using an assumption-based truth maintenance system. In *Proceedings of the 25th International Workshop on Description Logics (DL 2012)*, pages 290–300. CEUR Workshop Proceedings Vol-846, 2012.
14. Zoran Nikoloski, Sergio Grimbs, Joachim Selbig, and Oliver Ebenhöh. Hardness and approximability of the inverse scope problem. In *International Workshop on Algorithms in Bioinformatics*, pages 99–112. Springer, 2008.
15. Hossein Saiedian and Thomas Spencer. An efficient algorithm to compute the candidate keys of a relational database schema. *The Computer Journal*, 39(2):124–132, 1996.
16. James Schmolze and Wayne Snyder. Detecting redundant production rules. In *Proceedings of AAAI-97*, pages 417–423, 1997.
17. Stuart C. Shapiro and Haythem O. Ismail. Anchoring in a grounded layered architecture with integrated reasoning. *Robotics and Autonomous Systems*, 43:97–108, 2003.
18. Abhishek Sharma and Kenneth Forbus. Automatic extraction of efficient axiom sets from large knowledge bases. In *Proceedings of AAAI-13*, pages 1248–1254, 2013.
19. Takeaki Uno and Hiroko Satoh. An efficient algorithm for enumerating chordless cycles and chordless paths. In *International Conference on Discovery Science*, pages 313–324. Springer, 2014.
20. Ralf Wastl. Linear derivations for keys of a database relation schema. *Journal of Universal Computer Science*, 4(12):883–897, 1998.
21. Marcel Wild. Generating all cycles, chordless cycles, and hamiltonian cycles with the principle of exclusion. *Journal of Discrete Algorithms*, 6:93–102, 2008.