University of Massachusetts Boston
ScholarWorks at UMass Boston

Management Science and Information Systems Faculty Publication Series | Management Science and Information Systems

9-1-2012

Craig interpolation for networks of sentences

H Jerome Keisler
University of Wisconsin Colleges

Jeffrey M. Keisler
University of Massachusetts Boston, jeff.keisler@umb.edu

Follow this and additional works at: http://scholarworks.umb.edu/msis_faculty_pubs

Recommended Citation
Keisler, H Jerome and Keisler, Jeffrey M., "Craig interpolation for networks of sentences" (2012). *Management Science and Information Systems Faculty Publication Series.* Paper 31.
http://scholarworks.umb.edu/msis_faculty_pubs/31

This Article is brought to you for free and open access by the Management Science and Information Systems at ScholarWorks at UMass Boston. It has been accepted for inclusion in Management Science and Information Systems Faculty Publication Series by an authorized administrator of ScholarWorks at UMass Boston. For more information, please contact library.uasc@umb.edu.

CRAIG INTERPOLATION FOR NETWORKS OF SENTENCES

H. JEROME KEISLER AND JEFFREY M. KEISLER

Abstract. The Craig Interpolation Theorem can be viewed as saying that in first order logic, two agents who can only communicate in their common language can cooperate in building proofs. We obtain generalizations of the Craig Interpolation Theorem for finite sets of agents with the following properties. (1) The agents are vertices of a directed graph. (2) The agents have knowledge bases with overlapping signatures. (3) The agents can only communicate by sending to neighboring agents sentences that they know and that are in the common language of the two agents.

1. Introduction

We may formulate the Craig Interpolation theorem in the following way to describe a pair of agents who can cooperate in building proofs.[1].

Fact 1.1. *(Craig* [10] *(1957)) In first order logic, suppose we have two signatures* $L(x), L(y)$ *and corresponding knowledge bases* $\mathcal{K}(x) \subseteq [L(y)]$, $\mathcal{K}(y) \subseteq [L(y)]$. *For any sentence* $D \in [L(y)]$ *that is provable from the combined knowledge base* $\mathcal{K}(x) \cup \mathcal{K}(y)$, *there is a sentence* C *in the common language* $[L(x) \cap L(y)]$ *such that* C *is provable from* $\mathcal{K}(x)$, *and* D *is provable from* $\mathcal{K}(y)$ *and* C.

In this paper we consider what happens when there are more than two agents. Our setup is motivated by the example of an organization with finitely many agents (perhaps individuals, departments, groups of people, or programs) who are identified with the vertices of a directed graph. The agents have different but overlapping signatures (or vocabularies), and can only communicate in the restricted way described below.

We assume that the directed graph has least one agent d, called a decider, such that for every other agent x there is at least one path from x to d. There are no edges from an agent to itself, but cycles are allowed. Each agent x has a signature $L(x)$ and a knowledge base

Date: February 10, 2012.

[1]Unexplained terms will be defined later in this paper

$\mathcal{K}(x) \subseteq [L(x)]$. We say that a sentence D in a decider's language $[L(d)]$ is **provable** if it is provable from the union of all the knowledge bases.

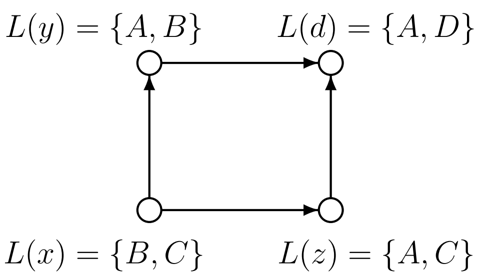
The agents can only communicate in the following restricted way: For each directed edge (x, y) in the graph, agent x can report to agent y a sentence that is in the common language of x and y and is provable from x's knowledge base and sentences that have been reported to x. We say that the sentence D is **report provable** at a decider d if after finitely many of these restricted communications, d can prove D from its own knowledge base and sentences that have been reported to d. It is clear that every report provable sentence is provable from the union.

By a pointed graph we will mean a directed graph with at least one decider. Let us fix a pointed graph, and fix a signature for each agent, but leave the knowledge bases unspecified. We will call this a **signature network** $\mathbb{S}$. When we also give each agent a knowledge base, we obtain a **knowledge base** $\mathbb{K}$ over $\mathbb{S}$. We say that a signature network $\mathbb{S}$ is **report complete** if for *every* knowledge base over $\mathbb{S}$ and every decider d, every sentence in d's language that is provable is report provable at d. So for a given signature network, report completeness is a guarantee that for every knowledge base and decider, every provable sentence is report provable. Report completeness is the central notion in this paper.

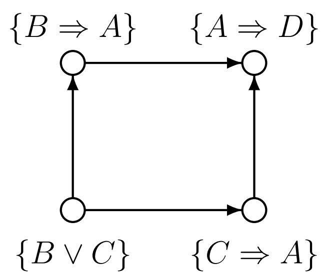
Craig's Theorem says that every signature network with just two agents is report complete. For this reason, we view report completeness as a natural generalization of Craig interpolation to a setting where we have a network of sentences. Our title is meant to convey this circle of ideas as briefly as possible.

The following examples illustrate some of the things that can happen.

Example 1.2. $\mathbb{S}$ is a signature network with exactly one decider d. $\mathbb{K}$ is a knowledge base over $\mathbb{S}$.



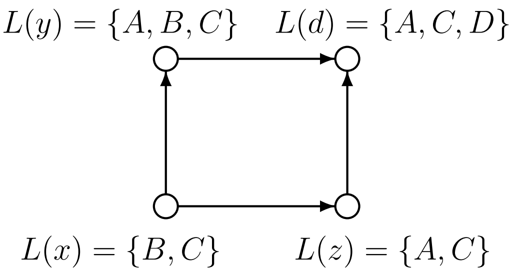
$\mathbb{S}$



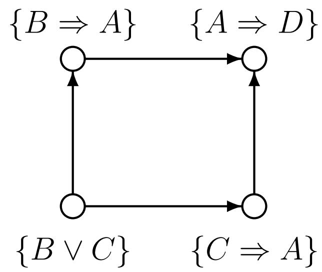
$\mathbb{K}$

It is easy to see that the sentence D is provable from the combined knowledge bases of the four agents in $\mathbb{K}$. But D is not report provable in $\mathbb{K}$, so the signature network $\mathbb{S}$ is not report complete. The agent x can't report anything (except logically valid sentences) to y because the symbol C is not in y's signature. Similarly, x can't anything to z, y can't report anything to d, and z can't report anything to d.

Example 1.3. In this example we change $\mathbb{S}$ to $\mathbb{R}$ by adding the symbol C to the upper two signatures, but keep the same knowledge bases.



$\mathbb{S}$

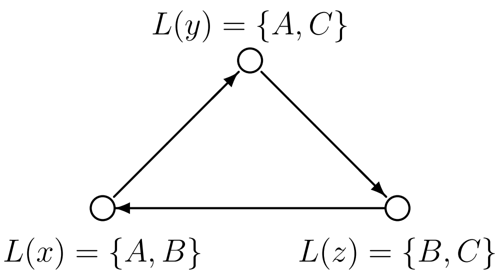


$\mathbb{K}$

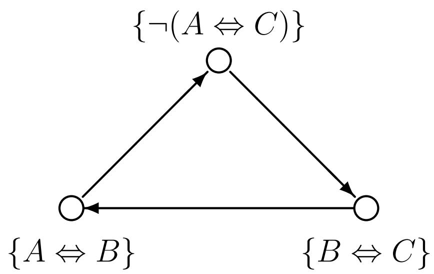
Now the sentence D is report provable. In fact, as we will see later, for every knowledge base network over $\mathbb{R}$, every provable sentence is report provable, so $\mathbb{R}$ is report complete.

Agent x still can't report anything to z. But x can report the sentence $B \vee C$ to y, and z can report the sentence $C \Rightarrow A$ to d. Then y can prove the sentence $A \vee C$ and report it to d, and finally, d can prove the sentence D.

Example 1.4. $\mathbb{S}$ is a signature network in which every agent is a decider. $\mathbb{K}$ is a knowledge base over $\mathbb{S}$.



$\mathbb{S}$



$\mathbb{K}$

The reader can check that the false sentence $\bot$ is provable in $\mathbb{K}$, but at each decider, $\bot$ is not report provable in $\mathbb{K}$. So at each decider, $\mathbb{S}$ is not report complete.

Example 1.5. *Change the signature network* $\mathbb{S}$ *in Example 1.4 to* $\mathbb{R}$ *by replacing any one of the signatures by* $\{A, B, C\}$. *We will see that* $\mathbb{R}$ *is report complete.*

An important property for a signature network is the **Peak Property**, that for each set σ of symbols, the subgraph consisting of those agents whose signature contains σ has a decider. We show in Theorem 3.5 that the Peak Property is a necessary condition for a signature network to be report complete. For a signature network on a tree, the Peak Property is equivalent to the running intersection property in the literature (e.g. see [2], [9], [11]), but these properties are different on arbitrary signature networks. Each of the examples above has the Peak Property.

Amir and McIlraith [2] generalized Craig's Theorem to trees of agents. Let us state their result in our setting. By a **signature tree** we mean a signature network with the Peak Property whose graph is a directed tree. Note that a directed tree has a unique decider.

Fact 1.6. *([2]) Every signature tree is report complete.*

This gives a sufficient condition for report completeness. In the terminology of [2], it says that the forward message-passing algorithm is complete. An easy consequence of Fact 1.6 is that every signature network that contains a signature tree with the same agents and signatures is report complete. For instance, in Example 1.3, $\mathbb{R}$ contains the signature tree obtained by removing the bottom edge, and thus is report complete. Similarly, 1n Example 1.5, $\mathbb{R}$ contains a signature tree and is report complete.

In Theorem 4.3 we obtain a converse to Fact 1.6: A pointed graph has no weak cycles if and only if every signature network on the graph that has the Peak Property is report complete.

We introduce two other conditions and prove that on every pointed graph, these conditions are also necessary for report completeness. These are the **Twin Peaks Property** (which implies the Peak Property), and the **Linked Chain Property**.

The Twin Peaks Property is hard to state in general, but in the case of a directed acyclic graph it is equivalent to the property that every agent x with parents has a **dominant parent**– a parent whose signature contains every symbol that belongs to the signatures of x and some parent of x. In Theorem 6.6 we show that for signature networks on a directed acyclic graph, the Twin Peaks Property, report completeness, and containing a signature tree are equivalent. For instance, the signature networks in Examples 1.2 and 1.3 have directed acyclic graphs. The network in 1.3 has the Twin Peaks Property, but the network in 1.2 does not.

The simplest case of the Linked Chain Property says that for any three disjoint sets of symbols, if each pair is within the signature of some agent, then so are all three. In Example 1.4, the graph has a directed cycle; $\mathbb{S}$ has the Twin Peaks Property but does not have the Linked Chain Property. But in Example 1.5, $\mathbb{R}$ does have the Linked Chain Property.

Theorem 9.7 says that for signature networks on a connected graph,[2] report completeness, having both the Peak Property and Linked Chain Property, and containing a signature tree are equivalent. Its proof uses two other results, Theorems 7.4 and 9.5. Theorem 7.4 shows that every signature network is report complete at a decider whose signature contains every symbol that occurs in the signature of more than one agent. Theorem 9.5 is a decomposition result that is of interest in its own right and may have other applications where one needs to break a large network into smaller pieces.

Example 4.2 shows that there are signature networks that are report complete but do not contain a signature tree. Example 8.4 shows that there are signature networks that have both the Twin Peaks Property and the Linked Chain Property but are still not report complete. In the last section we will list some questions that remain open.

This work is related to but has a different focus than the paper [2] of Amir and McIlraith. The aim of that paper is to develop partition-based reasoning as a method for creating efficient algorithms for automated reasoning. They consider how signatures might be chosen with that aim in mind, rather than being given in advance. For an arbitrary graph, they give an algorithm that simultaneously removes edges and enlarges signatures to form a signature tree. By contrast, our results say when it is possible to cut a signature network down to a signature tree by keeping the signatures the same and only removing edges.

This work is also related the area of peer-to-peer networks, where again the main focus has been to find efficient algorithms for automated reasoning in a decentralized setting. The paper Adjiman et. al. [1] gives an algorithm and proves a completeness theorem for propositional resolution proofs in peer-to-peer networks. In effect, that system allows agents to avoid the Linked Chain Property by making queries that they cannot prove. This leads to a very different completeness result that requires only the Peak Property on an arbitrary connected graph.

Our aim is to determine when it is possible, as in the Craig Interpolation case, to have report completeness where the signatures are given in advance and the agents can only report sentences they know. This

[2]By a connected graph we mean a graph with a symmetric edge relation such that any two vertices are connected by a path.

is a natural problem from mathematical logic which has intrinsic interest in its own right. There are also situations where one might expect agents to act in this way. One example is a decentralized organization where established departments report their findings to others and have vocabularies that cannot easily be changed.

For surveys of Craig interpolation and its uses see van Bentham [5], Feferman [14], McMillan [18], Renardel de Lavalette [21].

Prerequisites: We assume familiarity with the notions of sentence and proof in first order logic with equality. For background see Enderton [12], Chang and Keisler [8].

The notation $\mathcal{K} \vdash B$ means that the sentence B is provable from the set of sentences $\mathcal{K}$, and $\vdash B$ means that B is provable. A **signature** (or **vocabulary**) is a set L of constant, relation, or function symbols, and the **language of** L is the set $[L]$ of first order sentences built from L. A **knowledge base**, or **theory**, in L is a subset of $[L]$. First order logic is formulated so that the true sentence $\top$ and false sentence $\bot$ belong to $[L]$ for every signature L. To avoid certain trivial exceptions, if there are constant symbols we will also require that there is a distinguished constant symbol 0 that can be used in every $[L]$ (this is needed in Definition 3.3). The symbols $\bot, \top, =, 0$ do not count as part of the signatures. A set of sentences $\mathcal{K}$ is **consistent** if it is not the case that $\mathcal{K} \vdash \bot$. We use $B \Rightarrow D$ as an abbreviation for $\neg B \vee D$. We will use script capital letters for sets of sentences, and blackboard bold letters for networks.

The only properties of first order logic that we use in this paper are:

- for each signature L, $[L]$ is closed under the connectives $\wedge, \vee, \neg$ and contains $\top, \bot$;
- all propositional tautologies are provable;
- (Deduction) $\mathcal{K} \vdash B \Rightarrow D$ if and only if $\mathcal{K} \cup \{B\} \vdash D$;
- (Compactness) a set of sentences $\mathcal{K}$ is consistent if and only if every finite subset of $\mathcal{K}$ is consistent;
- (Craig Interpolation) if $B \in [L_1]$, $D \in [L_2]$, and $B \vdash D$, then there exists $C \in [L_1 \cap L_2]$ such that $B \vdash C$ and $C \vdash D$.

Thus all of our results also hold for any other logic that has these properties. Some examples of such logics are propositional logic, first order logic without quantifiers, first order formulas (with free variables), many-sorted first order logic (e.g. Feferman [13]), and various modal logics (e.g. Bilkova [6]).

For other logics, interpolation theorems have been obtained with restrictions. These include infinitary logic (Barwise and van Bentham

[4]), intuitionistic logic (Gabbay [15]), and logic with probability quantifiers (Hoover [16]). There are many interpolation results for fragments of first order logic (for example Carbone [7], Lyndon [17], Otto [19], Popescu et. al. [20], Rodenberg [22]). We leave open the problem of generalizing all these results to networks of sentences.

2. Report Completeness

In this section we formally define the notions of signature network and report completeness. We first review some terminology about directed graphs.

By a **(simple) directed graph** (V, E) we mean a non-empty finite set V of **vertices**, and a set $E \subseteq V \times V$ of **edges** (or arcs) (x, y) such that $x \neq y$. (We do not allow more than one edge from a vertex x to a vertex y, we do not allow edges from a vertex to itself, and we distinguish between the pair (x, y) and the pair (y, x).) A **(directed) path** of length n from x to y is a sequence $(x_0, \ldots, x_n)$ of vertices such that $x_0 = x, x_n = y$, and for each $i < n$, $(x_i, x_{i+1}) \in E$. (In particular, for each vertex x, (x) is a path of length 0 from x to itself.) A **directed cycle** of length n is a sequence $(x_0, \ldots, x_{n-1}, x_n)$ of vertices such that $x_0, \ldots, x_{n-1}$ is a directed path, $x_n = x_0$, and $(x_{n-1}, x_n) \in E$. A **directed acyclic graph** is a directed graph with no directed cycles.

Definition 2.1. *In a directed graph, by a* ***decider*** *we mean a vertex d such that for every other vertex x, there is at least one path from x to d. By a* ***pointed graph*** *we mean a directed graph (V, E) with at least one decider.*

Lemma 2.2. *For each directed graph (V, E), the following are equivalent:*

(i) (V, E) is pointed;

(ii) For every pair of vertices $x, y \in V$ there exists a vertex $z \in V$, a path form x to z, and a path from y to z.

Proof. It is clear that (i) implies (ii). Assume (ii), and let $V = \{x_0, \ldots, x_n\}$. Let $y_0 = x_0$ and inductively take y_{m+1} to be a vertex such that there is a path from y_m to y_{m+1} and a path from x_{m+1} to y_{m+1}. Then y_n is a decider for (V, E), so (V, E) is pointed. □

Remark 2.3. *Let (V, E) be a directed graph and let $x \in V$. Let (U, F) be the graph*

$$U = \{y \in V : \text{ there is a path from } y \text{ to } x\}, \quad F = E \cap U \times U.$$

Then (U, F) is a pointed graph and x is a decider for (U, F). If x is already a decider for (V, E) then $U = V$.

Hereafter we will always assume that (V,E) is a pointed graph, and that d is a decider for (V,E). This will cause no loss in generality and will allow us to avoid exceptional cases in the statements of results.

A **source** is a vertex x such that there are no edges $(y,x) \in E$, and a **sink** is a vertex x such that there are no edges $(x,y) \in E$. Note that a pointed graph has at most one sink, and if there is a sink then it is the unique decider. In particular, a pointed directed acyclic graph has a unique sink and decider d. On the other hand, if there are no sinks then there must be more than one decider, and every decider must be on a directed cycle.

By a **tree** we will mean a pointed directed acyclic graph such that for every vertex $x \neq d$ there is a unique edge $(x,y) \in E$. It follows that for every vertex x there is a unique path from x to d.

We will now attach signatures and knowledge bases to the vertices of pointed graphs. From now on we will call the vertices **agents**.

Definition 2.4. *A* ***signature network*** *on* (V,E) *is an object*

$$\mathbb{S} = (V, E, L(\cdot))$$

where (V,E) *is a pointed directed graph with a labeling* $L(\cdot)$ *that assigns a signature* $L(x)$ *to each agent* $x \in V$*. We let* $L(V) = \bigcup_{x \in V} L(x)$*, and call the set* $L(V)$ *the* ***combined signature****.*

Given a signature network $\mathbb{S} = (V, E, L(\cdot))$*, a* ***knowledge base*** *(over* $\mathbb{S}$*) is an object*

$$\mathbb{K} = (V, E, L(\cdot), \mathcal{K}(\cdot))$$

where $\mathcal{K}(\cdot)$ *is a labeling that assigns a knowledge base* $\mathcal{K}(x) \subseteq [L(x)]$ *to each agent* $x \in V$*. For each set* $U \subseteq V$ *we write* $\mathcal{K}(U) = \bigcup_{x \in U} \mathcal{K}(x)$*, and we call the set* $\mathcal{K}(V)$ *the* ***combined knowledge base****.*

Note that in the above definition, each symbol that occurs in a sentence of $\mathcal{K}(x)$ must belong to $L(x)$, but we allow the possibility that $L(x)$ also has additional symbols.

We now formalize the notion of report provability.

Definition 2.5. *Let*

$$\mathbb{K} = (V, E, L(\cdot), \mathcal{K}(\cdot))$$

be a knowledge base over a signature network $\mathbb{S}$*. A sentence* C *is* **0-*reportable*** *in* $\mathbb{K}$ *along an edge* (x,y) *if*

$$C \in [L(x) \cap L(y)] \text{ and } \mathcal{K}(x) \vdash C.$$

C *is* $(n+1)$***-reportable*** *in* $\mathbb{K}$ *along an edge* (x,y) *if*

$$C \in [L(x) \cap L(y)] \text{ and } \mathcal{K}(x) \cup \mathcal{R} \vdash C,$$

where $\mathcal{R}$ is a set of sentences each of which is n-reportable along some edge (z, a).

The word "reportable" means n-reportable for some n", and "reportable to y" means "reportable along (x, y) for some x".

Given a decider d for $\mathbb{S}$, a sentence $D \in [L(d)]$ is ***report provable*** *in $\mathbb{K}$ at d if D is provable from $\mathcal{K}(d)$ and a set $\mathcal{R}$ of sentences each of which is reportable to d in $\mathbb{K}$.*

Verbally, at each stage, for each edge (x, y), agent x can report to agent y a sentence C in their common language, where C is provable from the knowledge base $\mathcal{K}(x)$ and sentences reported to x at earlier stages. Finally, D is provable from the knowledge base $\mathcal{K}(d)$ and sentences reported to d. Thus the sentence D is established using only proofs within the languages $[L(x)]$ of single agents x, and communications along edges (x, y) in common language $[L(x) \cap L(y)]$.

Example 1.2 give a sentence that is provable but not report provable.

Let us check that report provability implies provability.

Lemma 2.6. *Suppose d is a decider and a sentence $D \in [L(d)]$ is report provable in a knowledge base $\mathbb{K}$ on a signature network $\mathbb{S}$. Then D is provable from the combined knowledge base, $\mathcal{K}(V) \vdash D$.*

Proof. The following can be proved by induction on n:

(1) For every agent y, every sentence that is n-reportable to y is provable from $\mathcal{K}(U)$, where U is the set of all agents z such that there is a path from z to y.

Then for some n, D is provable from $\mathcal{K}(d)$ and a set of sentences in $\mathcal{K}(V)$. □

Definition 2.7. *Let $\mathbb{S}$ be a signature network and d a decider in $\mathbb{S}$. A knowledge base on a signature network is* ***report complete at*** *d if every sentence $D \in [L(d)]$ that is provable from the combined knowledge base $\mathcal{K}(V)$ is report provable in $\mathbb{K}$ at d. A signature network $\mathbb{S}$ is* ***report complete at*** *d if every knowledge base on $\mathbb{S}$ is report complete at d. $\mathbb{S}$ is* ***report complete*** *if $\mathbb{S}$ is report complete at every decider d.*

The Craig Interpolation Theorem shows that every signature network on a pointed graph with two agents is report complete. Here is an easy converse result showing that on every pointed graph with more than two agents, there is a signature network that is not report complete.

Theorem 2.8. *Let (V, E) be a pointed graph. The following are equivalent:*

(i) *Every signature network on* (V, E) *is report complete.*
(ii) V *has cardinality* ≤ 2.

Proof. Craig's Theorem (Fact 1.1) gives (ii) $\Rightarrow$ (i).

For (i) $\Rightarrow$ (ii), consider a signature network $\mathbb{S}$ with at least three agents x, y, z with

$$\begin{array}{lllll} L(x) & = & \{A, B\}, & \mathcal{K}(x) & = & \{A \Rightarrow B\}, \\ L(y) & = & \{B, C\}, & \mathcal{K}(y) & = & \{B \Rightarrow C\}, \\ L(z) & = & \{C, A\}, & \mathcal{K}(z) & = & \{\neg[C \Rightarrow A]\}, \end{array}$$

and with all other signatures and knowledge bases empty. The false sentence $\bot$ is provable from the combined knowledge base $\mathcal{K}(V)$. However, every sentence that is reportable to an agent is equivalent to the true sentence $\top$. Therefore $\bot$ is nowhere report provable, and $\mathbb{S}$ is not report complete. □

The next lemma gives a slightly simpler equivalent formulation of report completeness.

Definition 2.9. *A knowledge base* $\mathbb{K}$ *on a signature network is* ***report inconsistent*** *at a decider* d *if the sentence* $\bot$ *is report provable in* $\mathbb{K}$ *at* d.

Lemma 2.10. *For each signature network* $\mathbb{S}$*, and each decider* d *in* $\mathbb{S}$*, the following are equivalent:*

(i) $\mathbb{S}$ *is report complete;*
(ii) *For every knowledge base* $\mathbb{K}$ *over* $\mathbb{S}$ *such that the combined knowledge base* $\mathcal{K}(V)$ *is inconsistent,* $\mathbb{K}$ *is report inconsistent at* d.

Proof. Given a knowledge base $\mathbb{K}$ over $\mathbb{S}$ and a sentence $D \in [L(d)]$, let $\mathbb{K}'$ be the new knowledge base over $\mathbb{S}$ formed by adding the sentence $\neg D$ to $\mathcal{K}(d)$. Note that D is provable from $\mathcal{K}(V)$ if and only if $\mathcal{K}'(V)$ is inconsistent, and D is report provable in $\mathbb{K}$ at d if and only if $\mathbb{K}'$ is report inconsistent at d. □

We observe that if condition (ii) in Lemma 2.10 holds for some decider, it holds for every decider. So we have:

Corollary 2.11. *A signature network is report complete at some decider if and only if it is report complete at every decider.*

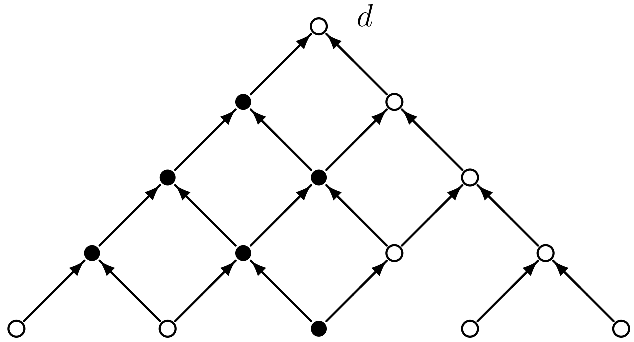
3. The Peak Property

In this section we formally define the Peak Property, and show every report complete signature network has the Peak Property.

Definition 3.1. *We call a subset P of V a* ***peak*** *if either P is empty, or P is non-empty and the subgraph $(P, E \cap (V \times V))$ is a pointed graph.*

By Lemma 2.2, P is a peak if and only if for each $x, y \in P$ there exists $z \in P$ such that P contains a path from x to z and a path from y to z.

For example every singleton $\{x\}$ in V is a peak, every path from an agent x to an agent y is a peak, and every cycle is a peak. In general, a peak $P \subseteq V$ looks like one of the peaks in a mountain range. In the following diagram, the set of agents represented by black circles is a peak.



Definition 3.2. *Let*

$$\mathbb{S} = (V, E, L(\cdot))$$

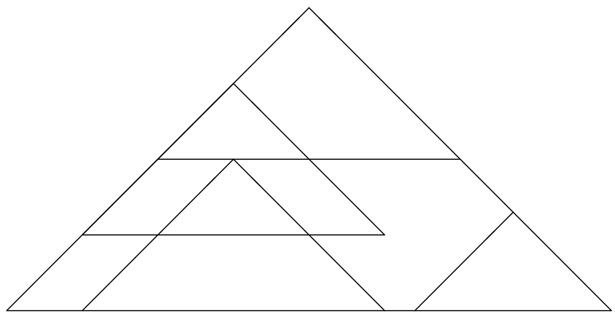
be a signature network. For each finite set σ of symbols in the combined signature $L(V)$, let $V[\sigma] = \{x \in V : \sigma \subseteq L(x)\}$ be the set of agents $x \in V$ whose signature contains σ.

We say that $\mathbb{S}$ has the ***Peak Property*** *if $V[\sigma]$ is a peak for each σ.*

Note that by Lemma 2.2, $\mathbb{S}$ has the Peak Property if and only if for every pair of agents x, y there exists an agent z such that $V[L(x) \cap L(y)]$ contains a path from x to z and a path from y to z.

As we mentioned in the Introduction, on a tree the Peak Property is equivalent to the running intersection property. The next figure gives a conceptual picture of a signature network with the Peak Property. The triangles represent sets of the form $V[\sigma]$ where σ is a set of symbols. Each $V[\sigma]$ looks like a mountain peak.

The Peak Property



We now introduce some special sentences that will be useful in proving necessary conditions for report completeness.

Definition 3.3. *For each symbol S, let $\widehat{S}$ be a sentence whose only non-logical symbol is S (i.e. $\widehat{S} \in [\{S\}]$) such that both $\widehat{S}$ and $\neg\widehat{S}$ are consistent.*

For every finite non-empty set σ of symbols, Let $Ev(\sigma)$ be the sentence that says that the number of $S \in \sigma$ such that $\widehat{S}$ holds is even.

For example, if S is a binary relation symbol, we can take $\widehat{S}$ to be $\forall x S(x,x)$. If S is the constant symbol c, we can take $\widehat{S}$ to be $c = 0$. This is the place where we need the convention that if there are constant symbols, then the distinguished constant 0 belongs to every signature.

Lemma 3.4. *Let σ be a finite non-empty set of symbols, and let A be a sentence in which at least one symbol $S \in \sigma$ does not occur. If either $Ev(\sigma) \vdash A$ or $(\neg Ev(\sigma)) \vdash A$, then $\vdash A$.*

Proof. Suppose $S \in \sigma$ but S does not occur in A. Also suppose that not $\vdash A$. Then $Ev(\sigma)$ is not provable from $\neg A$, because any model of $\neg A$ in which $Ev(\sigma)$ holds can be converted to a model of $\neg A$ in which $Ev(\sigma)$ fails by changing the interpretation of S and leaving everything else alone. Therefore A is not provable from $\neg Ev(\sigma)$. A similar argument shows that A is not provable from $Ev(\sigma)$. □

The next theorem shows that the Peak Property is necessary for report completeness.

Theorem 3.5. *Every signature network that is report complete has the Peak Property.*

Proof. Let $\mathbb{S}$ be a signature network that does not have the Peak Property. Then there is a finite set σ of symbols and two agents x, y such that $V[\sigma]$ contains x and y but there is no agent z such that $V[\sigma]$ contains a path from x to z and a path from y to z.

Now let $\mathbb{K}$ be the knowledge base over $\mathbb{S}$ such that

$$\mathcal{K}(x) = \{Ev(\sigma)\}, \quad \mathcal{K}(y) = \{\neg Ev(\sigma)\},$$

and $\mathcal{K}(z) = \{\top\}$ for every other agent z. Then the sentence $\bot$ is provable from $\mathbb{K}(V)$ and belongs to $[L(d)]$. Let $U(x)$ be the set of all agents $z \in V[\sigma]$ such that $V[\sigma]$ contains a path from x to z, and let $U(y)$ be the corresponding set for y. Then $U(x) \cap U(y) = \emptyset$.

Using Lemma 3.4, one can see by induction that for each n, each agent z, and each sentence C, if C is n-reportable to z then either:

- $\vdash C$;
- each symbol in σ occurs in C, $Ev(\sigma) \vdash C$, and $z \in U(x)$;
- or each symbol in σ occurs in C, $(\neg Ev(\sigma)) \vdash C$, and $z \in U(y)$.

In all cases, the false sentence $\bot$ is never reportable to an agent. Therefore $\bot$ is not report provable in $\mathbb{K}$. □

Example 1.3 gives a report complete signature network with the Peak Property. Example 1.2 gives a signature network that has the Peak Property but is not report complete. So the Peak Property is not sufficient for report completeness. The following two examples are signature networks that do not have the Peak Property, and hence by Theorem 3.5, are not report complete.

Example 3.6. *(Peak Property fails, $\mathbb{S}$ is not report complete).*
The directed edges and signatures are as shown:

$$\{B, C\} \longrightarrow \{B, D\} \longleftarrow \{C, D\}.$$

The Peak Property fails because the set $V[\{C\}]$ is not a peak.

Example 3.7. *(Peak Property fails, $\mathbb{S}$ is not report complete).*
The directed edges and signatures are as shown:

$$\{B, C\} \longrightarrow \{C, D\} \longrightarrow \{B, D\}.$$

The Peak Property fails because the set $V[\{B\}]$ is not a peak.

Theorem 3.5 shows that if the Peak Property fails for a signature network $\mathbb{S}$, then $\mathbb{S}$ is not report complete, so there exists a knowledge base over $\mathbb{S}$ that is not report complete. However, for every signature network $\mathbb{S}$ there also exist knowledge bases over $\mathbb{S}$ that are report complete. A trivial example is the knowledge base where every agent has knowledge base $\{\top\}$. Here is another example.

Example 3.8. *(Peak Property fails, $\mathbb{K}$ is report complete).*

The directed graph and signatures are the same as in Example 3.6, and the knowledge bases are:

$$\{C \Rightarrow B\} \longrightarrow \{B \vee D\} \longleftarrow \{C \Rightarrow D\}.$$

As before, $V[\{C\}]$ is not a peak. But any sentence in the language of $\{B, D\}$ that is provable from $\mathcal{K}(V)$ is already provable from $B \vee D$, so this knowledge base is report complete.

4. Signature Trees

By a **signature tree** we will mean a signature network with the Peak Property that is on a tree. It is easily seen that in a tree (V, E), the intersection of two peaks is again a peak. It follows that on a tree, the Peak Property holds if and only if $V[\sigma]$ is a peak for every singleton σ.

In the Introduction we stated the following result of [2], which is the starting point for this work:

Fact 1.6 (restated). *([2]) Every signature tree is report complete.*

For the convenience of the reader, and because the terminology in the paper [2] is quite different from ours, we give a proof of this result in the Appendix.

As in the Introduction, we say that a signature network $\mathbb{S}$ **contains** a signature network $\mathbb{T}$ if $\mathbb{S}$ and $\mathbb{T}$ have the same agents and signatures, and the directed graph of $\mathbb{T}$ can be obtained from the directed graph of $\mathbb{S}$ by removing edges. Fact 1.6 immediately gives a sufficient condition for any signature network to be report complete.

Corollary 4.1. *If $\mathbb{S}$ is a signature network and $\mathbb{S}$ contains a signature tree $\mathbb{T}$, then $\mathbb{S}$ is report complete.*

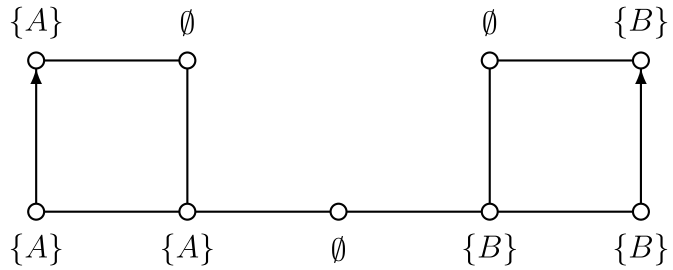
Proof. Let $\mathbb{K}$ be a knowledge base over $\mathbb{S}$, and $\mathbb{H}$ be the knowledge base over $\mathbb{T}$ obtained by removing edges and leaving everything else unchanged. By Fact 1.6, $\mathbb{T}$ is report complete. It is clear that for every agent x, every sentence reportable to x in $\mathbb{H}$ is also reportable to x in $\mathbb{K}$, and it follows that $\mathbb{S}$ is also report complete. □

Intuitively, a signature network $\mathbb{S}$ contains a signature tree $\mathbb{T}$ with decider d if there is a way of assigning to each agent $x \neq d$ an "immediate supervisor" $s(x)$ such that $(x, s(x)) \in E$ and, for every knowledge base $\mathbb{K}$ over $\mathbb{S}$, every provable sentence can be proved by d from its knowledge base and sentences reportable to d in $\mathbb{K}$ when agents can only report sentences to their immediate supervisors.

The following example shows that the converse of Corollary 4.1 fails; there are report complete signature networks that do not contain a signature tree.

Convention: *Throughout this paper, plain lines in a figure will always indicate edges in both directions.*

Example 4.2. *A universal tree-blocker:* This signature network $\mathbb{S}$ is report complete, and every agent is a decider, but $\mathbb{S}$ does not contain a signature tree.



Suppose $\mathbb{R}$ is any report complete signature network in which the symbols A, B do not occur. If we connect the left half of $\mathbb{S}$ to one agent in $\mathbb{R}$ and the right half of $\mathbb{S}$ to another, the new signature network will still be report complete but will not contain a signature tree.

The next theorem improves Fact 1.6 by showing exactly which pointed graphs (V, E) satisfy that condition that every signature network on (V, E) with the Peak Property is report complete.

By a **weak cycle** in a pointed graph (V, E) we mean a finite sequence of agents that can be made into a directed cycle of length ≥ 3 by reversing the direction of some of the edges. Formally, a weak cycle is a sequence of agents $(x_0, \ldots, x_n)$ such that $n \geq 3$, $x_0, \ldots, x_{n-1}$ are distinct, $x_n = x_0$, and for each $j < n$, at least one of the pairs $(x_j, x_{j+1}), (x_{j+1}, x_j)$ is an edge.

Theorem 4.3. *For each pointed graph (V, E), the following are equivalent:*

(i) Every signature network on (V, E) with the Peak Property is report complete.

(ii) Every signature network on (V, E) with the Peak Property contains a signature tree.

(iii) (V, E) has no weak cycles.

Proof. Assume (iii). Let d be a decider in (V, E). Then for each $x \in V$, there is a unique path from x to d. Let F be the set of all edges $(x, y) \in E$ such that y is on the unique path from x to d. Then

(V,F) is a tree. Moreover, for every signature network $\mathbb{S}$ over (V,E), $\mathbb{S}$ contains the signature network $\mathbb{T}$ obtained by replacing E by F. One can easily see that every peak in $\mathbb{S}$ is a peak in $\mathbb{T}$. So if $\mathbb{S}$ has the Peak Property, then $\mathbb{T}$ has the Peak Property, and hence $\mathbb{T}$ is a signature tree and (ii) holds. (ii) implies (i) by Corollary 4.1.

Now suppose (iii) fails, so (V,E) is a pointed graph with a weak cycle $(x_0,\ldots,x_n)$. Let $\mathbb{S}$ be the signature network over (V,E) with $L(x_i) = \{S_i, S_{i-1}\}$ for $1 \le i \le n$, $L(x_0) = \{S_0, S_n\}$, and with all other agents having the empty signature. Then $\mathbb{S}$ has the Peak Property. However, $\mathbb{S}$ is not report complete, because if $\mathbb{K}$ is the knowledge base with $\mathcal{K}(x_i) = \{Ev(S_i, S_{i-1})\}$ for $1 \le i \le n$, and $\mathcal{K}(x_0) = \{\neg Ev(S_0, S_n)\}$, then $\mathcal{K}(V)$ is inconsistent but $\mathbb{K}$ is report consistent. Therefore (i) fails, so (i) implies (iii). ☐

5. Directed Acyclic Graphs

In this section we give a necessary and sufficient condition for report completeness on a directed acyclic graph. We show that a signature network on a directed acyclic graph is report complete if and only if it has the Peak Property and every agent with parents has a dominant parent, and also if and only if it contains a signature tree.

We first make some easy observations. For each edge (x,y) in a graph, we will say that y is a **parent** of x, and that x is a **child** of y. Then x is a sink if and only if x has no parents, and x is a source if and only if x has no children.

Suppose (V,E) is a pointed directed acyclic graph. Recall that (V,E) has a unique sink that is also the unique decider. So the Decider d has no parents. (V,E) has at least one source. A directed acyclic graph (V,E) is a tree if and only if each agent $x \neq d$ has exactly one parent. If P is a peak in a pointed directed acyclic graph, then P has an element x such that for every $y \in P$, P contains a path from y to x.

Definition 5.1. *Let $\mathbb{S}$ be a signature network. We say that an agent y is a **dominant parent** of an agent x if y is a parent of x such that*

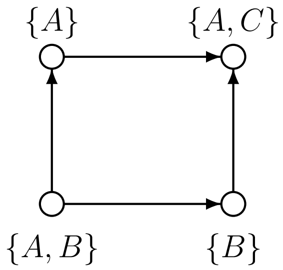
$$L(x) \cap L(y) \supseteq L(x) \cap L(z)$$

*for every other parent z of x. We say that $\mathbb{S}$ **has dominant parents** if every agent x with parents has a dominant parent.*

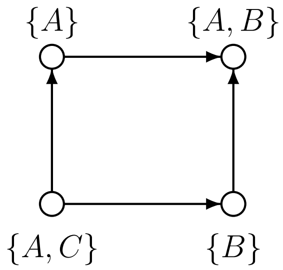
If an agent has only one parent, then that parent is trivially a dominant parent. Therefore every signature tree has dominant parents. But an agent with more than one parent may or may not have a dominant parent. Example 1.2 gives signature a network on a directed acyclic graph that has the Peak Property, but does not have dominant parents.

Example 1.3 has the Peak Property and has dominant parents. Here are two more examples.

Example 5.2. Both $\mathbb{R}$ and $\mathbb{S}$ are signature networks on a directed acyclic graph. $\mathbb{R}$ has the Peak Property but the lower left agent lacks a dominant parent. $\mathbb{S}$ has the Peak and has dominant parents. $\mathbb{R}$ is not report complete, and $\mathbb{S}$ is report complete.



$\mathbb{R}$



$\mathbb{S}$

Theorem 5.3. *Let $\mathbb{S}$ be a signature network on a directed acyclic graph. The following are equivalent:*
(i) $\mathbb{S}$ is report complete.
(ii) $\mathbb{S}$ has the Peak Property and has dominant parents.
(iii) $\mathbb{S}$ contains a signature tree $\mathbb{T}$.

To prove that (ii) implies (iii) we will show that when (ii) holds, the following algorithm will produce a signature tree $\mathbb{T}$ contained in $\mathbb{S}$.

Algorithm 5.4. *GET.TREE.1*($\mathbb{S}$)*:*

- List the edges,
$$E = \langle (x_1, y_1), \ldots, (x_n, y_n) \rangle.$$
- For each $k \leq n$, if y_k is a dominant parent of x_k and there is no integer $j < k$ such that $x_j = x_k$ and the edge (x_j, y_j) is green, then color the edge (x_k, y_k) green, and otherwise color the edge (x_k, y_k) red.
- Output the signature tree $\mathbb{T} = (V, F, L(\cdot))$ where F is the set of all green edges.

Proof of Theorem 5.3. Let d be the unique decider in (V, E). By Fact 1.6 and Corollary 4.1, (iii) implies (i).

We now assume (ii) fails and prove that (i) fails. By Theorem 3.5, $\mathbb{S}$ has the Peak Property. Since (ii) fails, there is an agent $z \neq d$ that does not have a dominant parent. Then z has a parent x such that $\sigma = L(x) \cap L(z)$ is maximal among parents of z. But x is not a dominant parent of z, so z has another parent y such that $L(x)$ does not contain $\tau = L(y) \cap L(z)$. Since σ is maximal, there is no parent u of z such that $(\sigma \cup \tau) \subseteq L(u)$. Since (V, E) has no cycles, there is no directed path from x to z, and no directed path from y to z.

Let $\mathbb{K}$ be the knowledge base over $\mathbb{S}$ such that

$$\mathcal{K}(x) = \{Ev(\sigma)\}, \quad \mathcal{K}(y) = \{Ev(\tau)\}, \quad \mathcal{K}(z) = \{\neg Ev(\sigma) \vee \neg Ev(\tau)\},$$

and all the other knowledge bases are empty. We note that $\mathcal{K}(V) \vdash \bot$. We will show that $\bot$ is not report provable at d in $\mathbb{K}$, so that $\mathbb{K}$ is not report complete at d.

It is clear that $Ev(\sigma) \wedge Ev(\tau)$ is consistent.

Claim. For each $n \geq 0$, whenever a sentence B is n-reportable to an agent v in $\mathbb{K}$, we must have $Ev(\sigma) \wedge Ev(\tau) \vdash B$, and either

- $\vdash B$, or
- there is a path from x to v, or
- there is a path from y to v.

Proof of Claim: We argue by induction on n.

$n = 0$: Suppose B is 0-reportable along an edge (u, v) in $\mathbb{K}$. Then $\mathcal{K}(u) \vdash B$. If $u \notin \{x, y, z\}$ then $\mathcal{K} = \{\top\}$, so $\vdash B$. If $u = z$, then $\neg Ev(\sigma) \vee \neg Ev(\tau) \vdash B$. But v is a parent of z, so there must be a symbol in $S \in \sigma \cup \tau$ that does not belong to $L(v)$. Say $S \in \sigma$. Then $\neg Ev(\sigma) \vdash B$ and S does not occur in B, so by Lemma 3.4 we again have $\vdash B$. If $u = x$, then (x, v) is a path from x to v, and $Ev(\sigma) \vdash B$, so $Ev(\sigma) \wedge Ev(\tau) \vdash B$. The case $u = y$ is similar.

Now suppose that $n > 0$ and the Claim holds for $n-1$. Also suppose B is n-reportable along an edge (u, v). Then $B \in [L(u) \cap L(v)]$, and there is a set $\mathcal{R}$ of sentences such that each $A \in \mathcal{R}$ is $n-1$-reportable to u and $\mathcal{K}(u) \cup \mathcal{R} \vdash B$. By the Claim for $n-1$, we have $Ev(\sigma) \wedge Ev(\tau) \vdash A$ for each $A \in \mathcal{R}$, and hence

$$\mathcal{K}(u) \cup \{Ev(\sigma) \wedge Ev(\tau)\} \vdash B.$$

Assume first that there is no path from either x or y to u. Using the Claim for $n - 1$, we have $\vdash A$ for each $A \in \mathcal{R}$, so $\mathcal{K}(u) \vdash B$ and hence B is 0-reportable along (u, v). The Claim for $n = 0$ now gives the required conditions.

Now assume that there is a path from x to u. Then there is a path from x to v. Since (V, E) is acyclic, we have $u \neq z$, so $Ev(\sigma) \wedge Ev(\tau) \vdash \bigwedge \mathcal{K}(u)$. Therefore $Ev(\sigma) \wedge Ev(\tau) \vdash B$.

Similarly, if there is a path from y to u, then there is a path from y to v and $Ev(\sigma) \wedge Ev(\tau) \vdash B$.

This completes the induction, and the Claim is proved.

By the Claim with $v = d$, any sentence that is reportable to d in $\mathbb{K}$ must be provable from $Ev(\sigma) \wedge Ev(\tau)$. Since z has parents, $d \neq z$. Therefore $\bigwedge \mathcal{K}(d)$ is also provable from $Ev(\sigma) \wedge Ev(\tau)$. Hence any sentence that is report provable at d is provable from $Ev(\sigma) \wedge Ev(\tau)$. But $Ev(\sigma) \wedge Ev(\tau)$ is consistent, so $\perp$ is not report provable at d in $\mathbb{K}$. This completes the proof that if (ii) fails then (i) fails.

Finally, we assume (ii) and prove (iii). We show that *GET.TREE.1*($\mathbb{S}$) outputs a signature tree contained in $\mathbb{S}$. By Lemma 6.5, for every agent $x \neq d$ there is a first edge (x, y) such that y is a dominant parent of x. Put $p(x) = y$. Let

$$\mathbb{T} = (V, F, L(\cdot))$$

where F is the set of edges

$$F = \{(x, p(x)) : x \neq d\}.$$

Then $F \subseteq E$, and hence $\mathbb{S}$ contains $\mathbb{T}$. Each agent $x \neq d$ has a unique parent in (V, F), so (V, F) is a tree. Let S be a symbol in $L(V)$ and write $V[S]$ for $V\{S\}]$. Let $x, z \in V[S]$ and $x \neq z$. Then $V[S]$ contains an agent u, a path from x to u in (V, E), and a path from z to u in (V, E). Hence $V[S]$ contains at least one parent of x in (V, E). Therefore $V[S]$ contains the first dominant parent $p(x)$ of x, and $(x, p(x)) \in F$. It follows that $V[S]$ contains the path $(x_1, \ldots, x_n)$ in (V, F) where $x_1 = x$, $x_n = u$, and $x_{i+1} = p(x_i)$ for each $i < n$. Similarly, $V[S]$ contains a path from z to u in (V, F). This shows that $V[S]$ is a peak in $\mathbb{T}$, so $\mathbb{T}$ has the Peak Property. Thus (ii) implies (iii). □

As a by-product of the proof of Theorem 5.3, we see that for directed acyclic graphs, report completeness for first order logic is equivalent to report completeness for propositional logic. To make this precise, we say that a signature network $\mathbb{S}'$ in propositional logic is a **propositional copy** of a signature network $\mathbb{S}$ in first order logic if $\mathbb{S}'$ is obtained from $\mathbb{S}$ by replacing each symbol in $L(V)$ by a propositional letter in a one-to-one fashion. (Recall that the symbols in $L(V)$ may be constant, relation, or function symbols, and the special symbols $=$ and 0 do not belong to $L(V)$).

Corollary 5.5. *Let $\mathbb{S}$ be a signature network over a directed acyclic graph, and let $\mathbb{S}'$ be a propositional copy of $\mathbb{S}$. Then $\mathbb{S}'$ is report complete if and only if $\mathbb{S}$ is report complete.*

Proof. It is clear that $\mathbb{S}'$ has the Peak Property if and only if $\mathbb{S}$ does, and that $\mathbb{S}'$ has dominant parents if and only if $\mathbb{S}$ does. The proof of

Theorem 5.3 works for propositional logic as well as first order logic. So $\mathbb{S}'$ is report complete if and only if $\mathbb{S}$ is report complete. □

6. The Twin Peaks Property

In this section we introduce the Twin Peaks Property. On a directed acyclic graph, it is equivalent to having dominant parents and the Peak Property. But it is different and better behaved on signature networks with cycles. The property of having dominant parents behaves badly on signature networks with cycles, even when the Peak Property holds.

The following examples shows that for arbitrary signature networks, or even for signature networks with the Peak Property and no cycles of length greater than two, having dominant parents is neither necessary nor sufficient for report completeness.

Example 6.1. *The signature network*

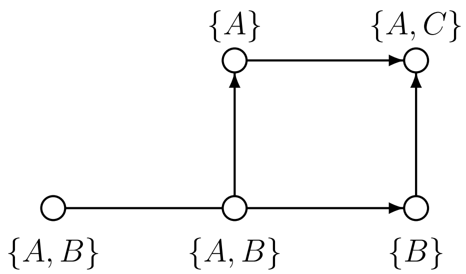
$$\mathbb{R}: \qquad \{B\} \;\text{———}\; \{A, B\} \longrightarrow \{A\}$$

contains the signature tree

$$\mathbb{S}: \qquad \{B\} \longrightarrow \{A, B\} \longrightarrow \{A\},$$

so $\mathbb{R}$ *is report complete.* $\mathbb{R}$ *does not have dominant parents, because the middle agent lacks a dominant parent. Note that* $\mathbb{R}$ *has the Peak Property, and has just one cycle of length two.*

Example 6.2. The signature network below has the Peak Property and has dominant parents. Like the signature network $\mathbb{R}$ in Example 5.2, it is not report complete. Note that the graph has a cycle of length two.



We say that a property is **preserved under adding edges** if whenever $\mathbb{S}$ has the property and $\mathbb{S}$ is contained in $\mathbb{R}$, then $\mathbb{R}$ also has the property. Intuitively, a property is preserved under adding edges if it cannot be destroyed by adding additional communication links. It is easy to see that report completeness is preserved under adding edges.

So is the property of d being a decider, the Peak Property, and the property of containing a signature tree.

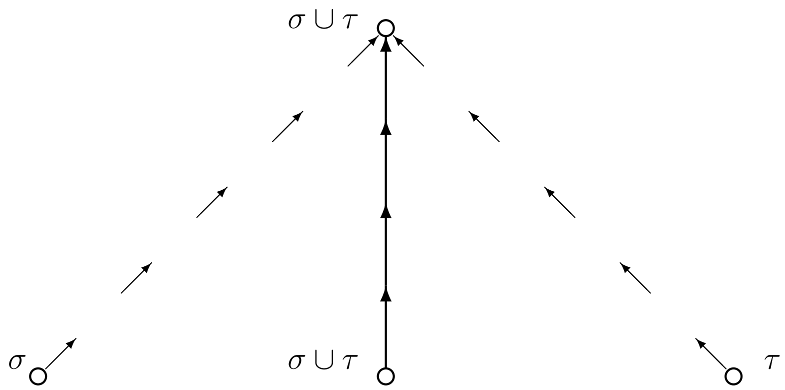
Example 6.1 shows that the property of having dominant parents is not preserved under adding edges, even in the presence of the Peak Property. ($\mathbb{S}$ has dominant parents, but $\mathbb{R}$ does not.)

We now introduce the Twin Peaks Property. It corrects the defects in the property of having dominant parents, but is not as easy to understand. This is because having dominant parents only involves agents and their parents, while the Twin Peaks Property involves paths.

Definition 6.3. *A signature network $\mathbb{S}$ has the* ***Twin Peaks Property*** *if or every two sets of symbols σ, τ and all agents $x \in V[\sigma], y \in V[\tau], z \in V[\sigma \cup \tau]$, there is an agent $u \in V[\sigma \cup \tau]$ such that:*

- *$V[\sigma \cup \tau]$ contains a path from z to u;*
- *either $V[\sigma]$ contains a path from x to u, or $V[\tau]$ contains a path from y to u.*

By taking $\sigma = \tau$ and $x = y$, we see at once from Lemma 2.2 that every signature network with the Twin Peaks Property has the Peak Property. It is easy to see that the Twin Peaks Property is preserved under adding edges. The following picture gives an intuitive view of the Twin Peaks Property. The solid line is a path in $V[\sigma \cup \tau]$, and at least one of the dotted lines is a path in $V[\sigma]$ or $V[\tau]$.



The signature network in Example 6.1 has the Twin Peaks Property but, as we have seen, does not have dominant parents. The signature network in Example 6.2 has the Peak Property and has dominant parents but does not have the Twin Peaks Property.

We now show that the Twin Peaks Property is necessary for report completeness. The proof is similar to the proof of (i) $\Rightarrow$ (ii) in Theorem 5.3.

Theorem 6.4. *Every signature network that is report complete has the Twin Peaks Property.*

Proof. Let $\mathbb{S}$ be a report complete signature network. By Theorem 3.5, $\mathbb{S}$ has the Peak Property.

Assume that $\mathbb{S}$ does not have the Twin Peaks Property. Then there are sets of symbols σ, τ and agents $x \in V[\sigma], y \in V[\tau], z \in V[\sigma \cup \tau]$ such that there is no path from x to an agent $v \in V[\sigma \cup \tau]$ that is entirely within $V[\sigma]$, and also no path from y to an agent $t \in V[\sigma \cup \tau]$ that is entirely within $V[\tau]$. By the Peak Property, $\sigma \neq \sigma \cup \tau$ and $\tau \neq \sigma \cup \tau$. Thus neither $\sigma \supseteq \tau$ nor $\tau \supseteq \sigma$.

As in Theorem 5.3, we let $\mathbb{K}$ be the knowledge base over $\mathbb{S}$ such that

$$\mathcal{K}(x) = \{Ev(\sigma)\}, \quad \mathcal{K}(y) = \{Ev(\tau)\}, \quad \mathcal{K}(z) = \{\neg Ev(\sigma) \vee \neg Ev(\tau)\},$$

and all the other knowledge bases are empty. We note that $\mathcal{K}(V) \vdash \perp$. We will show that $\perp$ is not report provable at d, contradicting the fact that $\mathbb{S}$ is report complete. This will show that $\mathbb{S}$ does have the Twin Peaks Property after all.

Claim. Suppose $n \geq 0$, $v \in V$, and B is a finite conjunction of sentences that are n-reportable to v in $\mathbb{K}$. Then either $\vdash B$, or exactly one of the following holds;

(1) $\mathcal{K}(z) \vdash B$ and $V[\sigma \cup \tau]$ contains a path from z to v;
(2) $\mathcal{K}(x) \vdash B$ and $V[\sigma]$ contains a path from x to v;
(3) $\mathcal{K}(y) \vdash B$ and $V[\tau]$ contains a path from y to v.

Proof of Claim: If any two of conditions 1.–3. hold, then $v \in V[\sigma \cup \tau]$ and either $V[\sigma]$ contains a path from x to v or $V[\tau]$ contains a path from y to v. By the Peak Property, there is an agent w such that $V[\sigma \cup \tau]$ contains a path from v to w and a path from z to w. But then either $V[\sigma]$ contains a path from x to w or $V[\tau]$ contains a path from y to w, contrary to hypothesis. Therefore in all cases, at most one of the conditions 1.–3. can hold. So it suffices to prove that when B is a single sentence n-reportable to v in $\mathbb{K}$, either $\vdash B$ or at least one of the conditions 1.–3. holds. We now argue by induction on n.

$n = 0$: Suppose B is 0-reportable along an edge (u, v) in $\mathbb{K}$. Then $\mathcal{K}(u) \vdash B$. If $u \notin \{x, y, z\}$ then $\mathcal{K} = \{\top\}$, so $\vdash B$. If $u = z$, then $\neg Ev(\sigma) \vdash B$ and $\neg Ev(\tau) \vdash B$. Hence by Lemma 3.4, either $\vdash B$, or every symbol in $\sigma \cup \tau$ occurs in B, so (z, v) is a path in $V[\sigma \cup \tau]$ and 1. holds. If $u = x$, then $Ev(\sigma) \vdash B$, so by Lemma 3.4, either $\vdash B$, or (x, v) is a path in $V[\sigma]$ and 2. holds. The case $u = y$ is similar.

Now suppose that $n > 0$ and the Claim holds for $n-1$. Also suppose B is n-reportable along an edge (u, v) in $\mathbb{K}$. Then $B \in [L(u) \cap L(v)]$, and there is a finite conjunction A of sentences that are $n-1$-reportable to u in $\mathbb{K}$ such that $\mathcal{K}(u) \cup \{A\} \vdash B$. By the Claim for $n-1$, either $\vdash A$ or one of conditions 1.–3. holds for u and A.

Case 0. $\vdash A$. In this case, $\mathcal{K}(u) \vdash B$, so B is 0-reportable to v in $\mathbb{K}$ and the Claim for n follows from the Claim for 0. Hereafter suppose not $\vdash A$.

Case 1. $\mathcal{K}(z) \vdash A$ and $V[\sigma \cup \tau]$ contains a path from z to u. Then $u \in V[\sigma \cup \tau]$, so $u \neq x$ and $u \neq y$. Hence $\mathcal{K}(z) \vdash \bigwedge \mathcal{K}(u)$, so $\mathcal{K}(z) \vdash B$. We also have $(\neg Ev(\sigma)) \vdash A$ and $(\neg Ev(\tau)) \vdash A$, so by Lemma 3.4, every symbol in $\sigma \cup \tau$ occurs in A. Therefore $v \in V[\sigma \cup \tau]$, so $V[\sigma \cup \tau]$ contains a path from z to v.

Case 2. $\mathcal{K}(x) \vdash A$ and $V[\sigma]$ contains a path from x to u. Then $u \in V[\sigma]$, so $u \neq y$. We have already shown that $V[\sigma]$ cannot contain a path from x to an agent in $V[\sigma \cup \tau]$, so $u \neq z$. Hence $\mathcal{K}(x) \vdash \bigwedge \mathcal{K}(u)$, so $\mathcal{K}(x) \vdash B$. By Lemma 3.4, every symbol in σ occurs in A. Therefore $v \in V[\sigma]$, so $V[\sigma]$ contains a path from x to v.

Case 3. $\mathcal{K}(y) \vdash A$ and $V[\sigma]$ contains a path from y to u. Similar to Case 2.

This completes the induction, and the Claim is proved.

By the Claim with $v = d$, we see that if B is reportable to d in $\mathbb{K}$ then either $\mathcal{K}(z) \vdash B$, $\mathcal{K}(x) \vdash B$, or $\mathcal{K}(x) \vdash B$. But each of the sets $\mathcal{K}(z), \mathcal{K}(x)$, and $\mathcal{K}(y)$ is consistent, so $\perp$ is not reportable to d in $\mathbb{K}$. $\square$

The next lemma shows that on a direct acyclic graph, the Twin Peaks Property is equivalent to having both the Peak Property and having dominant parents.

Lemma 6.5. *Let $\mathbb{S}$ be a signature network on a directed acyclic graph with decider d. Then $\mathbb{S}$ has the Twin Peaks Property if and only if it has the Peak Property and has dominant parents.*

Proof. If $\mathbb{S}$ has both the Peak and Dominance Properties, then $\mathbb{S}$ is report complete by Theorem 5.3], and hence $\mathbb{S}$ has the Twin Peaks Property by Theorem 6.4.

Suppose that $\mathbb{S}$ has the Twin Peaks Property. Then $\mathbb{S}$ has the Peak Property. Suppose some agent $z \neq d$ does not have a dominant parent. Then there are two parents x, y of z such that neither of the sets $L(z) \cap L(x), L(z) \cap L(y)$ contains the other, but there is no parent u of z such that $L(z) \cap L(u)$ contains $(L(z) \cap L(x)) \cup (L(z) \cap L(y))$. Let $\sigma = (L(z) \cap L(x))$ and $\tau = (L(z) \cap L(y))$. By the Twin Peaks Property, there is an agent $v \in V[\sigma \cup \tau]$ such that $V[\sigma \cup \tau]$ contains a path from z to v, and either $V[\sigma]$ contains a path from x to v, or $V[\tau]$ contains a path from y to v, say the former. Then $z \neq v$, so the path from z to v in $V[\sigma \cup \tau]$ must pass through some parent u of z. Hence $\sigma \cup \tau \subseteq L(u)$, contrary to hypothesis. Therefore $\mathbb{S}$ has dominant parents. □

Theorem 6.6. *Let $\mathbb{S}$ be a signature network on a directed acyclic graph. The following are equivalent:*
(i) $\mathbb{S}$ is report complete.
(ii) $\mathbb{S}$ has the Twin Peaks Property.
(iii) $\mathbb{S}$ has Peak Property and has dominant parents.
(iv) $\mathbb{S}$ contains a signature tree $\mathbb{T}$.

Proof. By Theorem 5.3 and Lemma 6.5. □

Example 8.4 will show that for arbitrary signature networks, or even for signature networks with no cycles of length greater than two, the Twin Peaks Property does not imply report completeness.

7. Deciders with Large Signatures

In this section we show that a signature network with a decider d that has a "very large signature" is report complete and contains a signature tree. To do this we will look at well-behaved paths in signature networks.

Definition 7.1. *In a signature network $\mathbb{S}$, a* ***good path*** *is a path $(x_0, \ldots, x_n)$ that is contained in $V[L(x_0) \cap L(x_n)]$, and an* ***excellent path*** *is a path $(x_0, \ldots, x_n)$ such that $(x_i, \ldots, x_n)$ is a good path for every $i \leq n$.*

Remark 7.2. *(i) Every excellent path is good.*
(ii) If $(y_0, \ldots, y_p)$ is an excellent path then every tail $(y_j, \ldots, y_p)$ is also an excellent path.
(iii) A path $(x_0, \ldots, x_n)$ is excellent if and only if for all $j < n$, $(L(x_j) \cap L(x_n)) \subseteq L(x_{j+1})$.

Lemma 7.3. *In a signature network $\mathbb{S}$, let d be an agent such that for every agent x there exists a good path from x to d. Then for every agent y there exists an excellent path form y to d.*

Proof. We argue by induction on the cardinality $|L(d) \setminus L(y)|$. If $|L(d) \setminus L(y)| = 0$, then every good path from y to d is excellent, so there exists an excellent path from y to d. Assume that $n > 0$, and there is an excellent path from y to d whenever $|L(d) \setminus L(y)| < n$. Suppose $|L(d) \setminus L(x)| = n$. Let $(x_0, \ldots, x_m)$ be a good path from x to d. Since $x_m = d$, there is a least $j \leq m$ such that $|L(d) \setminus L(x_j)| < n$. By inductive hypothesis there is an excellent path $(y_0, \ldots, y_p)$ from x_j to d. Then $(x_0, \ldots, x_{j-1}, y_0, \ldots, y_p)$ is an excellent path from x to d. □

Theorem 7.4. *Let $\mathbb{S}$ be a signature network on a directed graph, and let d be a decider such that every symbol that is in $L(x)$ for more than one agent $x \in V$ belongs to $L(d)$. Then the following are equivalent:*

(i) *For every agent x there exists a good path from x to d.*
(ii) *For every agent x there exists an excellent path from x to d.*
(iii) *$\mathbb{S}$ contains a signature tree $\mathbb{T}$ with decider d.*

This theorem will be needed twice in Section 9, once with a connected graph, and once with a directed graph. The hard part is (ii) implies (iii). The proof of the theorem will show that the following algorithm will output a signature tree contained in $\mathbb{S}$ with decider d provided that (ii) holds.

Algorithm 7.5. *GET.TREE.2*($\mathbb{S}, d$)*:*

- Color every edge $(x, d) \in E$ green, and color all other edges red.
- Until every agent x is on a green edge, find an edge (x, y) such that:
  - x is not on a green edge,
  - y is on a green edge,
  - $L(x) \cap L(d) \subseteq L(y)$,

  and color the edge (x, y) green.
- When every agent x is on a green edge, output the signature tree $\mathbb{T} = (V, F, L(\cdot))$, where F is the set of green edges.

Proof of Theorem 7.4. Lemma 7.3 shows that (i) implies (ii).

(iii) implies (i): The signature tree $\mathbb{T}$ has the Peak Property, so for every $x \in V$ there is a good path from x to d in $\mathbb{T}$. This path is also a good path in $\mathbb{S}$.

To prove that (ii) implies (iii), we show that the algorithm works. Note that at each stage in the algorithm, for every green edge (x, y) we have $L(x) \cap L(d) \subseteq L(y)$, so $y \in V[L(x) \cap L(d)]$. Therefore, every path from x to d consisting of green edges is excellent. By hypothesis, for every agent $x_0 \neq d$ there is an excellent path $(x_0, x_1, \ldots, x_n)$ from x to d. If x_0 is not on a green edge, then (x_0, x_1) is red and (x_{n-1}, x_n) is green, so there exists $i < n$ such that $x = x_i$ is not on a green edge and $y = x_{n+1}$ is on a green edge. Since the path is excellent, $L(x) \cap L(d) \subseteq L(y)$. The set E of edges is finite, so the algorithm will eventually terminate and return a signature network $\mathbb{T}$ that is contained in $\mathbb{S}$. By induction on the steps, for each x there is at most one y such that $(x, y) \in F$, and if $(x, y) \in F$ then there is a path from x to d in (V, F). At the end, for each x there exists y such that $(x, y) \in F$, so (V, F) is a tree with decider d. Moreover, the unique path from x to d

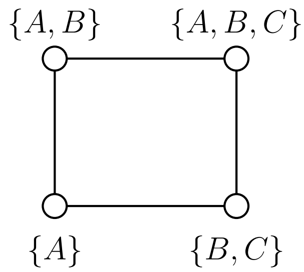
in $\mathbb{T}$ is excellent. To prove that $\mathbb{T}$ has the Peak Property, consider a set of symbols σ and suppose $u, v \in V[\sigma]$ and $u \neq v$. Then each symbol in σ occurs in both signatures $L(u)$ and $L(v)$, and therefore must be in $L(d)$. Hence $d \in V[\sigma]$. Since the paths from u to d and v to d in $\mathbb{T}$ are excellent, they must be contained in $V[\sigma]$. This proves that $\mathbb{T}$ has the Peak Property, and thus is a signature tree. □

Corollary 7.6. *Let $\mathbb{S}$ satisfy the hypotheses of Theorem 7.4, and let $\mathbb{S}'$ be a propositional copy of $\mathbb{S}$. Then $\mathbb{S}'$ is report complete if and only if $\mathbb{S}$ is report complete.*

Proof. It is clear that $\mathbb{S}'$ has the Peak Property if and only if $\mathbb{S}$ does, and that $\mathbb{S}'$ has Linked Chain Property if and only if $\mathbb{S}$ does. Moreover, the proof of Theorem 5.3 works for propositional logic as well as first order logic. □

The following two examples illustrate Theorem 7.4. In these examples, the graph has cycles, so Theorem 6.6 is not applicable.

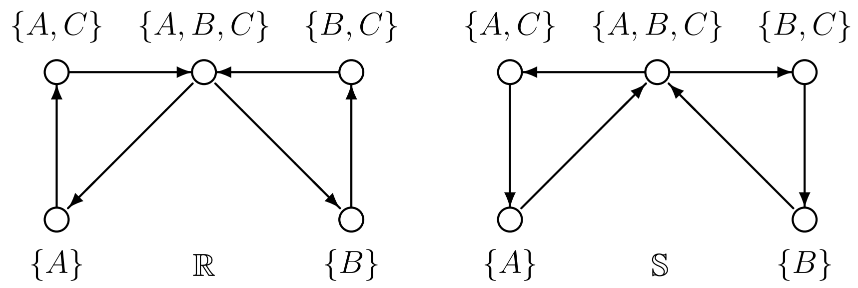
Example 7.7. $\mathbb{S}$ is on a connected graph. Let d be the upper right agent. $L(d)$ contains every symbol, and there is an excellent path from any agent to d. By Theorem 7.4, $\mathbb{S}$ is report complete and contains a signature tree with decider d.



Example 7.8. In both of the signature networks $\mathbb{R}$ and $\mathbb{S}$, every agent is a decider. Moreover, $L(d)$ contains every symbol, where d be the top center agent.

In $\mathbb{R}$, there is an excellent path from every agent to d. Theorem 7.4 says that $\mathbb{R}$ contains a signature tree with decider d, and hence is report complete. To get this tree, remove the two diagonal edges.

In $\mathbb{S}$, all the arrows are reversed. $\mathbb{S}$ has the Peak Property, but $\mathbb{S}$ does not have the Twin Peaks Property, and hence is not report complete. There is no good path from the agent with $\{A, C\}$ to d, and $\mathbb{S}$ does not contain a signature tree.



8. The Linked Chain Property

In this section we give another necessary condition for report completeness, the Linked Chain Property. This property involves only the set of agents and signatures, and does not depend on the edges in the directed graph.

Definition 8.1. *A* ***linked chain*** *in a signature network* $\mathbb{S}$ *is a finite sequence of sets of symbols* $(\sigma_0, \ldots, \sigma_p)$ *such that:*

- $p \geq 2$;
- σ_j *is a non-empty subset of* $L(V)$ *for each* $j \leq p$;
- *the sets* σ_j *are pairwise disjoint;*
- *for each* $j < p$ *there is an agent* x_j *such that* $L(x_j) \supseteq \sigma_j \cup \sigma_{j+1}$;
- *there is an agent* x_p *such that* $L(x_p) \supseteq \sigma_p \cup \sigma_0$.

We say that a linked chain $(\sigma_0, \ldots, \sigma_p)$ *in* $\mathbb{S}$ *is* ***good*** *if there are three distinct* i, j, k *in* $\{0, \ldots, p\}$ *and an agent* $x \in V$ *such that*

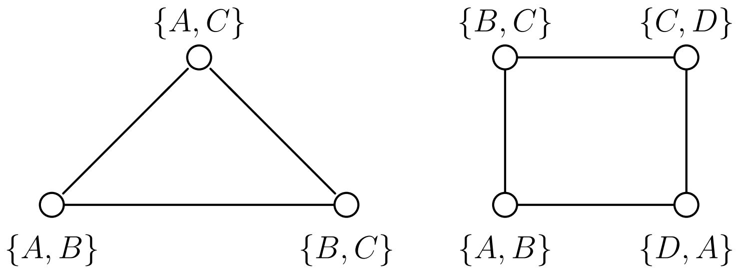
$$L(x) \supseteq \sigma_i \cup \sigma_j \cup \sigma_k,$$

and ***bad*** *otherwise.*

$\mathbb{S}$ *has the* ***Linked Chain Property*** *if every linked chain* $(\sigma_0, \ldots, \sigma_p)$ *in* $\mathbb{S}$ *is good.*

Note that if $\mathbb{S}$ has an agent d such that $L(d)$ contains every symbol occurring in more than one signature (as in the preceding section), then $\mathbb{S}$ automatically has the Linked Chain Property. Here are two examples of signature networks that have the Twin Peaks Property but are not report complete and do not have the Linked Chain Property.

Example 8.2. The graphs are a triangle and a square with edges in both directions, and signatures as shown.



The triangle is not report complete, because the knowledge base

$$\mathcal{K}(x) = \{Ev(B, C)\}, \quad \mathcal{K}(y) = \{\neg Ev(A, C)\}, \quad \mathcal{K}(d) = \{Ev(A, B)\}$$

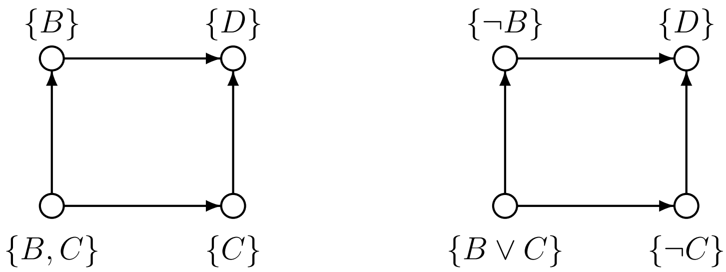
is report consistent but $\mathcal{K}(V)$ is not consistent. The triple of sets of symbols

$$(\{A\}, \{B\}, \{C\})$$

is a bad linked chain in $\mathbb{S}$, so the Linked Chain Property fails.

The square is similar, but with a bad linked chain of length 4.

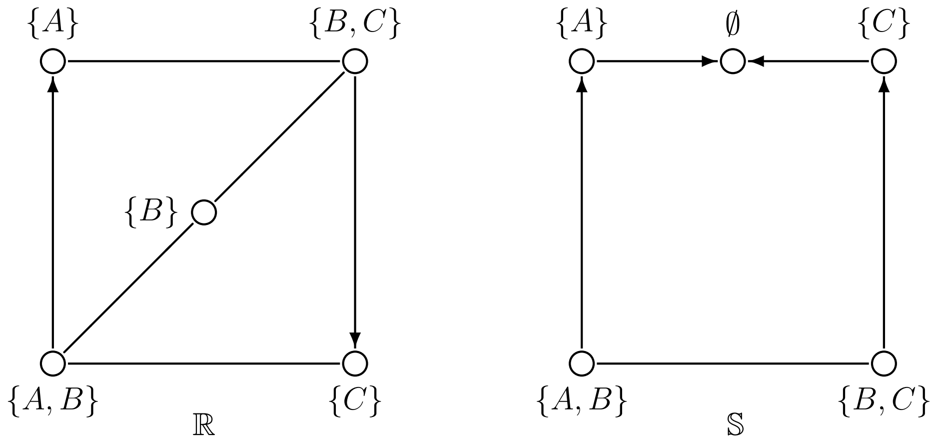
Example 8.3. This is a signature network on a directed acyclic graph that has the Linked Chain Property and the Peak Property, but does not have the Twin Peaks Property, and hence is not report complete. The false sentence $\perp$ is provable but not report provable. If an edge is added from the upper right corner to the lower left corner, we obtain a signature network with the same properties on a directed graph with cycles.



$\mathbb{S}$ $\mathbb{K}$

The following example shows that the Twin Peaks and Linked Chain Properties together do not imply report completeness.

Example 8.4. Both the signature networks $\mathbb{R}$ and $\mathbb{S}$ have the Twin Peaks Property and the Linked Chain Property but are not report complete and do not have the Dominance Property. Note that $\mathbb{S}$ has just one cycle, and that cycle has length two. By inserting a new agent with signature $\{A, B\}$ on the left sides of $\mathbb{R}$ and $\mathbb{S}$, and a new agent with signature $\{B, C\}$ on the right sides of $\mathbb{R}$ and $\mathbb{S}$, we get examples that also have dominant parents.



The following theorem shows that the Linked Chain Property is necessary for report completeness.

Theorem 8.5. *If $\mathbb{S}$ is a report complete signature network, then $\mathbb{S}$ has the Linked Chain Property.*

Proof. Suppose $\mathbb{S}$ does not have the Linked Chain Property. Let $(\sigma_0, \dots, \sigma_p)$ be a bad linked chain in $\mathbb{S}$. For convenience put $\sigma_{p+1} = \sigma_0$.

For each $j \le p$ let x_j be an agent such that $L(x_j) \supseteq (\sigma_j \cup \sigma_{j+1})$. For each $j \le p$ let A_j be the sentence $Ev(\sigma_j \cup \sigma_{j+1})$. Let $\mathbb{K}$ be the knowledge base over $\mathbb{S}$ such that

$$\mathcal{K}(x_j) = \{A_j\} \text{ for each } j < p,$$

$$\mathcal{K}(x_p) = \{\neg A_p\}, \text{ and } \mathcal{K}(x) = \emptyset \text{ for } x \notin \{x_0, \dots, x_p\}.$$

Since each symbol is counted twice, the combined theory $\mathcal{K}(V)$ is inconsistent. However, for each $j \le p$, the set of sentences $\mathcal{K}(V) \setminus \mathcal{K}(x_j)$ is consistent.

By Lemma 3.4, for each $j \le p$ and each sentence B, if $A_j \vdash B$ then either $\vdash B$ or every symbol in $\sigma_j \cup \sigma_{j+1}$ occurs in B.

It follows by induction on n that for each $j \le p$ and every agent x, either there is a unique $j \le p$ such that $L(x) \supseteq \sigma_j \cup \sigma_{j+1}$ and every sentence that is n-reportable to x is provable from A_j, or nothing is n-reportable to x. Therefore $\mathbb{K}$ is report consistent, and hence $\mathbb{S}$ is not report complete. □

Corollary 8.6. *A signature network $\mathbb{S}$ on a directed acyclic graph that has the Twin Peaks Property has the Linked Chain Property.*

Proof. By Theorems 8.5 and 3.5. □

9. Connected Graphs and Trees

By a **connected (undirected) graph** we mean a pointed graph (V, E) where E symmetric, that is, for each pair x, y of distinct agents, $(x, y) \in E$ if and only if $(y, x) \in E$ (so (x, y, x) is a 2-cycle).

Note that in a connected graph, there is a path from x to y if and only if there is a path from y to x. Since a connected graph is pointed and thus has at least one decider, it follows that there is a path from any agent to any other agent, and hence every agent is a decider. For a signature network on a connected graph, the Peak Property holds if and only $V[\sigma]$ is connected for every set of symbols σ. It easily follows that every signature network on a connected graph that has the Peak Property also has the Twin Peaks Property.

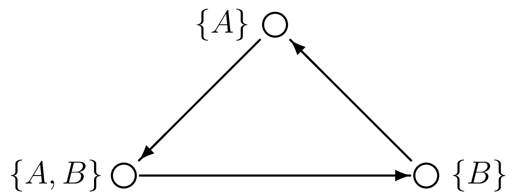
Lemma 9.1. *Let $\mathbb{S}$ be a signature network on a connected graph. If $\mathbb{S}$ contains a signature tree, then for every agent $d \in V$, $\mathbb{S}$ contains a signature tree with decider d.*

Proof. Suppose $\mathbb{S}$ contains a signature tree $\mathbb{T}$ with decider e, and let $d \in V$. Let $\mathbb{T}'$ be the signature network obtained from $\mathbb{T}$ by reversing the arrows on the path from d to e and keeping everything else the same. It is easily seen that $\mathbb{S}$ contains $\mathbb{T}'$, that $\mathbb{T}'$ has the Peak Property, and that d is the unique decider for $\mathbb{T}'$. □

This is illustrated by Example 7.7. The graph is connected and $\mathbb{S}$ contains a signature tree whose decider is the upper right agent, so for every agent x it contains a signature tree with decider x.

The following an example shows that Lemma 9.1 does not hold for all signature networks.

Example 9.2. The signature network $\mathbb{S}$ shown in the picture is report complete, and every agent is a decider, but $\mathbb{S}$ only contains a signature tree where the lower right agent is a decider.



Our main result in this section, Theorem 9.7, will show that if $\mathbb{S}$ is a signature network on a connected graph and $\mathbb{S}$ has the Peak Property and the Linked Chain Property, then $\mathbb{S}$ contains a signature tree and is report complete. Our plan for doing this is as follows. Pick an agent d. First, we will show in Lemma 9.3 every agent x connected to d by a an excellent path. This, together with Theorem 7.4, gives us Theorem 9.7 in the special case that some $L(d)$ is so big that it contains every symbol that occurs in more than one signature. Then we will show in Theorem 9.5 that when $L(d)$ is not that big, the signature network $\mathbb{S}$ can be partitioned into a sequence of smaller signature networks $\mathbb{S}_0, \dots, \mathbb{S}_k$ in such a way that each $\mathbb{S}_i$ has the Peak Property and the Linked Chain Property, and each symbol that is not in $L(d)$ appears in exactly one of the $\mathbb{S}_i$. All these results will be used in the proof of Theorem 9.7. Arguing by induction on the number of agents, we assume that each $\mathbb{S}_i$ contains a signature tree $\mathbb{T}_i$. We will then construct another signature tree $\mathbb{T}'$ whose agents are the indices $i \leq k$, and use $\mathbb{T}'$ to combine the signature trees $\mathbb{T}_i$ into a big signature tree $\mathbb{T}$ that is contained in $\mathbb{S}$.

Lemma 9.3. *Let $\mathbb{S}$ be a signature network on a connected graph such that $\mathbb{S}$ has the Peak Property. Then for each $x, y \in V$ there exists an excellent path from x to y.*

Proof. Since $\mathbb{S}$ has the Peak Property and E is symmetric, for each $x, y \in V$ there is a good path from x to y. Then by Lemma 7.3, for each x, y there is an excellent path from x to y. □

Given a signature network $\mathbb{S}$ on a connected graph, and an agent d in V, let

$$\alpha = \bigcap \{(L(d) \cap L(x)) \,:\, x \in V \text{ and } x \neq d\},$$

$$\beta = \bigcap \{(L(x) \cap L(y) \,:\, x, y \in V \text{ and } x \neq y\} \setminus \alpha.$$

Then $L(d)$ contains α and is disjoint from β. Let $\sim$ be the finest equivalence relation on β such that $P \sim Q$ whenever there is an agent $x \in V$ with $\{P, Q\} \subseteq L(x)$. Let $\{\beta_1, \dots, \beta_k\}$ be the set of equivalence classes under $\sim$. Note that

$$\beta = \beta_1 \uplus \cdots \uplus \beta_k,$$

and for each agent $x \neq d$, the set $L(x) \cap \beta$ is contained in some β_i. When we need to show the dependence on $\mathbb{S}$ and d, we write $\alpha(\mathbb{S}, d)$ for α, and similarly for β and β_i.

By the **restriction** of $\mathbb{S}$ to a connected set $U \subseteq V$ we mean the signature network $\mathbb{R} = (U, F, L'(\cdot))$ where $F = E \cap (U \times U)$ and $L'(x) = L(x)$ for all $x \in U$. Note that (U, F) is a connected graph and $\mathbb{R}$ is a signature network over (U, F). We define $L(U) = \bigcup_{x \in U} L(x)$, and for a set σ of symbols we define $U[\sigma] = \{x \in U \,:\, \sigma \subseteq L(x\}$.

Definition 9.4. *A* ***report decomposition*** *of* $\mathbb{S}$ *at* d *is a sequence of signature networks* $\mathbb{S}_0, \ldots, \mathbb{S}_k$ *such that, with* $\alpha = \alpha(\mathbb{S}, d)$ *etc., we have*

- *Each* $\mathbb{S}_i$ *is a restriction of* $\mathbb{S}$ *to a connected set* V_i *and has the Peak Property and the Linked Chain Property.*
- *The sets of agents* V_i *are pairwise disjoint and have union* V,

$$V = V_0 \uplus \cdots \uplus V_k.$$

- $d \in V_0$ *and* $L(V_0) \cap \beta = \emptyset$.
- *For* $1 \leq i \leq k$, $L(V_i) \cap \beta \subseteq \beta_i$.
- *For each* $1 \leq i \leq k$ *there is an agent* $v \in V_i$ *such that*

$$(L(x) \cap \alpha) \subseteq (L(v) \cap \alpha)$$

  for all $x \in V_i$. *We say that* v *is* ***high*** *in* V_i.

Theorem 9.5. *Let* $\mathbb{S}$ *be a signature network* $\mathbb{S}$ *on a connected graph* (V, E) *such that* $\mathbb{S}$ *has the Peak Property and Linked Chain Property. Then* $\mathbb{S}$ *has a report decomposition at every agent* $d \in V$.

The proof of Theorem 9.5 will show that the following algorithm returns a report decomposition of $\mathbb{S}$ at d whenever $\mathbb{S}$ is on a connected graph and has the Peak Property and Linked Chain Property.

Algorithm 9.6. *DECOMP*$(\mathbb{S}, d)$:

- Initially all agents are colored red.
- Find the sets of symbols $\alpha, \beta, \beta_1, \ldots, \beta_k$ for $(\mathbb{S}, d)$.
- Color d green and put d in V_0.
- For each $x \in V$ such that $L(x) \cap \beta_i \neq \emptyset$, color x green and put x in V_i.
- Do the following until all agents are green:
  - Take an agent x that is still red, and choose an excellent path $(x_0, \ldots, x_p)$ from x to d.
  - Let x_ℓ be the first green agent on the path.
  - Color the agents $x = x_0, \ldots, x_{\ell-1}$ green and put them into the set V_i that contains x_ℓ.
- Return the sequence $\mathbb{S}_0, \ldots, \mathbb{S}_k$, where $\mathbb{S}_i$ is the restriction of $\mathbb{S}$ to V_i.

Proof of Theorem 9.5. Let $\alpha = \alpha(\mathbb{S}, d)$, and similarly for β and $\beta_i, i = 0, \ldots, k$. By Lemma 9.3, there is an excellent path from each agent x to d, and it follows that Algorithm *DECOMP*$(\mathbb{S}, d)$ will terminate and return a sequence $\mathbb{S}_0, \ldots, \mathbb{S}_k$. We must show that this sequence is a report decomposition of $\mathbb{S}$ at d. Let

$$U_0 = \{x \in V : L(x) \cap \beta = \emptyset\},$$

and for $1 \le i \le k$ let

$$U_i = \{x \in V : \emptyset \neq L(x) \cap \beta \subseteq \beta_i\}.$$

Then $d \in U_0$, and $U_i, i \le k$ forms a partition of V into disjoint non-empty sets.

It is easy to see that:

- $V_0, \ldots, V_k$ partitions V into non-empty disjoint sets.
- $V_0 \subseteq U_0$, and for $i \ge 1$, $V_i \supseteq U_i$.
- $L(V_0) \cap \beta = \emptyset$, and for $1 \le i \le k$, $L(V_i) \cap \beta \subseteq \beta_i$.

Claim 1. For each $i \le k$, $\mathbb{S}_i$ is a signature network on a connected graph.

Proof of Claim 1: It is clear that for each $x \in V_0$, V_0 contains a path from x to d, so (V_0, E_0) is connected.

Let $1 \le i \le k$ and let $x, y \in V_i$. By the definition of V_i, V_i contains a path from x to an agent $x' \in U_i$, and path from y to an agent $y' \in U_i$. There is a finite sequence of agents $(x_0, \ldots, x_m)$ in U_i such that $x' = x_0, y' = x_m$, and for each $j < m$ there is a symbol $S_j \in L(x_j) \cap L(x_{j+1}) \cap \beta$. By the Peak Property for $\mathbb{S}$, for each $j < m$ the set $V[\{S_j\}]$ contains a path from x_j to x_{j+1}. But $V[\{S_j\}] \subseteq U_i$, so U_i contains a path from x_j to x_{j+1}, and hence U_i contains a path from x' to y'. Therefore V_i contains a path from x to y, so (V_i, E_i) is connected. □

Claim 2. For $i \le k$ there exists $v_i \in U_i$ such that $(L(v_i) \cap \alpha) \supseteq (L(x) \cap \alpha)$ for all $x \in U_i$, and $v_0 = d$.

Proof of Claim 2: It is clear that the claim holds for $i = 0$.

Suppose $1 \le i \le k$ and the claim fails for i. Then there are $x, y \in U_i$ such that $L(x) \cap \alpha$ and $L(y) \cap \alpha$ are maximal but different. Hence there is no agent $z \in U_i$ such that $L(z)$ contains both $L(x) \cap \alpha$ and $L(y) \cap \alpha$. So there is no agent $z \in V$ such that $L(z)$ contains $L(x) \cap \alpha$, $L(y) \cap \alpha$, and some symbol $S \in \beta_i$. For each $P \in L(x) \cap \beta$ and $Q \in L(y) \cap \beta$, we have $P, Q \in \beta_i$, and therefore there is a sequence of symbols $(P_0, \ldots, P_m)$ in β_i such that $P_0 = P, P_m = Q$, and for each $j < m$ there is an agent $z_j \in V$ with $\{P_j, P_{j+1}\} \subseteq L(z_j)$. It follows that there is a least integer p such that there is a sequence of symbols $(S_0, \ldots, S_p)$ in β_i and a sequence of agents $(x', u_0, \ldots, u_{p-1}, y')$ in V where:

- $L(x')$ contains $L(x) \cap \alpha$ and S_0;
- $L(y')$ contains $(L(y) \setminus L(x)) \cap \alpha$ and S_p;
- for $0 \le j < p$, $L(u_j)$ contains S_j and S_{j+1}.

Since $L(x'), L(y')$, and each $L(u_j)$ contains a symbol in β_i, x', y', and each u_j must belong to U_i. Since p is least, there is no agent $z \in V$ such that $L(z)$ contains two non-consecutive symbols in the list $(S_0, \ldots, S_p)$ (otherwise one or more symbol can be skipped and p would not be least). For the same reason, there is no agent $z \in V$, such that $L(z)$ contains $L(x) \cap \alpha$ and a symbol S_j with $j > 0$, and there is no agent $z \in V$, such that $L(z)$ contains $(L(y) \setminus L(x)) \cap \alpha$ and a symbol S_j with $j < p$. Since there is also no agent $z \in V$ such that $L(z)$ contains $L(x) \cap \alpha$, $L(y) \cap \alpha$, and some symbol $S \in \beta_i$, the sequence of sets

$$(L(x) \cap \alpha, \{S_1\}, \ldots, \{S_p\}, ((L(y) \setminus L(x)) \cap \alpha)$$

is a bad linked chain in $\mathbb{S}$, contradicting the hypothesis that $\mathbb{S}$ has the Linked Chain Property. □

Claim 3. For each $i \leq k$, v_i is high in V_i.

Proof of Claim 3: By the construction of V_i, for each $x \in V_i$ there is an excellent path $(y_0, \ldots, y_n)$ from x to d such that some y_j belongs to U_i. Then $(L(x) \cap L(d)) \subseteq (L(y_j) \cap L(d))$, so by Claim 2,

$$(L(x) \cap \alpha) \subseteq (L(y_j) \cap \alpha) \subseteq (L(v_i) \cap \alpha).$$

□

Claim 4. For each $i \leq k$, $\mathbb{S}_i$ has the Linked Chain Property.

Proof of Claim 4: $\mathbb{S}_0$ has the Linked Chain Property because any symbol that belongs to more than one signature in $\mathbb{S}_0$ belongs to $L(d)$. Let $i \geq 1$, and suppose $(\sigma_0, \ldots, \sigma_p)$ is a linked chain in $\mathbb{S}_i$. Then $(\sigma_0, \ldots, \sigma_p)$ is a linked chain in $\mathbb{S}$. Since $\mathbb{S}$ has the Linked Chain Property, there is an agent $x \in V$ and integers j, k, ℓ such that $0 \leq j < k < \ell \leq p$ and $L(x) \supseteq \sigma_j \cup \sigma_k \cup \sigma_\ell$. If $x \in V \setminus V_i$, then by Claim 3, any symbol that belongs to $L(x)$ and the signature of an agent in V_i belongs to $L(v_i) \cap \alpha$, so $L(v_i) \supseteq \sigma_j \cup \sigma_k \cup \sigma_\ell$. This shows that $(\sigma_0, \ldots, \sigma_p)$ is a good linked chain in $\mathbb{S}_i$. □

Claim 5. For each $i \leq k$, $\mathbb{S}_i$ has the Peak Property.

Proof of Claim 5: For each $x \in V_0$, V_0 contains an excellent path from x to d, and it follows that $\mathbb{S}_0$ has the Peak Property.

Let $i \geq 1$, let σ be a non-empty set of symbols, and let $x, y \in V_i[\sigma]$ and $x \neq y$. Then $\sigma \subseteq (L(x) \cap L(y))$, so $\sigma \subseteq ((L(v_i) \cap \alpha) \cup \beta_i)$. We also have $x, y \in V[\sigma]$, and since $\mathbb{S}$ has the Peak Property, $V[\sigma]$ contains a path from x to y. If σ meets β_i, then $V[\sigma] = V_i[\sigma]$, so $V_i[\sigma]$ contains a path from x to y.

Suppose instead that $\sigma \cap \beta_i = \emptyset$. Then $\sigma \subseteq (L(v_i) \cap \alpha)$. Since there is an excellent path from x to d whose first element outside of U_0 belongs to U_i, we may assume that $x \in U_i$. Similarly, we may assume that $y \in U_i$. Then there are symbols P, Q in β_i such that $L(x)$ contains σ and P, and $L(y)$ contains σ and Q. So there is a sequence of symbols

$(P_0, \ldots, P_m)$ in β_i such that $P_0 = P, P_m = Q$, and for each $j < m$ there is an agent $z_j \in V$ with $\{P_j, P_{j+1}\} \subseteq L(z_j)$. It follows that there is a least integer m such that there exists a sequence of symbols $(S_1, \ldots, S_m)$ in β_i and a sequence of agents $(u_0, \ldots, u_m)$ in V where:

- $L(u_0)$ contains σ, P, and S_1 (possibly $P = S_1$);
- $L(u_m)$ contains σ, Q, and S_m (possibly $Q = S_m$);
- for $0 < j < m$, $L(u_j)$ contains S_j and S_{j+1}.

Case 1: $m = 1$. In this case we have $S_1 = S_m$. By the Peak Property for $\mathbb{S}$, $V[\sigma \cup \{P\}]$ contains a path from x to u_0, $V[\sigma \cup \{S_1\}]$ contains a path from u_0 to u_1, and $V[\sigma \cup \{Q\}]$ contains a path from u_1 to y. Since P, Q, and S_1 belong to β_i, all of these paths are within V_i, and together they form a path from x to y in $V_i[\sigma]$.

Case 2: $m > 1$. In this case, the sequence

$$(\sigma, \{S_1\}, \ldots, \{S_m\})$$

is a linked chain in (V, E). Since m is least, there is no $z \in V$ such that $L(z)$ contains two non-consecutive symbols in the sequence $(S_1, \ldots, S_m)$. Take the least h such that $1 < h \leq m$ and there exists an agent $w \in V$ where $L(w)$ contains σ and S_h. Then the sequence

$$(\sigma, \{S_1\}, \ldots, \{S_h\})$$

is a linked chain in (V, E). By the Linked Chain Property for $\mathbb{S}$, there is an agent $t \in V$ such that $L(t)$ contains three distinct sets in this linked chain. Since h is least, the only possibility is that $m = h = 2$ and $L(t)$ contains σ, S_1, and S_2.

By the Peak Property for $\mathbb{S}$, $V[\sigma \cup \{P\}]$ contains a path from x to u_0, $V[\sigma \cup \{S_1\}]$ contains a path from u_0 to t, $V[\sigma \cup \{S_2\}]$ contains a path from t to u_1, and $V[\sigma \cup \{Q\}]$ contains a path from u_1 to y. Since P, Q, S_2, and S_2 belong to β_i, all of these paths are within V_i, and together they form a path from x to y in $V_i[\sigma]$.

In both cases we have a path from x to y in $V_i[\sigma]$, so $\mathbb{S}_i$ has the Peak Property. This completes the proof. □

Theorem 9.7. *For every signature network $\mathbb{S}$ on a connected graph, the following are equivalent:*

(i) $\mathbb{S}$ *is report complete.*
(ii) $\mathbb{S}$ *has the Peak Property and the Linked Chain Property.*
(iii) $\mathbb{S}$ *contains a signature tree* $\mathbb{T}$.

The proof of Theorem 9.7 will show that the following algorithm returns a signature tree $\mathbb{T}$ contained in $\mathbb{S}$ with decider d whenever $d \in V$ and condition (ii) holds.

Algorithm 9.8. *GET.TREE.3*$(\mathbb{S}, d)$*:*

- Run *DECOMP*$(\mathbb{S}, d)$, which returns a report decomposition $\mathbb{S}_0, \ldots, \mathbb{S}_k$.
- If $k = 0$, run *GET.TREE.2*$(\mathbb{S}_0, d)$, which returns a signature tree $\mathbb{T}$ contained in $\mathbb{S}_0 = \mathbb{S}$ with decider d, and return $\mathbb{T}$.
- If $k > 0$, then for each $i \leq k$, find a high element v_i in V_i and run *GET.TREE.3*$(\mathbb{S}_i, v_i)$, which returns a signature tree $\mathbb{T}_i$ contained in $\mathbb{S}_i$ with decider v_i.
- Form the signature network $\mathbb{S}' = (V', E', L'(\cdot))$ where $V' = \{v_0, \ldots, v_k\}$, $L'(v_i) = L(V_i)$, and $(v_i, v_j) \in E'$ if and only if there is an edge $(x, y) \in E$ such that x is high in V_i, $y \in V_j$, and $(L(x) \cap \alpha) \subseteq (L(y) \cap \alpha)$.
- Run *GET.TREE.2*$(\mathbb{S}', d)$, which returns a signature tree $\mathbb{T}' = (V', F', L'(\cdot))$ contained in $\mathbb{S}'$ with decider d.
- For each $i > 0$ and edge $(v_i, v_j) \in F'$, find a high element x_i in V_i and an edge $(x_i, y_i) \in E$ such that $y_i \in V_j$, and $(L(x_i) \cap \alpha) \subseteq (L(y_i) \cap \alpha)$.
- Reverse arrows where necessary in $\mathbb{T}_i$ to get a signature tree $\mathbb{T}'_i$ with decider x_i.
- Let F be the union of all the edges in $\mathbb{T}'_i$ for $i \geq 0$ and the edges (x_i, y_i) for $i \geq 1$.
- Return $\mathbb{T} = (V, F, L(\cdot))$.

Proof of Theorem 9.7. (i) implies (ii) by Theorems 3.5 and 8.5. Corollary 4.1 says that (iii) implies (i).

To prove that (ii) implies (iii), we show by induction on the number of agents that the algorithm *GET.TREE.3*$(\mathbb{S}, d)$ returns a signature tree $\mathbb{T}$ contained in $\mathbb{S}$ with decider d. This is trivial when $\mathbb{S}$ has two agents. Let $N > 2$ and assume that the result holds for every signature network with fewer than N agents. Suppose $\mathbb{S}$ satisfies (ii) and has N agents. Choose an element $d \in V$. We prove that the algorithm returns a signature tree $\mathbb{T}$ contained in $\mathbb{S}$ with decider d.

By Theorem 9.5, $\mathbb{S}$ has a report decomposition $\mathbb{S}_0, \ldots, \mathbb{S}_k$ at d. If $k = 0$, then $\mathbb{S} = \mathbb{S}_0$ and $L(d)$ contains every symbol that occurs in more than one signature. By Lemma 9.3, for each agent x there is an excellent path from x to d, so by Theorem 7.4 the algorithm returns a signature tree contained in $\mathbb{S}$ with decider d.

Assume $k > 0$. Then for each $i \leq k$, there is a high element $v_i \in V_i$, and $\mathbb{S}_i$ satisfies (ii) and has fewer than N agents, so by the inductive hypothesis, the algorithm with input $(\mathbb{S}_i, v_i)$ returns a signature tree $\mathbb{T}_i = (V_i, F_i, L_i(\cdot))$ contained in $\mathbb{S}_i$ with decider v_i. The algorithm will then construct the directed graph (V', E'), with the agents $V' = \{v_0, \ldots, v_k\}$.

We will show that $\mathbb{S}' = (V', E', L'(\cdot))$ satisfies the hypotheses of Theorem 7.4. at d. To do this, we need the notion of a high path. We call a path π from x to d in $\mathbb{S}$ a **high path** if π is an excellent path, and whenever π enters a set V_i, it stays in V_i until it reaches a high agent $z \in V_i$.

Claim 1. For each $x \in V$ there exists a high path from x to d.

Proof of Claim 1: We argue by induction on the number $a(x) = |\alpha \setminus L(x)|$ of symbols in α that are missing from $L(x)$. Note that $a(d) = 0$, and a path $(x_0, \ldots, x_n)$ from x to d is excellent if and only if $a(x_0) \geq a(x_1) \geq \cdots \geq a(x_n)$. By Lemma 9.3, for every $x \in V$ there is an excellent path from x to d. Clearly, if $a(x) = 0$ then every excellent path from x to d is a high path, so the result holds when $a(x) = 0$.

Assume that for every $y \in V$ such that $a(y) < M$, there is a high path from y to d. Suppose $x \in V$ and $a(x) = M$. Let $(x_0, \ldots, x_m)$ be an excellent path from x to d. We must show that there is a high path from x to d. Suppose $(x_0, \ldots, x_m)$ is not already a high path. Then there is a first agent $y = x_j$ such that $(x_0, \ldots, x_m)$ enters some V_i at y, and later leaves V_i without reaching a high agent $z \in V_i$. By Lemma 9.3, V_i contains an excellent path $(y_0, \ldots, y_n)$ from y to v_i. Since y is not high, we have $a(x) \geq a(y) > a(v_i)$, so $a(v_i) < M$. By the inductive hypothesis, there is a high path $(z_0, \ldots, z_p)$ from v_i to d. Since $(y_0, \ldots, y_m)$ is excellent and v_i is high in V_i, for each $j \leq m$ we have

$$(L(y_j) \cap \alpha) = (L(y_j) \cap L(v_i) \cap \alpha) \subseteq (L(y_{j+1}) \cap L(v_i) \cap \alpha) = (L(y_{j+1}) \cap \alpha).$$

Therefore $a(y_0) \geq \cdots \geq a(y_m)$, so the combined path

$$(x_0, \ldots, x_j = y_0, y_1, \ldots, y_n = z_0, z_1, \ldots, z_p)$$

is an excellent path, and in fact a high path, from x to d. □

Claim 2. (V', E') is a pointed graph with d as a decider, and for each $v_i \in V'$, there is an excellent path from v_i to d in (V', E').

Proof of Claim 2: By Claim 1, for each $v_i \in V'$ there is a high path $(x_0, \ldots, x_m)$ from v_i to d. Let $(x_{s_0}, \ldots, x_{s_n})$ be the subsequence of $(x_0, \ldots, x_m)$ composed of those x_ℓ such that the edge $(x_\ell, x_{\ell+1})$ crosses from one set V_j to another. For each $h \leq n$ let $y_h = v_j$ where V_j is the set such that $x_{s_h} \in V_j$. Then $(y_0, \ldots, y_n)$ is an excellent path from v_i to d in (V', E'). □

It is easily seen from the definitions that:

Claim 3. Every symbol that appears in more than one signature in $\mathbb{S}'$ belongs to α, and therefore belongs to $L'(d)$.

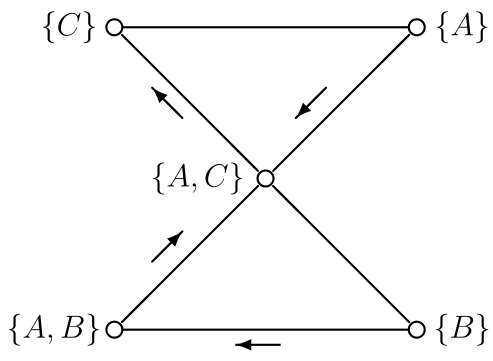
By Claims 2 and 3 and Theorem 7.4, the algorithm *GET.TREE.2*$(\mathbb{S}', d)$ will return a signature tree $\mathbb{T}' = (V', F', L'(\cdot))$ contained in $\mathbb{S}'$ with decider d. The remaining steps of the algorithm can then be carried out, and will return a signature network $\mathbb{T}$ that is contained in $\mathbb{S}$ and is on a tree (V, F) with decider d.

It remains to show that $\mathbb{T}$ has the Peak Property. Let $x, y \in V$. If $x, y \in V_i$ for some $i \geq 1$, then in $\mathbb{T}_i$ the set $V_i[L(x) \cap L(y)]$ contains paths from x to v_i and y to v_i. Therefore in $\mathbb{T}'_i$, $V_i[L(x) \cap L(y)]$ contains paths from x to x_i and from y to x_i, and hence in $\mathbb{T}$, $V[L(x) \cap L(y)]$ contains these paths. If $x \in V_i$ and $y \in V_j$ with $i \neq j$, then $(L(x) \cap L(y)) \subseteq \alpha$, and in $\mathbb{T}$, $V[L(x) \cap L(y)]$ contains paths from x to d and from y to d. This shows that $\mathbb{T}$ has the Peak Property, and hence is a signature tree. □

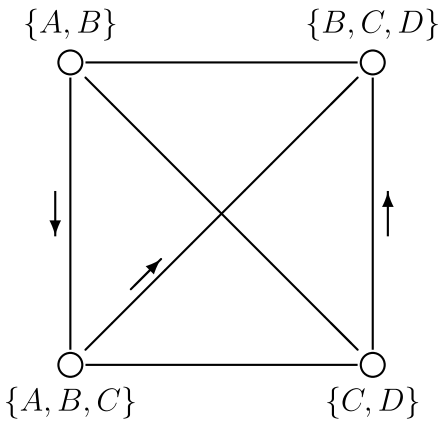
Corollary 9.9. *Let $\mathbb{S}$ be a signature network over a connected graph, and let $\mathbb{S}'$ be a propositional copy of $\mathbb{S}$. Then $\mathbb{S}'$ is report complete if and only if $\mathbb{S}$ is report complete.*

The following examples illustrate Theorem 9.7.

Example 9.10. $\mathbb{S}$ is on a connected graph and has the Peak Property and Linked Chain Property. By Theorem 9.7 and Lemma 9.1, $\mathbb{S}$ is report complete, and for every agent d it contains a signature tree with decider d. The arrows indicate one such signature tree.



Example 9.11. $\mathbb{S}$ is on a complete graph and has the Linked Chain Property. Every complete graph has the Peak Property.



Amir and McIlraith [2] deal with signature networks $\mathbb{S}$ on a connected graph in the following way. An algorithm called *CUT-CYCLES* produces a signature tree $\mathbb{T}' = (V, F', L'(\cdot))$ by removing edges to form a tree and adding symbols to the signatures to satisfy the Peak Property. In general, $L'(x)$ will properly contain the original signature $L(x)$. (Actually, they put signatures on the edges, but when the Peak Property holds, the signature on an edge will be the intersection of the signatures of the two agents on the edge.) Theorem 9.7 shows that if $\mathbb{S}$ has the Peak Property and the Linked Chain Property, then $\mathbb{S}$ contains a signature tree $\mathbb{T} = (V, F, L(\cdot))$ with the original signatures $L(x)$ rather than with enlarged signatures.

Like Theorem 6.6, Theorem 9.7 can be interpreted as a statement about immediate supervisors. We are given signature network $\mathbb{S}$ over a connected graph. Suppose that for every knowledge base $\mathbb{K}$ over $\mathbb{S}$, every provable sentence can be proved by Decider from sentences reportable to Decider when each agent can report to any neighboring agent a sentence that he knows and that is in the common language of the two agents. Then there is a way of assigning an immediate supervisor to each agent $x \neq d$ so that for every knowledge base $\mathbb{K}$ over $\mathbb{S}$, every provable sentence can be proved by d from sentences reportable to d when agents can only report sentences they know to their immediate supervisors in their common language.

10. Open Questions

Our main focus in this paper has been on the following three general questions.

Question 10.1. *Which signature networks are report complete?*

We have seen in Theorems 6.4 and 8.5 that every report complete signature network has both the Twin Peaks Property and the Linked Chain Property. Theorems 4.3, 6.6, 7.4, and 9.7 give sufficient conditions for report completeness.

Question 10.2. *Which signature networks contain signature trees?*

Corollary 4.1 shows that a signature network that is not report complete cannot contain a signature tree. Theorem 6.6 shows that every report complete signature network on a pointed directed acyclic graph contains a signature tree. Theorem 9.7 shows that every report complete signature network on a connected graph contains a signature tree. But Example 4.2 shows that there are many report complete signature networks that do not contain a signature tree.

Question 10.3. *If* $\mathbb{S}$ *is a signature network and the propositional copy of* $\mathbb{S}$ *is report complete, must* $\mathbb{S}$ *be report complete?*

Corollaries 5.5 and 9.9 show that the answer is yes for signature networks on pointed directed acyclic graphs and on connected graphs.

Here are some other questions related to this work.

Question 10.4. *Can one generalize the Lyndon Interpolation Theorem (*[17]*) to signature networks?*

E. Amir asked this question in a personal communication. Amir and McIlraith [3] applied the Lyndon Interpolation Theorem to strengthen the report completeness property of a signature tree in a way that takes into account the parity of signatures. One can hope for analogous results for signature networks.

Theorem 4.3 tells us that the pointed graphs with no weak cycles are exactly the pointed graphs on which the Peak Property implies report completeness. We can ask the corresponding questions for the Twin Peaks Property and the Linked Chain Property.

Question 10.5. *On which pointed graphs does the Twin Peaks Property imply report completeness?*

Question 10.6. *On which pointed graphs does the Linked Chain Property imply report completeness?*

In summary, the main questions we have addressed in this paper are Questions 10.1–10.3. We have solved each of these questions for signature networks on pointed directed acyclic graphs and for signature networks on connected graphs. However, these questions are still mostly open for signature networks on arbitrary pointed directed graphs.

Appendix A. Appendix: Proof of Fact 1.6

Fact 1.6 (restated). *([2]) Every signature tree is report complete.*

Proof of Fact 1.6. We argue by induction on the number $|V|$ of agents. The result is trivial if there is only one agent.

Assume, as an inductive hypothesis, that every signature tree with n agents is report complete. Let

$$\mathbb{S} = (V, E, L(\cdot)), \quad \mathbb{K} = (V, E, L(\cdot))$$

be a signature tree with $n+1$ agents, let d be the unique decider, and let $\mathbb{K}$ be a knowledge base over $\mathbb{S}$. Suppose $\mathcal{K}(V)$ is inconsistent. We must show that $\mathbb{K}$ is report inconsistent at d, that is, the false sentence $\bot$ is report provable at d.

Let x be a source agent in $\mathbb{K}$. Since V has more than one element, $x \neq d$, and there is a unique edge $(x, y) \in E$. Let $C \in [L(x) \cap L(y)]$ be a sentence to be specified later. Let

$$\mathbb{S}' = (V', E', L(\cdot), \mathcal{K}'(\cdot))$$

be the knowledge base over the signature network with n agents formed from $\mathbb{K}$ by removing the agent x and the edge (x, y), and adding the sentence C to the knowledge base of y. Thus $V' = V \setminus \{x\}$, $E' = E \setminus \{(x, y)\}$, $\mathcal{K}'(y) = \mathcal{K}(y) \cup \{C\}$, and everything else is unchanged. By the Peak Property for $\mathbb{S}$, any symbol that belongs to both $L(x)$ and $L(V')$ also belongs to $L(y)$. Thus

$$L(x) \cap L(y) = L(x) \cap L(V').$$

It is clear that

$$\mathbb{S}' = (V', E', L(\cdot))$$

still has the Peak Property, and is still a signature tree with decider d. Let $\mathcal{K}(V') = \bigcup_{v \in V'} \mathcal{K}(v)$. Then

$$\mathcal{K}(V) = \mathcal{K}(x) \cup \mathcal{K}(V') \text{ and } \mathcal{K}'(V') = \mathcal{K}(V') \cup \{C\}.$$

Since $\mathcal{K}(V) \vdash \bot$, we have

$$\mathcal{K}(x) \cup \mathcal{K}(V') \vdash \bot.$$

Therefore $\mathcal{K}(x) \cup \{B\} \vdash \bot$ for some sentence B that is a finite conjunction of sentences in $\mathcal{K}(V')$. B belongs to $[L(V')]$. We now define the sentence C. By the Craig Interpolation Theorem, we may take C to be a sentence $C \in [L(x) \cap L(V')]$ such that

$$\mathcal{K}(x) \vdash C \text{ and } C \cup \{B\} \vdash \bot.$$

Then $C \in [L(x) \cap L(y)]$ as required, and

$$\mathcal{K}'(V') = \mathcal{K}(V') \cup \{C\} \vdash \bot.$$

By the inductive hypothesis, $\perp$ is report provable in $\mathbb{K}'$ at d. Since C is 0-reportable along (x, y) in $\mathbb{K}$, it follows that $\perp$ is also report provable in $\mathbb{K}$ at d. □

References

[1] Adjiman, P., Chatalic, P., Goasdoue, F., Rousset, M.-C., and Simon, L. Distributed Reasoning in a Peer-to-peer Setting. Journal of Artificial Intelligence Research 25, 269-314 (2006).

[2] Amir, E. and S. McIlraith. Partition-Based Logical Reasoning for First-Order and Propositional Theories. Artificial Intelligence Journal, 162 49-88 (2005).

[3] Amir, E. and S. McIlraith. Strategies for Focusing Structure-based Theorem Proving (2008), Submitted, www.reason.cs.uiuc.edu/eyal/papers.html.

[4] Barwise, J. and J. van Bentham. Interpolation, Preservation, and Pebble Games. Journal of Symbolic Logic 64, 881-903 (1999).

[5] van Bentham, J. The Many Faces of Interpolation. Synthese vol. 165, 451-460 (2008).

[6] Bilkova, M. Uniform Interpolation and Propositional Quantifiers in Modal Logic, Studia Logica 85 1-31 (2007).

[7] Carbone, A. The Craig Interpolation Theorem for Schematic Systems. Colloquium Logicum, vol. 2, pp. 87-100 (1996).

[8] Chang, C. C. and H. J. Keisler. Model Theory (3rd Ed.). North Holland (1990), Dover (2012).

[9] Cowell, R, A. P. Dawid, S. Lauritzen, and D. Spiegelhalter. Probabilistic Networks and Expert Systems, Springer (2007).

[10] Craig, W. Linear Reasoning. Journal of Symbolic Logic 22 250-268. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. Ibid. 269-285 (1957).

[11] Dechter, R. Constraint Processing. Morgan Kauffman (2003).

[12] Enderton, H. A Mathematical Introduction to Logic. Harcourt Academic Press (2001)

[13] Feferman, S. Application of Many-Sorted Interpolation Theorems. Proceedings of the Tarski Symposium, American Math. Society, 205–223, ed. by L. Henkin (1974).

[14] Feferman, S. Harmonious logic: Craig's interpolation theorem and its descendants. Synthese, 164 (2008), 341 357.

[15] Gabbay, D. Craig Interpolation Theorem for Intuitionistic Logic and Extensions, I, II. Logic Colloquium '69, North-Holland, 391–410, ed. by R. Gandy (1971).

[16] Hoover, D. N. A Probabilistic Interpolation Theorem. Journal of Symbolic Logic 50, 708-713 (1985).

[17] Lyndon, R. An Interpolation Theorem in the Predicate Calculus. Pacific Journal of Mathematics 9, 129-142 (1959).

[18] McMillan, K. Applications of Craig Interpolation in Model Checking. In Tools and Algorithms for the Construction and Analysis of Systems, 10th International Conference, Springer, 1-12 (2005).

[19] Otto, M. An Interpolation Theorem. Bulletin of Symbolic Logic 6, 447-462 (2000).
[20] Popescu, A., T. F. Serbanuta, and G. Rosu. A Semantic Approach to Interpolation. J. of Theoretical Computer Science 410 (2009) 1109-1128.
[21] Renardel de Lavalette, G.R. Interpolation in Computing Science: The Semantics of Modularization. Synthese 164 (2008), 437-450.
[22] Rodenburg, P. H. A Simple Algebraic Proof of the Equational Interpolation Theorem. Algebra Universalis 28, 48-51 (1991).

Department of Mathematics, University of Wisconsin-Madison, Madison, WI 53706, keisler@math.wisc.edu, www.math.wisc.edu/~keisler

Department of Management Science and Information Systems, University of Massachusetts-Boston, Boston MA 02125, jeff.keisler@umb.edu