Character-Based Cladistics and Answer Set Programming

Daniel R. Brooks[1], Esra Erdem[2], James W. Minett[3], and Donald Ringe[4]

[1] Department of Zoology, University of Toronto, Toronto, Canada
[2] Institute of Information Systems, Vienna University of Technology, Vienna, Austria
[3] Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong
[4] Department of Linguistics, University of Pennsylvania, Philadelphia, USA

Abstract. We describe the reconstruction of a phylogeny for a set of taxa, with a character-based cladistics approach, in a declarative knowledge representation formalism, and show how to use computational methods of answer set programming to generate conjectures about the evolution of the given taxa. We have applied this computational method in two domains: to historical analysis of languages, and to historical analysis of parasite-host systems. In particular, using this method, we have computed some plausible phylogenies for Chinese dialects, for Indo-European language groups, and for *Alcataenia* species. Some of these plausible phylogenies are different from the ones computed by other software. Using this method, we can easily describe domain specific information (e.g. temporal and geographical constraints), and thus prevent the reconstruction of some phylogenies that are not plausible.

1 Introduction

Cladistics (or phylogenetic systematics), developed by Willi Henig [17], is the study of evolutionary relations between species based on their shared traits. Represented diagrammatically, these relations can form a tree whose leaves represent the species, internal vertices represent their ancestors, and edges represent the genetic relationships between them. Such a tree is called a "phylogenetic tree" (or a "phylogeny"). In this paper, we study the problem of reconstructing phylogenies for a set of taxa (taxonomic units) with a character-based cladistics approach.[1]

In character-based cladistics, each taxonomic unit is described with a set of "(qualitative) characters"—traits that every taxonomic unit can instantiate in a variety of ways. The taxonomic units that instantiate the character in the same way are assigned the same "state" of that character. Here is an example from [31]. Consider the languages English, German, French, Spanish, Italian, and Russian. A character for these languages is the basic meaning of 'hand':

English	German	French	Spanish	Italian	Russian
hand	*Hand*	*main*	*mano*	*mano*	*ruká*

[1] See [12] for a survey on the other methods for phylogeny reconstruction.

Since the English and German words descended from the same word in their parent language, namely Proto-Germanic **handuz*, by direct linguistic inheritance, those languages must be assigned the same state for this character. The three Romance languages must likewise be assigned a second state (since their words are all descendants of Latin *manus*) and Russian must be assigned a third:

English	German	French	Spanish	Italian	Russian
1	1	2	2	2	3

In character-based cladistics, after describing each taxonomic unit with a set of characters, and determining the character states, the phylogenies are reconstructed by analyzing the character states. There are two main approaches: one is based on the "maximum parsimony" criterion [7], and the other is based on the "maximum compatibility" criterion [3]. According to the former, the goal is to infer a phylogeny with the minimum number of character state changes along the edges. With the latter approach, the goal is to reconstruct a phylogeny with the maximum number of "compatible" characters. Both problems are NP-hard [14, 5]. In this paper we present a method for reconstructing a phylogenetic tree for a set of taxa, with the latter approach.

Our method is based on the programming methodology called answer set programming (ASP) [26, 33, 21]. It provides a declarative representation of the problem as a logic program whose answer sets [15, 16] correspond to solutions. The answer sets for the given formalism can be computed by special systems called answer set solvers. For instance, CMODELS [20] is one of the answer set solvers that are currently available.

We apply our method of reconstructing phylogenies using ASP to historical analysis of languages, and to historical analysis of parasite-host systems.

Histories of individual languages give us information from which we can infer principles of language change. This information is not only of interest to historical linguists but also of interest to archaeologists, human geneticists, physical anthropologists as well. For instance, an accurate reconstruction of the evolutionary history of certain languages can help us answer questions about human migrations, the time that certain artifacts were developed, when ancient people began to use horses in agriculture [24, 25, 32, 35].

Historical analysis of parasites gives us information on where they come from and when they first started infecting their hosts. The phylogenies of parasites, with the phylogenies of their hosts, and with the geographical distribution of their hosts, can be used to understand the changing dietary habits of a host species, to understand the structure and the history of ecosystems, and to identify the history of animal and human diseases. This information allows predictions about the age and duration of specific groups of animals of a particular region or period, identification of regions of evolutionary "hot spots" [2], and thus can be useful to make more reliable predictions about the impacts of perturbations (natural or caused by humans) on ecosystem structure and stability [1].

With this method, using the answer set solver CMODELS, we have computed 33 phylogenetic trees for 7 Chinese dialects based on 15 lexical characters, and 45 phylogenetic trees for 24 Indo-European languages based on 248 lexical, 22 phonological and 12 morphological characters. Some of these phylogenies are plausible from the point of view of historical linguistics. We have also computed 21 phylogenetic trees for 9 species of *Alcataenia* (a tapeworm genus) based on their 15 morphological characters, some of which are plausible from the point of view of coevolution—the evolution of two or more interdependent species each adapting to changes in the other, and from the point of view of historical biogeography—the study of the geographic distribution of organisms.

We have also computed most parsimonious trees for these three sets of taxa, using PARS (available with PHYLIP [13]). Considering also the most parsimonious trees published in [30] (for Indo-European languages), [27] (for Chinese dialects), and [18, 19] (for *Alcataenia* species), we have observed that some of the plausible trees we have computed using the compatibility criterion are different from the most parsimonious ones. This shows that the availability of our computational method based on maximum compatibility can be useful for generating conjectures that can not be found by other computational tools based on maximum parsimony.

As for related work, one available software system that can compute phylogenies for a set of taxa based on the maximum compatibility criterion is CLIQUE (available with PHYLIP), which is applicable only to sets of taxa where a taxonomic unit is mapped to state 0 or state 1 for each character. This prevents us from using CLIQUE to reconstruct phylogenies for the three sets of taxa mentioned above since, in each set, there is some taxonomic unit mapped to state 2 for some character. Another system is the Perfect Phylogeny software of [31], which can compute a phylogeny with the maximum number of compatible characters only when all characters are compatible. Otherwise, it computes an approximate solution. In this sense, our method is more general than the existing ones that compute trees based on maximum compatibility.

Another advantage of our method over the existing ones mentioned above is that we can easily include in the program domain specific information (e.g. temporal and geographical constraints) and thus prevent the reconstruction of some trees that are not plausible.

We consider reconstruction of phylogenies as the first step of reconstructing the evolutionary history of a set of taxa. The idea is then to reconstruct (temporal) phylogenetic networks, which also explain the contacts (or borrowings) between taxonomic units, from the reconstructed phylogenies. The second step is studied in [29, 9, 10].

For more information on the semantics of the ASP constructs used in the logic program below, and on the methodology of ASP, the reader is referred to [22].

2 Problem Description

A *phylogenetic tree* (or *phylogeny*) for a set of taxa is a finite rooted binary tree $\langle V, E\rangle$ along with two finite sets I and S and a function f from $L \times I$ to S, where L is the set of leaves of the tree. The set L represents the given taxonomic units whereas the set V describes their ancestral units and the set E describes the genetic relationships between them. The elements of I are usually positive integers ("indices") that represent, intuitively, qualitative characters, and elements of S are possible states of these characters. The function f "labels" every leaf v by mapping every index i to the state $f(v, i)$ of the corresponding character in that taxonomic unit.

For instance, Fig. 1 is a phylogeny with $I = \{1, 2\}$ and $S = \{0, 1\}$; $f(v, i)$ is represented by the i-th member of the tuple labeling the leaf v.

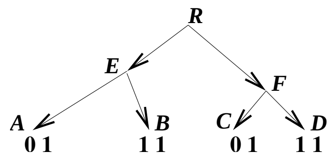

Fig. 1. A phylogeny for the languages A, B, C, D.

A character $i \in I$ is *compatible* with a phylogeny (V, E, L, I, S, f) if there exists a function $g : V \times \{i\} \mapsto S$ such that

(i) for every leaf v of the phylogeny, $g(v, i) = f(v, i)$;
(ii) for every $s \in S$, if the set

$$V_{is} = \{x \in V : \; g(x, i) = s\}$$

is nonempty then the digraph $\langle V, E \rangle$ has a subgraph with the set V_{is} of vertices that is a rooted tree.

A character is *incompatible* with a phylogeny if it is not compatible with that phylogeny. For instance, Character 2 is compatible with the phylogeny of Fig. 1, but Character 1 is incompatible.

The computational problem we are interested in is, given the sets L, I, S, and the function f, to build a phylogeny (V, E, L, I, S, f) with the maximum number of compatible characters. This problem is called the *maximum compatibility problem*. It is NP-hard even when the characters are binary [5].

To solve the maximum compatibility problem, we consider the following problem: given sets L, I, S, a function f from $L \times I$ to S, and a nonnegative integer n, build a phylogeny (V, E, L, I, S, f) with at most n incompatible characters if one exists.

3 Describing the Problem as a Logic Program

We formalize the problem of phylogeny reconstruction for a set of taxa (as described in Section 2) as a logic program. The inputs to this problem are

- a set L of leaves $0, \ldots, k$ ($k > 0$), representing a set of taxa,
- a set I of (qualitative) characters,
- a set S of (character) states,
- a function f mapping every leaf, for every character, to a state, and
- a nonnegative integer n .

The output is a phylogeny (V, E, L, I, S, f) for L with at most n incompatible characters, if one exists.

The logic program describing the problem has two parts. In the first part, rooted binary trees whose leaves represent the given taxa are generated. In the second part, the rooted binary trees according to which there are more than n incompatible characters are eliminated.

Part 1. First note that a rooted binary tree $\langle V, E\rangle$ with leaves L has $2k+1$ vertices, since $|L| = k+1$. Then V is a set of $2k+1$ vertices. We identify the vertices in V by the numbers $0, \ldots, 2k$. For a canonical representation of a rooted binary tree, i.e., a unique numbering of the internal vertices in V, we ensure that (1) for every edge $(x, y) \in E$, $x > y$, and (2) for any two internal vertices x and y, $x > y$ iff the maximum of the children of x is greater than the maximum of the children of y. We call such a canonical representation of a rooted binary tree an *ordered binary tree*, and describe it as follows.

Suppose that the edges (x, y) of the tree, i.e., elements of E, are described by atoms of the form $edge(x, y)$. The sets of atoms of the form $edge(x, y)$ are "generated" by the rule[2]

$$2 \leq \{edge(x, y) : y \in V, x > y\}^c \leq 2 \leftarrow \qquad (x \in V \setminus L). \tag{1}$$

Each set describes a digraph where there is an edge from every internal vertex to two other vertices with smaller numbers, thus satisfying condition (1). Note that, due to the numbering of the internal vertices above, the in-degree of Vertex $2k$ is 0. Therefore, Vertex $2k$ is the root of the tree.

These generated sets are "tested" with some constraints expressing that the set describes a tree: (a) the set describes a connected digraph, and (b) the digraph is acyclic.

To describe (a) and (b), we "define" the reachability of a vertex y from vertex x in $\langle V, E\rangle$:

$$\begin{array}{ll} reachable(x, y) \leftarrow edge(x, y) \qquad (x, y \in V) \\ reachable(x, y) \leftarrow edge(x, z), reachable(z, y) \qquad (x, y, z \in V). \end{array} \tag{2}$$

For (a), we make sure that every vertex is reachable from the root by the constraint

$$\leftarrow not\ reachable(2k, x) \qquad (x \in V \setminus \{2k\}). \tag{3}$$

For (b), we make sure that no vertex is reachable from itself:

$$\leftarrow reachable(x, x) \qquad (x \in V). \tag{4}$$

To make sure that condition (2) above is satisfied, we first "define" $max_Y(x, y)$ ("Child y of vertex x is larger than the sister of y")

$$max_Y(x, y) \leftarrow edge(x, y), edge(x, y_1) \qquad (x, y, y_1 \in V, y > y_1) \tag{5}$$

and express that a vertex x is larger than another vertex x_1 if the maximum child of x is larger than that of x_1:

$$\leftarrow max_Y(x, y), max_Y(x_1, y_1) \qquad (x, x_1, y, y_1 \in V, y > y_1, x < x_1). \tag{6}$$

Part 2. We eliminate the rooted binary trees $\langle V, E\rangle$, generated by Part 1 above, with more than n incompatible characters as follows. First we identify, for a rooted binary tree $\langle V, E\rangle$, the characters such that, for some function $g : V \times I \mapsto S$, condition (i) holds but condition (ii) does not. Then we eliminate the rooted binary trees for which the number of such characters is more than n.

[2] Rule (1) describes the subsets of the set $\{edge(x, y) : y \in V, x > y\}$ with cardinality 2.

Take any such function g. According to condition (i), g coincides with f where the latter is defined:

$$g(x, i, s) \leftarrow \qquad (x \in L, f(x, i) = s). \tag{7}$$

The internal vertices are labeled by exactly one state for each character by the rule

$$1 \leq \{g(x, i, s) : s \in S\}^c \leq 1 \leftarrow \qquad (x \in V \setminus L, i \in I). \tag{8}$$

To identify the characters for which condition (ii) does not hold, first we pick a root x for each character i and for each state s such that $V_{is} \neq \emptyset$ by the choice rule

$$\{root_{is}(x, i, s)\}^c \leftarrow g(x, i, s) \qquad (x \in V, i \in I, s \in S). \tag{9}$$

We make sure that exactly one root is picked by the constraints

$$\leftarrow root_{is}(x, i, s), root_{is}(y, i, s) \qquad (x, y \in V, x \neq y, i \in I, s \in S) \tag{10}$$

$$\leftarrow \{root_{is}(x, i, s) : x \in V\}\ 0,\ g(y, i, s) \qquad (y \in V, i \in I, s \in S), \tag{11}$$

and that, among the vertices in V_{is}, this root is the closest to the root of the tree by the constraint

$$\leftarrow root_{is}(x, i, s), g(y, i, s), reachable(y, x) \qquad (x, y \in V, i \in I, s \in S). \tag{12}$$

After defining the reachability of a vertex in V_{is} from the root:

$$reachable_{is}(x, i, s) \leftarrow root_{is}(x, i, s) \qquad (x \in V, i \in I, s \in S) \tag{13}$$

$$reachable_{is}(x, i, s) \leftarrow g(x, i, s), reachable_{is}(z, i, s), edge(z, x) \qquad (x, z \in V, i \in I, s \in S) \tag{14}$$

we identify the characters for which condition (ii) does not hold:

$$incompatible(i) \leftarrow g(x, i, s), not\ reachable_{is}(x, i, s) \qquad (x \in V, i \in I, s \in S). \tag{15}$$

We make sure that there are at most n incompatible characters by the constraint

$$\leftarrow n + 1 \leq \{incompatible(i) : i \in I\}. \tag{16}$$

The following theorem shows that the program above correctly describes the maximum compatibility problem stated as a decision problem.

Let Π be the program consisting of rules (1)–(16). Let E_k denote the set of all atoms of the form $edge(x, y)$ such that $0 \leq y < x \leq 2k$.

Correctness Theorem for the Phylogeny Program *For a given input (L, I, S, f, n), and for a set E of edges that is a rooted binary tree with the leaves L, E describes a phylogeny (V, E, L, I, S, f) with at most n incompatible characters iff E can be represented by the ordered binary tree $Z \cap E_k$ for some answer set Z for Π. Furthermore, every rooted binary tree with the leaves L can be represented like this in only one way.*

The proof is based on the splitting set theorem and uses the method proposed in [11].

Note that constraints (11) and (12) can be dropped from Π, if the goal is to find the minimum n such that Π has an answer set. In our experiments, we drop constraint (11) for a faster computation.

4 Useful Heuristics

We can use the answer set solver CMODELS with the phylogeny program described above to solve small instances of the maximum compatibility problem. Larger data sets, like the Indo-European dataset (Section 7), require the use of some heuristics.

Sometimes the problem for a given input (L, I, S, f, n) can be simplified by making the set I of characters smaller. In particular, we can identify the characters that would be compatible with any phylogeny constructed for the given taxa. For instance, if every taxonomic unit is mapped to a different state at the same character, i.e., the character does not have any "essential" state,[3] then we do not need to consider this character in the computation. Similarly, if every taxonomic unit is mapped to the same state at the same character then the character has only one essential state, and that character can be eliminated. Therefore, we can consider just the characters with at least 2 essential states. Such a character will be called *informative* since it is incompatible for some phylogeny. For instance, for the Indo-European languages, out of 275 characters, we have found out that 21 are informative.

Due to condition (ii) of Section 2, every nonempty V_{is} forms a tree in $\langle V, E\rangle$. In each such tree, for every pair of sisters x and y, such that $x, y \in V_{is}$, x and y are labeled for character i in the same way as their parent is labeled. Therefore, to make the computation more efficient, while labeling the internal vertices of the rooted binary tree in Part 2, we can propagate common labels up. For instance, for the *Alcataenia* species, this heuristic improves the computation time by a factor of 2.

In fact, as described in [9, Section 5], we can use partial labelings of vertices, considering essential states, instead of a total one. For instance, for the Indo-European languages, this heuristic improves the computation time by a factor of 3.

Due to the definition of a (partial) perfect network in [9], a character i is compatible with respect to a phylogeny (V, E, L, I, S, f) iff there is a partial mapping g from $V \times \{i\}$ to S such that $(V, E, \emptyset, g)$ is a partial perfect network built on the phylogeny $(V, E, L, \{i\}, S, f|_{L\times\{i\}})$. Then, Propositions 4 and 5 from [9] ensure that no solution is lost when the heuristics above are used in the reconstruction of a phylogeny with the maximum number of compatible characters.

5 Computation and Evaluation of Phylogenetic Trees

We have applied the computational method described above to three sets of taxa: Chinese dialects, Indo-European languages, and *Alcataenia* (a tapeworm genus) species. Our experiments with these taxa are described in the following three sections.

To compute phylogenies, we have used the answer set solver CMODELS with the programs describing a set of taxa, preprocessing of the taxa, and reconstruction of a phylogeny. Since the union of these programs are "tight" on their models of completion [8], CMODELS transforms them into a propositional theory [23], and calls a SAT solver to compute the models of this theory, which are identical to the answer sets for

[3] Let (V, E, L, I, S, f) be a phylogeny, with $f : L \times I \mapsto S$. A state $s \in S$ is *essential* with respect to a character $j \in I$ if there exist two different leaves l_1 and l_2 in L such that $f(l_1, j) = f(l_2, j) = s$.

Character	Xiang	Gan	Wu	Mandarin	Hakka	Min	Yue
'feather'	1	2	2	1	2	1	2
'give'	1	1	2	3	4	5	2
'grease'	1	2	1	3	2	2	2
'know'	1	1	1	2	2	2	2
'say'	1	3	2	2	1	1	1

Fig. 2. The character states of some informative characters for seven Chinese dialects.

the given programs [20]. In our experiments, we have used CMODELS (Version 2.10) with the SAT solver ZCHAFF (Version Z2003.11.04) [28], on a PC with a 733 MHz Intel Pentium III processor and 256MB RAM, running SuSE Linux (Version 8.1).

In the following, we present the computed trees in the Newick format, where the sister subtrees are enclosed by parentheses. For instance, the tree of Fig. 1 can be represented in the Newick format as ((A, B), (C, D)).

We compare the computed phylogenetic trees with respect to three criteria. First, we identify the phylogenies that are plausible. For the Chinese dialects and Indo-European languages, the plausibility of phylogenies depends on the linguistics and archaeological evidence; for *Alcataenia*, the plausibility of the phylogeny we compute is dependent on the knowledge of host phylogeny (e.g. phylogeny of the seabird family *Alcidae*), chronology of the fossil record, and biogeographical evidence. Since our method is based on maximum compatibility, the second criterion is the number of incompatible characters: the more the number of compatible characters the better the trees are. As pointed out earlier in Section 1, we view reconstructing phylogenies as the first step of reconstructing the evolutionary history of a set of taxonomic units. The second step is then, to obtain a perfect (temporal) phylogenetic network from the reconstructed phylogeny by adding some lateral edges, in the sense of [29, 9, 10]. Therefore, the third criteria is the minimum number of lateral edges (denoting contacts such as borrowings) required to turn the phylogeny into a phylogenetic network.

We also compare these trees to the ones computed by a maximum parsimony method. Usually, to compare a set of trees with another set, "consensus trees" are used. A consensus tree "summarizes" a set of trees by retaining components that occur sufficiently often. We have used the program CONSENSE, available with PHYLIP [13], to find consensus trees.

6 Computing Phylogenetic Trees for Chinese Dialects

We have applied the computational method described above to reconstruct a phylogeny for the Chinese dialects Xiang, Gan, Wu, Mandarin, Hakka, Min, and Yue. We have used the dataset, originally gathered by Xu Tongqiang and processed by Wang Feng, described in [27]. In this dataset, there are 15 lexical characters, and they are all informative. Each character has 2–5 states. For some characters, their states are presented in Fig. 2. After the inessential states are eliminated as explained in Section 4, each character has 2 essential states.

	Phylogenies	m
15	((Hakka, Min), (Yue, (Gan, (Xiang, (Wu, Mandarin)))))	2
18	((Yue, (Hakka, Min)), (Mandarin, (Wu, (Xiang, Gan))))	3
23	((Hakka, Min), (Yue, ((Xiang, Gan), (Wu, Mandarin))))	3
24	((Yue, (Hakka, Min)), (Gan, (Xiang, (Wu, Mandarin))))	2
27	((Hakka, Min), (Yue, (Mandarin, (Wu, (Xiang, Gan)))))	3

Fig. 3. Phylogenies computed for Chinese dialects, using CMODELS, that are plausible from the point of view of historical linguistics. Each of these trees has 6 incompatible characters, and requires m lateral edges to turn into a perfect phylogenetic network.

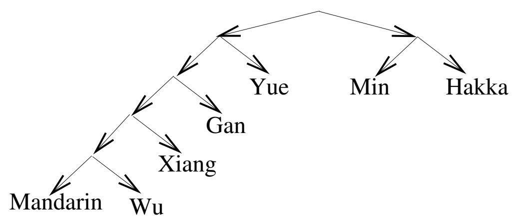

Fig. 4. A plausible phylogeny for Chinese dialects, constructed by CMODELS.

With this dataset, we have computed 33 phylogenies with 6 incompatible characters and found out that there is no phylogeny with less than 6 incompatible characters, in less than an hour. The sub-grouping of the Chinese dialects is not yet established. However, many specialists agree that there are a Northern group and a Southern group. That is, for the dialects we chose in our study, we would expect a (Wu, Mandarin, Gan, Xiang) Northern grouping and a (Hakka, Min) Southern grouping. (It is not clear which group Yue belongs to.) Out of the 33 trees, 5 are more plausible with respect to this hypothesis. One of these plausible trees, Phylogeny 15, is presented in Fig. 4. Among these 5 plausible phylogenies, 2 require at least 2 lateral edges (representing borrowings) to turn into a perfect phylogenetic network; the others require at least 3 edges.

With the dataset above, we have constructed 5 most parsimonious phylogenies using the phylogeny reconstruction program PARS, and observed that none of these phylogenies is consistent with the hypothesis about the grouping of Northern and Southern Chinese dialects.

Using the program CONSENSE, we have computed the majority-consensus tree for our 33 phylogenies: ((Yue, (Hakka, Min)), ((Gan, Xiang), (Wu, Mandarin))). Both this tree and the majority-consensus tree for the 55 most parsimonious trees of [27] are consistent with the more conventional hypothesis above, grouping Yue with the Southern dialects.

All of the 33 phylogenies we have computed correspond to the trees of Types I–III in [27]. Each of the remaining 22 trees of [27] has 7 incompatible characters, but they have the same degree of parsimony as the other 33 trees. This highlights the difference between a maximum parsimony method and a maximum compatibility method.

Character	Ancient Greek	Old Church Slavonic	Old English	Old High German	Latin	Old Persian
'child'	3	8	10	18	12	15
'father'	2	1	2	2	2	2
'free'	3	8	10	10	3	14
'laugh'	2	7	9	9	11	14
'tear'	2	4	2	2	2	7

Fig. 5. The character states of some informative characters for six Indo-European languages.

7 Computing Phylogenetic Trees for Indo-European Languages

We have applied the computational method described above to reconstruct a phylogeny for the Indo-European languages Hittite, Luvian, Lycian, Tocharian A, Tocharian B, Vedic, Avestan, Old Persian, Classical Armenian, Ancient Greek, Latin, Oscan, Umbrian, Gothic, Old Norse, Old English, Old High German, Old Irish, Welsh, Old Church Slavonic, Old Prussian, Lithuanian, Latvian, and Albanian. We have used the dataset assembled by Don Ringe and Ann Taylor, with the advice of other specialist colleagues. This dataset is described in [31].

There are 282 informative characters in this dataset. Out of 282 characters, 22 are phonological characters encoding regular sound changes that have occurred in the prehistory of various languages, 12 are morphological characters encoding details of inflection (or, in one case, word formation), and 248 are lexical characters defined by meanings on a basic word list. For each character, there are 2–24 states. Some of the character states for some Indo-European languages are shown in Fig. 5.

To compute phylogenetic trees, we have treated as units the language groups Balto-Slavic (Lithuanian, Latvian, Old Prussian, Old Church Slavonic), Italo-Celtic (Oscan, Umbrian, Latin, Old Irish, Welsh), Greco-Armenian (Ancient Greek, Classical Armenian), Anatolian (Hittite, Luvian, Lycian), Tocharian (Tocharian A, Tocharian B), Indo-Iranian (Old Persian, Avestan, Vedic), Germanic (Old English, Old High German, Old Norse, Gothic), and the language Albanian.

For each language group, we have obtained the character states by propagating the character states for languages up, similar to the preprocessing of [9]. After propagating character states up, we have found out that grouping Baltic and Slavic makes 1 character incompatible, and grouping Italic and Celtic makes 6 characters incompatible. (For the purposes of this experiment we accept the Italo-Celtic subgroup as found in [31] largely on the basis of phonological and morphological characters.) Other groupings do not make any character incompatible. Therefore, we have not considered these 7 characters while computing a phylogenetic tree, as we already know that they would be incompatible with every phylogeny.

Then we have identified the characters that would be compatible with every phylogeny built for these 7 language groups and the language Albanian. By eliminating such characters as explained in Section 4, we have found out that, out of $282 - 7$ characters, 21 characters are informative. Out of those 21, 2 are phonological ('P2' and 'P3') and 1 is morphological ('M5'). Each character has 2–3 essential states.

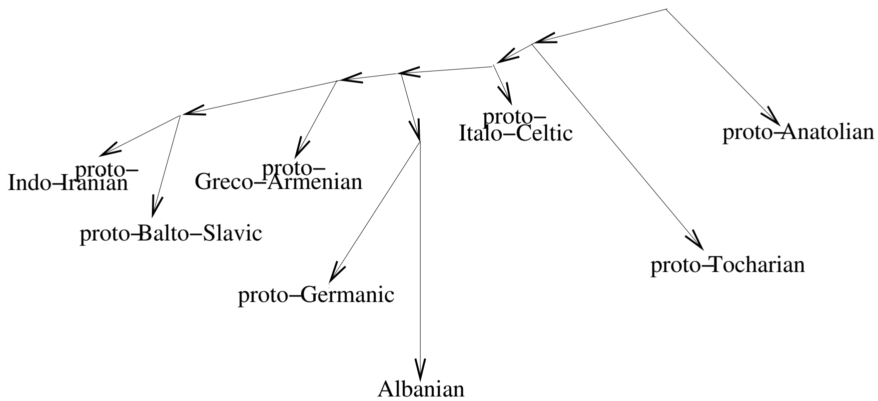

Fig. 6. A plausible phylogeny computed for Indo-European languages, using CMODELS.

While computing phylogenetic trees for the 7 language groups and the language Albanian, we have ensured that each tree satisfies the following domain-specific constraints: Anatolian is the outgroup for all the other subgroups; within the residue, Tocharian is the outgroup; within the residue of that, Italo-Celtic, and possibly Albanian are outgroups, but not necessarily as a single clade; Albanian cannot be a sister of Indo-Iranian or Balto-Slavic.

The domain-specific information above can be formalized as constraints. For instance, we can express that Anatolian is the outgroup for all the other subgroups by the constraint

$$\leftarrow \mathit{not}\ \mathit{edge}(2k, 6)$$

where $2k$ is the root of the phylogeny, and 6 denotes proto-Anatolian.

Another piece of domain-specific information is about the phonological and morphological characters. The phonological and morphological innovations (except 'P2' and 'P3') considered in the dataset are too unlikely to have spread from language to language, and that independent parallel innovation is practically excluded. Therefore, while computing phylogenetic trees, we have ensured that these characters are compatible with them. This is achieved by adding to the program the constraint

$$\leftarrow \mathit{incompatible}(i) \qquad (i \in IC \cap MP)$$

where MP is the set of all morphological and phonological characters except 'P2' and 'P3'.

With 21 informative characters, each with 2–3 essential states, we have computed 45 phylogenetic trees for the 7 language groups above and the language Albanian, in a few minutes. Out of the 45 phylogenies computed using CMODELS, 34 are identified by Don Ringe as plausible from the point of view of historical linguistics. Fig. 6 shows the most plausible one with 16 incompatible characters. This phylogeny is identical to the phylogeny presented in [31], which was computed with a greedy heuristic using the

Character	*A. Longicervica*	*A. Cerorhincae*	*A. Pygmaeus*	*A. Meinertzhageni*	*A. Campylacantha*
uterus	1	1	1	1	1
size of hooks	1	0	1	2	2
position in host	1	0	1	1	0
position of hooks	1	0	0	2	1

Fig. 7. The character states of some characters for five *Alcataenia* species.

Perfect Phylogeny software in about 8 days (Don Ringe, personal communication), and used in [29, 9, 10] to build a perfect phylogenetic network for Indo-European.

With the same Indo-European dataset obtained after preprocessing (with 21 informative characters, each with 2–3 essential states), we have also computed a most parsimonious phylogeny using the computational tool PARS: (Anatolian, Tocharian, (Greco-Armenian, ((Albanian, ((Italo-Celtic, Germanic), Balto-Slavic)), Indo-Iranian))). Some other most parsimonious phylogenies constructed for Indo-European languages are due to [30], where the authors use PAUP [34] with the dataset Isidore Dyen [6] to generate phylogenies. None of these most parsimonious trees is consistent with the domain-specific information described above, and thus none is plausible from the point of view of historical linguistics. On the other hand, we should note that Dyen's dataset is not very reliable since it is a purely lexical database from modern languages.

8 Computing Phylogenetic Trees for *Alcataenia* Species

With the computational method presented above, we can also infer phylogenies for some species, based on some morphological features. Here we have considered 9 species of *Alcataenia*—a tapeworm genus whose species live in alcid birds (puffins and their relatives): *A. Larina* (LA), *A. Fraterculae* (FR), *A. Atlantiensis* (AT), *A. Cerorhincae* (CE), *A. Pygmaeus* (PY), *A. Armillaris* (AR), *A. Longicervica* (LO), *A. Meinertzhageni* (ME), *A. Campylacantha* (CA). We have used the dataset described in [19].

In this dataset, there are 15 characters, each with 2–3 states. For some characters, their states are presented in Fig. 7. After preprocessing, we are left with 10 informative characters, each with 2 essential states.

According to [19], the outgroup for all *Alcataenia* species is *A. Larina*. We have expressed this domain-specific information by the constraint

$$\leftarrow not\ edge(2k, 0)$$

where $2k$ is the root of the phylogeny, and 0 denotes *A. Larina*.

With the dataset obtained after preprocessing, we have found out that, for *Alcataenia*, there is no phylogeny with less than 5 incompatible characters. Then we have computed 18 phylogenies, with 5 incompatible characters, for *Alcataenia*, in less than 30 minutes. One of these phylogenies is presented in Fig. 8.

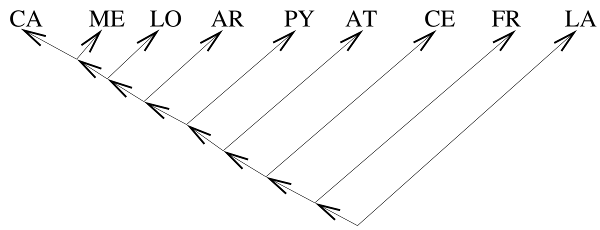

Fig. 8. A plausible phylogeny computed, using CMODELS, for *Alcataenia* species.

For the plausibility of the phylogenies for *Alcataenia*, we consider the phylogenies of its host *Alcidae* (a seabird family) and the geographical distributions of *Alcidae*. This information is summarized in Table 3 of [19]. For instance, according to host and geographic distributions over the time, diversification of *Alcataenia* is associated with sequential colonization of puffins (parasitized by *A. Fraterculae* and *A. Cerorhincae*), razorbills (parasitized by *A. Atlantiensis*), auklets (parasitized by *A. Pygmaeus*), and murres (parasitized by *A. Armillaris*, *A. Longicervica*, and *A. Meinertzhageni*). This pattern of sequential colonization is supported by the phylogeny of *Alcidae* in [4]. Out of the 18 trees we have computed, only two are consistent with this pattern. (One of them is shown in Fig. 8.) Both trees are plausible also from the point of view of historical biogeography of *Alcataenia* in *Alcidae*, summarized in [19]. Each plausible tree needs 3 lateral edges to turn into a perfect phylogenetic network.

With the *Alcataenia* dataset described above, we have computed a most parsimonious tree using PARS, which is very similar to the phylogeny of Fig. 8, and to the most parsimonious phylogeny computed for the *Alcataenia* species above (except *A. Atlantiensis*) by Eric Hoberg [18][Fig. 1].

According to [18, 19], a more plausible phylogeny for *Alcataenia* is the variation of the phylogeny of Fig. 8 where *A. Armillaris* and *A. Longicervica* are sisters. We can express that *A. Armillaris* and *A. Longicervica* are sisters by the constraint

$$\leftarrow not\ sister(2,4)$$

where 2 and 4 denote *A. Armillaris* and *A. Longicervica* respectively. By adding this constraint to the problem description, we have computed 3 phylogenies, each with 6 incompatible characters, in less than 10 minutes; their strict consensus tree is identical to the one presented in Fig. 5 of [19]. It is not the most parsimonious tree.

9 Conclusion

We have described how to use answer set programming to generate conjectures about the phylogenies of a set of taxa based on the compatibility of characters. Using this method with the answer set solver CMODELS, we have computed phylogenies for 7 Chinese dialects, and for 24 Indo-European languages. Some of these trees are plausible from the point of view of historical linguistics. We have also computed phylogenies for

9 *Alcataenia* species, and identified some as more plausible from the point of view of coevolution and historical biogeography.

Some of the plausible phylogenies we have computed (e.g. the ones computed for Indo-European) using CMODELS are different from the ones computed using other software, like PARS of PHYLIP, based on maximum parsimony. This shows that the availability of our computational method based on maximum compatibility can be useful for generating conjectures that can not be found by other computational tools.

One software that can compute phylogenies for a set of taxa based on the maximum compatibility criterion is CLIQUE (available with PHYLIP), which is applicable only to sets of taxa where a taxonomic unit is mapped to state 0 or state 1 for each character. Another one is the Perfect Phylogeny software of [31], which can compute a phylogeny with the maximum number of compatible characters only when all characters are compatible. Our method is applicable to sets of taxa (like the ones we have experimented with) where a taxonomic unit can be mapped to multiple states. Also, it guarantees to find a tree with the maximum number of compatible characters, if one exists, when all characters may not be compatible. In this sense, our method is more general than the existing ones that compute trees based on maximum compatibility.

Another advantage of our method over the existing ones mentioned above is due to answer set programming. Its declarative representation formalism allows us to easily include in the program domain specific information, and thus to prevent the reconstruction of some phylogenetic trees that are not plausible. Moreover, well-studied properties of programs in this formalism allow us to easily prove that the maximum compatibility problem is correctly described as a decision problem by the phylogeny program.

Acknowledgments We have had useful discussions with Selim Erdoğan and Vladimir Lifschitz on the formalization of the problem, and with Wang Feng on the plausibility of phylogenies for Chinese dialects. Eric Hoberg, Luay Nakhleh, William Wang, and Tandy Warnow have supplied relevant references. Brooks was supported by an NSERC Discovery Grant to DRB. Ringe was supported by NSF BCS 03-12911. Erdem was supported in part by FWF P16536-N04; part of this work was done while she visited the University of Toronto, which was made possible by Hector Levesque and Ray Reiter.

References

1. D. R. Brooks, R. L. Mayden, and D. A. McLennan. Phylogeny and biodiversity: Conserving our evolutionary legacy. *Trends in Ecology and Evolution*, 7:55–59, 1992.
2. D. R. Brooks and D. A. McLennan. *Phylogeny, Ecology, and Behavior: A Research Program in Comparative Biology*. Univ. Chicago Press, 1991.
3. J. H. Camin and R. R. Sokal. A method for deducing branching sequences in phylogeny. *Evolution*, 19:311–326, 1965.
4. R. M. Chandler. *Phylogenetic analysis of the alcids*. PhD thesis, University of Kansas, 1990.
5. W. H. E. Day and D. Sankoff. Computational complexity of inferring phylogenies by compatibility. *Systematic Zoology*, 35(2):224–229, 1986.
6. I. Dyen, J. B. Kruskal, and P. Black. An Indoeuropean classification: a lexicostatistical experiment. *Transactions of the American Philosophical Society*, 82:1–132, 1992.
7. A. W. F. Edwards and L. L. Cavalli-Sforza. Reconstruction of evolutionary trees. *Phenetic and Phylogenetic Classification*, pp. 67–76, 1964.

8. E. Erdem and V. Lifschitz. Tight logic programs. *TPLP*, 3(4–5):499–518, 2003.
9. E. Erdem, V. Lifschitz, L. Nakhleh, and D. Ringe. Reconstructing the evolutionary history of Indo-European languages using answer set programming. In *Proc. of PADL*, pp. 160–176, 2003.
10. E. Erdem, V. Lifschitz, and D. Ringe. Temporal phylogenetic networks and answer set programming. In progress, 2004.
11. S. T. Erdoğan and V. Lifschitz. Definitions in answer set programming. In *Proc. of LPNMR*, pp. 114–126, 2004.
12. J. Felsenstein. Numerical methods for inferring evolutionary trees. *The Quarterly Review of Biology*, 57:379–404, 1982.
13. J. Felsenstein. PHYLIP (Phylogeny inference package) version 3.6.
14. L. R. Foulds and R. L. Graham. The Steiner tree problem in Phylogeny is NP-complete. *Advanced Applied Mathematics*, 3:43–49, 1982.
15. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP/SLP*, pp. 1070–1080, 1988.
16. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
17. W. Hennig. *Grundzuege einer Theorie der Phylogenetischen Systematik*. Deutscher Zentralverlag, 1950.
18. E. P. Hoberg. Evolution and historical biogeography of a parasite-host assemblage: *Alcataenia* spp. (Cyclophyllidea: Dilepididae) in Alcidae (Chradriiformes). *Canadian Journal of Zoology*, 64:2576–2589, 1986.
19. E. P. Hoberg. Congruent and synchronic patterns in biogeography and speciation among seabirds, pinnipeds, ans cestodes. *J. Parasitology*, 78(4):601–615, 1992.
20. Yu. Lierler and M. Maratea. Cmodels-2: SAT-based answer sets solver enhanced to non-tight programs. In *Proc. of LPNMR*, pp. 346–350, 2004.
21. V. Lifschitz. Answer set programming and plan generation. *AIJ*, 138:39–54, 2002.
22. V. Lifschitz. Introduction to answer set programming. Unpublished draft, 2004.
23. J. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 1984.
24. V.H. Mair, editor. *The Bronze Age and Early Iron Age Peoples of Eastern Central Asia*. Institute for the Study of Man, Washington, 1998.
25. J.P. Mallory. *In Search of the Indo-Europeans*. Thames and Hudson, London, 1989.
26. V. Marek and M. Truszczyński. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: a 25-Year Perspective*, pp. 375–398, 1999.
27. J. W. Minett and W. S.-Y. Wang. On detecting borrowing: distance-based and character-based approaches. *Diachronica*, 20(2):289–330, 2003.
28. M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proc. of DAC*, 2001.
29. L. Nakhleh, D. Ringe, and T. Warnow. Perfect phylogenetic networks: A new methodology for reconstructing the evolutionary history of natural languages. *Language*, 2005. To appear.
30. K. Rexova, D. Frynta, and J. Zrzavý. Cladistic analysis of languages: Indo-European classification based on lexicostatistical data. *Cladistics*, 19:120–127, 2003.
31. D. Ringe, T. Warnow, and A. Taylor. Indo-European and computational cladistics. *Transactions of the Philological Society*, 100(1):59–129, 2002.
32. R.G. Roberts, R. Jones, and M.A. Smith. Thermoluminescence dating of a 50,000-year-old human occupation site in Northern Australia. *Science*, 345:153–156, 1990.
33. P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *AIJ*, 138:181–234, 2002.
34. D.L. Swofford. PAUP* (Phylogenetic analysis under parsimony) version 4.0.
35. J.P. White and J.F. O'Connell. *A Prehistory of Australia, New Guinea, and Sahul*. Academic Press, New York, 1982.