A logic approach to modelling nomenclatural change

Nico M. Franz, Chao Zhang and Joohyung Lee

School of Life Sciences, Arizona State University, PO Box 874501, Tempe, AZ 85287-4501, USA; School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, PO Box 878809, Tempe, AZ 85287-8809, USA

Accepted 10 March 2017

Abstract

We utilize an Answer Set Programming (ASP) approach to show that the principles of nomenclature are tractable in computational logic. To this end we design a hypothetical, 20 nomenclatural taxon use case, with starting conditions that embody several overarching principles of the International Code of Zoological Nomenclature, including Binomial Nomenclature, Priority, Coordination, Homonymy, Typification and the structural requirement of Gender Agreement. The use case ending conditions are triggered by the reinterpretation of the diagnostic features of one of 12 type specimens anchoring the corresponding species-level epithets. Permutations of this child-to-parent reassignment action lead to 36 alternative scenarios, where each scenario requires a set of 1–14 logically contingent nomenclatural emendations. We show that an ASP transition system approach can correctly infer the Code-mandated changes for each scenario, and visually output the ending conditions. The results provide a foundation for further developing logic-based nomenclatural change optimization and validation services, which could be applied in global nomenclatural registries. More generally, logic explorations of nomenclatural and taxonomic change scenarios provide a novel means of assessing design biases inherent in the principles of nomenclature, and can therefore inform the design of future, big data-compatible identifier systems that recognize and mitigate these constraints.

Introduction

The origins of modern biological nomenclature trace back to the age of the Renaissance (Minelli, 2003), culminating in the foundational works of Linnaeus (1753, 1758). The Linnaean rules were subsequently embedded, expanded and refined in the Codes (e.g. ICZN, 1999; McNeill et al., 2012), thus having remained in use for more than 250 years (Schuh, 2003; Dayrat, 2010). In spite of such remarkable persistence, the merits of Linnaean nomenclature are continuously under scrutiny (e.g. Bryant and Cantino, 2002; Ereshefsky, 2007; Dayrat, 2010; Dubois, 2011; Vences et al., 2013; Remsen, 2016). Indeed, the demands on any system for identifying the diversity of life are not trivial, and may include: precise and reliable connections to physical vouchers; congruence with central tenets in evolutionary and systematic theory; responsiveness to changing taxonomic and phylogenetic perspectives; proper attribution of authorship—original or revisionary; stability in usage across spatial and temporal dimensions; transparency in application; support for community-sanctioned rules and updates that accommodate new naming challenges; and facilitation of everyday use among human speakers and general alignment with evolutionarily constrained human cognitive strengths (Atran, 1998; Sterner and Franz, 2017). No system of nomenclature can fulfill all of these and other potentially conflicting criteria to the highest degree.

Here we introduce a novel element into the longstanding discourse about the Linnaean nomenclature, i.e. its relationship to computational logic. Methods of knowledge representation and reasoning (Brachman and Levesque, 2004) are on the rise in biomedical and evolutionary disciplines (Smith et al., 2007; Franz and Thau, 2010; Panahiazar et al., 2013; Deans et al., 2011).
We can conceive of at least four reasons why a better integration of nomenclatural practices with logic representation and reasoning is desirable. (1) The language controlled by the rules of nomenclature shapes the ways in which humans communicate our collective knowledge about life (Patterson et al., 2010). Hence an exploration of the extent to which these rules are amenable to logic may be of interest in its own right. (2) In light of the current trend towards ontology-driven representation of biodiversity data (e.g. Mungall et al., 2010; Midford et al., 2013; Walls et al., 2014), it appears relevant to assess whether nomenclatural entities and relationships can, and should, be integrated with these efforts. (3) Taxonomic communities are in the process of creating registries for names, nomenclatural acts and related information (e.g. Pyle and Michel, 2008). New submissions could be vetted, with logic reasoners configured to enact compliance with nomenclatural rules, resulting in improved quality control (Patterson et al., 2016). (4) Lastly, due to the complex interactions between evolving taxonomic perspectives and nomenclatural emendations, the ways in which such changes are implemented can have varying effects on nomenclatural stability (e.g. van der Linde and Houle, 2008; ICZN, 2010; Nicholson et al., 2012). Whenever alternatives exist to reconcile taxonomic and nomenclatural changes, logic reasoners can be deployed to maximize nomenclatural stability or other criteria (Vences et al., 2013). In summary, representing nomenclatural rules in computational logic may yield both theoretical and practical benefits. What do we mean by “nomenclatural rules”? To clarify, the domain-specific Codes of nomenclature are not a monolithic body of rules applicable to all life (Hawkesworth, 2001; David et al., 2012). The particularities of each rule set, combined with the histories of certain taxonomic names, can lead to unique complexity challenges, frequently requiring expert knowledge, interpretation and deliberate choice to attain nomenclatural resolution (e.g. Vane-Wright, 2003; Rhodin and Carr, 2009). However, in spite of inter-Code differences and sometimes bewildering or ambiguous use cases, there are several overarching principles of Linnaean nomenclature—such as Priority and Typification—that can be translated into general, conditional if–then statements, and thus become tractable for logic-based reasoning. Our focus is on representing several of these foundational nomenclatural principles in logic.

Here we create a hypothetical 20 nomenclatural taxon use case and logically model its transition following a specific taxonomic action (Fig. 1). The starting conditions embody several core principles (henceforth capitalized) of the International Code of Zoological Nomenclature (ICZN, 1999). These include: Binominal Nomenclature (Article 5), Priority (Article 23), Coordination (Article 36), Homonymy (Article 52), Typification (Article 61) and the structural requirement of Gender Agreement (Article 31.2). The starting conditions are encoded into an Answer Set Programming (ASP) language and program (Gelfond and Lifschitz, 1988; Gelfond, 2008; Brewka et al., 2011). In a subsequent step, modelled with a transition system approach (Lifschitz and Turner, 1999), one of 12 species-level names (child) and the corresponding type specimen are transferred to another genus name (parent). Permutations of this single child-to-parent reassignment action lead to 36 alternative scenarios, where each ending condition requires a specific set of 1–14 logically contingent nomenclatural emendations. We show that an ASP reasoner (Gebser et al., 2011) can correctly infer the Code-mandated changes for each input and transition scenario, and visually output the ending conditions. In the Discussion, we assess the outcomes in relation to a broader discourse about the interaction of nomenclature, taxonomy and logic-based representations of systematic change. We also address how feasible and desirable a wider implementation of logic-enabled nomenclatural services would be.

Materials and methods

To our knowledge this is the first application of an ASP transition system approach to model an inference process in the nomenclatural–taxonomic domain (but see Brooks et al., 2007; Gebser et al., 2008; and Franz et al., 2016a; for different uses of ASP in biology). We regard ASP as an alternative or complementary solution to the challenge of knowledge representation and reasoning in biology, where Description Logics is the prevailing paradigm (Grenon et al., 2004; Smith et al., 2007; Baader et al., 2008; Deans et al., 2015; Thessen et al., 2015). To bring the readership into this field, we first describe the properties of ASP in relation to the inference needs for the use case (Fig. 2). This review sets the stage for modelling of the nomenclatural change scenario.

Properties of the Answer Set Programming approach

The term Answer Set Programming was coined in the late 1990s for a declarative form of logic programming rooted in non-monotonic reasoning and stable model semantics (Lifschitz, 2008; Eiter et al., 2009; Brewka et al., 2011). Problems are solved in ASP by

2015; Thessen et al., 2015). But this dynamic has not yet permeated the realm of nomenclature (but see Woodger, 1937; Gregg, 1954; Mahner and Bunge, 1997; Sereno, 2005; Tuominen et al., 2011; Chawutthai et al., 2013; Cellinese and Lapp, 2015; Dmitriev and Yoder, 2017).
specifying sets of rules and constraints and then generating stable models that satisfy these. Computational problems are solved by grounding the specified input domains, variables and conditions during the reasoning process. Application of the Davis–Putnam–Logemann–Loveland (DPLL) procedure satisfies the given constraints (Gomes et al., 2008). Using ASP solvers, we can generate all or as many answer sets as needed.

The ASP approach has several desirable properties in connection to the complex rules of nomenclature, as follows. (1) ASP uses an expressive language resembling that of other Semantic Web languages (Eiter et al., 2008). (2) There are powerful ASP reasoners (answer set solvers) such as the Potassco Answer Set Solving Collection (Gebser et al., 2011). (3) Unlike Description Logic, the approach is based on the modelling conventions of the generate-and-test paradigm (Dixon and Simmons, 1983) and applies the closed world assumption, which permits deciding that conditions not proven to be true are false (“negation as failure”; Gelfond and Lifschitz, 1988). (4) It offers elaboration tolerance, i.e. the ability to retain reasoning abilities while taking into account additional constraints (McCarthy, 1998). (5) It also supports default reasoning, i.e. the ability to stipulate the truth of certain conditions unless specified otherwise (Reiter, 1980). (6) ASP can represent transition systems through automated reasoning about action domains, by generating plans for such domains. (7) Under certain conditions, First-Order Logic and Description Logic can be accommodated in stable model semantics, although not necessarily vice versa (Eiter et al., 2008; Kutz et al., 2010). Jointly these properties make ASP highly suitable for performing complex, exception-rich, rule-based reasoning in the biological domain. However, (8) at present most ASP applications lack an end user interface as mature as (e.g.) the Protégé platform (http://protege.stanford.edu/), instead requiring knowledge of command line and logic program coding.

Starting conditions

To set the stage for this use case, we must first specify the nomenclatural and taxonomic boundaries and reasoning intentions (Figs 1 and 2; Tables 1 and 2). The ICZN—International Commission on Zoological Nomenclature (1999) contains 90 articles. Of these, only a subset can be represented here. We also recognize that certain nomenclatural changes are triggered independently of new taxonomic insights (e.g. a lapsus calami; Article 32.5). Conversely, not all taxonomic changes are mirrored in nomenclatural adjustments (Franz, 2005; Franz et al., 2008, 2016a,b; Remsen, 2016). These various dependencies between nomenclatural and taxonomic emendations make it necessary to characterize the taxonomic starting and change conditions for the use case as unambiguously as possible. We thereby model specific principles and articles of the ICZN that guide the reasoning process towards succinct ending conditions.

The taxonomic names themselves are another set of variables in need of prior specification. Given a taxonomic change scenario, the reasoner can infer that certain nomenclatural emendations are necessary. However, the reasoner has no capacity to conceive which names (letter strings) should be used. For instance, under the proper rules and change conditions, the program can infer that the species-level epithet tertius will require replacement with a new, valid, masculine genus name. But it cannot decide whether the name string Zyzzyzus (Brickmann-Voss and Calder, 2013) is preferentially suited for this purpose—in terms of availability, Latin grammar (Casadio and Lambek, 2005), and perhaps also cognitive or aesthetic considerations. Consequently, we must provide the reasoner upfront with a set of Code-compliant name options that can be applied—as logically required—to execute the transition.

The use case entails two temporal conditions or steps [time (t) = 0, 1]. One condition occurs prior to the taxonomic change (t = 0). The other condition occurs subsequent to the taxonomic change (t = 1). The pre-change phase is set in the year 1985, whereas the post-change phase is set in the year 2000. Prior to the change, 20 ranked, nomenclaturally connected names exist: one each at the family and subfamily level (endings -idae and -inae); two at the tribal level (-ini); four at the genus level; and 12 at the species level (Fig. 1a). Each name is assigned a unique numerical identifier, displayed in square brackets: [1], [2], . . . , [20]. Lower-level child names are allocated in equal numbers among their respective parent names, i.e. each generic name contains three species names, and each tribal name contains two generic names. The 12 species-level names were validly published in two preceding time clusters, referred to as (1) the “type period”—extending from 1775 to 1790—and (2) the “non-type period”—extending from 1950 to 1985. The publication years for species-level epithets are specified such that each generic name accommodates exactly one epithet from the earlier type period and two epithets from the later non-type period. No two species-level names were published in the same year, thereby rendering judgments of Priority unambiguous.

As a consequence of Coordination, the distribution of type period publication years at the species level generates four distinct lineages of Priority, namely: (1) Agenus primus, 1775—[9], [5], [3], [2] and [1]; (2) Egena secunda, 1780—[12] and [6]; (3) Igena prima, 1785—[15], [7] and [4]; and Ogenus secundus, 1790—[18] and [8]. Two generic names are masculine ([5], [8]) and two
are feminine ([6], [7]), respectively. Moreover, two pairs of epithets share the same Latin root, either prim- ([9], [15]) or second- ([12], [18]). This circumstance has implications for Homonymy under certain change scenarios.

Each of the species-level names is anchored by a single, vouched, unambiguously assigned type specimen (Fig. 1a). Note, however, that “assignment” is used here in the nomenclatural sense, i.e. there is agreement regarding which specimen is being referred to (Witteveen, 2015, 2016). Such an unambiguous nomenclatural assignment differs from agreeing on the taxonomic identity. More precisely, this leaves room for interpretation regarding certain taxonomic traits that the nomenclatural type specimen in question may or may not exhibit, according to an expert’s re-/assessment.

Because the logic of reasoning ultimately falls back on the expert-perceived taxonomic identity of type specimens, we need to stipulate traits for the latter as

![Diagram](image-url)
Fig. 1. (a) Starting conditions for the hypothetical 20 nomenclatural taxon use case at time = 0. The internal nodes and terminal edges of the topology are identified by numbers in square brackets. Each ranked taxonomic name and year of origin are specified to establish Priority. Nomenclatural type relationships are indicated with the “*” symbol; for example, the species-level name *Agenus primus*, 1775, typifies the genus-level name *Agenus*, 1775. The coloured shapes located to the right of each species-level name represent the corresponding type specimens, as designated and described at time = 0. (b) Representation of 12 of the 36 possible change scenarios, focusing on the corresponding type specimen reinterpretations (see also Table 2). The four represented types are: *A. primus* (scenarios 1–3), *Igena secunda* (scenarios 10–12), *Igena prima* (scenarios 19–21) and *Ogenus secundus* (scenarios 28–30)—all as recognized at time = 0. Each type also has Priority over its other two congeners at time = 0, thus making these change scenarios more nomenclaturally complex than the 24 other change scenarios. The type's shape as interpreted at time = 0 is shown to the left, whereas the reinterpreted shape at time = 1 (also coloured green) is shown to the right. (c) Representation of change scenario 5 with ending conditions at time = 1, involving the reinterpretation of the type of *Agenus tertius*, 1950 (10), as recognized at time = 0 to have a hemi-elliptical instead of a square shape (see also Table 2). Because the epithet *tertius*, 1950, does not have Priority over all other epithets associated with either the source genus-level name *Agenus* [5] or the target genus-level name *Igena* [7], scenario 5 is one of the 24 scenarios requiring only minimal nomenclatural change—the creation of *Igena tertia*, (1950), new combination (coloured in red). In this case, the new combination also involves changing the epithet's ending from *-us to -a*, to comply with Gender Agreement. [Colour figure can be viewed at wileyonlinelibrary.com].

Fig. 2. Overview of the ASP transition system approach, as applied to the nomenclatural use case; this is modified from Eiter (2008: 5). The Problem Instance I—including nomenclatural rules and taxonomic starting, change and ending constraints—is encoded as a non-monotonic Logic Program P (Data S1). The latter is submitted to the ASP solver clingo (combining the grounder gringo with the solver claspD; Data S2), which produces an output of 36 Stable Model Solutions. We display the clingo output both as text (output directly by clingo; Data S3) and as GraphViz visualizations (Data S5) generated via a translational Java program (Data S4).

well (Fig. 1a). Accordingly, each type specimen within the same genus-group name varies in one species-level taxonomic trait related to its pigmentation: (1) unpigmented, (2) lightly pigmented and (3) strongly pigmented. In addition, all three type specimens assigned to the same genus-group name prior to the change share a second, genus-level diagnostic feature, as follows: (i) the type specimens that anchor the three *Agenus* species names have a square shape; (ii) those of the three *Igena* species names have a rectangular shape; (iii) those of the three *Igena* species names have a hemi-elliptical shape; and (iv) those of the three *Ogenus* species names have a pentagonal shape. The combination of three species-level and four genus-level diagnostic traits establishes each of the 12 type specimens as taxonomically unique in the original assessment. Jointly, the above conditions specify the pre-change status of the use case, which is complete as of 1985.

The change–type specimen reinterpretation

The transition from the pre- to the post-change phase is triggered by the reinterpretation of the taxonomic identity of one individual type specimen (Fig. 1b). This reinterpretation occurs in the year 2000. Recall that the respective shapes of the four specimens anchoring the type period epithets (1775–1790)—square, rectangular, hemi-elliptical and pentagonal—formed the diagnostic foundation for delimiting the corresponding genus-level concept to which additional species-level entities were subsequently assigned. These (eight) additions were performed under the assumption of taxonomic correspondence, thus effectively reaffirming the original shape descriptions. However, in 2000 matters are reassessed with a different outcome. The original interpretation of a type specimen’s diagnostic feature is now found to have been erroneous, creating a case of “reference by misdescription” (Donnellan, 1966). Instead, the type specimen is diagnosed to exhibit one of the other shapes present in species-level entities assigned to one of the respective three generic entities (Fig. 1b). The use case entails all possible permutations of type specimen rediagnoses to match alternative shapes.

While the example is contrived, in practice such type reinterpretations are not infrequent (e.g. Ribot et al., 1996; Woodley et al., 2011; Cappellini et al., 2013; Laloy et al., 2013; Witteveen, 2015). Reassessments of the taxonomic identity of type specimens are very frequently involved in determining heterotypic synonymy. In the present use case, the type reinterpretation...
requires an adjustment that achieves taxonomic congruence across the named entities, such that similarly shaped type specimens are assigned to the same genus-level name.

The reinterpretation of the type specimen has specific consequences. In particular, the 1:1 cardinality relationship between type specimens and valid species-level epithets is not altered through the transition, remaining at 12:12 for each phase and permutation. With regards to taxonomic or topological congruence, a genus-level concept (parent) that previously contained three species-level concepts (children) now entails one species-level concept fewer, whereas another parent entails one additional child. Considering the 3-3-3-3 numerical allocation of specific to generic entities in the pre-change phase (Fig. 1a), this means that exactly two of these numbers shift to 2 and 4, respectively, whereas the others remain at 3. Because the asserted shape of the type specimen anchoring each of the 12 species-level epithets is potentially “corrigible” to match any of the three alternative shapes (Fig. 1b), there are 36 possible ending conditions for the use case (12 type specimens × 3 alternative shapes).

**Ending conditions**

Each of the 36 ending scenarios produced in the year 2000 (t = 1) requires application of a set of logically contingent, Code-mandated nomenclatural rules, and therefore triggers specific but different nomenclatural emendations. The particular rules and changes to be enacted depend (1) on the individual type specimen under reinterpretation and (2) the morphological shape newly ascribed to the specimen under reinterpretation. Differential sets of principles and articles in the ICZN—International Commission on Zoological Nomenclature (1999) are invoked accordingly.

The outcomes fall into two broad categories. Twenty-four of the 36 ending conditions are caused by reinterpretations of type specimens anchoring eight species-level epithets that were created in the later (non-type) phase, i.e. [10], [11], [13], [14], [16], [17], [19] and [20]. These names lack Priority in relation to other same-ranked names typifying both the source and the target genus-level names between which the names transition. Thus, each of these transfers requires only a limited set of nomenclatural changes: specifically, the creation of one new combination, with or without epithet changes to comply with Gender Agreement (Fig. 1c).

The remaining 12 transition scenarios are nomenclaturally more complex (Figs 3–6). Here the act of reinterpretation affects type specimens anchoring one of four species-level epithets created in the earlier (type) phase, i.e. [9], [12], [15] and [18]. Priority and Coordination considerations now drive the nomenclatural outcomes, through multi-level dependencies. First, the epithets whose types are reinterpreted have Priority over each of the two epithets accommodated under the respective (starting phase) genus-level name (Fig. 1a). We may call this kind of Priority relative to initially congeneric epithets “local Priority”. An example of an epithet with local Priority is secundus, 1790—relative to nomus, 1980, and decimus, 1985. “Global Priority”, in turn, holds throughout the entire nomenclatural hierarchy. The epithet primus, 1775, has this status among all 12 epithets. Hence, in deciding on the appropriate nomenclatural emendations for the 12 more complex cases, one must assess the relative Priority among paired epithets that each have local Priority: for example, (1) primus, 1775, versus (2) secundus, 1780; or (1) prima, 1785, versus (2) secundus, 1790. Whichever epithet lacks Priority in the pairwise comparison effectively transitions into a new combination, where the post-transition genus-level name is that corresponding to the Priority-carrying epithet.

Second, we need to account for the principle of Coordination, which further complicates matters. Consider the transition involving an epithet with initial, local Priority—e.g. secundus, 1790—in relation to another such epithet, which has Priority in the more expansive context—e.g. secunda, 1780 (Fig. 4c). In such cases the two epithets’ respective genus-level names—Egena, 1780, and Ogenus, 1970—also enter into synonymy: only one name remains valid. In some instances, the coordinated tribe-level names must enter into synonymy as well; see ICZN—International Commission on Zoological Nomenclature (1999), Article 40.1 [see also Article 40.a.i in the preceding ICZN (1985)]. In addition, certain epithets that originated in the non-type period will newly acquire local Priority, and thereby trigger the creation of new genus- and/or tribe-level names, new typifications of these, and new genus name/epithet combinations (e.g. Fig. 4c).

Lastly, no taxonomic change in the use case can affect the validity of the nomenclatural lineage extending from Agenini, 1775, to A. primus, 1775 ([3] → [5] → [9]), which has global Priority (Fig. 3). In three of the 12 complex cases where the type specimen anchoring the epithet primus, 1775, is reinterpreted to have a non-square shape, this novel insight will not affect the validity of the globally Priority-carrying name lineage. Instead, it is required that the other lineage in the comparison, having local Priority, move into synonymy. The two additional, non-type period epithets subsumed under that latter lineage are thus also newly combined with the genus-level name Agenus. Moreover, the two names Agenus teritus, 1950, and Agenus quartus, 1955, no longer represent valid combinations (given their square-shaped types), and trigger the creation and typification of a new genus-level name.
Table 1
High-level ASP pseudocode for the 20 nomenclatural taxon use case; the actual clingo-compatible code, input commands, supporting (Java) scripts and output results (clingo text, GraphViz visualizations) are presented in the Data S1–S5, respectively

| Section | Constraint or action modelled in Answer Set Programming logic |
|---------|---------------------------------------------------------------|
| 1.      | Define the transition system domains, variables and starting conditions |
| 1.1.    | Define a system that has two phases or steps: 0 = original; 1 = revised. |
| 1.2.    | Define that the system has 20 nodes, labelled node 1 to node 20. One or more nodes can share the same rank (family to species). |
| 1.3.    | Define that rank-specific nodes are part of the entire set of nodes. |
| 1.4.    | Specify that higher-level names are constrained by coordinated (type-lineage) identities (Coordination). Generic names and epithets are constrained by gender identities and corresponding spellings (Gender Agreement). |
| 1.5.    | Define multiple domains for nodes: generally, and for each taxonomic rank (family to species). |
| 1.6.    | Specify the original tree shape at time = 0, with 19 edges connecting the 20 nodes. |
| 1.7.    | Specify the original generic name/epithet combinations for the corresponding nodes at time = 0, with the corresponding publication years and gender identities. |
| 1.8.    | Constrain each epithet to remain assigned to its original node; thereby, species-level node numbers remain consistent throughout the two phases of the transition scenario. |
| 1.9.    | Specify the coordinated (type-lineage) identities for sets of names—including each name’s prefix and rank-specific endings—for the starting and ending conditions. |
| 1.10.   | Constrain the publication year for each name, including new names created in the year 2000 (but excluding epithets which were previously specified in 1.7.) |
| 2.      | Specify the globally applicable uniqueness and existence constraints |
| 2.1.    | Constrain that each node is assigned a single, rule-compliant name—either mono- or binomial—as well as the year and the edge to the parent node, except the root node which has no parent node. |
| 3.      | Specify the globally applicable inertia conditions for entities and relationships that remain consistent unless change is mandated by the transition model |
| 3.1.    | Constrain that all edges, node-name-year-gender assignments, and node years specified as valid at time = 0 will remain so at time = 1, ranging from the family level down to the genus level (however, species-level edges may change). |
| 4.      | Specify that the Priority of an epithet-level node and its coordinated higher-level nodes is established by having the more senior publication year in comparison to another epithet-level node (Priority) |
| 5.      | Assign nodes persistently to their respective names and years at each taxonomic rank (family to species). |
| 6.      | Infer the rule-mandated nomenclatural changes at time = 1, for persisting and newly valid names (ending conditions) |
| 6.1.    | Constrain that the valid generic name (at any time) has the senior, Priority-carrying epithet. |
| 6.2.    | Constrain that epithets persist from time = 0 to time = 1; however, the corresponding generic name is determined by the identity of the senior, Priority-carrying epithet. |
| 6.3.    | Infer that an epithet is combined with a new generic name if it newly acquires Priority at time = 1 through the transfer of a previously Priority-carrying epithet (at time = 0). |
| 6.4.    | Exception to 6.3.: The node of the more senior, Priority-carrying epithet is not transferred in the transition from time = 0 to time = 1. Instead, another, not Priority-carrying node is transferred “inversely”. |
| 6.5.    | Infer that the tribe-level name is coordinated (newly, if required) with the Priority-carrying genus-level name. |
| 6.6.    | Infer that the subfamily-level name is coordinated (newly, if required) with the Priority-carrying tribe-level name. |
| 6.7.    | Infer that the family-level name is coordinated (newly, if required) with the Priority-carrying subfamily-level name. |
| 7.      | Specify the available new names and their respected years of publication, as required by previous constraints |
| 8.      | Specify the changed taxonomic conditions (stable models) that are valid at time = 1: choose exactly one species-level node to be transferred as the child of another parent genus-level node than that which it had at time = 0 |
| 8.1.    | Select one species-level node to change its parent genus-level node at time = 1. |
| 8.2.    | Exception to 8.1.: The change is not executed if the new node placement affects the same original edge (i.e. no effective change). |
| 8.3.    | Establish that the new placement of the species-level node creates an edge to the corresponding genus-level node. |
| 9.      | Identify and aggregate the required nomenclatural changes at time = 1 |
| 9.1.    | Record a new combination when the node names at times = 0 and 1 are not identical. |
| 9.2.    | Record a new placement strictly in recognition of the relative Priority of locally and globally senior children; in particular, the latter are not transferred (as specified in 6.4.). |
| 9.3.    | Record all nomenclaturally valid and synonymous names at time = 1. |
| 9.3.1.  | Create a species-level synonym if an epithet combined with a generic name at time = 0 is combined with another generic name at time = 1. |
| 9.3.2.  | Create a genus-level synonym if a locally Priority-carrying epithet at time = 0 is combined with another generic name at time = 1. |
| 9.3.3.  | Create a tribe-level synonym if a locally Priority-carrying generic name at time = 0 is transferred to a tribal name with a globally Priority-carrying generic name at time = 1. |
| 9.4.    | Create all required new names at time = 1. |
| 9.4.1.  | Categorize all unchanged names as those whose corresponding nodes experience no name change at time = 1. |
| 9.4.2.  | Create a new generic name, with Gender Agreement, for those genus-level nodes that do not fall under 9.4.1. |
| 9.4.3.  | Create a new tribal name, with Gender Agreement, for those tribe-level nodes that do not fall under 9.4.1. |
Thus, even though the name A. primus, 1775, is nomenclaturally stable across all 36 scenarios, reinter-
pretations of the type specimen anchoring this name can have considerable and cascading nomenclatural
effects for other names and relationships modelled in the use case permutations (Fig. 3).

Documentation of logic approach and outcomes

We assume that the ASP approach is novel to most
readers, and therefore make an effort to present the
methods and results in an accessible way. Our analysis
entails two main components: (1) the specification of a
transition system, with starting and ending conditions
that require application of multiple principles of zoo-
logical nomenclature (Figs 1 and 2); and (2) the trans-
lation of these conditions and action rules into a logic
program that an ASP reasoner can analyse to generate
the 36 possible outcomes (Tables 1 and 2). The finer
details of our logic program encodings are probably
not of great interest to the systematics readership, and
therefore make an effort to present the
methods and results in an accessible way. Our analysis
entails two main components: (1) the specification of a
transition system, with starting and ending conditions
that require application of multiple principles of zoo-
logical nomenclature (Figs 1 and 2); and (2) the trans-
lation of these conditions and action rules into a logic
program that an ASP reasoner can analyse to generate
the 36 possible outcomes (Tables 1 and 2). The finer
details of our logic program encodings are probably
not of great interest to the systematics readership, and
are therefore presented in the Data S1–S5.

The code for this program was initially written in the
input language F2lp (Lee and Palla, 2009). F2lp is an
implementation of the First-Order Logic stable model
semantics by Ferraris et al. (2011) that translates this
logic into the input language of the ASP solvers. The final
code version directly interfaces with software tools
produced by Potassco—the Potsdam Answer Set Solv-
ing Collection (Gebser et al., 2011). The code is pro-
cessed with the system clingo, version 3.0.5., which
combines the variable grounder gringo with the Answer
Set solver claspD to process disjunctive logic programs
(Gebser et al., 2015). The code comprises fewer than
600 lines, including annotations (Data S1).

The output inferred by the ASP solver contains
both hierarchical taxonomic and nomenclatural infor-
mation, in particular: (1) relationships (edges) among
nodes in the single starting tree and set of 36 ending
trees, and (2) information on the names and nomen-
clatural relationships, actions and changes that reflect
the ending conditions for each node. To visually rep-
resent and confirm the correctness of the outcomes,
we configured a simple Java program that translates the
clingo text string output into.dot files. The latter
can be displayed as taxonomic trees with edges and
named nodes in the open source GraphViz visualization
software (Gansner and North, 2000). The nomenclatural (relationship) information is output
directly in textual format by clingo. Information on
the corresponding clingo commands, textual reasoner
output, Java visualization program and generated
GraphViz visualizations is provided in the Data S2–
S5, respectively—all are also available via the Dryad
Digital Repository and GitHub.

The Results are separated into two main sections.
The first of these focuses on presenting the principal
components and actions of the logic program as a
high-level pseudocode. This non-technical code review
illustrates in principle how our ASP approach succeeds
in representing the taxonomic and nomenclatural start-
ing conditions and rule-compliant transitions. The sec-
ond section presents the use case outcomes, empha-
sizing the relationship of nomenclatural and
taxonomic changes across the 36 transition scenarios.
To this end, we aggregate the numbers of new combi-
nations, new names (species, genus, tribe), new syn-
nonyms (genus, tribe) and new typifications (genus,
tribe), at the respective taxonomic ranks. For new
combinations and new species names, we also report

Table 1
(Continued)

| Section | Constraint or action modelled in Answer Set Programming logic |
|---------|---------------------------------------------------------------|
| 9.4.4.  | If the transfer at time = 1 creates a new generic name/epithet combination that is homonymous with another (valid) one, then modify the epithet of the new combination, with Gender Agreement, to prohibit this outcome. |
| 9.5.    | Create all required new typifications at time = 1. |
| 9.5.1.  | Establish new genus-level typifications by assigning the Priority-carrying epithet to the new genus-level name at time = 1. |
| 9.5.2.  | Establish new tribe-level typifications by assigning the Priority-carrying genus-level name to the new tribe-level name at time = 1. |
| 9.6.    | Aggregate all nomenclatural novelties at time = 1. |
| 9.6.1.  | Aggregate all new generic name/epithet combinations. |
| 9.6.2.  | Aggregate all new placements of species-, genus- and tribe-level names. |
| 9.6.3.  | Aggregate all genus- and tribe-level synonyms. |
| 9.6.4.  | Aggregate all new genus- and tribe-level names. |
| 10.     | Output the ending conditions: taxonomy visualization constraints and aggregated nomenclatural changes |
| 10.1.   | Prohibit the display of non-terminal nodes (at any rank) that lack children (family to species). |
| 10.2.   | Show valid species-level names by providing the genus-level name, gender-agreeing epithet and year of publication. |
| 10.3.   | Constrain the new taxonomic hierarchy to show only the set of revised edges, with all nodes showing the Priority-carrying names. |
| 10.4.   | Show, for each stable model, the changed species-level placement (specified in 8.1.), indicate whether 6.4. (“inverse transfer”) is invoked and output the aggregate nomenclatural changes. |
Overview of the 36 transition scenarios for the 20 nomenclatural taxon use case that involve the reinterpretation of one type specimen, with resulting taxonomic and nomenclatural emendations

| Scenario | Type being rediagnosed (target generic name) | Intended transfer | Inverse move | Primary new nomenclatural relationship (rediagnosis) | Comb. nov. (Gen. coord.) | Nomen nov. | Syn. nov. | Typ. nov. | Totals |
|----------|---------------------------------------------|-------------------|--------------|--------------------------------------------------|-------------------------|------------|-----------|-----------|--------|
| 1        | Agenus primus (Egena)                       | 9 → 6             | 12 → 5       | Egena secunda → Agenus secundus                  | 5 (3)                   | 1          | 1         | 1         | 1      |
| 2        | Agenus primus (Egena)                       | 9 → 7             | 15 → 5       | Igena prima → Agenus primus                     | 4 (2)                   | 1 (1)      | 1         | 1         | 1      |
| 3        | Agenus primus (Ogenus)                      | 9 → 8             | 18 → 5       | Ogenus secundus → Agenus secundus               | 5 (0)                   | 1          | 1         | 1         | 1      |
| 4        | Agenus tertius (Egena)                      | 10 → 6            | –            | Agenus tertius → Egena tertia                   | 1 (1)                   | 1          | 1         | 1         | 1      |
| 5        | Agenus tertius (Igena)                      | 10 → 7            | –            | Agenus tertius → Igena tertia                   | 1 (1)                   | 1          | 1         | 1         | 1      |
| 6        | Agenus tertius (Ogenus)                     | 10 → 8            | –            | Agenus tertius → Ogenus tertius                 | 1 (0)                   | 1          | 1         | 1         | 1      |
| 7        | Agenus quartus (Egena)                      | 11 → 6            | –            | Agenus quartus → Egena quartia                  | 1 (1)                   | 1          | 1         | 1         | 1      |
| 8        | Agenus quartus (Igena)                      | 11 → 7            | –            | Agenus quartus → Igena quartia                  | 1 (1)                   | 1          | 1         | 1         | 1      |
| 9        | Agenus quartus (Ogenus)                     | 11 → 8            | –            | Agenus quartus → Ogenus quartus                 | 1 (0)                   | 1          | 1         | 1         | 1      |
| 10       | Egena secunda (Agenus)                      | 12 → 5            | –            | Egena secunda → Agenus secundus                 | 3 (1)                   | 1          | 1         | 1         | 1      |
| 11       | Egena secunda (Igena)                       | 12 → 7            | 15 → 6       | Igena prima → Egena prima                       | 5 (0)                   | 1          | 1         | 1         | 1      |
| 12       | Egena secunda (Ogenus)                      | 12 → 8            | 18 → 6       | Ogenus secundus → Egena secundula               | 4 (2)                   | 1 (1)      | 1         | 1         | 1      |
| 13       | Egena quinta (Agenus)                       | 13 → 5            | –            | Egena quinta → Agenus quinta                    | 1 (1)                   | 1          | 1         | 1         | 1      |
| 14       | Egena quinta (Igena)                        | 13 → 7            | –            | Egena quinta → Igena quinta                     | 1 (0)                   | 1          | 1         | 1         | 1      |
| 15       | Egena quinta (Ogenus)                       | 13 → 8            | –            | Egena quinta → Ogenus quinta                    | 1 (1)                   | 1          | 1         | 1         | 1      |
| 16       | Egena sexta (Agenus)                        | 14 → 5            | –            | Egena sexta → Agenus sexta                     | 1 (1)                   | 1          | 1         | 1         | 1      |
| 17       | Egena sexta (Igena)                         | 14 → 7            | –            | Egena sexta → Igena sexta                      | 1 (0)                   | 1          | 1         | 1         | 1      |
| 18       | Egena sexta (Ogenus)                        | 14 → 8            | –            | Egena sexta → Ogenus sexta                     | 1 (1)                   | 1          | 1         | 1         | 1      |
| 19       | Igena prima (Agenus)                        | 15 → 5            | –            | Igena prima → Agenus primus                    | 2 (1)                   | 1 (1)      | 1         | 1         | 1      |
| 20       | Igena prima (Igena)                         | 15 → 6            | –            | Igena prima → Egena prima                       | 3 (0)                   | 1          | 1         | 1         | 1      |
| 21       | Igena prima (Ogenus)                        | 15 → 8            | 18 → 7       | Ogenus secundus → Igena secunda                 | 5 (3)                   | 1          | 1         | 1         | 1      |
| 22       | Igena septima (Agenus)                      | 16 → 5            | –            | Igena septima → Agenus septima                  | 1 (1)                   | 1          | 1         | 1         | 1      |
| 23       | Igena septima (Igena)                       | 16 → 6            | –            | Igena septima → Igena septima                   | 1 (0)                   | 1          | 1         | 1         | 1      |
| 24       | Igena septima (Ogenus)                      | 16 → 8            | –            | Igena septima → Ogenus septima                  | 1 (1)                   | 1          | 1         | 1         | 1      |
| 25       | Igena octava (Agenus)                       | 17 → 5            | –            | Igena octava → Agenus octava                    | 1 (1)                   | 1          | 1         | 1         | 1      |
| 26       | Igena octava (Igena)                        | 17 → 6            | –            | Igena octava → Igena octava                     | 1 (0)                   | 1          | 1         | 1         | 1      |
| 27       | Igena octava (Ogenus)                       | 17 → 8            | –            | Igena octava → Ogenus octava                    | 1 (0)                   | 1          | 1         | 1         | 1      |
| 28       | Ogenus secundus (Agenus)                    | 18 → 5            | –            | Ogenus secundus → Agenus secundus               | 3 (0)                   | 1          | 1         | 1         | 1      |
| 29       | Ogenus secundus (Igena)                     | 18 → 6            | –            | Ogenus secundus → Egena secundula               | 2 (1)                   | 1 (1)      | 1         | 1         | 1      |
| 30       | Ogenus secundus (Ogenus)                    | 18 → 7            | –            | Ogenus secundus → Igena secunda                 | 3 (1)                   | 1          | 1         | 1         | 1      |
| 31       | Ogenus nonus (Agenus)                       | 19 → 5            | –            | Ogenus nonus → Agenus nonus                     | 1 (0)                   | 1          | 1         | 1         | 1      |
| 32       | Ogenus nonus (Igena)                        | 19 → 6            | –            | Ogenus nonus → Igena nona                       | 1 (1)                   | 1          | 1         | 1         | 1      |
| 33       | Ogenus nonus (Ogenus)                       | 19 → 7            | –            | Ogenus nonus → Ogenus nonus                     | 1 (1)                   | 1          | 1         | 1         | 1      |
| 34       | Ogenus decimus (Agenus)                     | 20 → 5            | –            | Ogenus decimus → Agenus decimus                 | 1 (0)                   | 1          | 1         | 1         | 1      |
| 35       | Ogenus decimus (Igena)                      | 20 → 6            | –            | Ogenus decimus → Igena decima                   | 1 (1)                   | 1          | 1         | 1         | 1      |
| 36       | Ogenus decimus (Igenus)                     | 20 → 7            | –            | Ogenus decimus → Igenus decima                  | 1 (1)                   | 1          | 1         | 1         | 1      |
| Totals   |                                            |                   |              |                                                  |                         | 68 (30)    | 4 (4)     | 12        | 4       | 12      | 4       | 154     |

Comb. nov., new combination (species level); Gen. coord., application of Gender Agreement changes gender of species epithet; nomen nov., new name (resolved by rank); syn. nov., new synonymy (resolved by rank); and typ. nov., new typification (resolved by rank). Bold font used to highlight numbers of emendations, excepting the applications of Gender Agreement.
on the number of cases where application of Gender Agreement requires changing the ending of an epithet from masculine to feminine, or vice versa.

While the present analysis is merely an exploratory step, the compiled numbers provide some general insights into the logical interdependencies of nomenclatural starting and ending conditions in light of taxonomic reassessment and change. As we discuss below, a better understanding of these dependencies may eventually lead to logic-enabled optimizations of nomenclatural and taxonomic change actions in biodiversity data environments.

**Results**

**Documentation of the ASP approach**

The general structure of our ASP programming approach is provided in Table 1. All sections are necessary to run the code with clingo and obtain the desired outcomes. However, because the ASP solver evaluates all constraints simultaneously, the sequence of the sections and subsections is relevant only to enhancing human readability of the code. Accordingly, Section 1 defines the overarching transition system domains, variables for subsequent if-then constraints and starting conditions; this includes two steps, 20 nodes, all node-associated ranks, node-associated names, Priority-constrained name lineages, mono- and binomial names, and name-associated years of publication. An important feature allowing further elaboration of this code is the semi-independent modelling of (1) name identities and relationships (“nomenclature”), and (2) node identities and relationships (“taxonomy”); our use of the term “semi-independent” corresponds to the complex cardinality interactions that occur in nomenclatural/taxonomic change scenarios (e.g. Vane-Wright, 2003; Franz, 2005; Remsen, 2016).

Sections 2–5 of the code establish several globally applicable conditions, i.e. conditions that must hold true for any step in the transition system. These include: assignment of a single valid name to each node, with rank, year and parent (except the root node); inertia conditions—all ending conditions are equal to the starting conditions unless the change triggered by the species-level node move overwrites this constraint; Priority among children nodes that have the same parent node; and persistence of taxonomic ranks throughout the transition.

Sections 6–8 represent the key rules needed to compliantly model the taxonomic change and nomenclatural emendations to be enacted at \( t = 1 \). The core change itself is specified in Section 8.1., by simply requiring one species-level node to not have the same parent node at \( t = 1 \) which it had at \( t = 0 \). This constraint in effect creates one new species-to-genus node edge at \( t = 1 \), while keeping the total number of nodes consistent. Sections 6 and 7 establish which existing and new names are held as valid at \( t = 1 \). These rules specify how the Priority (and hence validity) of a genus-level name is determined by the most senior epithet associated with that generic name at time = 0. They furthermore allow epithets to “newly become the most senior” for the respective parent genus name at \( t = 1 \), if the taxonomic change triggers such a “transition” in one of the more complex scenarios (see above). However, this latter constraint—and others logically tied to it—require an important exception (Section 6.4.): if the mandated node move violates the relative or global Priority status of generic names and specific epithets as established at \( t = 0 \), then an “inverse move” is executed (e.g. Fig. 3). This means that, in spite of the reassignment of a child node to a new parent node, the globally Priority-carrying binomial—e.g. \( A. \) primus, 1775, in all Fig. 3 scenarios—is maintained as valid. Moreover, the other locally Priority-carrying binomial will “move into \( Agenus \)”, even though neither its type specimen nor those of other congeners recognized at time = 0 were reinterpreted to exhibit another shape. Hence we say “inverse”, as in: initially intended to affect the status of another node/ name association. The remainders of Sections 6 and 7 constrain (e.g.) that Coordination propagates from the lowest to the highest represented taxonomic rank, and that any new genus-level name(s) requiring creation at \( t = 1 \) should start with the letters Nov- prefixing the previously valid name. For instance (Fig. 5), Novigena, 2000, is used as a new genus name for valid epithets (septima, octava) previously associated \( (t = 0) \) with the now synonymized \( (t = 1) \) genus name Igena. Lastly, if the move of a node and its associated epithet is bound to create two (binomial) homonyms, then this Code violation will be prohibited by adding the endings -ulus (e.g. prim-ulus instead of prim-us) or -ula (e.g. instead of second-a) to the epithet that will no longer have local Priority at \( t = 1 \) (see, e.g. Figs 4c and 5a). These cases are not recorded as new combinations, but instead are replacement names, and are recognized as new species-level names in Table 2.

Sections 9 and 10 respectively identify all resulting nomenclatural emendations and generate the corresponding output topologies, valid nodes names and nomenclatural status and relationship updates. The code of Section 9 includes if-then constraints that identify among the ending conditions \( (t = 1) \): new combinations and concomitant applications of Gender Agreement leading to new epithet endings; new synonyms at the generic and tribal levels; new species-, genus- and tribe-level names; and new typifications of higher-level names by their entailed, newly senior lower-level names. Added are commands that
aggregate totals for each of these emendations and for the corresponding stable model. The final Section 10 provides code to output each possible node topology as a set of “revised edges”, generating the valid node names for each rank, as well as the list of nomenclatural emendations. If an inverse move is indicated in the model, this information is also printed out (see Data S5).

Analysis of nomenclatural transition scenarios

The ASP programming approach correctly generates all 36 transition scenarios (stable models) and the required nomenclatural changes (Figs 1c, 3–6; Table 2). Given the use case constraints, exactly six of the 36 scenarios require an inverse move: three involving the node/binomial *A. primus*, 1775 (Figs 3a–c); two concerning the node/binomial *E. secunda*, 1780 (Figs 4b and 4c); and one including the node/binomial *I. prima*, 1785 (Fig. 5c). The node/binomial *O. secunda*, 1790, does not have Priority over the three aforementioned names, and thus cannot trigger an inverse move (Fig. 6).

The 12 complex scenarios involving reinterpretations of specimens designated during the type period (1775–1790) account for 114/154 (74%) of the recorded
The nomenclatural particularities of the 12 complex scenarios are well captured in Figs 3–6 and require little additional explanation. Each scenario requires the creation and typification of a new generic name at t = 1, as well as a genus-level synonymy (Table 2). In four cases, the node transfer affects an epithet associated with a source genus-level name (either I. prima, 1785, or O. secundus, 1790) at t = 0 that is homonymous with that of another epithet associated with the target genus-level name (A. primus, 1775, and E. secundus, 1780, respectively). These cases require the creation of new species-level replacement names in accordance with Priority, resulting in the new epithets primulus, 2000 (Figs 3b and 5a) or secundula, 2000 (Figs 4c and 6b). In two of these scenarios (Figs 3b and 5a) and two others (Figs 4b and 5b), Igena, 1785, is synonymized at t = 1 with another generic name in the Agenini, 1775 (either Agenus or Egena). Consequently, and in compliance with Coordination, the tribal name Igenini, 1775, is also synonymized with

![Diagram of change scenarios](image-url)
Nomenclatural changes

Igena → Igenini, new synonymy (tribe)
Igena → Igenus, new synonymy (genus)
Igena prima → Agenus primus, new species name1
Novigena, new genus name
Igena septima → Novigena septima, new comb.
Igena octava → Novigena octava, new comb.
Novigena septima → Novigena, new typ. (genus)
Ogenini, new tribe name
Ogenus → Ogenini, new typ. (tribe)

Complementary representation approach—taxonomic concept alignments

Elsewhere we have applied ASP to promote the taxonomic concept approach (Franz et al., 2015, 2016a, b). This approach represents each published taxonomy (t = 0, 1) separately, via taxonomic concept labels (name sec. author; where sec. means “according to”) and parent–child (is-a) relations, and then employs expert-provided Region Connection Calculus (RCC–5) articulations to express taxonomic congruence or non-congruence (inclusion, overlap, exclusion) between regions pertaining to multiple taxonomies. The RCC–5 articulations differ from names and nomenclatural relationships (homonymy, synonymy) in that “equivalence” is assessed in relation to the entire perceived referential extensions of taxonomic concept labels, with less emphasis on nomenclatural type identity (Dubois, 2005; Franz, 2005; Franz and Peet, 2009). An ASP reasoning toolkit (Chen et al., 2014) is available to (1) check for the logical consistency of the input constraints under certain taxonomic covering assumptions, (2) infer additional implied articulations and (3)
visualize consistent multi-taxonomy alignments. The last of these may be viewed as taxonomic meaning resolution maps that can guide information integration across the varying taxonomies. In this sense, the alignments can complement representations of “identity in light of change” as provided by Linnaean names and nomenclatural relationships. Simple RCC–5 metrics—namely “how many articulations of each type appear in an alignment?”—are also suitable for assessing the performance of taxonomic names as identifiers of congruent or non-congruent taxonomic meanings (Franz et al., 2008, 2016a,b).

Because the transition scenarios of this use case are strongly and heterogeneously affected by the application of Priority, Coordination, Homonymy, Typification and Gender Agreement—all key principles in the Linnaean tradition—it is illustrative to model the starting and ending conditions as RCC–5 alignments of two taxonomic concept hierarchies. The complete set of 36 alignments, each with relevant input and output data files, is provided in the Data S6–S8.

Four exemplary alignments are shown in Figs 7 and 8, corresponding to scenarios 1, 2, 4 and 5. Table 3 summarizes the alignment patterns obtained for the 36 scenarios. Interestingly, there are only three general patterns that can be categorized as follows: either the species-level node is transferred among genus-level nodes under two tribe-level nodes at $t = 0$ (Agenini and Igenini, respectively), and in that case it is (1) an intended move or (2) an inverse move; or (3)

---

**Fig. 6.** Representation of change scenarios 28–30, involving the reinterpretation of the type Ogenus secundus, 1790 [18], as recognized at time $t = 0$ (see also Fig. 1b). All other conventions are as in Fig. 3. See also Table 2. (a) Scenario 28: the type of O. secundus is square. (b) Scenario 29: the type of O. secundus is rectangular. (c) Scenario 30: the type of O. secundus is semi-elliptical. [Colour figure can be viewed at wileyonlinelibrary.com].
the transfer involves genus-level nodes placed within the same tribe-level node at $t = 0$ (Agenini or Igenini). Pattern (1) is exemplified by scenario 5 and shown in Fig. 7a (compare with Fig. 3a). This pattern holds for 20 scenarios, and in each case corresponds to alignments that have 15 congruent regions—three above the species level—and six overlapping articulations. Pattern (2) is true for scenario 2 and shown in Fig. 7b (compare with Fig. 3b). This pattern is present in four scenarios, where the alignments have 15 congruent regions and four overlapping articulations. Lastly, pattern (3) is represented by scenarios 4 and 1 and displayed in Fig. 8a and b (compare the latter with Fig. 3a). The pattern applies to 12 scenarios, each with an alignment that has 17 congruent regions—five above the species level—and only one overlapping articulation. Two of the 12 scenarios (scenarios 1 and 21) that constitute pattern (3) are inverse move scenarios; however, this has no effect on the alignment pattern (compare Fig. 8a and b).

Comparison of the RCC–5 alignment representations and the nomenclatural change scenarios shows two rather different approaches to modelling identity and change relationships for this use case (Table 3, last column). In particular, patterns (1) and (2) are by some measure the most taxonomically impacting: the reinterpretation of a type specimen triggers that a species-level entity is transferred from one tribe-level entity to another (as conceived at $t = 0$). This move consistently translates into six or four overlapping articulations among the affected genus- and tribe-level concepts, depending on whether it is carried out in the intended or inverse direction. However, often scenarios that fall under the same general pattern (among the three) yield rather dissimilar nomenclatural outcomes (compare Fig. 7a and b). Pattern (1) is variously associated with 1–11 nomenclatural changes (average: $2.9 \pm 3.0$), whereas pattern (2) involves 8–14 nomenclatural changes (average: $11.0 \pm 2.4$). In addition to being less homogenous, the direction of change between patterns (1) and (2) is somewhat opposed if we compare the nomenclatural representations (pattern 1: less change $\rightarrow$ pattern 2: more change, i.e. more nomenclatural differences) to the RCC–5 alignments (pattern 1: more change $\rightarrow$ pattern 2: less change, i.e. fewer instances of overlap). The greater average
The number of nomenclatural changes under pattern (2) is caused by global Priority constraints requiring inverse moves. Pattern (3) is taxonomically less impacting in comparison, because the reinterpretation of the type and associated species-level node transfer are confined to one tribe-level entity (as recognized at \( t = 0 \)). Nevertheless, this relatively minor reclassification translates into a range of 2–11 nomenclatural changes (average: 4.3 \pm 3.7), with either two or seven changes in transfer scenarios in the intended direction, and 11 changes in transfer scenarios in the inverse direction (compare Fig. 8a and b). This suggests that certain nomenclatural principles are not designed to track taxonomic change proportionally—a theme that we revisit in the Discussion.

Discussion

Compatibility of nomenclature and computational logic

We have shown, probably to an unprecedented degree of complexity, that many overarching principles of zoological nomenclature can be modelled with computational logic. Such logic can then be applied to sufficiently well-specified taxonomic change scenarios to yield nomenclaturally diverse but accurate reasoning outcomes. While our approach and encoded program are not immediately designed to resolve real-life reclassification challenges, they demonstrate that ASP representation and reasoning powers are fully adequate to do so. In short, the core principles of zoological nomenclature are logically tractable.

We reach this conclusion not because we have covered every principle, application and exception represented in the 90 articles of the ICZN—International Commission on Zoological Nomenclature (1999). For instance, our use case does not account for the principle of the First Reviser (Article 24) or for the use of the Commission’s Plenary Power (Article 81) to preserve or suppress names or other nomenclatural acts under certain conditions, and typically with the intent to conserve “prevailing usage” (Vences et al., 2013). We have also not addressed issues of uncertainty in taxonomic judgment (Confalonieri et al., 2012). Moreover, as several reviewers have pointed out, showing logic compatibility at the level of core nomenclatural principles is not the same as “automating” logic solutions for real-life use cases. To clarify, the point of our logic representations is neither to change existing nomenclatural rules, nor to somehow replace with computational logic the critical human judgments needed in their real-life application. Indeed, a necessary precondition for executing our use case is that taxonomic judgments are sufficiently well-specified upfront to then permit the logic reasoning. This circumstance has an obvious corollary: if particular nomenclatural rules are logically under-specified such that their execution requires additional human judgment, or if a particular use case potentially applies to
multiple reciprocally incompatible rules—and hence (again) human choice is needed to yield an action—then logic alone cannot resolve these cases. In the language of ASP, however, requirements for human judgments to resolve under-specified rules are simply additional constraints, each with specific trigger conditions. Logic cannot replace taxonomic judgment, and is also not suited to favour one valid nomenclatural solution pathway over another similarly valid pathway. But, logic can make valid proposals, or forbid invalid nomenclatural pathways once the taxonomic constraints are well specified. Thus an appropriate way to interface the Code and logic is to recognize where “fuzziness” in nomenclatural rules is particularly prone to yield too many answer sets, and to demand further constraints to be provided by the user in these cases. As needed, each novel constraint can be added to the elaboration-tolerant code to model increasingly complex situations. Other aspects, such as the size of the input taxonomies, can be augmented. Additional variables such as author names or page numbers can be included to resolve Homonymy or Priority disputes more finely. Constraints can be applied to specific taxonomic ranks only [“groups” in ICZN—International Commission on Zoological Nomenclature (1999) terminology]. More input classifications and more steps can be accommodated.

In summary, we see no elements in the ICZN—International Commission on Zoological Nomenclature (1999) that are principally incompatible with ASP representation and reasoning to produce valid stable models for complex nomenclatural/taxonomic transition scenarios. Once the appropriate rules, exceptions and points of ambiguity needing additional input are

| Scenario | Type being rediagnosed [species-level node] | Generic names involved (direction of move) | Across tribes | Inverse move | Overlapping articulations | Alignment pattern | Nomenclatural emendations |
|----------|---------------------------------------------|--------------------------------------------|--------------|--------------|----------------------------|-------------------|---------------------------|
| 5        | Agenus tertius [10]                         | Agenus → Igena                             | +            | −            | 6                          | 7A                | 2                         |
| 8        | Agenus quartus [11]                         | Agenus → Igena                             | +            | −            | 6                          | 7A                | 2                         |
| 19       | Agenus prima [15]                           | Agenus → Igena                             | +            | −            | 6                          | 7A                | 2                         |
| 22       | Igena septima [16]                          | Agenus → Igena                             | +            | −            | 6                          | 7A                | 2                         |
| 25       | Igena octava [17]                           | Agenus → Igena                             | +            | −            | 6                          | 7A                | 2                         |
| 9        | Agenus quartus [11]                         | Agenus → Ogenus                            | +            | −            | 6                          | 7A                | 1                         |
| 28       | Ogenus secundus [18]                        | Agenus → Ogenus                            | +            | −            | 6                          | 7A                | 6                         |
| 31       | Ogenus nonus [19]                           | Agenus → Ogenus                            | +            | −            | 6                          | 7A                | 1                         |
| 34       | Ogenus decimus [20]                         | Agenus → Ogenus                            | +            | −            | 6                          | 7A                | 1                         |
| 14       | Igena quinta [13]                           | Igena → Igena                              | +            | −            | 6                          | 7A                | 1                         |
| 17       | Igena sexta [14]                            | Igena → Igena                              | +            | −            | 6                          | 7A                | 1                         |
| 20       | Igena prima [15]                            | Igena → Igena                              | +            | −            | 6                          | 7A                | 9                         |
| 23       | Igena septima [16]                          | Igena → Igena                              | +            | −            | 6                          | 7A                | 1                         |
| 26       | Igena octava [17]                           | Igena → Igena                              | +            | −            | 6                          | 7A                | 1                         |
| 15       | Igena quinta [13]                           | Igena → Igena                              | +            | −            | 6                          | 7A                | 2                         |
| 18       | Igena sexta [14]                            | Igena → Ogenus                            | +            | −            | 6                          | 7A                | 2                         |
| 29       | Ogenus secundus [18]                         | Ogenus → Ogenus                            | +            | −            | 6                          | 7A                | 8                         |
| 32       | Ogenus nonus [19]                           | Ogenus → Ogenus                            | +            | −            | 6                          | 7A                | 2                         |
| 35       | Ogenus decimus [20]                         | Ogenus → Ogenus                            | +            | −            | 6                          | 7A                | 2                         |
| 2        | Agenus primus [9]                           | Agenus → Igena                             | +            | +            | 4                          | 7B                | 14                        |
| 3        | Agenus primus [9]                           | Agenus → Ogenus                            | +            | +            | 4                          | 7B                | 8                         |
| 11       | Igena secunda [12]                          | Igena → Igena                              | +            | +            | 4                          | 7B                | 11                        |
| 12       | Igena secunda [12]                          | Igena → Ogenus                            | +            | +            | 4                          | 7B                | 11                        |
| 4        | Agenus tertius [10]                         | Agenus → Igena                             | −            | +            | 1                          | 8A                | 2                         |
| 7        | Agenus quartus [11]                         | Agenus → Igena                             | −            | +            | 1                          | 8A                | 2                         |
| 10       | Igena secunda [12]                          | Igena → Igena                              | −            | +            | 1                          | 8A                | 7                         |
| 13       | Igena quinta [13]                           | Igena → Igena                              | −            | +            | 1                          | 8A                | 2                         |
| 16       | Igena sexta [14]                            | Igena → Igena                              | −            | +            | 1                          | 8A                | 2                         |
| 24       | Igena septima [16]                          | Igena → Ogenus                            | −            | +            | 1                          | 8A                | 2                         |
| 27       | Igena octava [17]                           | Igena → Ogenusz                           | −            | +            | 1                          | 8A                | 2                         |
| 30       | Ogenus secundus [18]                         | Igena → Ogenus                            | −            | +            | 1                          | 8A                | 7                         |
| 33       | Ogenus nonus [19]                           | Igenus → Ogenus                            | −            | +            | 1                          | 8A                | 2                         |
| 36       | Ogenus decimus [20]                         | Igenus → Ogenus                            | −            | +            | 1                          | 8A                | 2                         |
| 1        | Agenus primus [9]                           | Agenus → Igena                             | −            | +            | 1                          | 8B                | 11                        |
| 21       | Igena prima [15]                            | Igena → Ogenus                            | −            | +            | 1                          | 8B                | 11                        |

Bold font used in right-most column to highlight scenarios requiring the highest numbers of nomenclatural emendations.
encoded, the burden of producing viable stable models will depend largely on the systematists’ ability to specify sufficiently precise starting conditions upon which these logic constraints can act.

Feasibility and desirability of logic-enabled nomenclatural services

Demonstrating the ability to model nomenclatural change in logic is not the same as showing that such an approach is worth the effort. Hence we should ask: under what circumstances might the benefits of applying ASP logic outweigh the costs represented by encoding the rules, starting conditions and then selecting suitable answer sets? The best—but nevertheless tentative—answer we have points to globally scalable nomenclatural registry and validation services. The answer is grounded in future expectations of abundant and disruptive nomenclatural changes, which present an opportunity to create logic-enabled solutions that can work at increasing information scales.

Virtual environments such as ZooBank (Pyle and Michel, 2008; Penev et al., 2016; Pyle, 2016) aspire to act as globally coordinated repositories for taxonomic and nomenclatural acts. For instance, since 2012 registration with ZooBank is mandatory for electronic publications to acquire availability under ICZN rules (Krell and Pape, 2015). To render the above approach feasible, a suitable interface between ZooBank’s underlying data schema and a logic reasoning service is needed. Another necessary precondition is to populate ZooBank with sufficient legacy information to assess Priority and other nomenclatural conditions globally for each use case. We recognize that these are not simple challenges.

We see three kinds of benefits. In the case of publishing complex sets of nomenclatural acts—such as those under consideration for the fruit fly name *Drosophila* where approximately 1500 valid names are in play (van der Linde and Houle, 2008; O’Grady and Markow, 2009; ICZN—International Commission on Zoological Nomenclature, 2010)—ASP-encoded logic could be used to explore alternative valid change scenarios. Given multiple nomenclaturally viable options, the application logic could be further optimized to identify solutions that maximize stability under certain prioritized criteria (benefit no. 1). Such criteria might include: (1) minimize the total number of nomenclatural changes, or (2) minimize the number of changes at the generic level, or (3) maximize stability for the most frequently used binomials, etc. A probably less strict set of rules could nevertheless be applied for higher-level names where the Codes have no jurisdiction (Dubois, 2015), in order to maximize naming stability in light of taxonomic change at these *supra*-familiar levels.

In addition, ASP logic could be used as a means of verifying compliance of proposed emendations with nomenclatural rules, by blocking new submissions to the registry that are not consistent with any stable model (benefit no. 2). Following the interconnected rules of the ICZN—International Commission on Zoological Nomenclature (1999) is not always trivial, even for systematist users, and hence an automated service that checks for logic errors in human-proposed nomenclatural emendations could increase user confidence in enacting necessary changes and reduce rule-violating errors.

We realize that the suggested applications of ASP logic to nomenclatural registries require further development—conceptual, technical and social—to possibly lead to real-life implementations. This study is limited to illustrating the general compatibility of zoological nomenclature and computational logic. Yet we may also remind ourselves that the enterprise of discovering and naming nature’s diversity is far from finished. While estimates of global species-level richness remain uncertain, most indicate that many millions of such entities remain unnamed (Caley et al., 2014; Stork et al., 2015; Locey and Lennon, 2016). Future additions, synonymizations and rearrangements of entities at varying taxonomic levels may have minimal or rather dramatic nomenclatural consequences. Computational solutions would tend to manage nomenclatural entities, relationships and changes more robustly at large scales than human users (benefit no. 3). We hope that the projected scale of this challenge will motivate further work aimed at developing logic-enabled solutions, based on this foundation-laying study.

Computational logic as an assessment tool for nomenclatural constraints

Modelling nomenclatural change scenarios in ASP logic may also give us a new perspective on the nature of nomenclature itself. In particular, our contrived use case illustrates how Priority and Coordination constraints may create dissimilar nomenclatural outcomes for what could be regarded as taxonomically identical, or at least highly comparative, triggers of change. From a strictly taxonomic perspective, the reinterpretation of the type specimens of *A. primus* and *A. tertius* having a hemi-elliptical shape instead of a square shape (as originally assessed) may be viewed as “the same amount of taxonomic change”. In each case, it would appear in retrospect that a single, species-level entity had been misdiagnosed and hence taxonomically misplaced (and misnamed as well). However, Priority makes these two scenarios highly unequal, considering that *A. primus* was coined in 1775 and *A. tertius* was coined in 1950 (Fig. 1a). Consequently, the
reinterpretation of the older type “costs” 14 nomenclatural emendations (Figs 3b and 7a), whereas that of the younger type “costs” two such changes (Figs 1c and 7a).

Generally speaking, it appears that zoological nomenclature is designed so as to make changes to the taxonomic identities of names with older ages more costly, whereas those of younger names are less costly. Such a design increases stability in naming taxonomic concepts that nevertheless may have various non-congruent circumscriptions. Thus, older names tend to persist in part by virtue of their greater age, but this is not necessarily the same as optimizing change ratios for the interaction of nomenclature and taxonomy. For that to be the case, one might instead design a naming system whose behaviour in light of taxonomic change is less driven by Priority—which along with Coordination can produce highly unequal cascading effects—and more receptive to the “relative amount” of taxonomic change to be enacted.

The RCC–5 alignments shown in Figs 7 and 8 (see also Table 3) are aiming in this direction. They mitigate the more dramatic nomenclatural disparities up to a point, while still also modelling valid nomenclatural conditions for the starting and ending conditions. In particular, one might argue that type specimen reinterpretations that motivate taxonomic rearrangements of species-level entities spanning across multiple tribes (Fig. 7) should be consistently more costly than within-tribe moves (Fig. 8). After all, alignments for the former scenarios only generate 15 congruent Euler regions (“less congruence → more renaming”), whereas the latter generate 17 congruent Euler regions (“more congruence → less renaming”). However, such considerations of the relative taxonomic quantity of change are not primary design features of the ICZN—International Commission on Zoological Nomenclature (1999) principles and rules.

The history of Codes is deep and multi-faceted (Winsor, 2001; Minelli, 2003; Schuh, 2003; Dayrat, 2010; Dubois, 2011; David et al., 2012). For instance, Witteveen (2016) connects the origins or the modern type concept in nomenclature to the struggle between the “metropolitan establishment” and “provincial radicals” in the colonial 19th century United Kingdom. This struggle to avert the “chaos of synonymy” is ultimately grounded in human cognitive constraints. Atran (1998) argues that the Linnaean naming tradition retains important features of folk biology, and in this sense is well aligned with what he calls cognitive universals. The latter include evolutionarily constrained notions of taxonomic rank, generic species and rank-associated essences. Both folk biology, which is functional in localized ecological contexts, and Linnaean classification, which is global in scope, are designed to satisfy our human cognitive preferences to maximize inductive learning and reasoning potential for the given context (see also Platnick, 2009).

Our logic explorations show that the principles of nomenclature are not designed in the main to respond proportionally to complex nomenclatural and taxonomic change scenarios. The principles may perform best under the cognitively grounded premise that old and long-established names experience relatively minimal taxonomic reassignments at late stages in the history of human taxonomic making. If on the other hand the taxonomic identities of these early-period names are frequently redefined “late in the game”, then this may result in disproportionately many nomenclatural emendations. Another way of saying this is that the Linnaean system may be biased to favour early success in achieving natural, inductively projectible classifications. Such early success is certainly possible if perceived taxonomic groups are not very diverse and are otherwise favourably accessible to human cognition (considering, for example, macroscopic size, phenotypic heterogeneity, evolutionary rates and signal preservation). However, other complex groups may well experience abundant and significant late-stage reassessments—a circumstance that in some sense runs counter to both our cognitive biases and those (logically) embedded in the Codes of nomenclature. Many naming controversies in systematics reflect the tension of apparent early- versus late-stage success in recognizing the evolutionary identity of complex natural groups and relationships.

Applications of computational logic to nomenclatural/taxonomic change scenarios can inform future designs of identifier systems for systematics that find the best balance between human cognitive preferences and logic-informed representation and reasoning maxims (Sterner and Franz, 2017). We have shown that the ASP approach can provide both diagnostic and prescriptive input in this regard. It is not too late to pursue this path, and doing so may enable us to more fully bring nomenclature into the realm of computationally enabled big data science (Page, 2016).

Acknowledgments

We thank Stanley Blum, Neal Evenhuis, Thomas Pape, David Patterson, Richard Pyle and Francisco Welter-Schultes for their advice in conceiving the use case and in the application of nomenclatural rules. Bertram Ludäscher, Parisa Kianmajd and Shizuo Yu provided support for the Euler/X toolkit. We also thank the Associate Editor and two anonymous reviewers for constructive comments that have led to improvements of this paper. Support of the authors’ research through
the National Science Foundation is kindly acknowledged (N.M.F.: DEB-1155984, DBI-1342595; J.L.: IIS-1319794, IIS-1526301).

References

Atran, S., 1998. Folk biology and the anthropology of science: cognitive universals and cultural particulars. Behav. Brain Sci. 21, 547–569.

Baader, F., Horrocks, I., Sattler, U., 2008. Description logics. In: Baader, F., Horrocks, I., Sattler, U., (Eds.), Handbook of Knowledge Representation. Elsevier Academic Press, Amsterdam, pp. 135–179.

Brachman, R.J., Levesque, H.J., 2004. Knowledge representation and reasoning. Morgan Kaufmann Publishers, San Francisco, CA.

Brewka, G., Eiter, T., Truszczyński, M., 2011. Answer set programming at a glance. Commun. ACM 54, 92–103.

Brickmann-Voss, A., Calder, D.R., 2013. Zyzzyus rubusidaeus (Cnidaria, Hydrozoa, Tubulariidae), a new species of anthoathecate hydroid from the coast of British Columbia, Canada. Zootaxa 3666, 389–397.

Brooks, D.R., Erdem, E., Erdogan, S.T., Minett, J.W., Ringe, D., 2007. Inferring phylogenetic trees using Answer Set Programming. J. Autom. Reason. 39, 471–509.

Cappellini, E., Gentry, A., Palkopoulou, E., Ishida, Y., Cram, D., Deans, A.R., Lewis, S.E., Huala, E., Anzaldo, S.S., Ashburner, M., Dahdul, W., Das, S., Dececchi, T.A., Dettaia, A., Diogo, R., Druzinsky, R.E., Dumontier, M., Franz, N.M., Friedrich, F., Gkoutos, G.V., Haenel, M., Harmon, L.J., Hayamizu, T.F., He, Y., Hines, H.M., Ibrahim, N., Jackson, L.M., Jaiswal, P., James-Zorn, C., Köhler, S., Lecointre, G., Lapp, H., Lawrence, C.J., Le Novère, N., Lundberg, J.G., Macklin, J., Mast, A.R., Midford, P.E., Miko, I., Mungall, C.J., Oellrich, A., Osmi-Sutherland, D., Parkinson, H., Ramírez, M.J., Richter, S., Robinson, R.N., Ruttenberg, A., Schulz, K.S., Sgerdell, E., Seltmann, K.C., Sharkey, M.J., Smith, A.D., Smith, B., Specht, C.D., Squires, R.B., Thacker, R.W., Thessen, A.E., Fernández-Triana, J., Vihinen, M., Vize, P.D., Vogt, L., Wall, C.E., Walls, R.L., Westerfeld, M., Wharton, R.A., Winkler, C.S., Woolley, J.B., Yoder, M.J., Zorn, A.M., Mabee, P., 2015. Finding our way through phenotypes. PLoS Biol. 13, e1002033.

Dixon, J.R., Simmons, M.K., 1983. Computers that design: expert systems for mechanical engineers. Comp. Mech. Eng. 2, 10–18.

Dmitriev, D.A., Yoder, M., 2017. NOMEN—a nomenclatural ontology for biological names (not concepts). Available at https://github.com/SpeciesFileGroup/nomen. Accessed 25 January 2017.

Donnellan, K.S., 1966. Reference and definite descriptions. Phil. Rev. 77, 281–304.

Dudoit, A., 2005. Proposed Rules for the incorporation of nominate of higher-ranked zoological taxon in the International Code of Zoological Nomenclature. 1. Some general questions, concepts and terms of biological nomenclature. Zoosystema 27, 365–426.

Dufour, A., 2011. The International Code of Zoological Nomenclature must be drastically improved before it is too late. Bionomia 2, 1–104.

Dudoit, A., 2015. The Duplotsensional Nomenclatural System for higher zoological nomenclature. Dumerilia 5, 1–108.

Eiter, T., 2008. Answer Set Programming in a nutshell. Available at http://gradlog.informatik.uni-freiburg.de/gradlog/slides_ak/eiter_a.pdf. Accessed 28 January 2017.

Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tampits, H., 2008. Combining answer set programming with description logics for the Semantic Web. Artif. Intell. 172, 1495–1539.

Eiter, T., Ianni, G., Krennwallner, T., 2009. Answer Set Programming: a primer. In: Tessaris, S., Franconi, E., Eiter, T., Gutierrez, S., Handschuh, S., Rouset, M.-C., Schmidt, R.A., (Eds.), Reasoning Web. Semantic Technologies for Information Systems: 5th International Summer School 2009, 5th International Summer School 2009, Brixen-Bressanone, Italy, August 30—September 4, 2009, Tutorial Lectures. Lect. Notes Comput. Sc. 5689, pp.40–110.

Ereshefsky, M., 2007. Foundational issues concerning taxa and biological nomenclature terms for facilitating communication in biological nomenclature. Zoosystema 27, 365–426.

Ferraris, P., Lee, J., Lifschitz, V., 2011. Circumscription. Artif. Intell. 175, 236–263.

Franz, N.M., 2005. On the lack of good scientific reasons for the taxonomic age. Available at https://dx.doi.org/10.6084/m9.figshare.1401984. Accessed 25 January 2017.

Franz, N.M., Peet, R.K., 2009. Perspectives: towards a language for taxonomic concepts in support of biodiversity research and conservation. C19 BioScience 59, 232–235.

Franz, N.M., Peet, R.K., Weakley, A.S., 2008. On the use of logical model for taxonomic concepts for expanding knowledge of higher-ranked zoological taxa in the International Code of Zoological Nomenclature. 1. Some general questions, concepts and terms of biological nomenclature. Zoosystema 27, 365–426.

Franz, N.M., 2005. On the lack of good scientific reasons for the growing phylogeny/classification gap. Cladistics 21, 495–505.

Franz, N.M., Peet, R.K., 2009. Perspectives: towards a language for mapping relationships among taxonomic concepts. Syst. Biodiv. 7, 5–20.

Franz, N.M., Thau, D., 2010. Biological taxonomy and ontology development: scope and limitations. Biodiv. Inform. 7, 45–66.

Franz, N.M., Peet, R.K., Weakley, A.S., 2008. On the use of taxonomic concepts in support of biodiversity research and taxonomy. In: Wheeler, Q.D., (Ed.), The New Taxonomy, Systematics Association Special Volume Series 74. Taylor & Francis, Boca Raton, FL, pp. 63–86.

Franz, N.M., Chen, M., Yu, S., Kianmajd, P., Bowers, S., Ludäscher, B., 2015. Reasoning over taxonomic change: exploring alignments for the Perelleschus use case. PLoS ONE 10 (2), e0118247.

Franz, N.M., Chen, M., Kianmajd, P., Yu, S., Bowers, S., Weakley, A.S., Ludäscher, B., 2016a. Names are not good enough, reasoning over taxonomic change in the Andropogon complex. Semantic Web 7, 645–667.

Franz, N.M., Pier, N.M., Reeder, D.M., Chen, M., Yu, S., Kianmajd, P., Bowers, S., Ludäscher, B., 2016b. Two influential primate classifications logically aligned. Syst. Biol. 65, 561–582.
Additional Supporting Information may be found in the online version of this article:

**Data S1.** Answer Set Programming code (.txt), with extensive comments, to perform the 20 nomenclatural taxon use case in conjunction with the Potassco solver clingo. See also Table 1.

**Data S2.** Data file (.txt) with installation instructions and command line interface operations to run the ASP code and obtain the related reasoning and visualization outputs.

**Data S3.** Data file (.txt) with the complete textual clingo solver output for the 20 nomenclatural taxon use case. This output is used to produce the GraphViz visualizations.

**Data S4.** Java Archive file (.jar) needed to translate the clingo solver output into the corresponding GraphViz visualizations.

**Data S5.** Collated set of 36 GraphViz visualizations (transformed into .pdf) for the 20 nomenclatural taxon use case. To each of the Supporting Information sets S5–S8, we add the “Scenario 0” which represents no change. The remaining 36 outputs represent the change scenarios.

**Data S6.** Set of 36 Euler/X input data files for the respective alignments (summarized in Table 3). Each file is saved in .txt format and contains annotations and instructions for run commands to yield the alignments and input/output visualizations. The corresponding Euler/X .pdf input visualizations are also provided.

**Data S7.** Set of 36 Euler/X toolkit output Maximally Informative Relations (MIR) for the input data files provided in the Supporting Information S6. Each output file is saved in .csv format.

**Data S8.** Set of 36 Euler/X output alignment visualizations (.pdf) for the input data files provided in the Supporting Information S6.