Institutional Grammar 2.0 Codebook

Christopher K. Frantz* Saba N. Siddiki**

Version: 1.4 (20th October 2024)

Abstract

The Grammar of Institutions, or Institutional Grammar, is an established approach to encode policy information in terms of institutional statements based on a set of pre-defined syntactic components. This codebook provides coding guidelines for a revised version of the Institutional Grammar, the Institutional Grammar 2.0 (IG 2.0) (Frantz & Siddiki, 2021, 2022). IG 2.0 is a specification that aims at facilitating the encoding of policy to meet varying analytical objectives. To this end, it revises the grammar with respect to comprehensiveness, flexibility, and specificity by offering multiple levels of expressiveness (IG Core, IG Extended, IG Logico). In addition to the encoding of regulative statements, it further introduces the encoding of constitutive institutional statements, as well as statements that exhibit both constitutive and regulative characteristics. As a supplementary resource to the IG 2.0 specification, the codebook initially covers fundamental concepts of IG 2.0, before providing an overview of pre-coding steps relevant for document preparation. Detailed coding guidelines are provided for both regulative and constitutive statements across all levels of expressiveness, along with the encoding guidelines for statements of mixed form – hybrid and polymorphic institutional statements. The document further provides an overview of taxonomies used in the encoding process and referred to throughout the codebook. The codebook concludes with a summary and discussion of relevant considerations to facilitate the coding process. An initial Reader's Guide helps the reader tailor the content to her interest.

Contents

*Affiliation: Norwegian University of Science and Technology, E-mail: christopher.frantz@ntnu.no
**Affiliation: Syracuse University, E-mail: ssiddiki@maxwell.syr.edu

Acknowledgements

The authors wish to explicitly acknowledge the valuable input and feedback provided by Ute Brady, Edella Schlager, Matia Vannoni, and the IGRI interns as part of ongoing evaluation activities. We are further grateful for the constructive feedback on earlier codebook drafts received from Seth Frey, Bartosz Pieliński, and Marcello Ceci.

Document Revisions

All publicly released versions of this document can be retrieved under https://arxiv.org/abs/2008.08937.

Version	Description
1.4 (20th October 2024)	Refined selected IG Script examples to reflect current feature set, extended study checklist in Appendix A to offer clearer guidance for the development of project-specific coding instructions
1.3 (3rd March 2022)	Recoded examples in Section 4 using the IG Script notation, aligned taxonomies with book information, revised selected figures, conceptual and methodological overview
1.2 (20th June 2021)	Clarified definitions, restructured Context Taxonomy, introduced explicit guidelines for Attribute/Attribute Property differentiation, revised selected examples, added statement-level annotation characterizations, refined selected figures
1.1 (6th December 2020)	Refinement of component definitions and conceptual background; constitutive 'Modal'; refinement of institutional function characterization as regulative and constitutive functions respectively; addition of differentiation heuristics for constitutive and regulative statements; addition of 'Domain', 'Cause/Reason' and 'Event' categories in Context Taxonomy, Metatype Taxonomy; clarification of component-level nesting; extended example coding for constituted entity; introduction of Reader's Guide
1.0 (15th August 2020)	Initial codebook version (underlying concepts, pre-coding steps, coding guidelines for regulative and constitutive statements, constitutive-regulative statements, polymorphic statements, annotation taxonomies)

1. Introduction

1.1. The Codebook at a Glance

The following instructions are intended to provide guidance for the coding of institutional statements, the focal unit of analysis in the Institutional Grammar (Crawford & Ostrom, 1995, 2005), with specific focus on the Institutional Grammar 2.0 (IG 2.0) (Frantz & Siddiki, 2021, 2022), to which this text is supplementary.[1] *An institutional statement describes expected actions for actors within the presence or absence of particular constraints, or parameterizes features of an institutional system.* (Frantz & Siddiki, 2021) Institutional statements convey information that contextualizes their applicability. They vary in prescriptiveness and force, as reflected by the presence of information that more or less strongly compels behavior and by the presence of information that specifies payoffs for compliance, or non-compliance,

[1] In addition to Crawford and Ostrom (1995, 2005), the specification draws on the original IG codebook (Brady et al., 2018), and conceptual refinements (Frantz et al., 2013; Siddiki et al., 2011) comprehensively integrated into the IG 2.0 (Frantz & Siddiki, 2022).

with statements instructions. Varying in the inclusion of these various kinds of information, institutional statements typically take two functional forms: constitutive and regulative. *Constitutive statements* parameterize an institutional setting, and in this process, introduce, modify or otherwise constitute features of an institutional system and action situations embedded therein, including aspects such as actor positions and roles, associated affordances, processes, environmental characteristics, objects and artifacts inasfar as they are institutionally relevant. *Regulative statements* describe actors' duties and discretion linked to specific actions within certain contextual parameters.

According to the IG 2.0, institutional statements are commonly comprised of a set of syntactic components, with individual components associating with unique information, and which combine to convey a statement's institutional meaning. Broadly characterized,[2] *regulative statements* are composed of some or all of the following components with the corresponding syntactic labels: (i) a responsible actor, referred to as *Attributes*; (ii) action regulated by the statement, referred to as an *Aim*; (iii) statement context, referred to as *Context*; (iv) a receiver of action, referred to as an *Object*; (v) a prescriptive operator that describes how strongly an action is compelled or restrained, referred to as a *Deontic*; and (vi) a consequence of violating the regulated action, referred to as an *Or else*.

Constitutive statements are composed of some or all of the following components with the corresponding syntactic labels: (i) the entity that is being constituted or directly modified within a statement, referred to as a *Constituted Entity*; (ii) a parameterizing activity or function that introduces or otherwise characterizes the Constituted Entity in relation to the institutional setting and potential Constituting Properties, called the *Constitutive Function*; (iii) the statement context, referred to as *Context*; (iv) properties that serve as input to the constitutive function, called *Constituting Properties*; (v) a modal operator that defines to which extent the constitutive function of an institutional statement is required (necessary) or merely possible (optional), referred to as a *Modal*; and (vi) a consequence linked to the non-fulfillment of the function referenced in the constitutive function, referred to as an *Or else*.

The operational definition of an institutional statement is tied to the presence of certain syntactic components, or *necessary components*. To qualify as a complete regulative institutional statement, the statement must at least contain Attributes, Aim, and Context components (*necessary components*; however, they can be contextually implied or inferred[3]). The *Object*, *Deontic*, and *Or else* components are deemed *sufficient components*. Similarly, the presence of *Constituted Entity*, *Constitutive Function* and *Context* components is necessary to qualify as a constitutive institutional statement; *Constituting Properties*, *Modal* and *Or else* are sufficient components.

The elementary form of an institutional statement is the *atomic institutional statement*. An atomic institutional statement is a statement of regulative or constitutive kind that contains the corresponding necessary components, alongside potential optional components other than the *Or else*, and in which none of the components contains *component-level combinations* (e.g., multiple attributes, combinations of aims, etc.). Building on this concept, IG 2.0 introduces the concept of nested institutional statements in order to capture complex institutional configurations expressed in terms of institutional statements of various kinds interlinked in various forms, alongside additional features to facilitate structural and semantic decomposition of institutional arrangements at various levels of detail.

In this codebook, guidance is offered for the encoding of regulative and constitutive institutional statements along the aforementioned syntactic components at three levels of expressiveness: (1) *IG Core*; (2) *IG Extended*; and (3) *IG Logico*. The core definitions of syntactic components remain the same across levels of expressiveness. However, the level of granularity with which components are parsed differs across levels. Section 2 provides elaborated definitions of syntactic components that generalize across levels of expressiveness. This is followed by a terminological overview of concepts used in IG 2.0. Section 3 provides an overview of the pre-coding, or pre-processing steps relevant for document preparation prior to coding. Section 4 specifies the syntactic conventions used in this document, followed

[2] A precise definition of each component is provided in Section 2.1.

[3] This is discussed in greater detail as part of the component definitions provided in Section 2.

by the coding guidelines by syntactic component and level of expressiveness. This further includes the discussion of hybrid forms of institutional statements. Section 5 provides a comprehensive overview of taxonomies used for semantic annotations and referenced throughout the different sections. Section 7 offers a concluding discussion of considerations of relevance in the coding process, alongside references to additional resources.

To better guide the reader through the specification and guidelines provided in this document, the following section sketches potential 'pathways' through the codebook that emphasize aspects of interest tailored to specific audiences.

1.2. Reader's Guide

The codebook is intended to cover features of IG 2.0 comprehensively. While underlying conceptual principles will be discussed elsewhere,[4] a specific emphasis lies on providing concrete instructions to ensure a practical use, while aiming at being comprehensive at the same time. This codebook thus highlights features that are of primary concern for different audiences, such as the researcher aiming at gaining a comprehensive picture of the opportunities provided by IG 2.0, the study designer who is primarily concerned with methodological aspects, as well as operational coders of different backgrounds and experience. Finally, researchers from diverse disciplines may find interest in identifying opportunities that IG 2.0 provides more generally, but also to develop insights how the principles of the institutional grammar can be linked to concepts established in the researcher's domain. With those audiences in mind, in the following, we highlight selected *pathways through the codebook* to increase accessibility and streamline the contents for a given readership. Note that all references to sections, tables and other items throughout this document are clickable to support efficient navigation.

Pathway 'General Overview' Where readers intend to gather a high-level understanding of IG 2.0, the basic concepts, but limited concern for specifics of coding, we suggest consulting the following sections:

- *Section 2 Conceptual Foundations of the Institutional Grammar 2.0* provides a high-level conceptual backdrop to provide a basic understanding of all the involved concepts.
- To develop an intuitive understanding syntax and features across different levels, the review of *Section 4.2 Regulative Statement Coding* and *Section 4.3 Constitutive Statement Coding* is recommended.
- Where specifics related to operational coding are of interest, readers may narrow down specific features of interest using Table 1.[5]
- *Section 7 Conclusion* offers a summative overview, alongside important considerations for the use of IG 2.0, as well as pointers to further resources.

Pathway 'Research Design' Where readers seek to consider important aspects for study design for IG 2.0, such as a background for considering coding levels and statement types, as well as pre-processing guidelines for documents, we suggest consulting the following sections:

- *Section 2 Conceptual Foundations of the Institutional Grammar 2.0* offers an overview of the essential concepts and high-level operationalization.

[4] A more comprehensive treatment of motivation and conceptual underpinnings will be provided in forthcoming companion literature.

[5] The pathway 'Coder' highlights the exemplary use of Table 1.

- *Section 3 Pre-coding Steps* highlight all preprocessing steps the designer needs to consider for document preparation.
- *Section 4 Coding Guidelines* should be reviewed at high level in order to
  - determine the suitable level of encoding (IG Core, IG Extended, IG Logico, or combinations thereof – see *Section 2.6 IG Coding Levels*), and
  - identify the types of institutional statements to be considered in the study (regulative (*Section 4.2 Regulative Statement Coding*), constitutive (*Section 4.3 Constitutive Statement Coding*. Where combinations of both are relevant, the reader should further consider *Section 4.4 Hybrid & Polymorphic Institutional Statements*.

  Note that Table 1[6] supports the selection of relevant subsections corresponding to the desired feature set.

  Especially where both constitutive and regulative statements are considered, a specific emphasis should lie on *Section 2.5 Institutional Statement Type Heuristics*, to inform the development of project-specific coding guidelines for both statement types.
- Review *Section 4.6 IG Configurations (Institutional Grammar Profiles)* to identify the specific feature set to be used in the study.
- Where the handling of specific data structures is of concern (e.g., extraction of numerical information, collections), *Section 4.5 Data Structure Patterns* offers specific directions.
- *Section 7 Conclusion* highlights some considerations for the use of IG 2.0.

Pathway 'Coder' Where the reader is primarily concerned with the operational coding of statements in a given study, the selection of consulted sections varies based on background and study design. The 'Coder' pathway thus offers the most variable configuration of all suggested paths.

To tailor the reading accordingly, the coder should reflect on the following questions:

- Is the coder fundamentally acquainted with concepts of IG 2.0? If not, consult relevant sections in *Section 2 Conceptual Foundations of the Institutional Grammar 2.0* as identified in conjunction with the next question.
- Does the coding involve regulative and/or constitutive statements?
- Which level of expressiveness is the coding performed at (IG Core, IG Extended, IG Logico)?

  Depending on the combination of statements and level of expressiveness, choose the corresponding sections within this document by consulting Table 1 (Usage instructions are provided alongside the table).

  Example: Readers who wish to encode regulative statements on IG Extended level, should first consult relevant foundation sections, followed by *Section 4.1 Coding Symbols & Syntax* for an overview of the syntax employed in all coding examples. To engage in the actual coding, the reader should review *Section 4.2 Regulative Statement Coding*, and therein specifically *Section 4.2.1 IG Core Coding of Regulative Statements* and *Section 4.2.2 IG Extended Coding of Regulative Statements*. These sections provide instructions for IG Core as a coding basis, as well as IG Extended, which builds on the IG Core coding.

  Where, for example, both regulative and constitutive statements are of concern, the reader is encouraged to consult *Section 4.4 Hybrid & Polymorphic Institutional Statements*, in addition

[6] The pathway 'Coder' highlights the exemplary use of Table 1.

Relevant Section	Regulative	Constitutive	Hybrids	IG Core	IG Extended	IG Logico
Sections 2.1, 2.2, 2.4 & 2.6	●	●	●	●	●	●
Section 2.2.3			●		●	○
Section 2.3					●	○
Section 2.5	●	●	●			
Section 4.1	●	●	●	●	●	●
Section 4.2	●		●	●	●	●
Section 4.2.1	●		●	●	○	○
Section 4.2.2	●		●		●	○
Section 4.2.3	●		○			●
Section 4.3		●	●	●	●	●
Section 4.3.1		●	●	●	○	○
Section 4.3.2		●	●		●	○
Section 4.3.3		●	○			●
Section 4.4			●			
Section 5					○[1]	●

[1] The Context Taxonomy in Section 5.1 is relevant for IG Extended coding.

Usage Instructions: This table indicates the sections relevant to the coder based on feature sets of interest. Select the feature(s) of interest (types of statements; level of expression) in the columns to identify the relevant sections. Filled circles signal strong relevance for the subject of concern; hollow circle indicate indirect or partial relevance.

Table 1: Relevant Coding Instructions by Feature(s) of Interest

to the specific guidelines corresponding to statement type (*Section 4.2 Regulative Statement Coding* and *Section 4.3 Constitutive Statement Coding*) and sections relevant for specific levels of expressiveness.

- Note: If both regulative and constitutive statements are coded, *Section 2.5 Institutional Statement Type Heuristics* provides essential heuristics to ensure unambiguous characterization of statements, in addition to guidelines related to the coding of hybrid forms of institutional statements (*Section 4.4 Hybrid & Polymorphic Institutional Statements*).
- Additional information related to the coding of specific data structures is provided in *Section 4.5 Data Structure Patterns*.

Pathway 'Coder experienced in the Original IG' Where coders are experienced in the coding of the original institutional grammar, and intend to adapt their coding to IG 2.0 without consideration of extended feature sets, the following sections are of relevance:

- *Section 2 Conceptual Foundations of the Institutional Grammar 2.0* develops a basic understanding of the core concepts; specifically *Section 2.1 Syntactic Definitions of Institutional Statement Components*, and *Section 2.2 Institutional Statements* are important.

- To understand the basic coding of regulative syntax, review *Section 4.1 Coding Symbols & Syntax*, as well as the overview of statement structure in *Section 4.2 Regulative Statement Coding*, and the coding described in *Section 4.2.1 IG Core Coding of Regulative Statements* are relevant.
- To capture 'Context' (in loose correspondence to the Conditions characterization in the original IG (see Brady et al., 2018), the Item 'Context' in Table 8 (in Section 4.2.2) offers important clarifications, alongside the Context Taxonomy in Section 5.1.

Pathway 'Analytical Linkage' Where readers are primarily concerned with analytical opportunities associated with semantic annotations introduced in IG 2.0, e.g., in the form of domain-specific annotation sets, or attempt to draw relationships to concepts within their domain, the reader is encouraged to consult the following sections:

- *Section 2 Conceptual Foundations of the Institutional Grammar 2.0* develops a basic understanding of the core concepts.
- The forthcoming companion literature (linked here once available) will provide the analyst with a detailed overview of levels of expressiveness and associated analytical opportunities.
- The analyst should be clear about the level that corresponds to analytical needs and read sections accordingly, guided by the trade-off of shallow (IG Core) vs. deep structural coding (IG Extended), the need for semantic annotations (IG Logico), as well as the involved statement types, or any mix thereof (see *Section 2.6 IG Coding Levels*). To identify relevant sections, the reader is encouraged to draw on Table 1.
- Specifically relevant for the conceptual linkage to domain-specific frameworks or concepts is the consideration of semantic annotations more generally: IG Logico coding is shown in *Section 4.2.3 IG Logico Coding of Regulative Statements* for regulative statements, and in *Section 4.3.3 IG Logico Coding of Constitutive Statements* for constitutive statements; associated taxonomies can be found in *Section 5 Taxonomies*.

Where the reader cannot identify with a specific audience, a general guidance is to follow the 'General Overview' pathway, with focus on conceptual foundations initially (*Section 2 Conceptual Foundations of the Institutional Grammar 2.0*), followed by a selective overview of the coding principles for specific features in Section 4, and finally, the concluding section (Section 7).

2. Conceptual Foundations of the Institutional Grammar 2.0

IG 2.0 is premised on a set of syntactic definitions, conceptualizations of institutional statements, and assumptions regarding institutional statements, as well as various forms of nesting (statement-level and component-level nesting). While these definitions, conceptualizations, and assumptions generalize across levels of IG encoding, how they are captured depends on at which level the encoder is working. In this section, we will thus lay out these foundational syntactic definitions, concepts, and assumptions. We start by offering complete definitions of IG components more generally, and then move to defining key institutional statement concepts, including the various notions of nested institutional statements and associated assumptions. We further provide principles and operational guidelines for the identification of statements in as far as they relate to the fundamental concept *Action Situation*, namely context characterizations embedded in institutional statements and identification of institutional statement types in the first place. Concluding this section, we organize the coding levels based on involved features, along with providing principal guidelines for the coding process. Note that most concepts in this section are introduced with a pragmatic focus on providing essential foundational understanding to perform the coding of institutional statements. For an extended treatment of the conceptual foundations, refer to Frantz and Siddiki (2022).

2.1. Syntactic Definitions of Institutional Statement Components

The IG structure as referred to in this document relies on the elementary syntactic components of regulative and constitutive statements highlighted in Tables 2 and 3. Definitions of these components that hold across IG 2.0 encoding levels are provided alongside each syntactic component.

Syntactic Element	**Definition**
Attributes	An actor (individual or corporate) that carries out, or is expected to/to not carry out, the action (i.e., Aim) of the statement. The Attributes component may also contain descriptors of the actor. The presence of this component is *necessary* for any regulative statement.
Deontic	A prescriptive or permissive operator that defines to what extent the action of an institutional statement is compelled, restrained, or discretionary. In institutional statements varying levels of prescriptiveness or strength thereof are represented with different terms or language that situate along a continuum. The presence of this component is *optional* for regulative statements.
Aim	The goal or action of the statement assigned to the statement Attribute. The presence of this component is *necessary* for any regulative statement.
Object	The inanimate or animate part of an institutional statement that is the receiver of the action captured in the Aim. Objects can be of *direct* or *indirect* nature. Indirect objects are objects that are affected or targeted by the application of the Aim to direct objects. Objects can both be real-world entities, or abstract ones (e.g., beliefs, concepts, facts, procedures). Abstract objects here further include complex constructs, such as conditions/processes that the responsible actor is or will be acting upon. Such complex objects can potentially take the structural form of institutional statements themselves. The presence of this component is *optional* for regulative statements.
Context	The context component instantiates settings in which the focal action of a statement applies, or qualifies the action indicated in an institutional statement. The former type of Context is referred to as an *"Activation Condition"*. The latter type of Context is referred to as an *"Execution Constraint"*. Both can occur in a given institutional statement, including multiples of either type. The presence of this component is *necessary* for any regulative statement, but can be implied (as with the original institutional grammar). Where no explicit Activation Condition is specified, the context clause is by default "under all conditions". Where no explicit Execution Constraints are specified, the context clause is by default "no constraints". It is important to note that *Context* in institutional statements reflects the context specific to the coded statement (Statement Context), as opposed to capturing context in the wider sense, making reference to the context of the policy or the domain more generally.
Or else	A sanctioning provision associated with the action indicated in a particular institutional statement represented in a nested institutional statement (as defined in the following discussion). Where combinations of regulative and constitutive statements are applied (hybrid institutional statements), the consequence can be existential in kind (see Table 3 and Section 4.4). The presence of this component is *optional* for regulative statements.

Table 2: Definitions of Syntactic Elements for Regulative Statements

Syntactic Element	**Definition**
Constituted Entity	The entity being constituted, reconstituted, modified or otherwise directly affected within an institutional statement as characterized by the constitutive function. This can be an entity as relevant in the institutional setting, or the policy itself. The presence of this component is *necessary* for any constitutive statement.
Modal	An operator that signals necessity or (im-)possibility of constitution or modification of system features captured in the constitutive function. In institutional statements varying levels of necessity or strength thereof are represented with different terms or language that situate along a continuum. The presence of this component is *optional* for constitutive statements.
Constitutive Function	An expression that relates the constituted entity and the institutional setting, e.g., by defining, establishing, or modifying the former, including the conferral of status. Where constituting properties exist, constitutive functions functionally link Constituted Entity and Constituting Properties. The presence of this component is *necessary* for any constitutive statement.
Constituting Properties	Constituting Properties parameterize the Constituted Entity via the Constitutive Function, and can both be physical or abstract in kind. Constituting Properties may or may not be present in constitutive statements. The presence of this component is *optional* for constitutive statements.
Context	The Context instantiates settings in which the focal Constitutive Function of a statement applies, or qualifies the function indicated in an institutional statement. The former type of Context is referred to as an *"Activation Condition."* The latter type of Context is referred to as an *"Execution Constraint."* Both can occur in a given institutional statement, including multiples of either type. The presence of this component is *necessary* for any constitutive statement, but can be implied (as with the original institutional grammar). Where no explicit Activation Condition is specified, the context clause is by default "under all conditions". Where no explicit Execution Constraints are specified, the context clause is by default "no constraints". It is important to note that *Context* in institutional statements reflects the context specific to the coded statement (Statement Context), as opposed to capturing context in the wider sense, making reference to the context of the policy or the domain more generally.
Or else	A consequence characterization associated with the non-fulfilment or -enactment of the Constitutive Function indicated in a separate institutional statement nesting on the leading institutional statement (as defined in the following discussion). Consequences, as understood in the context of constitutive statements, can be existential in kind (e.g., invalidating policy), as opposed to the non-existential sanctioning provisions on the regulative side. The presence of this component is *optional* for constitutive statements.

Table 3: Definitions of Syntactic Elements for Constitutive Statements

2.2. Institutional Statements

Capturing both the regulative and constitutive nature of statements, in IG 2.0 *an institutional statement describes expected actions for actors within the presence or absence of particular constraints, or parameterizes features of an institutional system.*

To accommodate a more comprehensive representation of structure and content, the construction of institutional statements in IG 2.0 requires a differentiated understanding of institutional statements, both in terms of *atomic* and *nested institutional statements* of various forms.

2.2.1. Atomic Institutional Statements

Before highlighting notions of nested institutional statements, we reiterate the notion of an atomic institutional statement as the elementary form of an institutional statement: an *atomic institutional statement* is a statement of regulative or constitutive kind that contains the corresponding necessary components, alongside potential optional components other than the *Or else*, and in which none of the components contains multiple values (*component-level combinations*) or is otherwise nested (*component-level nesting* – see Section 2.2.3). Examples for component-level combinations are multiple attributes or combinations of aims in a given institutional statement. Examples for component-level nesting include context characterizations (with details to be introduced at a later stage) expressed as institutional statements themselves.

Illustrating this concept, the statement *"Organic farmers must commit to organic farming standards"* is *atomic* in kind. It identifies the following elements:

- Attributes: Organic farmers
- Deontic: must
- Aim: commit to
- Direct object: organic farming standards
- (Implied) Context: under all conditions.

It does not include an *Or else*, and neither of the components contains value combinations.[7]

In contrast, the statement *"Organic farmers must commit to their organic farming standards and accommodate regular reviews of their practices"* combines multiple distinctive activities and associated objects, namely *"commit to organic farming standards"* and *"accommodate regular reviews of their practices"*, as illustrated below by aggregating all phrases by component type (multiple phrases are separated by semicolon).

- Attributes: Organic farmers
- Deontic: must
- Aim: commit to; accommodate
- Direct Object:[8] organic farming standards; regular reviews of their practices
- (Implied) Context: under all conditions.

We refer to any such combination of component values as a *component-level combination*.

In consequence, this statement is *not atomic* in kind.

[7] Note that a component can contain phrases; however, the terms need to represent a semantic unit in the context of the concerned institutional statement.

[8] The distinction between direct and indirect object is discussed in greater detail at a later stage.

2.2.2. Nested Institutional Statements

Motivating the conception of nested institutional statements is the pragmatic observation that statements can reflect a linkage or embedding of multiple actors or regulated activities, reflecting complex configurational settings; rules in form are generally not expressed in the atomic form specified above. Lack of specificity of these linkages undermines the coder's ability to comprehensively, and accurately, capture institutional content for versatile downstream analysis.

IG 2.0 accommodates two forms of institutional statement nesting: *horizontal nesting* and *vertical nesting*. Generally, horizontal nesting allows for the representation of multiple institutional statements that convey co-occurring or alternative actions, actor or object involvement, i.e., any case in which multiple of the same syntax element occur in an institutional statement.

Generally, vertical nesting allows for the representation of multiple institutional statements that convey coupled actions that follow from one another in the form of a consequential relationship. It is particularly suited to representing the case of consequentially linked statements in which statement A delineates permitted, required, or forbidden activity (for regulative statements), or is constitutive in kind, and statement B delineates sanctions for non-conformance with statement A. In the IG 2.0 parlance, statement A is considered a "*monitored statement*," and statement B a "*consequential statement*." Horizontal nesting and vertical nesting are described in more detail below.

Horizontal Nesting: Horizontal nesting describes a logical combination of two or more statements to capture institutional content comprehensively. Exemplified in narrative form, a horizontally nested statement can combine two or more statements. Borrowing the example introduced earlier (*"Organic farmers must commit to their organic farming standards and accommodate regular reviews of their practices"*), horizontal nesting reflects the decomposition of such statement into two logically-combined atomic institutional statements, such as

"Organic farmers must commit to organic farming standards" **[AND]** *"Organic farmers must accommodate regular reviews of their practices"*

Horizontal nesting allows for more complex constructs linking an, in principle, arbitrary number of atomic institutional statements and combinations thereof, such as

("Organic farmers must commit to organic farming standards" **[AND]**
*"Organic farmers must accommodate regular reviews of their practices") **[XOR]***
("Organic farmers must **[NOT]** *sell their produce under the organic farming label").*

In this example, we can observe the linkage of three statements, in which two AND-combined statements apply as an alternative (XOR) to a third statement.

Note the use of parentheses to signal the precedence of individual statements where multiple logical operators are used.

Possible logical operators for horizontal nesting, or *statement combinations*, are **[AND]** (conjunction), **[OR]** (inclusive disjunction; colloquially: **AND/OR**), or **[XOR]** (exclusive disjunction; colloquially: **EITHER/OR**). Where negation is involved, those can be combined with the operator **[NOT]** (as highlighted in the previous example). An alternative equivalent representation is

("Organic farmers must commit to organic farming standards" **[AND]**
*"Organic farmers must accommodate regular reviews of their practices") **[XOR]***
[NOT] *("Organic farmers must sell their produce under the organic farming label").*

A common use case in practice is the decomposition of action combinations (e.g., *"... establish and maintain ..."*) or multiple actors (e.g., *"... producers or inspectors of the facility ..."*).[9]

[9] The decomposition of such component-level combinations is discussed as part of the coding guidelines in Section 4, specifically in Table 7, Item *Decomposition of component-level combinations*, as well as the same item in Table 8 for regulative statements. Table 10, Item *Decomposition of component-level combinations* provides guidelines for

Vertical Nesting: Vertical nesting describes a relationship of two or more statements, in which the leading statement (*monitored statement*) describes an action that is regulated by a second statement nested in the Or else component (*consequential statement*), rendering the Or else an abstract component backed by a separate institutional statement. More accurately, the Or else does not reflect an IG component, but rather a logical connective that links two institutional statements. In this linkage, the second statement reflects a consequence of violating the instructions captured in the monitored statement. Consequences generally involve some pay-off for non-compliance or compliance.[10] Exemplifying vertical nesting in narrative form, we can write

"Organic farmers must comply with organic farming regulations", **OR ELSE** *"Certifiers must revoke the organic farming certification".*

Note that both forms of nesting can be combined, i.e., monitored and consequential statements can embed horizontal nesting. Extending the previous example, we can state (visually supported by indentation to signal the corresponding forms of nesting)

("Organic farmers must comply with organic farming regulations" **AND**
"Organic farmers must accommodate regular review of their practices"),
OR ELSE *("Certifiers must suspend the organic farming certification"* **XOR**
"Certifiers must revoke the organic farming certification").

Note the use of parentheses to signal precedence of the respective statements. Vertical nesting can occur across an arbitrary number of levels (i.e., a consequential statement may be a monitored statement in deeper levels of nesting). Exemplifying multi-level nesting, we can state

("Organic farmers must comply with organic farming regulations" **AND**
"Organic farmers must accommodate regular review of their practices"),
OR ELSE *("Certifiers must suspend the organic farming certification"* **XOR**
"Certifiers must revoke the organic farming certification"),
OR ELSE *"USDA may revoke certifier's accreditation".*

The combination of both forms of nesting affords the representation of complex institutional arrangements, both in terms of institutional content (horizontal nesting) and enforcement (vertical nesting). The principles are schematically highlighted in Figure 1.[11] While exemplified here for regulative statements, the principles equally apply to constitutive, and in consequence, hybrid institutional statements (see Section 4.4).

2.2.3. Component-Level Nesting

In addition to the nesting of statements, IG 2.0 further recognizes the possibility of nesting on individual components. Specific components may be substituted with entire institutional statements, or generalizable instances of activity (e.g., statements of fact, observed behavior) expressed in the syntactic form of institutional statements. For regulative statements, those may characterize individuals with respect to their actor properties, embed statements as abstract objects (e.g., the belief that citizens must comply with the law), or express preconditions, i.e., statements whose fulfillment is a precondition for the application of a given statement (as exemplified below).

To motivate this approach explicitly, we will use the following example: *"Organic farmers may sell their produce under the organic label {under the condition that organic farmers apply for certification}"*, with the precondition (the *Activation Condition* as a more specific variant of the Context component,

constitutive statements. General operational guidelines for fine-grained decomposition can be found in Table 9, Item *Logical relationships among statement components*. Note: All preceding table item references are clickable links.

[10] Where constitutive statements are involved, consequences can further be existential in kind.

[11] An formal discussion of nesting can be found in Frantz et al. (2013) and an extended conceptual discussion is provided in (Frantz & Siddiki, 2022).

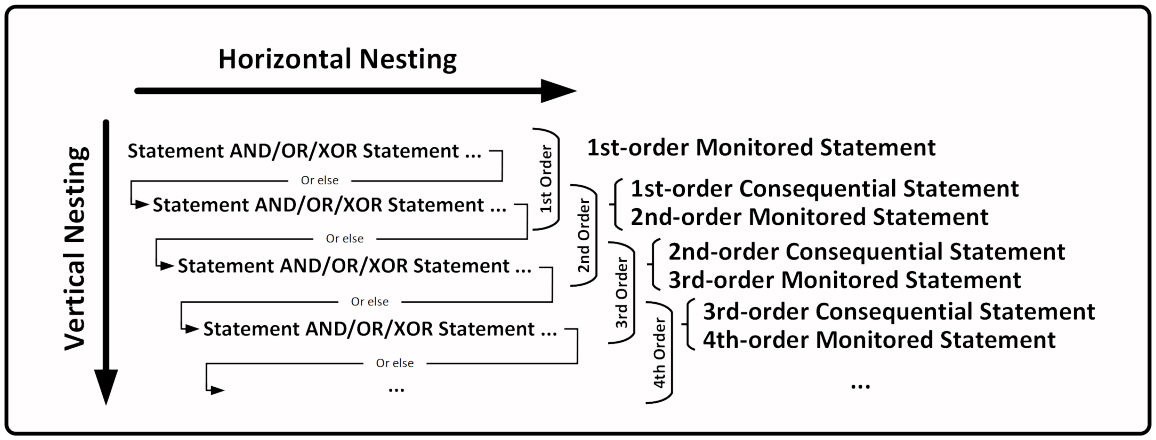

Figure 1: Institutional Grammar Nesting Principles

as to be discussed in Section 2.4) identified in braces. Rewriting this statement for illustration, we find the following component characterization:

- Attributes: Organic farmers
- Deontic: may
- Aim: sell
- Direct Object: their produce
- Execution Constraint (Context): under the organic farming label
- Activation Condition (Context):[12] under the condition that organic farmers apply for certification

Here the instance of behavior embedded in the activation condition reflects the precondition, but, upon closer inspection, may in itself be expressed in terms of syntactic components of institutional statements, either reflecting a specific action configuration (as statement of fact, behavioral instance, or observation) or an institutional statement in its own right (including combinations of statements).

We refer to cases where individual components of an institutional statement can be replaced by institutional statements (or statements that follow the syntactic structure of institutional statements) as *component-level nesting*. An institutional statement that contains any such nested structure (as initially indicated in Section 2.2.1) is not atomic in kind.

Expanding the statement above correspondingly, we arrive at the following structure exemplifying the nesting on the activation condition:

- Attributes: Organic farmers
- Deontic: may
- Aim: sell
- Direct Object: their produce
- Execution Constraint (Context): under the organic farming label
- Activation Condition (Context): under the condition that
  - Attributes: organic farmers
  - Aim: apply

[12] The activation condition and execution constraint components are a specialization of the Context component, which will be introduced at greater detail in Section 2.4.

  - Direct Object: for certification

The use case for such component-level nesting further includes the articulation of abstract concepts, including non-physical concepts (e.g., beliefs, suspicions, etc.) about individuals' behaviors inasfar as those are relevant to capture the institutional setting accurately (e.g., an official may sanction an individual if the official believes that the individual has performed a violation).

Exemplifying nesting on the object component, we can use the statement *"Inspectors must ensure that organic farmers comply with organic farming regulations"*. In this instance, the nesting occurs on the object component, i.e., the receiver of the action *ensure*, as visualized below.

- Attributes: Inspector
- Deontic: must
- Aim: ensure
- Direct Object:
  - Attributes: organic farmers
  - Aim: comply
  - Direct Object: organic farming regulations
- (Implied) Context: under all conditions

While not motivated further at this stage, component-level nesting (as statement-level nesting) equally applies to regulative and constitutive statements. Component-level nesting can apply to Attributes, Objects, Context components (Activation conditions and Execution constraints) in regulative statements, as well as to Constituted Entity, Constituting Properties and Context components in constitutive statements. It can further be applied in combination with all forms of statement-level nesting, including the further nesting within component-level nested statements. (For example, imagine a chain of preconditions expressed in nested statements).

The coding for all forms of nesting and combinations thereof, as conceptually described here, is specified in Section 4.

2.3. Object-Property Hierarchy

In addition to the nesting concepts, advanced coding relies on decomposing actors and objects into core descriptors and associated properties. For this purpose, we rely on the conceptual representation of an *Object-Property Hierarchy* as exemplified in Figure 2. In this visualization, statements such as *"... a written notification of proposed suspension or revocation of certification ..."* reflect an involved Object hierarchy centering on the *"notification"*, that has a property *"written"*. Looking at the context of the notification we recognize the concept of *"certification"* that has the property of being *"suspended"* or *"revoked"*, expressed as dependent Objects (*"suspension"*, *"revocation"*), whereas the latter concepts themselves have a shared property of being *"proposed"* in the first place. However, while the property *"written"* functionally depends on the *"notification"*, that is, writtenness alone does not make sense with an Object it refers to, the existence of a certification does not rely on the notification (i.e., it is functionally independent), and has a self-contained property hierarchy (suspended, revoked, proposed) as described above.

Interpreting complex Object specifications with this decomposition hierarchy in mind affords a uniform coding approach. The structure of this hierarchy for the specific statement is shown in Figure 2 (the dashed line signals relationships between functionally independent Objects). Note specifically the

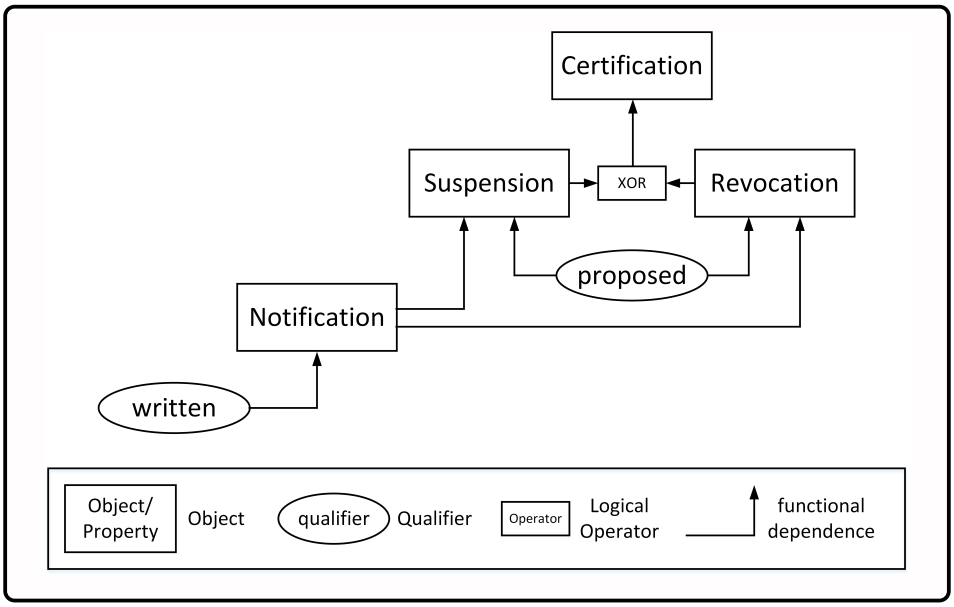

Figure 2: Object-Property Hierarchy for the given Example

potential use of logical operators (*"XOR"*), as well as the ability to reflect shared properties (*"proposed"*), to disambiguate the logical relationship between the identified properties, an aspect that is implicit in the textual representation.

Capturing all potential decomposition approaches, we can apply this scheme to functionally dependent properties (*"written"* in the previous example), functionally independent Objects or properties (*"certification"* in the previous example), and furthermore afford the substitution of Objects by complete institutional statements. Since the latter aspect relies on richer contextualization, it will be exemplified in the context of the coding instructions. The stylized general form of the *Object-Property Hierarchy* is shown in Figure 3. Note that logical operators apply to both functionally dependent and independent properties and on any level of decomposition.

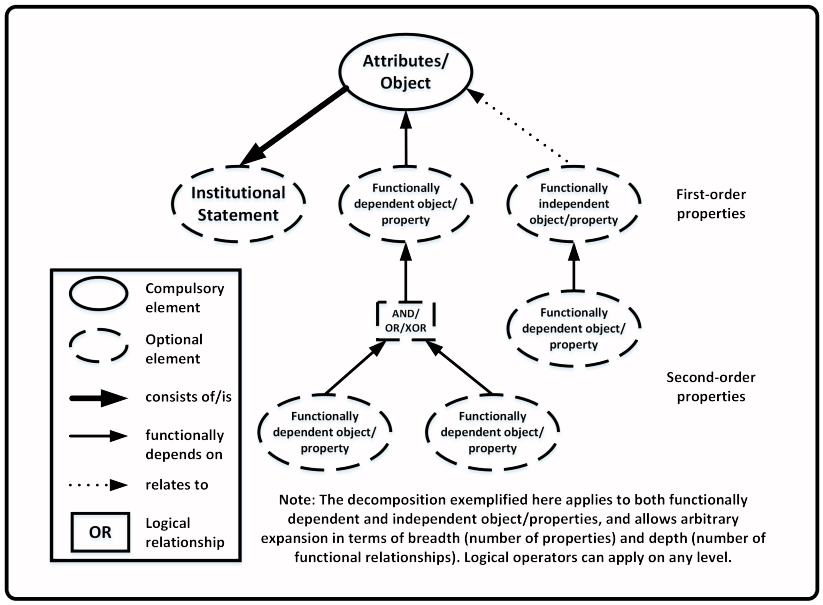

Figure 3: Object-Property Hierarchy

2.4. Relating Institutional Statements to Action Situations

A concept central to the coding with IG 2.0 is the action situation (Crawford & Ostrom, 2005). Basically, an action situation is defined as an institutionally governed setting in which two or more actors interact, in relation to which specific outcomes emerge. Action situations are governed by a configuration of different types of institutional statements, which can be regulative or constitutive in kind, with distinctive functional properties. The seven types of institutional statements, labeled in parentheses in terms of different types of "rules", convey: positions that actors can occupy within an action situation (position rules), eligibility criteria for occupying those positions (boundary rules), operational actions linked to actors occupying certain positions (choice rules), situational outcomes (scope rules), channels of information flow (information rules), guidance on collective decision making (aggregation rules), and incentives tied to particular actions (pay-off rules). Each action situation can be governed by multiple statements of a particular type. Action situations, and key action situation components, are schematically visualized in Figure 4. In the widest sense, action situations describe the context in which institutional statements operate, and in the context of regulative statements, specifically the mapping between actors, actions, outcomes and the associated payoffs.

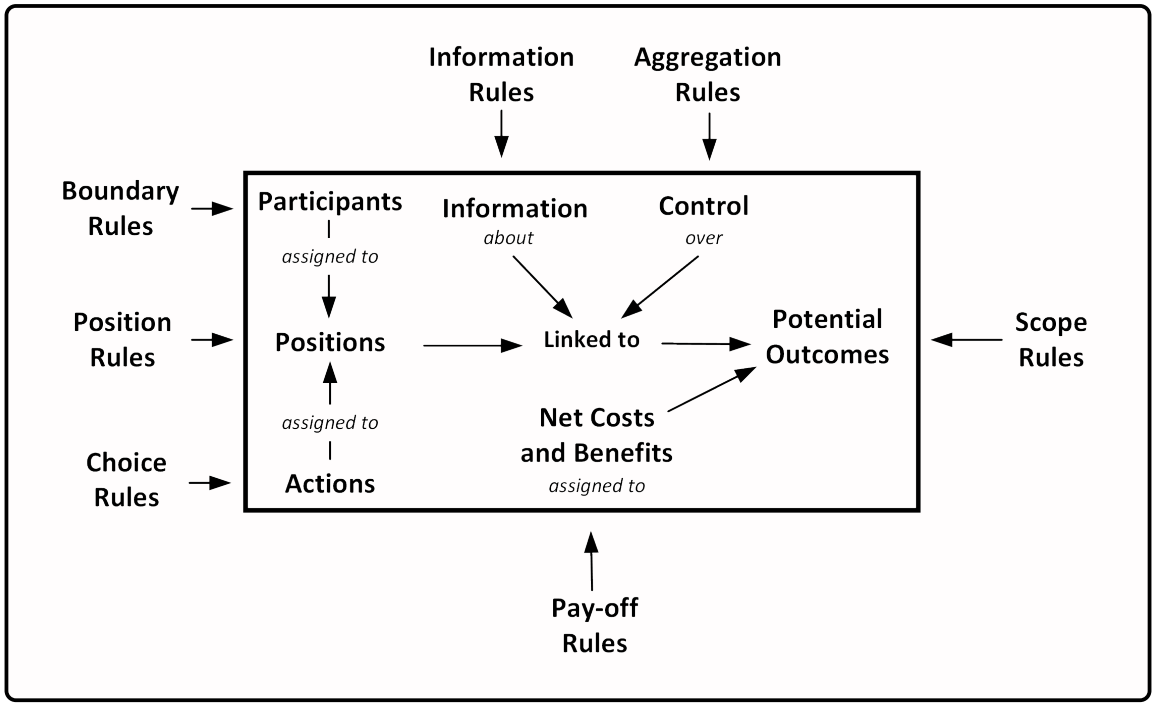

Figure 4: Schematic Visualization of the Action Situation (as per Crawford and Ostrom (2005))

Again, the rule type taxonomy associated with the action situation concept links to the IG insofar as whole institutional statements can be classified accordingly, depending on their functional properties.

The IG 2.0 specification leverages the action situation concept in recognizing generally that institutional statements characterize activity occurring in action situations, and in accounting through syntactic classification for ways that institutional statement information corresponding to the Context component contextualizes intra- and inter-statement activities.

Operationalized in the context of regulative statements, Context clauses can instantiate an action situation in which an Attribute acts on Objects in a particular manner, and which are governed by some configuration of institutional statements. By way of contrast, the Context clauses of other statements may simply constrain an Attribute's behavior in some way within a given action situation. As noted in Section 2.1, context clauses which serve an instantiation function, as well as Attribute or Object changes are referred to as Activation Conditions. Context clauses which qualify action are referred to

as Execution Constraints. Figure 5 schematically represents how Activation Conditions and Execution Constraints situate relative to action situations.

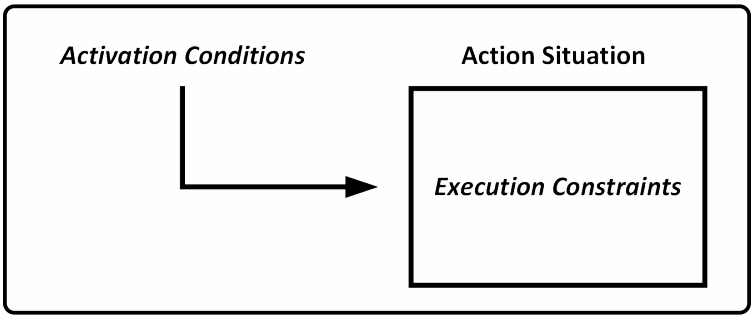

Figure 5: Activation Conditions and Execution Constraints Principles

Naturally, explicit characterization of how institutional statements relate to action situations necessitates understanding of the institutional domain. Without this, the coder may encounter difficulty in determining as to whether a specific Context descriptor refers to the action situation more generally, or the action specifically in the form of an action property. While we offer further elaboration as part of the coding guidelines in Section 4, we provide a brief example to motivate the distinction at this stage.

Inherent to the Activation Condition is reference to a set of exogenous variables; exogenous in the sense that it references states or actions that are beyond the actions that can be qualified within certain Execution Constraints in an instantiated environment (e.g., a new action situation, an environment in which Attributes change or take on new roles, or an environment in which Attributes act in an altered way upon Objects). In other words, Activation Conditions precede the regulated action and activate a given institutional statement in the first place. Conversely, Execution Constraints describe constraints on actions once enacted (and implicitly on actors and associated pre-/proscription as visualized in Figure 6).

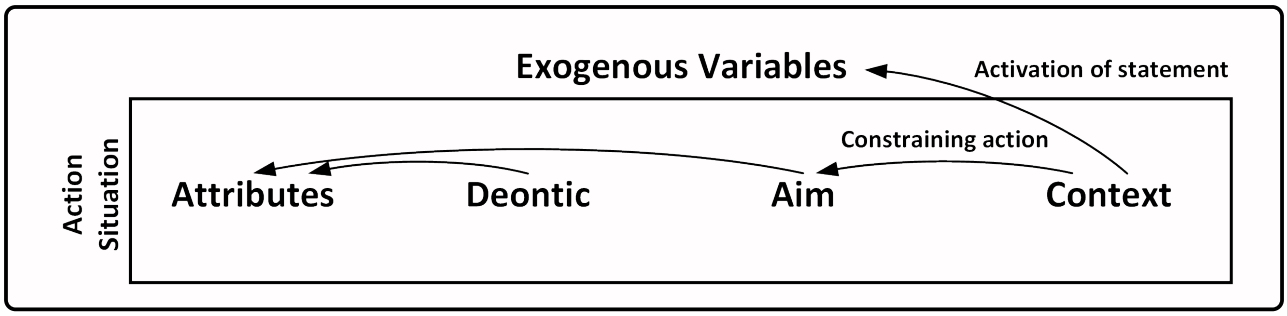

Figure 6: Context Relationships in the Action Situation

Procedurally, this implies different semantics for Activation Conditions and Execution Constraints. Activation Conditions represent Context external to the action situation an institutional statement is embedded in that activates the non-conditional part of an institutional statement, and possibly leading to the activation or modification of an action situation. Execution Constraints, in contrast, are directly attached to the institutional statement and thus reflect context embedded within the action situation itself. The discussed distinction is summarized in Figure 6.

Offered here is more operational guidance on the differentiation highlighted above, which starts with the identification of Context clauses in an institutional statement. To systematize the differentiation, we firstly provide a terminological basis. Linguistically, context clauses are generally modifiers, specifically qualifiers (*"usually"*, *"some"*, *"annually"*), adverbial clauses (*"When the traffic light turns from red to*

green, . . . ") and prepositional clauses (*"after midnight"*). Whereas qualifiers reliably signal Execution Constraints (action properties in the narrow sense), and adverbial clauses generally indicate Activation Conditions, depending on contextual interpretation, prepositional clauses can fall in either category and selectively signal Activation Conditions or reflect Execution Constraints (action properties in the wider sense).

The differentiated treatment for prepositional clauses is best described with an example: *"At 8am, farmers may begin selling their goods in accordance with market rules,"* contains two context clauses (*at 8am, in accordance with market rules*), one of which is a conditions clause (*at 8am*) and one of which is a constraints clause (*in accordance with market rules*). Context clauses may be implicit, and institutional statements are not constrained in the number of clauses for condition and constraint type. The remainder of the statement is the non-context clause of an institutional statement. Figure 7 highlights this decomposition of regulative institutional statements with respect to the Context component.

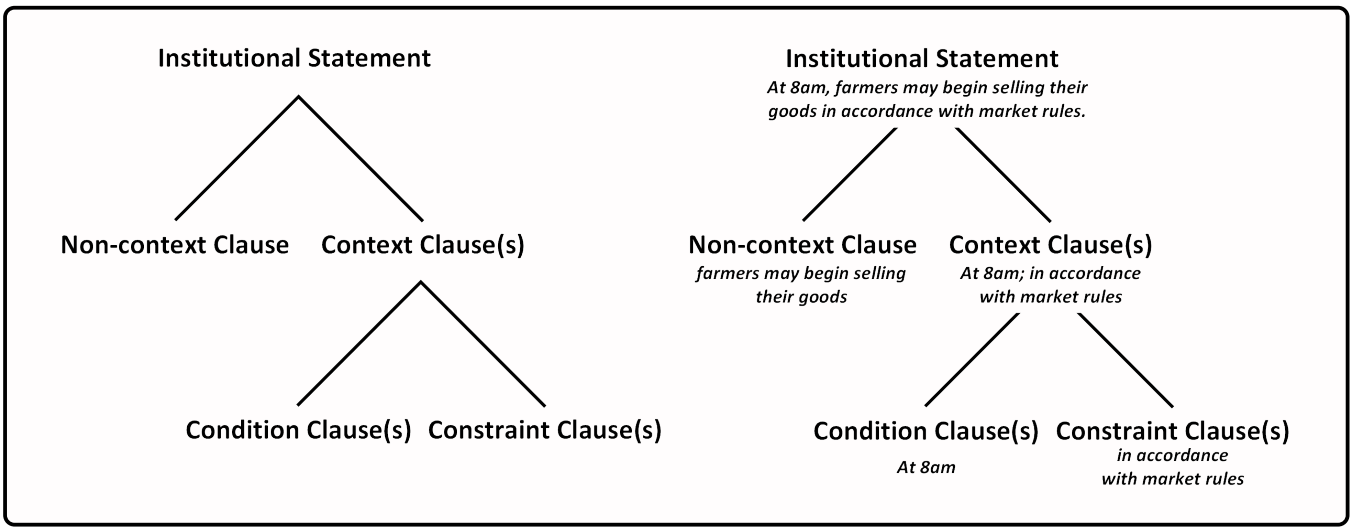

Figure 7: Context Clauses in Institutional Statements

Decision heuristics can be employed to aid in the identification of activation conditions and execution constraints. The following heuristics are particularly designed to help the analyst determine if a context clause in question is an activation condition, leaving the resultant classification of the clause as an execution constraint, if it is determined that it is not. Offered first is a heuristic that generalizes across regulative and constitutive statements. This is followed by heuristics specific to two different types of statements.

General Heuristic for Identifying Activation Conditions:

The clause instantiates a discrete setting (constrained temporally, spatially, or otherwise as shown below) and/or event that activates the non-condition clauses of the institutional statement (i.e., non-context clauses along with potential constraint clauses) as a whole. The following example statements contain activation conditions (underlined) for illustration.:

- *"Upon receiving final notice of non-compliance, farmers shall cease sale of any product bearing the USDA organic farming label."* This statement signals the instantiation of a novel attribute-object link described by the activity, and is positioned within a discrete temporal setting.
- *"Starting January 1, the Department of Agriculture is the certifying authority."* Here the statement makes explicit reference to an event that leads to the activation of an associated role change in a constitutive statement.

- *"Upon entry into the house, visitors must remove shoes."* (Event) vs. *"At home individuals must not wear shoes."* (Discretized setting). Whereas the first statement references a specific event (entry), the second statement describes a general discretized setting in which the statement holds at all times, i.e., the statement is activated at any time.

Heuristics for Identifying Activation Conditions in Regulative Statements:

Attributes: The clause instantiates a) a change in attributes linked to a statement's activity or b) a change in attribute role.

- *"Between the hours of 6pm and 6am on Mondays, members of neighborhood watch residing in blocks 7-10 will assume night patrol activities."* This example signals a change in attribute role within a specified time frame.

Objects: The clause instantiates a change of the object(s) linked to the statement's activity.

- *"Starting Dec. 15th, inspectors must exclusively use the revised inspection form."* (novel object use)

To support the classification more generally, we can further offer practical considerations to aid in the decision making:

- Regulative statements: More generally, in a regulative context with a given specification of pre/proscriptions, activation conditions constitute a discretized setting in which institutional content can in principle be adhered to or violated.
- Context-clause interdependencies: If the application of the clause of concern is contingent on the prior activation of another context clause, the former is an execution constraint, whereas the latter describes an activation condition in the context of the analyzed institutional statement. If only the satisfaction of both clauses leads to the activation of the non-context clause, both are sensibly identified as activation conditions.
  - Example: *"When live fish or viable gametes are sold, traded, taken or otherwise disposed of from an aquaculture facility, the permittee or operator shall, at the time of transfer of possession, give an invoice to the person receiving such fish or viable gametes."* Here *"when live fish . . . facility"* represents the activation condition; the subsequent provision of an invoice *"at the time of transfer of possession"* is an execution constraint.

Returning to the initial example *"At 8am, farmers may begin selling their goods in the farmer's market."*, the condition clause *"At 8am"* signals the instantiation of a discrete temporal setting in which the remaining statement is activated as a whole (i.e., permitting the sales of goods on the market). The clause *"in the farmer's market"* complements the description of the regulated content and neither affects attribute/role, object nor does it define a setting that activates the remaining statement. The resulting statement would thus be coded as follows:

Coded statement: "At 8am (Activation statement), farmers (Attribute) may (Deontic) begin selling (Aim) their goods (Object) at the farmer's market (Execution constraint)."

To highlight the distinction between activation conditions and execution constraints more clearly, we can review the following statement: *"Farmers may sell non-organic goods in the organic farmer's market only between the hours of 3 and 5pm."*

Here the time frame *"between the hours of 3 and 5pm"* signals a distinctively different relationship between attributes (farmers), aim and object, whereas the *"in the farmer's market"* complements the characterization of the institutional setting in which the permission holds.

This is in contrast to the following statement:

"Farmers must perform inventory of goods sold at farmers market daily." (execution constraint)

This statement signals a general obligation to provide inventory information (i.e., is activated at all times), but does not establish a specific discretized setting or event that triggers the obligation.

Contrasting this, the following example highlights such event, leading to the characterization of the conditional clause as activation condition:

"At the close of market each day (activation condition), farmers must perform inventory of goods sold."

Heuristics for Identifying Activation Conditions in Constitutive Statements:

- *Entity*: The clause instantiates a change in the Entity that is being constituted.
  - Example:

    "In the event that the Board Chair position becomes vacant, the Vice-Chair is the chief executive of the Council."

    (change in entity specification under event)

  Properties: The clause instantiates a change in the constituting properties of the entity that is constituted, reconstituted or otherwise affected in the institutional statement.
  - Example:

    "Starting Dec. 15th, organic farming is agricultural production that does not involve the use of synthetic chemicals or genetically modified organisms."

    (change in constituting properties of constituted entity)

Considerations for challenges cases:

In written institutional statements, it is not uncommon to find constructions that elevate a statement's pre- or proscriptiveness embedded in activation condition or execution constraint. A stylized example is the following statement:

"Drivers must comply with traffic rules, especially when encountering congested traffic."

A specific characteristic of such statement is the elevated prescriptiveness based on the context clause, with specific focus on the contextual characterization *"... especially when encountering congested traffic"*. As such, the context characterization signals a situational increase of prescriptiveness, beyond the generally expected compliance. Facing such statements, the analyst will need to evaluate, whether this statement exclusively highlights such contextual elevation, which is interpreted as an execution constraint, or characterizes a specific condition under which the non-context part of a statement applies as a whole. Typical terms associated with elevated prescriptiveness mediated via execution constraints include

- especially
- particularly
- specifically

Activation conditions are conventionally signaled using terms such as

- where possible
- under the condition
- when ...

2.5. Institutional Statement Type Heuristics

IG 2.0 has been developed in response to observed challenges in the encoding of policy, such as the inability to sensibly capture institutional statements of constitutive kind. In addition to affording the syntactic parsing of constitutive institutional statements, their unambiguous characterization is of central importance so as to ensure coding reliability.

While a characterization based on syntactic alignment has an intuitive appeal, not in all instances does this lead to correct statement classification, e.g., based on contextually implied actor associations or stylistic artifacts. To afford a reliable characterization of statements, we rely on a set of first principles expressed in a set of general guidelines that seek to identify both constitutive and regulative statements, alongside potential combinations of those, the expression of which IG 2.0 explicitly supports. This is then followed by a more specific set of questions, supporting the identification of individual features relevant for a distinctive statement characterization.

To lend best possible guidance, the guidelines, or heuristics, are expressed as closed questions, alongside clarifying explanations.

General Heuristics for Identifying Institutional Statement Types:

- Does the statement introduce or parameterize fundamental aspects of the action situation (boundaries, definition or modification of actors, actions, objects, artifacts and associated affordances; endowment of rights or authority/power), and in doing so, define positioning and constellation of entities in an institutional setting in which potentially regulated behavior is enacted? → If so, the statement is *constitutive* in kind.

- Does the statement signal unambiguously implied or explicit agency, while specifying duty and discretion, or sanctions for transgression? In doing so, does the statement draw on (i.e., makes implicit or explicit reference to) actors, objects or artifacts, including a potential modification of the institutional setting as an outcome of actors exercising activities regulated as part of the institutional statement? → If so, the statement is *regulative* in kind.

- If both criteria are satisfied (e.g., introduction of novel entity as byproduct of enacted agency), test for combination of regulative and constitutive statements under *consideration of the primary and secondary objective/purpose of the statement in the institutional setting as commonly signaled by aim or constitutive function respectively* (e.g., essential focus on regulation of behavior vs. parameterization of institutional setting/action situation – with further operational criteria provided in Table 4).

  Statements are then characterized as regulative-constitutive (where regulation is of primary concern), or constitutive-regulative hybrid (where parameterization is of primary concern), respectively (see Section 4.4 for more details on hybrid institutional statements).

  A special case are statements that can coded according to both syntactic forms. Where dual coding of statements is possible and desirable (i.e., coding as *both* constitutive and regulative statements), statements can be encoded as polymorphic institutional statements (see Section 4.4.4).

  Where a specific preference for regulative or constitutive forms exists based on analytical preferences, this poses challenges to coding reliability, an aspect that can be alleviated by defining a specific *interpretational scope* prior to coding (as described in the following).

Interpretational Scope of Institutional Statement Coding

To establish a reliable and consistent encoding of institutional statements, an important concern is the appropriate characterization of the intended *scope of interpretation* coders should put forth when establishing the structure and type of institutional statements. More specifically, while the encoding of text centers around individual institutional statements as units of analysis, tacit preferences in interpretation can become overt when distinguishing between constitutive and regulative statements.
Using the following set of (stylized) example statements for illustration, we can observe two leading statements of regulative kind, which, in alignment with the general heuristics expressed above, is signaled by the primary focus on behavior regulation. The last statement, in contrast, offers further contextualization for the previous statements.

(1) Organic farming operations must not utilize genetically-modified seeds.
(2) Organic farming operations may not process crops other than the ones specified in Appendix A.
. . .
(10) Paragraphs in this section do not apply to traces of genetically modified material.

Exploring the statements structurally, the last statement does not directly regulate behavior, but instead imposes operational constraints on the preceding statements. Given its broader overarching function and associated capacity for re-configuration, it may be interpreted as *parameterizing in nature*, since it affects the wider institutional setting regulated in the referenced statements that offer specific operational guidance. Such characterization would lead to a qualification of the statement as *constitutive* in kind.
Conversely, however, the statement can be reconstructed in a form that captures the operational character. Absent specific coding instructions (which are discussed in Section 4 onwards), the last statement could be reformulated as

Organic handling operations may not apply paragraphs in this section to traces of genetically modified material.

Following the general heuristics for statement classification, the reformulated statement primarily focuses on behavior regulation, suggesting its characterization as *regulative*.
Without further specification, a potential inconsistent interpretation of statements of such character can lead to reliability challenges, an aspect IG 2.0 seeks to alleviate.
To afford a consistent differentiation that a) establishes reliability, and b) ensures methodological consistency in response to analytical objectives, we differentiate forms of coding by *scope of interpretation*, where interpretation of *narrow scope* emphasizes an immediate interpretational focus on the statement of concern, including its semantics as well as the function it holds with respect to other statements expressed in the policy, or the policy at large; however, semantic linkages of the statement are not resolved beyond the statement itself.
Contrasting the narrow scope, applying a *wide scope* of interpretation expands the focus of analysis by resolving implied or explicit semantic links to other statements, and thereby introduces potential inferences that can affect the interpretation and representation of the analyzed statement both structurally (as showcased above for the reconstruction in regulative form) and semantically (e.g., by drawing deeper inferences draw from and even extend beyond the linked statement). While rooted in the original institutional statement of concern, coding based on wide scope implicitly invokes the interpretation from a systemic perspective, and thereby introduces a variable unit of analysis; this interpretation is necessarily anchored in, but invariably extends beyond, the originating institutional statement.

The choice of interpretational scope is an important consideration and determined as part of the design of the coding exercise, alongside aspects such as pre-coding steps (see Section 3), prior to operational coding. Justifications for either choice of interpretational scope can be based on the analytical objectives, such as an actor-centric perspective that favors the uniform representation in behavioral terms, as opposed to systemic analyses that emphasize conceptual and relational aspects of the institutional system. Justification can further occur on methodological grounds, such as reliability concerns as well as coder experience and contextual knowledge.
Where coders seek encoding that is agnostic of specific analytical biases, such statement can be systematically coded both with narrow and wide interpretational scope, leading to the characterization as *polymorphic institutional statements*, which are discussed at greater detail in Section 4.4.4.

Specific Heuristics for Identifying Institutional Statement Types:

With focus on cases for which the general characteristics are insufficiently precise, Table 4 lists various operational and in part interrelated heuristics that reflect on the distinctive purpose and semantics of individual institutional statements. The heuristics are ordered by specificity. Earlier heuristics are more broadly applicable, whereas later ones offer guidance relevant for more specific cases. Where useful and applicable, the table further includes explanatory support, while drawing links to related heuristics.

Heuristic	**Description**	**Statement Type**	**Related Heuristic(s)**
Function	Is the statement's focal purpose to explicitly define, introduce, or modify an entity (e.g., actor, action, role, object), or otherwise parameterize the analyzed institutional system? → If so, the statement is *constitutive* in kind. Is the statement's focal purpose to compel, restrain, permit, or otherwise assign expectations about, an individual or collective (e.g., corporate) actor's performance of a particular action or set of actions (i.e., regulate behavior)? → If so, the statement is *regulative* in kind.	constitutive or regulative	–
Consequence of Violation	Does the institutional statement refer to an activity the violation of which leads to a consequence that is, if specified, localized or systemic in nature, i.e., does the violation modify, or otherwise *re-parameterize*, the system? → If the consequence is systemic, i.e., signals a modification or re-parameterization of the system (e.g., an entity does not come about), then the statement is *constitutive* in kind. → Otherwise, the statement is *regulative* in kind.	constitutive or regulative	Function

Deontic vs. Non-deontic Modal	Does the statement contain a modal that explicitly characterizes discretionary actions or signals duty as imposed on the responsible actor? → If so, the statement is *regulative* in kind. Does the statement contain a modal that describes the general possibility or necessity of a constitution or modification of an entity described by the action? → If so, the statement is *constitutive* in kind. Example: Oftentimes, statements of such type attach obligations to objects (e.g., *"Access to mediation shall be maintained . . . "*). While these may contextually allow for the inference of the responsible actor, interpreted on statement level the modal 'shall' here signals the epistemic necessity of access to mediation (not an individual's duty to provide it).	constitutive or regulative	Consequence of Violation
Actor/Entity as Target of Conferral	Is the actor or constituted entity referred to in the statement an explicit recipient of a right, role and associated authority (power), or other form of status? → If so, the statement is *constitutive* in kind. Note: There may exist separate corresponding regulative statements expressing specific duties associated with a role or authority, or other actors' complementary duties ensuring the satisfaction of a right. However, the statement of concern is exclusively focused on conferral in terms of rights, authority or other forms of status, and is thus constitutive in kind.	constitutive	Function, Deontic vs. Non-deontic Modal

Table 4: Institutional Statement Type Heuristics

2.6. IG Coding Levels

The IG 2.0 identifies three levels of encoding to provide flexible accommodation of coding necessities based on the complexity of encoded data, as well as the analytical objectives of the coder: **IG Core**, **IG Extended**, and **IG Logico**.

IG Core: IG Core facilitates coding following a fundamental syntactic structure. This level best accommodates comparatively simple institutional statements that largely follow the basic regulative or constitutive structure, along with analytical objectives that involve the statistical assessment of references to the individual components (e.g., distribution of actor, action, object or deontic references).

IG Extended: The next higher level, IG Extended, focuses on capturing the syntactic structure of institutional statements in greater detail (*deep structure*). For regulative statements, this involves the fine-granular encoding of actors and objects, along with complex property relationships. Furthermore, it enables for both regulative and constitutive statements, a detailed encoding of context, such as the characterization of statement dependencies, and categorization based on circumstantial aspects of conditions and constraints (e.g., temporal, spatial, procedural aspects). Choosing to encode on this level may be motivated by the complexity of the encoded institution regulation (e.g., complex statements involving Object-Property Hierarchies (Section 2.3), or extensive statement interdependencies), but also by the analytical objectives, such as the operationalization of the extracted structure in advanced computational models that require the explicit representation of actor properties and context characterization.

IG Logico: The highest level of expressiveness, IG Logico, aims at enabling the analyst to derive more sophisticated understanding of semantic relationships embedded in and among institutional statements based on institutional statement classification across syntactic categories; for example, improved understanding of actor roles, explicit references between statements, as well as inference of actor obligations tacitly expressed in the coded document. As a point of contrast, whereas at the IG Core and IG Extended levels, syntactic classification of institutional statements is a final goal of the encoding exercise, at the IG Logico level the goal is to build on syntactic classification by leveraging this coding toward identification of institutional semantics that relay functional and/or relational information of interest to the institutional analyst.

Shared Assumptions:

All coding levels are backward-compatible, i.e., statements coded at higher levels of expressiveness can be reduced to any lower level of expressiveness. In other words, statement information corresponding to different syntactic components is simply more finely decomposed as one moves from lower to higher levels of expressiveness (e.g., IG Core to Extended), with the effect that moving in the other direction, the coder can simply collapse decomposed information. Methodologically, this accommodates a multi-pass approach towards coding: coding can commence at the lowest level, before being incrementally refined to accommodate syntactic parsing associated with the respective next higher level(s) of expressiveness, and conclude at the desired level of expressiveness set out during study design (which is informed by the nature of the coded document and analytical objectives as discussed above).

Mapping Prerequisites:

The different coding levels make varying use of the concepts highlighted in Section 2 as outlined in Table 5. Concepts specified at the IG Core level, apply to IG Extended and IG Logico, and concepts that apply to IG Extended apply to IG Logico. Statement level nesting applies at all levels. Given the multi-pass coding approach, concepts specified for respective lower levels apply to all higher levels (e.g, statement-level nesting applies to all levels).

Coding Level	Relevant Concepts
IG Core	• Horizontal and vertical nesting (Statement-level nesting) • Activation conditions, execution constraints • Conceptual understanding of Hybrid Institutional Statements (Section 4.4)
IG Extended	• Component-level nesting • Object-Property Hierarchy in regulative statements; equally applies to Constituted Entities/Properties in constitutive statements (both of which may have properties on their own) • Hybrid & Polymorphic Institutional Statements (Section 4.4)

Table 5: Relevant concepts on different coding levels

A high-level overview of the individual levels, along with the discussed objectives is captured in Figure 8, highlighting both the coder perspective as well as analytical perspective. The principles and objectives of the individual codings are discussed in detail in Section 4 in this Codebook; the analytical perspectives are covered elsewhere.[13]

3. Pre-coding Steps

Before discussing the coding of institutional statements in detail, in this section we lay out "pre-coding" steps that relate to familiarization with the institutional setting and document preparation, commencing with general pre-processing, followed by considerations specific to distinctive levels of expressiveness.

3.1. General Steps

1. Familiarization with institutional setting: Prior to embarking on any coding, the institutional analyst should carefully review the institution to be coded. A thorough pre-coding review (e.g., reading) of the institution to be coded is necessary for gaining a high-level understanding of institutional actors, actions, and institutional statement relationships that can be leveraged in the encoding process.

2. Selection of coding platform: One of the first steps the institutional analyst should engage in as she gains familiarity with the institutional setting is identifying the coding platform in which institutional data will be stored. The selection of a coding platform will be informed by the analyst's expectations regarding at which level of expressiveness institutional statements will be encoded, and related assessment of institutional complexity, as certain platforms are better equipped to

[13] A detailed introduction of the levels of expressiveness, their underlying motivation and appropriate analytical techniques are covered in Frantz and Siddiki (2022).

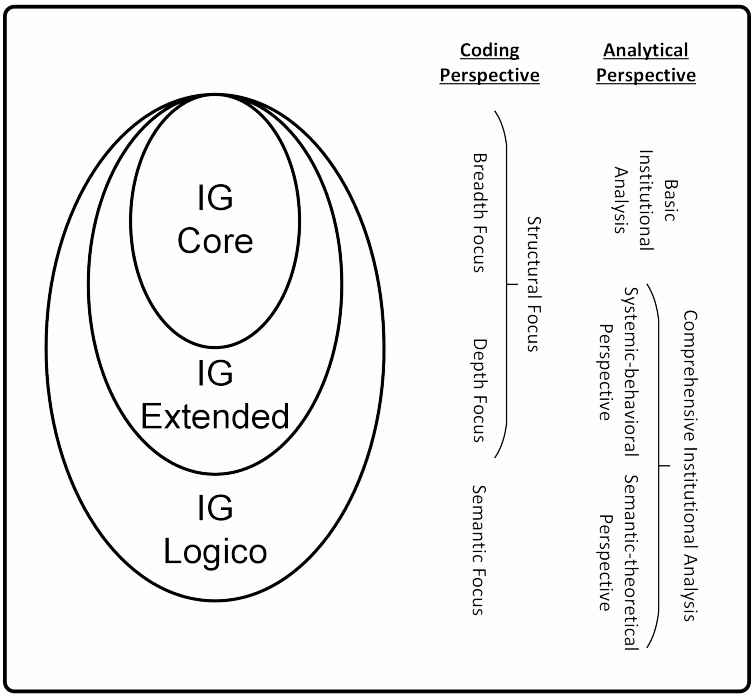

Figure 8: Overview of IG Coding Levels and Associated Objectives

capture institutional statement complexity. The selection of a coding platform will also be informed by the analyst's anticipated usage of institutional data; for example, whether stored data will later be engaged in computational applications. Platforms store data in forms that are more or less computer readable.

3. Initial organization of institutional information: Once the institutional analyst reviews the institution to be coded as part of step 1, she can start to organize its contents. Though variable across jurisdictional setting, policy content is typically organized according to (i) a preamble, that describes the motivation for the policy; (ii) key definitions, that provide descriptions for actors (e.g., *"'Secretary' means the Secretary of Agriculture"*) and other terms (e.g., *"'Prohibited substances' are substances that have been banned by the Dept. of Agriculture for use in organic food production"*) and abbreviations (e.g., *"'NOSB' means National Organic Standards Board"*) that aid in the effective interpretation of policy content; and (ii) Policy instructions organized by topic according to section and subsection headers.

   All three types of content (preambles, definitions, and policy instructions) should be coded. Preambles are likely to be comprised of self-referential statements that can convey the purpose of, or contextualize, the institution under examination more generally as well as regulative and/or constitutive statements. In most cases, definitions are constitutive, and can be useful for encoding institutional statements encountered in a policy document; e.g., when some statement clause references something that is defined in the definitions section of the policy document. This is computationally useful [in the coding process] because it allows the computer to reference particular definitions when certain terms inline. It also allows the computer to link statements that share common definitional information in the analysis process.

   The critical aspect of this coding step is to ensure that the institutional analyst identifies all relevant, codeable information – i.e., all information that is comprised of codeable institutional statements.

4. Verification and pre-processing of institutional statements: Following the identification of candi-

date statements in step 3, the analyst should engage in verification and pre-processing of institutional statements to enable their syntactic decomposition in step 5. Verification in this case means ascertaining that candidate statements accord with defining syntactic and semantic features of regulative and constitutive statements. Principally, this means verifying that statements presumed to be regulative in kind at least contain an Attribute, Aim, and Context, and that statements presumed constitutive in kind at least contain a Constituted Entity, Constitutive Function, and Context component. Institutional statements often do not align with sentences encountered in formal institutions, as a result of writing style (e.g., compound sentences) and punctuation (e.g., bulleted lists). Examples of excerpts of formal institutions that do and do not accord with the definition of regulative and constitutive statements are provided below. Importantly, text that does not classify as institutional statements should be retained and annotated as domain specific background. This information can be useful for institutional interpretation and implication, but has oftentimes informational character, without parameterizing or regulating function per se.

Institutional Statements:

Organic farming is hereby established as a practice regulated under the Department of Agriculture.

The Department of Agriculture shall promulgate regulations governing the practice of organic farming.

Pre-processing in this step also means organizing the content of institutional statements to both remove extraneous content from statements (i.e., punctuation that accompanies statements often reflecting institutional style or organization; for example, roman numerals, bullet points) as well as to begin to arrange statement content to offer additional clarity about how statements are to be coded in step 5. Provided below is text pre-processed to remove extraneous punctuation.

Unprocessed Excerpt

"(a) The producer of an organic livestock operation must establish and maintain year-round livestock living conditions which accommodate the health and natural behavior of animals, including:
(1) Year-round access for all animals to the outdoors, shade, shelter, exercise areas, fresh air, clean water for drinking, and direct sunlight, suitable to the species, its stage of life, the climate, and the environment: Except, that, animals may be temporarily denied access to the outdoors in accordance with §§ 205.239(b) and (c). Yards, feeding pads, and feedlots may be used to provide ruminants with access to the outdoors during the non-grazing season and supplemental feeding during the grazing season . . ."

Pre-processed Text for Institutional Statement Delineation and Punctuation Removal

The producer of an organic livestock operation must establish and maintain year-round livestock living conditions which accommodate the health and natural behavior of animals, including: Year-round access for all animals to the outdoors, shade, shelter, exercise areas, fresh air, clean water for drinking, and direct sunlight, suitable to the species, its stage of life, the climate, and the environment. Except, that, animals may be temporarily denied access to the outdoors in accordance with §§ 205.239(b) and (c). Yards, feeding pads, and feedlots may be used to provide ruminants with access to the outdoors during the non-grazing season and supplemental feeding during the grazing season.

5. Decomposition of institutional statements: Following the verification and preprocessing of institutional statements, the analyst should commence the syntax-based encoding process according to a selected level of expressiveness.

Pre-processing is an optional step in the encoding process but can significantly reduce the time and cognitive load associated with subsequent decomposition. Further, the analyst can choose degrees of pre-processing. More extensive pre-processing is particularly useful for encoding at higher levels of expressiveness. Below are pre-processing guidelines that are useful for encoding at any level of expressiveness, as well as guidelines that are level specific. The level specific guidelines build upon each other, rather than being exclusive, meaning that guidelines applicable for IG Core are relevant for IG Extended and IG Logico, and IG Extended guidelines are applicable for IG Logico. The specific steps outlined here pertain to a manual process. Where tool support, e.g., in the form of an automated parser, is employed, the steps may deviate (specifically for the aspects that are automated).

Generally, pre-processing, particularly of a more extensive kind, will be easier for analysts with greater familiarity with the IG, as they will likely be able to detect statement structure, components, and relations without engaging in even a preliminary decomposition of statements. Some degree of formal decomposition might be required of analysts less familiar with the IG to be able to discern these.

General pre-processing guidelines:

- Data cleaning: dealing with extraneous punctuation, fixing typos
- Delineation of text into institutional statements
- Delineation of nested statements (e.g., Or else statements)
- Preliminary classification of institutional statements as regulative, constitutive, regulative or-else, constitutive or-else, or combinations thereof (see Section 2.5 and Section 4.4)
- Preliminary organization of statements by identifiers capturing the institutional structure/ordering of institutional statements in the document. These identifiers can uniquely identify institutional statements, and statement linkages (i.e., nested statements), as well as policy sections or parts that can facilitate understanding of statement context and cross-statement or policy references.

3.2. Pre-processing Guidelines for IG Core

- To accommodate encoding of institutional statements at the core level, preliminary decomposition of institutional statements to account for multiple values within individual syntactic fields should be entertained during the preprocessing of institutional documents. Institutional statements often contain multiple Attributes, Aims, and/or Objects. For example:

  "The producer of an organic livestock operation must establish and maintain year-round livestock living conditions which accommodate the health and natural behavior of animals."

  This statement contains multiple Aims, *"establish"* and *"maintain."* Neither of these Aims is associated with unique values in other syntactic fields, and therefore, they can both be captured within a single institutional statement. However, downstream coding is facilitated by capturing multiple values individually within separate statements. With the recommended decomposition, the statement above is reflected as two:[14]

  "The producer of an organic livestock operation must establish year-round livestock living conditions which accommodate the health and natural behavior of animals."

  and

  "The producer of an organic livestock operation must maintain year-round livestock living conditions which accommodate the health and natural behavior of animals."

[14] The decomposition of *component-level combinations*, as identified here, is further discussed in Section 4.2.

- In preprocessing institutional statements for coding at the IG Core level, the analyst may consider reformulating statements into active form (where statements are originally captured in passive form while being careful to retain statement meaning from an institutional perspective). Note that conversion of statements from passive to active form typically requires some implication of values according with different syntactic fields. For example, the passive statement *"Notifications of compliance must be sent to farmers within 30 days of facility inspections"* converts to *"[Certifier] must send farmers notifications of compliance within 30 days of facility inspection,"* prompting the implication of *"Certifier"* as the relevant Attribute, or actor in charge of performing action. This implication requires understanding of institutional context obtained through step 1, as well potentially of the identification of a convention for notating implied information, such as for example, the use of brackets ([ ]) in the example included here.

- Where actions or actors are implied, those are inferred from the context and additionally specified as part of the coding in terms of the institutional statement structure. While found across a wide range of statements, this is commonly necessary in the context of statement combinations (combination of two actions performed by the same actor). The same applies to implied logical relationships (e.g., AND, OR, XOR).

- When facing complex sentence structures, statements should be thought of in terms of sequentially applied actions. For example, if statements report outcomes of actions without making reference to such actions, the coder should reconstruct the action sequence leading to such outcome in terms of institutional statements (see Section 2.1).

- At this stage, the analyst should flag additional semantic information that she wishes to capture in the syntactic decomposition of statements and associated label.

3.3. Pre-processing Guidelines for IG Extended

Additional pre-processing of institutional statements to accommodate their downstream coding at the IG Extended level involves some preliminary characterization of institutional statement linkages, particularly to capture action sequences. The coding in IG Extended affords richer decomposition of institutional statements into action sequences. Composite actions are often represented as exemplified in the following: *"When an inspection of an accredited certifying agent by the Program Manager reveals any noncompliance with the Act or regulations in this part, a written notification of noncompliance shall be sent to the certifying agent."*, where *"When an inspection of an accredited certifying agent by the Program Manager reveals any noncompliance with the Act or regulations in this part"* represents a conditional clause that does not overtly reflect an institutional statement due to the expression of actions in terms of nouns (conceptual reification). From a semantic perspective, this clause captures two linked action statements, namely the fact that a *"Program Manager inspects accredited certifying agents"* and that the *"Program Manager reveals non-compliance in this process"*. Retaining the essential institutional semantics, the original statement can thus be rewritten as (with inference of implied components) *"When Program Manager inspects accredited certifying agents and [the Program Manager] reveals non-compliance in this process, the Program Manager shall send a written notification of noncompliance to the certifying agent."* While possible to identify as part of the coding process, a specific consideration of IG Extended is to identify such action sequences, and potentially offload their reconstruction to the pre-processing process, the decision on which is subject to the analytical objective, nature of the coded policy, as well as coder background, used coding platform, and opportunities for automation of the reconstruction.

3.4. Pre-processing Guidelines for IG Logico

Additional pre-processing of institutional statements to accommodate their downstream coding at the IG Logico level involves some more extensive, albeit still preliminary, capturing of inter-statement relationships and embedded actions within institutional statements that the analyst may want to fully reconstruct in terms of institutional statements during the encoding process. Inter-statement relationships are often indicated with referential clauses that embed within institutional statements. In policy documents, these references are often to statement collections, the coded document at large, or third-party documents. The example statements below include types of referential clauses that embed within statements that the institutional analyst may wish to capture during the pre-processing phase. Embedded actions can generally be thought of as actions ancillary to that represented in the focal action of an institutional statement (reflected in the Aim or Constitutive Function for regulative and constitutive statements, respectively). Embedded actions, while referenced, are incompletely described. However, reconstruction of these embedded actions into complete institutional statements can afford a more complete depiction and understanding of the institutional domain being evaluated. The particular reconstruction the analyst pursues will depend on her analytical objectives, but also the specific types of institutional functions [constitutive (see Section 5.6) or regulative (see Section 5.5)] she wants to capture within explicit and reconstructed statements. In the pre-processing phase, the analyst might consider constructing a dictionary of terms they observe through preliminary review of institutional statements that signal different institutional functions. This prompts consideration of how different types of observed actions might link to different institutional functions of interest to the analyst. The example statements below include embedded actions that can be fully reconstructed during the encoding process.

Example Statements with Referential Clauses

Any operation that: (1) Knowingly sells or labels a product as organic, except in accordance with the Act, shall be subject to a civil penalty of not more than 3.91(b)(1)(xxxvii) of this title per violation.

A production or handling operation that sells agricultural products as "organic" but whose gross agricultural income from organic sales totals $5,000 or less annually is exempt from certification under subpart E of this part.

Any agricultural product that is sold, labeled, or represented as "100 percent organic," "organic," or "made with organic (specified ingredients or food group(s))" must be: (a) Produced in accordance with the requirements specified in §205.101 or §§205.202 through 205.207 or §§205.236 through 205.240 and all other applicable requirements of part 205.

Example Statements with Embedded Actions

A handler of organic products may use information provided by the certified operation to determine percentage of organic ingredients.

→ Embedded action: information provided by the certified operation (i.e., provision of information by certified operation)

A certifying agent must provide an applicant with a copy of the on-site inspection report, as approved by the certifying agent, for any on-site inspection performed.

→ Embedded action: approved by the certifying agent (i.e., approval of report by certifying agent)

A certifying agent whose accreditation is suspended by the Secretary under this section may at any time submit a request for reinstatement of its accreditation.

→ Embedded action: accreditation is suspended by the Secretary (i.e., accreditation suspension by Secretary)

The Program Manager may initiate suspension proceedings against a certified operation, when a certifying agent fails to take appropriate action to enforce the Act.

→ Embedded action: certifying agent fails to take appropriate action to enforce the Act (i.e., failure to act by certifying agent). In this case, the failure to act is also signalling a violation, or non-compliance of some kind, which could be marked as an institutional function of interest and even used in downstream coding toward the reconstruction of both direct statements and their logical inverses.

Concluding this overview on activities relevant for the design and planning of the coding exercise, the following section turns to the coding of institutional statements.

4. Coding Guidelines

In this section, we provide guidelines for coding institutional statements at the IG Core, IG Extended, and IG Logico Levels of Expressiveness. Following the specification of utilized syntax, we specify encoding principles for regulative and constitutive statements.

4.1. Coding Symbols & Syntax

4.1.1. Coding Symbols

The syntactic coding for examples in the remainder of this document relies on specific symbols, whose function depends on the applied context, i.e., grammar component vs. institutional statement, and respective coding level (IG Core, IG Extended, IG Logico). An overview of all symbols along with application context, minimum level of applicable encoding, description and examples is provided in Table 6. The specific syntactic notation referenced in this codebook (introduced in (Frantz & Siddiki, 2022)) is referred to as *IG Script*[15], a notation that manages the trade-off between computational representation on the one hand, and retaining general human readability of the encoded text on the other. Before exploring the encoding in practice more generally, initially all commonly referenced symbols are introduced, alongside specific features that are applicable in particular levels of expressiveness.

Throughout the remainder of this section we use color coding to signal the association/introduction of specific symbols for syntactic components or features with specific levels of expressiveness (as introduced in Section 2.6). Symbols associated with IG Core features are held in blue for regulative statements, and in purple for constitutive statements (specifically relevant from Table 10 onward). Symbols associated with IG Extended are held in green, and features associated with IG Logico are called out in orange. Symbols of general relevance across levels and regulative and constitutive statements (e.g., parentheses to signal precedence or nesting) are held in bold **black**. Naturally, the examples draw on features not introduced to this stage, but offer an illustration of the representations used throughout the subsequent guidelines.

Symbol/ Symbol Pairs	**Coding Context**	**Lowest applicable Level of Expressiveness & Statement Type**	**Description**	**Example**
( )	Component	IG Core, Regulative & Constitutive Statements	Component classification: The characterization of an expression as a component type is signaled through parentheses that contain the component type.	**A***(Certifier)* . . . , where **A** identifies the *certifier* as an attribute in a given institutional statement.

[15] A more detailed discussion of all syntactic features of IG Script are provided under https://github.com/chrfrantz/IG-Parser?tab=readme-ov-file#ig-script.

			Where used to combine individual components of the same type (in addition to annotation or the indication of statement combinations), parentheses signal component-level combinations (with logical operators discussed at the end of this table). Inner parentheses are only needed to indicate precedence for three or more elements linked by different logical operators.	*Attendees must not* **I(***eat* **[AND]** *drink***)** *on the train.* Note that this is equivalent to 'Attendees must not **I((**eat **[AND]** drink**))** on the train.', but the additional parentheses can be omitted, unless relevant to indicate precedence (e.g., combinations of three or more actions with differing logical operators), such as **I(**firstAction **[AND]** **(**secondAction **[OR]** thirdAction**))**.
[ ]	Component	IG Core, Regulative & Constitutive Statements	Tacit components: The explicit specification of implied components (e.g., actor(s)) is signaled with brackets.	*They [***A***(farmers)] must comply with the certification regulation . . . , where [***A***(farmers)] characterizes the inferred actor.*
[ ]			Where component annotations (e.g., animate, inanimate) are used, those are embedded in square brackets following the component symbol, but preceding the component content.	**A[type=animate]***(Certifier)* . . . , where **A** identifies the *certifier* as an attribute in a given institutional statement, and *animate* as a semantic annotation, a feature of specific relevance in IG Extended (applied in Section 4.2.2) and IG Logico (from Section 4.2.3 onward).
{ }	Statement	IG Core, Regulative & Constitutive Statements	Horizontally nested statements are represented using surrounding parentheses to emphasise the precedence of combined individual statements.	{ *stmt* **[AND]** *stmt* }; { *stmt* **[AND]** { *stmt* **[OR]** *stmt* }}, where stmt represents an institutional statement combined with other institutional statements using logical operators (**[AND]**, **[OR]**, **[XOR]**, and potentially **[NOT]**) – more details on logical operators below.

			Vertically nested statements are represented using braces that embed the respective consequential statement	*stmt1*{*stmt2*}, where *stmt1* represents a monitored statement, and *stmt2* the corresponding consequential statement.
{ }	Component	IG Extended, Regulative & Constitutive Statements	Component-level nesting is represented by embedding the component-substituting nested institutional statement in braces. In the case of component-level nesting, the component type specification follows the embedded nested statement.	**A***(Certifier)* **I***(believes)* **Bdir**{**A***(farmer)* **I***(violates)* **Bdir***(code of conduct)*} In this example, the direct object (**Bdir**) of a given institutional statement is substituted with another institutional statement.
A	Component	IG Core, Regulative Statements	Identifies the preceding expression as Attributes component.[2]	**A***(Certifier)*
I	Component	IG Core, Regulative Statements	Identifies the preceding expression as aim component.	**A***(Certifier)* **I***(monitors)* **Bdir***(farmers).*
Bdir	Component	IG Core, Regulative Statements	Identifies the preceding expression as direct object component.	**A***(Certifier)* **I***(administers)* **Bdir***(certifications).*
Bind	Component	IG Core, Regulative Statements	Identifies the preceding expression as indirect object component.	**A***(Certifier)* **I***(registers)* **Bdir***(certification)* *for* **Bind***(organic farmer).*
D	Component	IG Core, Regulative Statements	Identifies the preceding expression as a deontic modal representing duty.	**A***(Certifier)* **D***(must)* **I***(monitor)* **Bdir***(farmers).*

Cac/**Cac**	Component	IG Core, Regulative & Constitutive Statements	Identifies the preceding expression as an activation condition component.	Regulative: **Cac***(Upon accreditation)* **A***(certifier)* **D***(must)* **I***(monitor)* **Bdir***(farmers).* Constitutive: **Cac***(From 1st January onwards),* **E***(Council)* **M***(shall)* **F***(include)* **P***(organic farming representatives)* **Cex***(to review chemical allowances within organic food production standards).*
Cex/**Cex**	Component	IG Core, Regulative & Constitutive Statements	Identifies the preceding expression as an execution constraint component.	Regulative: **A***(Certifier)* **D***(must)* **I***(monitor)* **Bdir***(farmers)* **Cex***(at any time).* Constitutive: **Cac***(From 1st January onwards),* **E***(Council)* **M***(shall)* **F***(include)* **P***(organic farming representatives)* **Cex***(to review adherence with food production standards).*
E	Component	IG Core, Constitutive Statements	Identifies the preceding expression as constituted entity	**Cac***(From 1st January onwards),* **E***(Council)* **M***(shall)* **F***(include)* **P***(organic farming representatives)* **Cex***(to review chemical allowances within organic food production standards).*
P	Component	IG Core, Constitutive Statements	Identifies the preceding expression as constituting property	**Cac***(From 1st January onwards),* **E***(Council)* **M***(shall)* **F***(include)* **P***(organic farming representatives)* **Cex***(to review chemical allowances within organic food production standards).*

F	Component	IG Core, Constitutive Statements	Identifies the preceding expression as constitutive function	**Cac***(From 1st January onwards),* **E***(Council)* **M***(shall)* **F***(include)* **P***(organic farming representatives)* **Cex***(to review allowances within organic food production standards).*
M	Component	IG Core, Constitutive Statements	Identifies the preceding expression as a modal representing existential necessity (in contrast to duty or permission in regulative statements).	**Cac***(From 1st January onwards),* **E***(Council)* **M***(shall)* **F***(be responsible)* **Cex***(for adherence with food production standards).* Alternative example: **Cac***(From January 1st onward), there* **M***(shall)* **F***(be) a* **E***(National Organic Standards Advisory Council)* **P***(within the Department of Agriculture).*

p/ **p**	Attributes, Object, Entity and Property components	IG Core, Regulative & Constitutive Statements	Identifies properties of attributes and objects respective. The *prop* symbol is used in conjunction with the respective component identifier. If coding on IG Extended level, where multiple properties for a given component exist, they receive a numeric index suffix. IG Extended further supports the explicit encoding of object and property hierarchies. Where multiple levels of object/properties exist in the property hierarchy, those are contextualized with the objects/properties they refer to (i.e., they are appended to the component specification). Further details on property coding are provided in Table 8 and illustrated below.	IG Core: Regulative: **A**,**p***(Certified organic)* **A***(farmers)* **D***(must)* **I***(respond)* *to* **Bdir**,**p***(formal)* **Bdir***(certification requirements).* Constitutive: *The* **E***(Council)* **F***(consists of)* **P**,**p***(elected)* **P***(officials)* **P**,**p***(resident in the electorate).* IG Extended: Regulative: **A**,**p1***(Certified)* **A**,**p2***(organic)* **A***(farmers)* **D***(must)* **I***(respond)* *to* **Bdir**,**p1***(formal)* **Bdir***(certification requirements).* Constitutive: *The* **E***(Council)* **F***(consists of)* **P**,**p1***(elected)* **P***(officials)* **P**,**p2***(resident in the electorate).*

p/ p (ctd.)	Component	IG Extended, Regulative & Constitutive Statements	The example on the right highlights a complex object hierarchy structure previously discussed in the context of the Object-Property Hierarchy (Section 2.3). As mentioned above, where multiple objects on a given hierarchy level exist, they are uniquely identified with a numeric index (e.g., **B1**, **B2**, etc.). Where multiple properties on a given hierarchy level exist (where properties can be objects by themselves), they are uniquely identified with a numeric index (e.g., **p1**, **p2**, etc.). Where a single property applies to multiple properties, references to both objects/properties are maintained on this property (separated by semicolon).	... **B1**,**p1**,**p**;**B1**,**p2**,**p***(proposed)* **B1**,**p1***(suspension)* *or* **B1**,**p2***(revocation)* *of* **B1***(certification)* ...
[AND], **[OR]**, **[XOR]**, **[NOT]**	Statement, Component	IG Core, Regulative & Constitutive Statements	The logical operators identify the relationship between statement and/or components as either conjunction (**[AND]**), inclusive disjunction (**[OR]**), or exclusive disjunction (**[XOR]**), as well as any combinations thereof. Where negation is involved, the **[NOT]** operator is used (e.g., in deontics: must not; combination of exceptions: **[NOT]** (option 1 **[AND]** option 2)).	**A***(Certifiers)* **D***(must)* **I**(*review* **[AND]** *assess*) **Bdir***(applications).* **A***(Certifiers)* **D***(must)* **I**(*review* **[AND]** (*approve* **[XOR]** *reject*)) **Bdir***(applications).* **A***(Certifiers)* **D***(must)* **[NOT]** **I***(review)* **Bdir***(applications)* *by* **Bdir,p***(offenders).*

Table 6: Symbol Reference for IG Coding as applied in this document

4.1.2. Coding Syntax

Following the intuitive introduction of the individual symbols annotating particular features of an institutional statement (e.g., actors, actions), at this stage, we will highlight selected syntactic principles of IG Script more explicitly, since these will be referenced specifically in the more advanced coding examples. The base syntax of IG Script is as follows:

*componentSymbol***(** *. . . encoded content . . .* **)**

. . . where *componentSymbol* is a symbolic reference to the component of choice as introduced in the previous Section 4.1.1 (e.g., **A***(actor)*).

Where component values are combined as part of horizontal nesting, the linkage and the scope of the linkage is explicitly captured by parentheses (e.g., to indicate elements that apply to both linked values):

componentSymbol(shared value **(***left value* **[***logicalOperator***]** *right value***)** *shared value)*

Applied operationally, this can be exemplified as **I***(to* **(***revise* **[AND]** *resubmit***)***)*

All relational logical operators can be applied in this fashion, including a nested representation, in which additional parentheses indicate precedence in the case of different logical operators, such as exemplified below:

I((*revise* **[AND]** *resubmit***) [OR]** *revoke***)**

For logical operators of the same kind the indication of precedence can be omitted.

I(*revise* **[OR]** *resubmit* **[OR]** *revoke***)**

The logical operator **[NOT]** plays a particular value, and can annotate individual components as well as statements as a whole.

Taking the following example,

A*(Author)* **D***(must* **[NOT]***)* **I***(review)* **Bdir**,**p***(own)* **Bdir***(paper).*

the operator can invert a single component value.

Statements can be scoped using braces ({, }), e.g., to delimit multiple statements, or to reflect nesting structures. An example for a scope statement is

{**A***(Actor)* **D***(must)* **I***(conform)* **Bdir***(with policy).*}

Augmenting this statement with the inversion operator (where of analytical value) leads to the negated interpretation of the statement entirely:

{**A***(Actor)* **D***(must)* **I***(conform)* **Bdir***(with policy)* **[NOT]**}

A central feature linked to the use of braces is the concept of vertical nesting and component-level nesting, i.e., the substitution of a given component of an institutional statement with the syntactic form of an institutional statement (e.g., preconditions expressed in terms of institutional state characterizations). Where components are augmented with braces instead of parentheses, this signals component-level nesting.

In the case of vertical nesting, the **OR ELSE** (syntactically represented as **O**) is essentially substituted with an entire statement (identifying the consequences of violating the monitored statement), as shown in the following example:

A(*citizen*) **D**(*must*) **I**(*conform*) *with* **Bdir**(*policy*) **O**{**A**(*official*) **D**(*may*) **I**(*sanction*) **Bdir**(*citizen*) **Cex**(*immediately*)}.

Where nesting occurs on other components, this is annotated in equivalent form. Component-level nesting is commonly found in the context of the activation condition by expressing preconditions for a particular activity in the same structure as the activity itself. This is exemplified in the following:

A(*Official*) **D**(*must*) **I**(*impose*) **Bdir**(*fine*) **Cac**{**A**(*citizen*) **I**(*violates*) **Bdir**(*rule*)}.

Implied component values are indicated with brackets surrounding the inferred value. For example, the observed statement "they must comply" may be contextually interpreted as "citizens must comply". To make this inference explicit, it can be encoded as follows:

A([*citizens*]) **D**(*must*) **I**(*comply*).

A common observation in the context of complex statements is the unique linkage between particular components and associated properties, an aspect related to the Object-Property Hierarchy (Section 2.3). To indicate the direct linkage, the encoding relies on suffices associated with particular components in order to identify the linkage.

A(*Official*) **D**(*must*) **I**(*impose*) **Bdir1**,**p**(*monetary*) **Bdir1**(*fine*)
Cac{**A**(*citizen*) **I**(*violates*) **Bdir**(*rule*)}.

A feature referenced in the context of IG Logico is the augmentation of the encoded institutional statements with semantic annotations that facilitate epistemological linkages to concepts drawn from theory of interest, or general categorizations introduced as part of the taxonomies discussed in Section 5. Semantic annotations are captured in squared brackets following the component symbol, but preceding the parentheses scoping the component content specification. The following example showcases the use of semantic annotations:

A[**type=actor**](*Official*) **D**[**stringency=prescription**](*must*) **I**[**act=administer**](*impose*)
Bdir1,**p**[**prop=quals**](*monetary*) **Bdir1**[**object=sanction**](*fine*).

Where annotations pertain to the statement entirely, the specification precedes the statement scope as shown below (here exemplified based on the qualification of the statement as regulative):

[**regulativeStatement**]{**A**[**type=actor**](*Official*) **D**[**stringency=prescription**](*must*)
I[**act=administer**](*impose*) **Bdir1**,**p**[**prop=quals**](*monetary*) **Bdir1**[**object=sanction**](*fine*).}

4.2. Regulative Statement Coding

The base syntax of regulative statements (as shown in Figure 9) consist of necessary (in solid boxes) and sufficient components (in dashed boxes) as defined in Table 2 using the symbols introduced in Table 6. The figure organizes feature refinements across different levels of expressiveness.

Motivating the principal coding of regulative statements (without further elaboration at this stage), a stylized atomic regulative statement, coded in shorthand form, thus reads:

A,**p**(*Certified*) **A**(*farmers*) **D**(*must*) **Cex**(*strictly*) **I**(*adhere*) *to* **Bdir**(*organic farming practices*)
Cac(*following their certification*).

The following Tables 7 to 9 provide detailed coding guidelines for regulative statements for all individual components, organized by level of expressiveness, starting with IG Core through IG Logico,

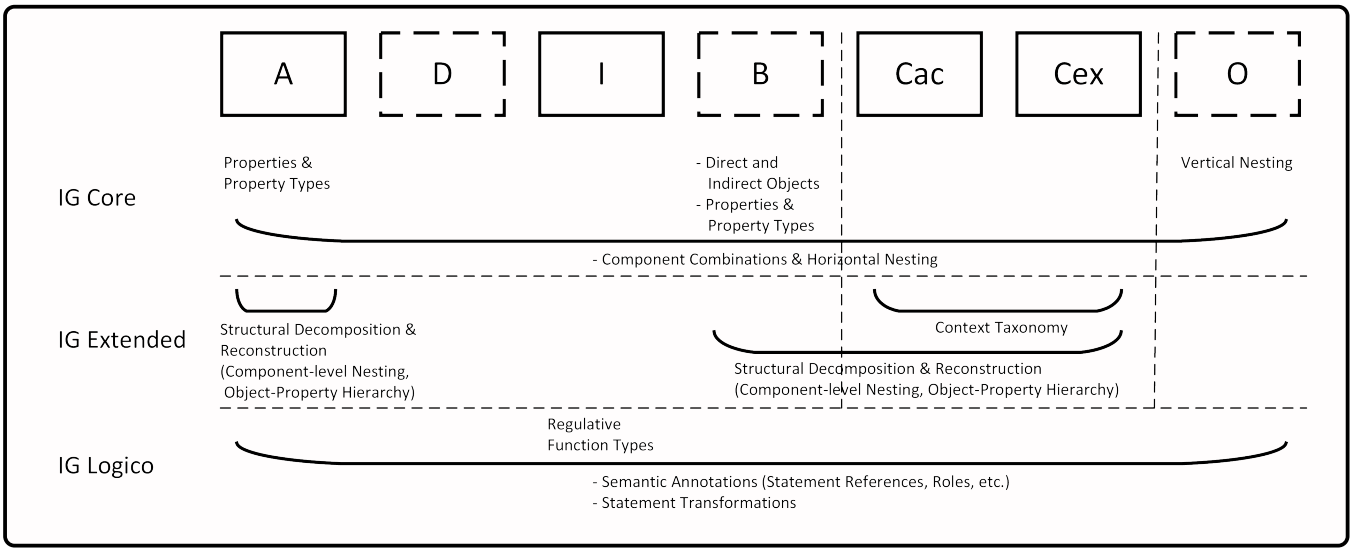

Figure 9: Syntax and Features of Regulative Statements by Level of Expressiveness

leveraging the notation convention captured in Section 4.1. As the institutional analyst commences coding at any level of expressiveness, the following general coding principles should be entertained.

- Where possible, the targeted level of encoding should be clarified at the beginning (see Section 2.6).
- The coder should acquaint oneself with the concepts relevant for the target level of encoding (see Section 2.6), as well as coding conventions, such as applied notation. A specific concern discussed as part of the planning process is the handling of absent syntactic components that are either implied or can be inferred from policy texts.
- Recall that coding can (but does not have to) occur iteratively, starting at one level that prompts less granular syntactic expressiveness (e.g., IG Core) moving with a subsequent coding pass to another level that prompts more granular coding (e.g., IG Extended). However, the specific approach is subject to coder experience, and of course available tool support, all of which should be considered in the planning phase (see Section 3 for further considerations).

4.2.1. IG Core Coding of Regulative Statements

Level of Expressiveness: IG Core

IG Core enables basic, structural analysis of institutional statements. Encoding at this level is designed to be human readable and moderately comprehensive in the detail with which syntactic properties of institutional statements are captured.

Syntactic Component	Treatment of Syntactic Components by Level of Encoding	Relevant Examples	Complete Syntactic Classification of Examples
Attributes	The encoding of Attributes, which can include an animate actor (individual or organizational) only or an animate actor and a property of this actor, differentiates between actor (Attributes) and actor property (Attribute Properties). An important decision to establish reliability is a shared understanding of the level of decomposition of Attributes and their corresponding properties performed as part of the coding process (or even pre-coding process), an aspect that relies on the domain, context, and nature of the coded policy. The example on the right illustrates the principles of decomposition assuming the most fine-grained possible decomposition. Guidelines to determine the appropriate level of decomposition are provided below.	Example statement: *Certified farmer must submit an organic system plan annually.* Attributes encoding: Attributes = *certified farmer* Actor = *farmer* Actor property = *certified*	**A**,**p***(Certified)* **A***(farmer)* **D***(must)* **I***(submit) an* **Bdir***(organic systems plan)* **Cex***(annually).*

Attributes (ctd.)

Deciding on the level of decomposition into Attributes and corresponding Attribute Properties:

The guidelines provided below equally apply to Object and Object Properties; Constituted Entity and Constituted Entity Properties; Constituting Properties and Constituting Properties Properties.

A general challenge across multiple components of the IG is the identification of appropriate heuristics for the decomposition of actor descriptors into Attribute and corresponding properties, and likewise Objects and Object Properties, etc. To support the study design, we propose a set of guidelines to inform the choice of decomposition levels as appropriate for a given study.

General note: Named entities (e.g., United States Department of Agriculture) are *not* decomposed into component properties, i.e., such terms are always annotated as Attributes, Object, Constituted Entity or Constituting Properties, but not their corresponding properties (i.e., Attributes Properties, Object Properties, Constituted Entity Properties, Constituting Properties Properties).

Guidelines for determining level of decomposition (in order of consideration):

- Policy documents often contain a 'Definitions' article or section, in which concepts of relevance to the policy are specified. Where this is the case, these definitions represent the desirable level of decomposition, i.e., if 'Organic farmer' is defined as part of the definitions, no further decomposition is required, unless analytical objectives motivate further decomposition (see next item).
- Where absent, the level of decomposition is contingent on the policy focus and context. If, for example, a regulation differentiates between organic and non-organic farmers, the decomposition into descriptor and associated properties is useful, in which both 'types' of farmers are distinguished by the properties. If, however, the regulation is exclusively concerned with organic farmers, the decomposition is of little analytical value at this level of encoding.
- Absent any other guidelines, decomposition should be performed on the most fine-granular level.

Object	The encoding of Objects identifies Direct Objects specified within institutional statements, along with their respective properties. Where statements are comprised of both direct and indirect object, it also entails the explicit identification of indirect objects, i.e., objects that are affected by the application of the aim to direct objects, e.g., which the action is targeted to. Objects can be concrete or abstract entities. Concrete entities can be animate (e.g., farmers) or inanimate (e.g., notification) in kind. Abstract entities (e.g., beliefs, goals, observations) are generally expressed in terms of statements of facts or institutional statements. The coding of such statements (*component-level nesting*) is discussed under IG Extended (see Table 8). Rules: • Identify object identifier • Identify object properties • Repeat for the indirect object Note: The separation of Objects and Object properties operates identical to the decomposition of Attributes and Attributes properties. See corresponding guidelines provided under the Attributes component specification (see here).	Example statement: *Organic certifier must send farmer notification of compliance within 30 days of inspection.* Direct Object = *notification of compliance* Indirect Object = *farmer* Object property = *organic* Note: The interpretation of property depends on the encoded policy (and, of course, the coder's analytical objectives). If a policy on organic farming exclusively refers to organic farmers as a proper noun, organic is not an attribute property, but part of the attribute; if the policy differentiates between organic and other types of farmers, capturing the specific characterisation as property is suggested. For the example here we highlight the second pathway.	**A**,**p(***Organic)* **A***(certifier)* **D***(must)* **I***(send)* **Bind***(farmer)* **Bdir***(notification of compliance)* **Cex***(within thirty days of inspection).*

Object (ctd.)

Potential pitfalls:

- *Objects vs. Constraints*: The introduction of the indirect object offers the benefit of capturing the functional interdependence of objects and implied directionality. This directionality is sometimes explicitly reflected, as in the following statement. When encountering such statements, the coder should be careful to not mischaracterize the indirect object preceded by a preposition as context. Example: "Parents must take children to school.", "school" is sensibly resolved as indirect object, but based on its prepositional embedding, could be mislabeled as constraint. For the characterization, we thus require an initial consideration of clauses containing objects (other than the direct object) as indirect objects, before characterizing those as a contextual descriptor (which primarily make reference to the contextual embedding of actions).

- *Object Properties vs. Constraints*: Differentiation between the object property and execution constraint can also sometimes be challenging in the encoding. When such confusion arises, the coder should ask herself to reflect on whether the statement words or clauses in question are qualifying the object or qualifying the action of the statement. Take the following statement for example that illustrates the referenced potential confusion: "Certifiers shall perform audits on product stock two times per year." The clause that may potentially give rise to confusion is "on product stock." This could be confused as an execution constraint relating to purpose, but in fact it describes the type of audit to be performed.

Aim

The encoding identifies the focal action of the statement.

Example statement:

Organic certifier must send farmer notification of compliance.

Aim = send

A,**p***(Organic)* **A***(certifier)* **D***(must)* **I***(send)* **Bind***(farmer)* **Bdir***(notification of compliance).*

Deontic	The encoding identifies the prescriptive operator that indicates whether the Aim (i.e., action) of the statement is required, allowed, or forbidden. Common Deontics indicating varying levels of prescriptive force include *must*, *may*, and *must not*.	Example statement: *Organic certifier must send farmer notification of compliance.* Deontic = must	**A**,**p***(Organic)* **A***(certifier)* **D***(must)* **I***(send)* **Bind***(farmer)* **Bdir***(notification of compliance).*
Context	The encoding identifies the Context of the institutional statement. The encoding differentiates between *"Activation Conditions,"* which are contextual clauses that specify preconditions under which the Aim is expected to occur or not occur, and *"Execution Constraints,"* which are contextual descriptors that qualify the Aim by assigning in relation to it temporal, spatial, procedural, and/or other constraining parameters.	Example statement: *Upon entrance into agreement with organic farmer to serve as his/her certifying agent, organic certifier must inspect farmer's operation within 60 days.* Context clauses: *Upon entrance into agreement with organic farmer to serve as his/her certifying agent; within 60 days.* Context encoding: Activation Condition: *Upon entrance into agreement with organic farmer to serve as his/her certifying agent* Execution Constraint: *within 60 days* Note: While coding the essential aspects of the statement, IG Core is limited with respect to capturing the nested complexity in the activation condition. We will revisit this statement in the context of IG Extended coding.	**Cac***(Upon entrance into agreement with organic farmer to serve as his/her certifying agent),* **A***(organic certifier)* **D***(must)* **I***(inspect)* **Bdir***(farmer's operation)* **Cex***(within 60 days).*

Or else

The encoding of Or else statements identifies consequences (e.g., payoffs) of compliance/non-compliance with institutional statements, or the conduct of Aim (i.e., activities) assigned to specific Attributes (i.e., actors) in institutional statements. The encoding captures these consequences, which generally take the form of regulative institutional statements that nest from the 'non-Or else' (monitored) statements.

Sometimes, the statements on which Or else, or consequential statements, nest convey a combination of monitoring activity and the associated payoff attached to outcomes of monitoring actions.

The encoding of Or else statements accommodates both vertical and horizontal nesting. Vertical nesting is applicable when there is one payoff activity that is specified within a distinct institutional statement as a consequence of an action indicated in another institutional statement. Horizontal nesting is applicable when there are two or more payoff activities that can be pursued as consequences of an action indicated in another institutional statement.

Example:
Certified organic farmers must not apply synthetic chemicals to crops at any time once organic certification is conferred, or else certifier will revoke certification from farmer.

Or else clause comprising statement: *or else certifier will revoke certification from farmer*

Or else statement nests on: *certified organic farmers must not apply synthetic chemicals to crops at any any time once organic certification is conferred*

The above encoding exemplifies vertical nesting. The following example, which is an extension of the above, exemplifies horizontal nesting within a vertically nested statement.

Example statement:
Certified organic farmers must not apply synthetic chemicals to crops at any time once organic certification is conferred, or else certifier will revoke certification from farmer or fine farmer.

In the following example, horizontal nesting is signaled using parentheses (**(** and **)**) around statements (as opposed to individual components), and vertical nesting is expressed using braces on the **OR ELSE** component ({ and }).

Vertical nesting:
A,**p1***(Certified)* **A**,**p2***(organic)* **A***(farmers)* **D***(must not)* **I***(apply)* **Bdir***(synthetic chemicals)* *to* **Bind***(crops)* **Cex***(at any time)* **Cac***(once organic certification is conferred),* **O**{**A***(certifier)* **D***(will)* **I***(revoke)* **Bdir***(certification)* *from* **Bind***(farmer)*}.

Or else (ctd.)

These statement combinations can signal

- alternative exclusive action options – XORs – (e.g., either suspending XOR revoking the certification),
- inclusive action options – ORs – (e.g., sanctions apply if a driver is caught speeding AND/OR on the phone) or
- co occurring action options – ANDs – (e.g., fining a transgression AND reporting to authorities).

Or else clauses comprising statements:

or else certifier will fine farmer or revoke certification from farmer.

Vertical and horizontal Or else nesting encoding:

Certified organic farmers must not apply synthetic chemicals to crops at any time once organic certification is conferred

[Vertical nesting]

or else certifier will revoke certification from farmer

[Horizontal nesting]

[XOR] (signaling exclusive or)

Or else certifier will fine farmer

Horizontal nesting within vertically-nested statement:

A,**p1**(*Certified*) **A**,**p2**(*organic*) **A**(*farmers*) **D**(*must not*) **I**(*apply*) **Bdir**(*synthetic chemicals*) *to* **Bind**(*crops*) **Cex**(*at any time*) **Cac**(*once organic certification is conferred*), **O** {{**A**(*certifier*) **D**(*will*) **I**(*revoke*) **Bdir**(*certification*) *from* **Bind**(*farmer*)} **[XOR]** {**A**(*certifier*) **D**(*will*) **I**(*fine*) **Bdir**(*farmer*)}}.

Note:

- Where component-level combinations exist (. . . fine farmer *or* revoke . . .), those have to be signaled explicitly by parentheses, or be decomposed into separate logically-combined complete atomic institutional statements. Further details are provided under *General IG Extended Instructions*, Item *Decomposition of component-level combinations* in Table 8.
- Ambiguities with respect to the linguistic use of logical operators (exclusive and inclusive or) are to be resolved as part of this process.

General IG Core Instructions for Regulative Statements

Additional annotations for Attributes, Objects, and Context

(Note: This feature is analogous to the specification of additional annotations for Constituted Entity, Constitutive Function and Constituting Properties components in the context of constitutive statements.)

In addition to the identification of properties embedded in the original statements, components can further be annotated using additional annotation labels. Such labels can follow the categories listed in Section 5, or be specific to the project objectives.

A systematic approach to labelling entities is discussed under *IG Logico Instructions*, Item *Cross-Component Semantic Annotations* in Table 9. This is particularly recommended if annotations are of strong relevance for the coding and of diverse nature.

Example statement:
Organic certifier must send farmer notification of compliance.

Subject to analytical necessity, additional annotations can for instance relate to the identification of aspects such as the characterisation of encoded objects with respect to their animacy as either animate or inanimate – signified in brackets in the coded example. Where indicated, the annotation should be separated from the component specification by semicolon and have the structure "*label=*", followed by the annotation.
Note that "*label*" is a characterizing prefix, either based on a specific taxonomy (see Section 5 for an overview), or a custom coder-defined category specification (e.g., defined as part of the project-specific guidelines). In this example, the Animacy Taxonomy has the prefix *anim* as defined in Section 5 and is used correspondingly.

While exemplified here for regulative statements, this equally applies to constitutive statements.

A[anim=animate]*(Organic certifier)* **D***(must)* **I***(send)* **Bind[anim=animate]***(farmer)* **Bdir[anim=inanimate]***(notification of compliance).*

Decomposition of component-level combinations (Note: This applies to regulative and constitutive statements, and is discussed here with focus on the regulative perspective.)	Where combinations of components (component-level combinations) are observed that are not explicitly decomposed as in the case of vertical nesting, or not explicitly identified as component-level combinations (e.g., using parentheses), these can be decomposed into logically-combined statements. Other than for aims, the decomposition is optional for IG Core. Extended details are provided under *General IG Extended Instructions*, Item *Decomposition of component-level combinations* in Table 8; IG Core Examples can be found in the following.

Decomposition of component-level combinations (ctd.)

Operationally, combinations of components are evidenced by the presence of multiple attributes, objects, aims or execution constraints within a single institutional statement. Examples of each case are provided in the next column.

Decomposition essentially entails constructing an individual statement to capture each of the unique components represented in multiples within institutional statements, noting the relation to the original statement in which multiple components are reflected. Information from component fields, other than that containing multiple components, is simply carried over to all related institutional statements.

Importantly, where decomposition actually changes the meaning of the original institutional statement containing multiple components within a particular syntactic field, the statement should not be decomposed. In such cases, multiples are typically intended to exist in coupled form. An example is provided in the next column.

Note: These guidelines highlight the motivation for the decomposition, and exemplify the process. Depending on the use of annotation means and tool support, the decomposition may be signaled by annotation of component combinations, and thus occur automated without requiring explicit decomposition by users.

Multiple Attributes Example Statement:
Certifiers and Inspectors must seek accreditation annually.

Decomposed as:

Statement 1: *Certifiers must seek accreditation annually.*
[AND]
Statement 2: *Inspectors must seek accreditation annually.*

Multiple Aims Example Statement:
Inspectors must sign and file farm inspection reports following site visits.

Decomposed as:

Statement 1: *Inspectors must sign farm inspection reports following site visits.*
[AND]
Statement 2: *Inspectors must file farm inspections reports following site visits.*

Multiple Conditions Example Statement:
Inspectors must conduct site visits in person twice per year.

Decomposed as:

Statement 1: *Inspectors must conduct site visits in person.*
[AND]
Statement 2: *Inspectors must conduct site visits twice per year.*

Decomposition of component-level combinations (ctd.)	Coupled Component Example Statement not to be decomposed: *Farmers must pay Certifier $250 for application and service fees upon entry into contract for certification services.* The reason for foregoing decomposition in such case lies in the inseparability of application and services fees, since they are reported as a combined fee of $250.

Table 7: Coding Guidance on Syntactic Elements for IG Core as Level of Expressiveness (Regulative Statements)

4.2.2. IG Extended Coding of Regulative Statements

Level of Expressiveness: IG Extended

IG Extended enables more detailed structural analysis of institutional data than IG Core and accommodates computational application to aid in institutional coding and analysis. Encoding at this level is designed to be human readable, moderately computationally tractable, and moderately comprehensive in the detail with which syntactic properties of institutional statements are captured.
Coding institutional statements on this level enforces many of the features that have been optional in IG Core and affords a fine-grained decomposition of statements. This includes a richer context characterisation based on predefined taxonomies, the expansion of component-level combinations to reconstruct atomic statements and their relationships, the decomposition of components containing embedded institutional statements or statements of fact (component-level nesting). Finally, IG Extended captures the decomposition of hierarchical relationships amongst explicitly highlighted actors, object and their respective properties.

Syntactic Component	Treatment of Syntactic Components by Level of Encoding	Relevant Examples	Complete Syntactic Classification of Examples
Attributes	Key features: • Property Hierarchy Decomposition • Component-level Nesting Building on the IG Core coding that identifies attributes and their respective properties, IG Extended affords a more comprehensive decomposition of the attributes component into Attributes, properties, along with their respective functional descriptors (higher-order properties). In addition, we can identify related attribute objects (so called to maintain the reference to the attribute) that carry their own properties and parameters. This approach is explored in the first example. Within this property hierarchy, elements may be substituted by an institutional statement or statement of fact, referred to as *component-level nesting* (which likewise applies to other components such as objects, conditions and constraints). We explore this approach in the second example. Note that in all cases, named entities (e.g., United States Department of Agriculture) are not decomposed.	Example Statement: *A certified farmer whose certification is suspended by the Secretary under this section may at any time submit a recertification request.* In this example, the attribute *"A certified farmer whose certification is suspended by the Secretary under this section . . ."* embeds aspects central to the institutional configuration. Beyond the identification of properties of the attribute (*"certified"*), it highlights a complex property in the form of a nested institutional statement signaled by the involvement of another entity. Reformulated as *"Secretary suspends certified farmer's certification"*, we can construe this second property as a nested statement with the corresponding component characteristics, as shown in the coding of the example. While objects related to attributes are objects, they are coded with a reference to the attribute to ensure unambiguous association with the attributes component in the institutional statement.	*A* **A**,**p1***(certified)* **A***(farmer)* **A**,**p2**{*whose* **Bdir***(certification) is suspended* **I***([suspends]) by the* **A***(Secretary)* **Cex***(under this section)*} **D***(may)* **Cac***(at any time)* **I***(submit) a* **Bdir,p***(recertification)* **Bdir***(request).*

Attributes (ctd.)

Where properties or parameters apply to multiple objects or entities, the respective references are separated by semicolon (e.g., **A1**,**p1**; **A2**,**p1**). This coding is exemplified in the context of the object component.
Properties can operate on an arbitrary level of depth. For example, properties of properties (second-order properties) are coded as (**A**,**p1**,**p1**), where property identifier are unique on each level.

Using a further example, we can showcase the coding of beliefs or assessments more generally in the form of second-order property hierarchies that rely on component-level nesting as shown before:

Example Statement:
Program managers who believe that a certified operation has violated the Act may pursue revocation proceedings.

A*(Program managers)* **A**,**p1***(who believe)* **A**,**p1**,**p1**{*that a* **A***(certified operation)* **I***(has violated) the* **Bdir***(Act)*} **D***(may)* **I***(pursue)* **Bdir***(revocation proceedings).*

In contrast to the previous statement that highlights complex property arrangements on a given level, this statement emphasizes a hierarchical organization amongst properties, where the second-order property contains a complete institutional statement.

Attributes (ctd.)	Nested institutional statement = *certified operation has violated the Act* Similar to the previous coding, attributes are decomposed into their identifier and properties. Here, the nested statement is captured in the second-order property (**A**,**p**,**p**), since the statement describes the content of the belief (**A**,**p**).	

Object

Key features:

- Object-Property Hierarchy Decomposition
- Component-level Nesting

Building on the differentiation into direct and indirect object along with their respective properties in IG Core, in IG Extended the coding is refined to capture relationships between elements in object specifications. Specific focus lies on the decomposition of hierarchical relationships between objects and properties, respectively. In addition, embedded objects without direct functional relationship are identified, i.e., objects that are referred to but are not explicitly acted on in the context of the institutional statement.
As highlighted for the attributes component, elements of the object-property hierarchy may be substituted by an institutional statement, in which case component-level nesting applies for the encoding (which is exemplified following the object-property decomposition).

Rules for object-property decomposition:

- The identification of the direct or indirect Object respectively.

Example statement:

The Program Manager shall send a written notification of proposed suspension or revocation of certification to certified organic farmer.

Similar to the attributes coding in IG Extended, this statement decomposes complexity embedded in the object component:

Object = *written notification of proposed suspension or revocation of certification*

The direct object in this statement is the *"notification"*, which has the property *"written"*, and makes reference to another related functionally independent object *"certification"*. Functional independence here refers to the fact that the certification does not depend on the notification. The certification itself is characterised by further dependent objects such as the *"suspension"* and *"revocation"*. Both objects are dependent, since they depend on the existence of a certification. Both dependent objects share the property of being *"proposed"*.

The **A***(Program Manager)* **D***(shall)* **I***(send)* *a* **Bdir**, **p***(written)* **Bdir***(notification)* *of* **Bdir**,**p1**,**p1**,**p**; **Bdir**,**p1**,**p2**,**p***(proposed)* **Bdir**,**p1**,**p1***(suspension)* *or* **Bdir**,**p1**,**p2***(revocation)* *of* **Bdir**,**p1***(certification)* *to* **Bind**,**p1***(certified)* **Bind**,**p2***(organic)* **Bind***(farmer).*

Object (ctd.)

- The identification of properties associated with the Object, both including directly functionally dependent (e.g., descriptors) and conceptually independent properties. If an object is a named entity (e.g., United States Department of Agriculture), it is not decomposed.
- For each property, functional relationships are identified (i.e., which Object or property they rely on). Where such relationships exist, these properties become child properties of the properties they functionally depend on. This process produces an Object hierarchy that may or may not directly involve the Object component itself.
- Where multiple children exist, implicit or explicit logical relationships (conjunction, disjunction, negation) between different branches of the emerging Object hierarchy are retained.
- Non-functional relationships are established between the root node of the Object hierarchy and the direct or indirect Object established in the first step.

Establishing these relationships, we can code the object and respective properties as follows:

Related functionally-independent object = *certification*

Functionally independent objects are identified as individual objects with unique alphabetical index, e.g., (**Bdir1**), (**Bdir2**), etc.

Dependent objects = *suspension, revocation*

Dependent objects are coded with reference to the object they depend on and unique numeric index, e.g., (**Bdir1,p1**), (**Bdir1,p2**), etc.

Dependent object properties = *proposed*

Dependent object properties are identified with reference to objects they relate to, with references to different objects separated by semicolon, e.g., (**Bdir,p1,p**;**Bdir,p2,p**), etc.

Where multiple properties exist for an object, they are, as in IG Core coding, uniquely identified by numeric index, e.g., (**Bdir,p1,p1**), (**Bdir,p1,p2**), etc.

Object (ctd.)

Where higher-order properties exist, these are coded according as specified in the context of the Attributes component.

Where object hierarchies are specified outside the object component (latent objects, e.g., as part of the context component), the same process applies, even if no reference is made to direct or indirect object. Note, however, that those are coded with lower priority, i.e., following object hierarchies that relate to direct or indirect object.

Object hierarchies outside the object component:

Taking the example *"**A**(Certifier) **D**(must) **I**(send) **Bdir**(notification) to **Bind**(inspector) **Cex**[**ctx=pur**](to produce a written report of the assessment)"*, we find an object hierarchy within the context (execution constraint) component.

In this example, first-order object is the report that is written and relates to an assessment, without being functionally dependent – as reflected in the coding.

A(Certifier) **D**(must) **I**(send) **Bdir**(notification) to **Bind**(inspector) **Cex**[**ctx=pur**](to produce a **B,p1,p1,p1**(written) **B,p1,p1**(report) of the **B,p1**(assessment)).

In this example, we use parentheses to clearly delineate the scope of the execution constraint that contains the external object hierarchy linked to the latent object "assessment".

Component-level nesting:

Another feature of IG Extended is the notion of component-level nesting, i.e., the decomposition of individual components that are expressed in terms of institutional statements or statements of fact. This approach affords the conceptual representation of mental constructs, such as beliefs, opinions or behavioral assessments. Note that component-level applies to various components (e.g., Attributes, Objects, Conditions, Constraints), but is exemplified for objects at this stage.

Component-level nesting on objects:

Reviewing the following example "*Inspectors must ensure that certified organic farmers report annually on their agricultural practices.*", the object of the statement ("*certified organic farmers report annually on their agricultural practices*") reflects nested institutional statement in its own right.

A(Inspectors) **D**(must) **I**(ensure) that **Bdir**{**A**,**p1**(certified) **A**,**p2**(organic) **A**(farmers) **I**(report) **Cex**(annually) on their **Bdir**(agricultural practices)}.

Aim	The coding of the Aim is identical to IG Core. See also *General IG Extended Instructions*, Item *Decomposition of component-level combinations* in Table 8 for associated decomposition rules.
Deontic	The coding of the Deontic is identical to IG Core.

Context

Key features:

- Context Taxonomy
- Component-level Nesting

In IG Extended, Activation conditions (**Cac**) and Execution constraints (**Cex**) (as specified in IG Core) are further characterised in terms of ontological categories, specifically capturing the nature of circumstances the conditions or constraints describe. Those are grouped into taxonomies comprehensively described in Section 5 (along with subcategories). The central taxonomy in this context is the *Context Taxonomy*, an extract of which is provided in the following, along with corresponding annotations:

Excerpt from Context Taxonomy:

- Temporal (tmp)
- Spatial (loc)
- Domain (dom)
- Procedural (prc)
- Purpose (pur)
- ...

A comprehensive overview of the Context Taxonomy including subcategories (e.g., 'point in time', 'time frame') is provided in Section 5.1, alongside further specific abbreviations (e.g., *tfr* for time frame).

Example 1:

Upon entrance into agreement with organic farmer to serve as his/her certifying agent, organic certifier must inspect farmer's operation within 60 days.

This statement is rather complex and includes both component-level nesting, along various further constraint specifications.

Retracing this coding is best achieved by reformulating this statement as follows:

A*(Organic certifier)* **D***(must)* **I***(inspect)* **Bdir***(farmer's operation)* **Cex[ctx=tfr]***(within 60 days)* **Cac[ctx=prc]**{*under the condition that the* **A***(organic farmer)* **I***(enters)* **Bdir**,**p***(an)* **Bdir***(agreement)* **Bind***(with the organic certifier)* **Cex[ctx=pur]***(to serve as his/her certifying agent)*}.

Here the execution constraint for the initial inspection (**Cex[ctx=tfr]**) is activated by the preceding procedural activation condition (**Cac[ctx=prc]**), expressed as a nested institutional statement, that the farmer enters an agreement, whose purpose is defined as an execution constraint (**Cex[ctx=pur]**).

Cac[ctx=prc]{**I***(Upon entrance)* **Bdir***(into agreement)* *with* **A***(organic farmer)* **Cex[ctx=pur]***(to serve as his/her certifying agent)*}, **A***(organic certifier)* **D***(must)* **I***(inspect)* **Bdir***(farmer's operation)* **Cex[ctx=tfr]***(within 60 days)*.

Context (ctd.)

In addition to the contextual characterisation, conditions or constraints can further be expressed as primitive statements (e.g., at 8am), or complex expressions embedding complete institutional statements (e.g., if an organic farmer violates conditions for certification, . . .).

Note: While the use of the annotation prefix label *ctx* is recommended for context annotations (as specified in Section 5.1 and further discussed under *IG Logico Instructions*, Item *Cross-Component Semantic Annotations* in Table 9), where no other labels are applied to activation conditions and execution constraints, the prefix can be omitted as part of the shorthand coding.

Where logical relationships between conditions or constraints exist, these are made explicit by annotating those as **[AND]**, **[OR]**, and **[XOR]** relationships. Where negation exists, this is to be likewise identified (**[NOT]**).

Rules:

- Identify involved actions (explicit and tacit) by reformulating statements in active terms, and expand into (potentially multiple) statements.

Example 2:

When rebuttal is unsuccessful or correction of the noncompliance by certified organic farmer is not completed within the prescribed time period, the Program Manager shall send a written notification of proposed suspension or revocation of certification to certified organic farmer.

This statement, similar to the previous example, includes the encoding of a nested institutional statement. In contrast, it highlights the combination of multiple conditions statements, where one is of implicit nature and the second one explicit (apart from minor rephrasing).

Coding this example, we first identify the top-level institutional statement:

Top-level institutional statement = *the* **A***(Program Manager)* **D***(shall)* **I***(send) a* **Bdir,p***(written)* **Bdir***(notification) of* **Bdir,p1,p1,p***;* **Bdir,p1,p2,p***(proposed)* **Bdir,p1,p1***(suspension) or* **Bdir,p1,p2***(revocation) of* **Bdir,p1***(certification) to* **Bind***(certified organic farmer).*

{**Cac[ctx=evt]**{*When [***A,p1***(certified)* **A,p2***(organic)* **A***(farmer)] ***I***([rebuts])* **Cex[ctx=evt]** *(unsuccessfully)*}
[OR]
Cac[ctx=tfr]{**Bdir,p***(correction) of the* **Bdir***(noncompliance) by* **A***(certified organic farmer) is not completed* **I***([does not complete])* **Cex[ctx=tfr]***(within the prescribed time period)*}},
the **A***(Program Manager)* **D***(shall)* **I***(send) a* **Bdir,p***(written)* **Bdir***(notification) of* **Bdir,p1,p1,p***;* **Bdir,p1,p2,p***(proposed)* **Bdir,p1,p1***(suspension) or* **Bdir,p1,p2***(revocation) of* **Bdir,p1***(certification) to* **Bind***(certified organic farmer).*

Context (ctd.)

- Hint: Where statements contain an aim that is linked to an object as noun, it is indicative of a missing or implied actor specification. Such case is signaled by passive tense (e.g., notification is received). The actor specification is either implied from context or potentially signaled by prepositional clauses such as "*by actor*"; Example: "*Notification by certifier is received by farmer*". In such cases, the statement requires reformulation in active terms along with the injection of an inferred explicit action an explicit action, e.g., "*certifier sends notification to farmer*". In some cases, this may require the expansion of a single statement into separate statements carrying separate actions (see Example 3).
- Identify logical relationships (**[AND]**, **[OR]**, **[XOR]**) and dependencies (sequence) amongst actions.
- Reconstruct complete statement using component-level nesting of statements where relevant.

In this statement we observe principles of the Attributes/Object-property hierarchy (see Section 2.3), which are encoded based on the instructions provided for objects.

Context clause = *When rebuttal is unsuccessful or correction of the noncompliance by certified organic farmer is not completed within the prescribed time period*

The context clause (which contains two activation conditions) can be decomposed into two statements that are logically combined (**[OR]**). The passive aim on an object (rebuttal is unsuccessful) signals an implied action, and requires reformulation.

First condition statement = *When [**A**(organic farmer)] **I**([rebuts]) **Cex**[**ctx=met**](unsuccessfully)*

The second condition ("*correction of the noncompliance by certified organic farmer is not completed within the prescribed time period*") can be reformulated in active terms as "*if certified organic certifier does not complete correction of noncompliance within the prescribed time period*".

Context (ctd.)

Doing so, the structure of the nested statement becomes overt:

"if **A,p1***(certified)* **A,p2***(organic)* **A***(certifier) does not* **[NOT]** **I***(complete)* **Bdir,p***(correction) of* **Bdir***(non-compliance)* **Cex[ctx=tfr]***(within the prescribed time period)"*

Both conditions are further logically related by an inclusive disjunction (AND/OR), which is annotated explicitly using **[AND]**, **[OR]**, or **[XOR]**, respectively.
The composition of all three statements is shown in the example.

Example 3:
When an inspection of an accredited certifying agent by the Program Manager reveals any noncompliance with the Act or regulations in this part, a written notification of noncompliance shall be sent to the certifying agent.

Note: While highlighted here for activation conditions, such logical combination equally applies to statements containing multiple execution constraints.

Cac{*When [***A***(Program Manager)]* **I***(reveals) any* **Bdir***(non-compliance) [by the* **Bind,p1***(accrediting)* **Bind,p2***(certifying)* **Bind***(agent)]* **Cex***(with the Act or regulations in this part)* **Cac**{*[under the condition that]* **A***(Program Manager)* **I***([performs])* **Bdir***(inspection) of an* **Bind,p1***(accredited)* **Bind,p2***(certifying)* **Bind***(agent)*}}, **A***([Program Manager])* **D***(shall)* **I***([send])* ***a*** **Bdir,p1***(written)* **Bdir***(notification)* **Bdir,p2***(of noncompliance) to the* **Bind,p1***(certifying)* **Bind***(agent).*

Context (ctd.)

As highlighted before, for the purpose of coding the reformulation of statements in active terms may be useful to facilitate coding. In the first statement "*When an inspection of an accredited certifying agent by the Program Manager reveals any noncompliance with the Act or regulations in this part*", the acting party is the Program Manager (actor), but the aim reveals relates to the object inspection. This is indicative of a tacit action (captured as conceptual reification in the object) on the part of the program manager (captured in the prepositional clause). In this case, we can decompose the compound conditional statement into two statements:

Program Manager [performs] inspection (where the performance is tacit)
[AND]
[Program Manager] reveals any noncompliance ... (in which case the attribute is tacit).

Context (ctd.)

In addition, the second statement depends on the activation of the first action. The conditional statement can thus be decomposed into two institutional statements, as follows:

"When **A***([Program Manager])* **I***(reveals) any* **Bdir***(non-compliance) [by the* **Bind**,**p1***(accrediting)* **Bind**,**p2***(certifying)* **Bind***(agent)]* **Cex***(with the Act or regulations in this part) {[under the condition that]* **A***(Program Manager)* **I***([performs])* **Bdir***(inspection) of an* **Bind**,**p1***(accredited)* **Bind**,**p2***(certifying)* **Bind***(agent), . . ."*

The final part of the statement (the main statement), likewise reformulated in active terms, implies that *"[Program Manager] shall [send] a* **Bdir**,**p1***(written)* **Bdir***(notification) of* **Bdir**,**p2***(noncompliance) to the* **Bind**,**p1***(certifying)* **Bind***(agent)"*.

	Structurally, in this statement we observe two levels of component-level nesting in the form *ABDIC*{*ABDIC*{*ABDIC*}}, where the main statement's (shall send notification; first *ABDIC*) activation relies on revealing potential non-compliance (second *ABDIC*), which in itself relies on the inspection in the first place (last *ABDIC*).
Or else	The coding of the Or else is identical to IG Core. The internal structure of nested statements is encoded according to IG Extended instructions (Table 8).

General IG Extended Instructions for Regulative Statements

Decomposition of component-level combinations

In the presence of multiple Attributes, actions (aims) and objects in a given statement, such statements are to be decomposed into individual statements that are combined with the corresponding logical operator (following the principles of horizontal nesting). At face value this mirrors the approach taken in the context of or Else statements. However, while in the context of Or else components, combinations designate logical relationships amongst action alternatives, in this context the purpose is to disambiguate the relationship amongst particular actors, actions and objects and to resolve a potential incongruence between the linguistic and logical use of conjunctives.
When a disaggregation or aggregation fundamentally alters the meaning of the statement (e.g., if a payoff associated with the institutional statement cannot be unambiguously associated with an individual entity), or if expressions are intentionally coupled (e.g., chips and fish) or proper names (e.g., Smith and Sons), the statement is not to be decomposed.

Example Statement:
Certified operations or handlers must comply with organic farming regulations.

This can be decomposed into

Certified operations must comply with organic farming regulations
[AND]
Certified handlers must comply with organic farming regulations.

Note that the interpretation of the logical operator is contextual. In this example, it carries the understanding that the specified obligation applies to both certified operations and certifier handlers.
Naturally, this approach can lead to the decomposition into a large number of additional statements, e.g., attribute and action combinations - as exemplified below. For practical reasons, the explicit coding can be substituted by an additional annotation that reflects the need for decomposition.

{{**A**,**p**(*Certified*) **A**(*operations*) **D**(*must*) **I**(*comply*) *with* **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)}
[AND]
{**A**,**p**(*Certified*) **A**(*handlers*) **D**(*must*) **I**(*comply*) *with* **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*).}}

Recall that the minimal institutional statement presumes the existence of context specifications, which – in absence of specific encoding – resolves to "under all circumstances" (for activation conditions), "without any constraints" (for execution constraints).

Decomposition of component-level combinations (ctd.)

Example Statement:

Certified operations or handlers must accept and comply with organic farming regulations.

{{**A**,**p**(*Certified*) **A**(*operations*) **D**(*must*) **I**(*accept*) **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)}
[AND]
{**A**,**p**(*Certified*) **A**(*handlers*) **D**(*must*) **I**(*accept*) **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)}
[AND]
{**A**,**p**(*Certified*) **A**(*operations*) **D**(*must*) **I**(*comply*) *with* **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)}
[AND]
{**A**,**p**(*Certified*) **A**(*handlers*) **D**(*must*) **I**(*comply*) *with* **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*).}}

Decomposed:

Certified operations must accept organic farming regulations
[AND]
Certified handlers must accept organic farming regulations
[AND]
Certified operations must comply with organic farming regulations
[AND]
Certified handlers must comply with organic farming regulations.

Practical considerations:

If labelling is performed manually, a practical consideration for such decomposition is to keep track of the relationships of such statements, e.g., by introducing sub-identifiers. For example, assuming the coded statement is Statement 10, the decomposed statements could be annotated as 10.1, 10.2, etc.

Decomposition of component-level combinations (ctd.)		Alternatively, annotation can be performed in shorthand form, or rely on tool-specific support offered by the applied text annotation tool that allows the indication of decomposition during the encoding. In shorthand form, the decomposed statements can be grouped by parentheses to signal component-level combinations as shown below. *Example:* **A**(**(***Operators* **[AND]** *Certifiers***)**) **D**(*must*) **I**(*comply*) *with* **Bdir**(**(***regulations* **[AND]** *best practices***)**). In expanded form (shown on the right), parentheses are then used to signal the association of the decomposed statements.	Grouping of statements using parentheses (in bold font): {{**A**,**p**(*Certified*) **A**(*operations*) **D**(*must*) **I**(*accept*) **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)} **[AND]** {**A**,**p**(*Certified*) **A**(*handlers*) **D**(*must*) **I**(*accept*) **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)} **[AND]** {**A**,**p**(*Certified*) **A**(*operations*) **D**(*must*) **I**(*comply*) *with* **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)} **[AND]** {**A**,**p**(*Certified*) **A**(*handlers*) **D**(*must*) **I**(*comply*) *with* **Bdir**,**p**(*organic farming*) **Bdir**(*regulations*)}}
Logical relationships among statement components	Where statements make tacit reference to multiple actions, conditions or constraints, these are likewise resolved using the instructions provided as part of the *IG Logico Instructions* in Table 9, Item *Logical relationships among statement components*. For IG Extended this provision is recommended, but optional.		

Regulative-constitutive Hybrids	The introduction of constitutive statements as part of IG 2.0 (see Section 4.3) provides the basis for encoding statements that consist of structural elements both of regulative and constitutive statements. Details are discussed in Section 4.4.

Table 8: Coding Guidance on Syntactic Elements for IG Extended as Level of Expressiveness (Regulative Statements)

4.2.3. IG Logico Coding of Regulative Statements

Level of Expressiveness: IG Logico

IG Logico is designed to support semantic analysis of institutional statements wholly relying on computational tools. Encoding at this level is designed to be moderately human readable, computationally tractable and comprehensive in the detail with which syntactic properties of institutional statements are captured.
In contrast to IG Core and Extended that focus on the encoding of specific grammar components, IG Logico emphasises refinements across individual components and further establishes explicit references to related statements to establish computational tractability, as well as the ability to perform logical transformations on institutional statements.

Syntactic Component	Treatment of Syntactic Components by Level of Encoding	Relevant Examples	Complete Syntactic Classification of Examples
Relation-centric Semantic Annotations	To establish relationships amongst statements and policies more generally, the initial refinement refers to the identification of statement references, e.g., between individual statements, collections thereof, or policies more generally. To this end, an additional annotation identifies all instances of references to other statements. Note that cross-statement references are not specific to any statement components, but apply across complex component types, including attributes, objects and context. Rules: • Identify references in coded statements • Add annotation of structure (*ref=value*), where *ref* signals the reference nature, and value contains an identifier of referenced statement, collection, or policy.	Exploring this approach, we borrow the complex example statement previously coded in the context of IG Extended: Example Statement: *When an inspection of an accredited certifying agent by the Program Manager reveals any noncompliance with the Act or regulations in this part, a written notification of noncompliance shall be sent to the certifying agent.* Coded form: **Cac[ctx=prc]**{**(***When [***A***(program manager)* **I***(performs)] an* **Bdir***(inspection) of an* **Bind,p1***(accredited)* **Bind,p2***(certifying)* **Bind***(agent)***)** *[***[AND]***]* **(A***(Program Manager)* **I***(reveals) any* **Bdir***(noncompliance)* **Cex***(with the Act or regulations in this part)***)**}, *[***A***(Program Manager)] a* **Bdir,p1***(written)* **Bdir***(notification)* **Bdir,p2***(of noncompliance)* **D***(shall) be sent* **I***([send])* **Bind***(to the certifying agent).*	**Cac[ctx=prc]**{{*When [***A***(Program Manager)* **I***(performs)] an* **Bdir***(inspection) of an* **Bind,p1***(accredited)* **Bind,p2***(certifying)* **Bind***(agent)*} *[***[AND]***]* {**A***(Program Manager)* **I***(reveals) any* **Bdir***(non-compliance)* **Cex[ref=["policy","section"]]** *(with the* **(***Act [***[OR]***] regulations in this part***)**) }}, **A***([Program Manager]) a* **Bdir,p1***(written)* **Bdir***(notification)* **Bdir,p2***(of non-compliance)* **D***(shall) be sent* **I***([send])* **Bind***(to the certifying agent).*

Relation-centric Semantic Annotations (ctd.)

This statement makes reference to "*the Act*" and "*regulations in this part*", the first of which makes reference to the coded policy in its entirety, whereas the second one focuses on a specific section of the policy.
Both are coded by providing an additional annotation (*ref=value*), along with the scope reference ("*value*"), i.e., an identifier of relevance in the context of the encoded policy. The identifiers need to be unambiguous within the given document, including statement IDs, section headers, or documents as a whole, etc. The corresponding convention should be decided as part of project-specific coding guidelines. If occurring in conjunction with existing component classification, the reference specification is appended (e.g., **Bdir[ref="policy"]**(*Act*)).
As stated before, while, in this specific example, cross-statement reference apply to constraints components, such references can likewise occur in other components.

{{**Cac**{*When* [**A**(*Program Manager*)] **I**(*reveals*) **Bdir**,**p**(*any*) **Bdir**(*non-compliance*) [*by the* **Bind**,**p1**(*accrediting*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)] **Cex[ref="policy"]** (*with the Act*)}

Cac{[*under the condition that*]

{**A**(*Program Manager*) [**I**(*performs*)] **Bdir**(*inspection*) *of an* **Bind**,**p1**(*accredited*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)}

[OR]

{**A**(*Program Manager*) [**I**(*performs*)] **Bdir**(*inspection*) *of an* **Bind**,**p1**(*accredited*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)}

[OR]

{**A**(*Program Manager*) [**I**(*performs*)] **Bdir**(*inspection*) *of an* **Bind**,**p1**(*accredited*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)}
}},
A([*Program Manager*]) **D**(*shall*) **I**([*send*]) *a* **Bdir**,**p1**(*written*) **Bdir**(*notification*) **Bdir**,**p2**(*of non-compliance*) *to the* **Bind**,**p1**(*certifying*) **Bind**(*agent*).}

[OR] (← decomposition of "*Act*" and "*regulations in this part*")
...

Logical relationships among statement components

The objective of decomposition of lists, or other forms of implied conjunctions, such as multiple conditions/constraints is to make logical relationships explicit. In such cases enumerations are decomposed into individual atomic institutional statements and combined using the corresponding logical operator.
Note that this is similar to the expansion of multi-entity components in IG Extended with specific emphasis on attributes and objects. In this case, the review operates across all component types and explicitly focuses on implied logical relationships.

Rules:

- Identify action alternatives embedded in lists or enumeration, or in conditions/constraints
- Identify associated atomic statement
- Establish logical operator and precedence where needed
- Expand statement via horizontal nesting to capture individual action alternatives

We use a modified example previously explored under IG Extended, Context component.

Example Statement:
When an inspection, review, or investigation of an accredited certifying agent by the Program Manager reveals any non-compliance with the Act or regulations in this part, a written notification of non-compliance shall be sent to the certifying agent.

Following the decomposition patterns for Context specification established previously (IG Extended, Context), we observe that an inspection of a program manager has to be performed, and may reveal non-compliance, and arrived at the following coding:

Cac{*When* **A**(*[Program Manager]*) **I**(*reveals*) *any* **Bdir**(*non-compliance*) *[by the accrediting)* **Bind**,**p2**(*certifying*) **Bind**(*agent*)*]* **Cex**(*with the* **(***Act [***[OR]** *]regulations in this part***)**) **Cac**{*[under the condition that]* **A**(*Program Manager*) **I**(*[performs]*) **Bdir**(*inspection*) *of an* **Bind**,**p1**(*accredited*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)}}, ...

...

{**Cac**{*When* **A**(*[Program Manager]*) **I**(*reveals*) **Bdir**,**p**(*any*) **Bdir**(*non-compliance*) *[by the* **Bind**,**p1**(*accrediting*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)*]* **Cex** **[ref="section"]** *(with regulations in this part)*

Cac{*[under the condition that]*

{**A**(*Program Manager*) **I**(*[performs]*) **Bdir**(*inspection*) *of an* **Bind**,**p1**(*accredited*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)}

[OR]

{**A**(*Program Manager*) **I**(*[performs]*) **Bdir**(*inspection*) *of an* **Bind**,**p1**(*accredited*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)}

[OR]

{**A**(*Program Manager*) **I**(*[performs]*) **Bdir**(*inspection*) *of an* **Bind**,**p1**(*accredited*) **Bind**,**p2**(*certifying*) **Bind**(*agent*)}
}},
A(*[Program Manager]*) **D**(*shall*) **I**(*[send]*) *a* **Bdir**,**p1**(*written*) **Bdir**(*notification*) **Bdir**,**p2**(*of non-compliance*) *to the* **Bind**,**p1**(*certifying*) **Bind**(*agent*)*.*}

Logical relationships among statement components (ctd.)

. . . **A***([Program Manager])* **D***(shall)* **I***([send])* *a* **Bdir**,**p1***(written)* **Bdir***(notification)* **Bdir**,**p2***(of non-compliance) to the* **Bind**,**p1***(certifying)* **Bind***(agent).*

In this example, the program manager's notification is contingent on an inspection, a review, or investigation, i.e., a variation of instruments for assessment. Logically, the initial task (inspection) that is prerequisite for further action. The original statement is

Cac{*[under the condition that]* **A***(Program Manager)* **I***([performs])* **Bdir***(inspection)* **Bind**,**p1***(of an accredited)* **Bind**,**p2***(certifying)* **Bind***(agent)*}

Given that we now have three alternative actions, the statement requires expansion into three separate tasks.
To achieve this, the logical relationship between the tasks needs to be established. Subject to context, the coder can interpret the relationship as either an inclusive disjunction (**[OR]**) or exclusive disjunction (**[XOR]**).

Logical relationships among statement components (ctd.)

Since realistically (based on interpretation of application context) a combination of any of such tasks could equally lead to the detection of non-compliance, suggesting the combination via **[OR]**. This example showcases the importance of coding context and interpretation, which makes an explicit specification necessary for analytical treatment of action alternatives.
In consequence, the statement is expanded into **[OR]**-combined statements as follows (Note: while not necessary in this case, the logical combination of the statements is signaled using surrounding parentheses):

. . . [under the condition that]

{{**A***(Program Manager)* **I***([performs])* **Bdir***(inspection)* **Bind**,**p1***(of an accredited)* **Bind**,**p2***(certifying)* **Bind***(agent)*}

[OR]

{**A***(Program Manager)* **I***([performs])* **Bdir***(review) of an* **Bind**,**p1***(accredited)* **Bind**,**p2***(certifying)* **Bind***(agent)*}

[OR]
. . .

Logical relationships among statement components(ctd.)

. . .

{**A***(Program Manager)* **I***([performs]) of an* **Bdir***(investigation)* **Bind**,**p2***(certifying)* **Bind**,**p1***(accredited)* **Bind***(agent)*}}

This coding is embedded in the complete statement coding as shown on the right (with formatting adjustments so as to make the discussed coding easily accessible).
Note: While not applicable in this case, where necessary, precedence of specific combinations has to be signaled using parentheses (e.g., if inspection OR review is permitted, or as an exclusive alternative, the investigation, which would be (simplified) represented as **((***inspection* **[OR]** *review***) [XOR]** *investigation***)**. Another aspect that requires explicit coding in this example is the reference to the scope of violation, i.e., non-compliance with the Act or regulations in this part, the logical relationship of which (here: "or"), subject to coder interpretation, has to be coded explicitly.

Logical relationships among statement components (ctd.)

In this example the relationship is characterized by an inclusive disjunction (**[OR]**), since regulations in this part are part of the Act. As with all other cases, this annotation is made explicit, and consequently, requires decomposition of the statement by duplication into corresponding statement variants:

{*When* **A***([Program Manager])* **I***(reveals)* **Bdir**,**p***(any)* **Bdir***(non-compliance)* *[by the* **Bind**,**p1***(accrediting)* **Bind**,**p2***(certifying)* **Bind***(agent)]* **Cex**[**ref**=**"policy"**] *(with the Act)*}

. . . (remainder of statements) . . .

[OR]

{*When* **A***([Program Manager])* **I***(reveals)* **Bdir**,**p***(any)* **Bdir***(non-compliance)* *[by the* **Bind**,**p1***(accrediting)* **Bind**,**p2***(certifying)* **Bind***(agent)]* **Cex**[**ref**=**"section"**]*(with regulations in this part)*}

. . . (remainder of statements) . . .

Logical relationships among statement components (ctd.)	Such decomposition affords a systematic assessment of the individual variants in the context of a specific situation, but likewise offers automation potential. For example, given the assumption that regulations in this part are a subset of the Act, we could ignore the statement variant that assesses the compliance with regulations in this part. However, for the sake of comprehensive illustration of the embedded institutional complexity (and subject to alternative interpretations), we decompose this example comprehensively.

Cross-Component Semantic Annotations

In addition to explicit encoding of properties as specified in the underlying statement (e.g., explicit identification of "certified" as a property of an operation), semantic qualities can be enriched by providing additional annotations that capture a differentiation of components with respect to different ontological categories captured in different taxonomies (see Section 5). In contrast to the property annotations used in IG Core and Extended, the annotations introduced here apply across all components, and explicitly emphasize extensibility both with respect to additional categories within the given taxonomies, as well as specification of further taxonomies, e.g., based on domain-specific or analytical necessities.
Furthermore, multiple annotations of different categories can be applied to a given component at the same time, e.g., animacy descriptors, such as animate, can be combined with role or metatype descriptors.
For the exemplified taxonomies here, categories include

- Animacy: animate, inanimate
- Metatype: abstract, concrete
- Role: Source, Recipient, Possessor, Experiencer, Beneficiary

An extended listing of the associated taxonomies along with their description is provided in Section 5.

Complementing the characterisation of context by circumstance, entities can further be annotated by the role they play in a particular setting, as well as further properties, such as physical types. These categorizations can be extended beyond the specified types, apply across all component types, and furthermore allow the introduction of additional taxonomies, beyond the ones highlighted in Section 5.
To annotate components with additional categories, the component coding is extended by key-value pairs, with the key specifying the taxonomy, and the value the corresponding categorization(s). In the coding highlighted here, the category specifications are separated from component specification by semicolon. Where multiple categories for a given taxonomy apply, these are separated by comma.
The example on the right highlights this approach with respect to various taxonomies specified in Section 5.

Cac{*When [A*[**anim=animate; role=experiencer; metatype=concrete**]*(Program Manager)] I(reveals)* **Bdir,p***(any)* **Bdir**[**anim=inanimate; metatype=abstract**]*(non-compliance) [by the* **Bind,p1***(accrediting)* **Bind,p2***(certifying)* **Bind**[**anim=animate; role=originator; metatype=concrete**]*(agent)]* **Cex**[**metatype=abstract**]*(with the* [**anim=inanimate; ref="policy"; metatype=concrete**] *Act /*[**OR**]*/* [**ref="section"; metatype=concrete**]*regulations in this part), [A*[**anim=animate; role=originator; metatype=concrete**]*(Program Manager)] D(shall) I([send]) a* **Bdir,p1***(written)* **Bdir**[**anim=inanimate; metatype=concrete**] *(notification)* **Bdir,p2***(of noncompliance) to the* **Bind,p1***(certifying)* **Bind**[**anim=animate; role=recipient; metatype=concrete**]*(agent).*

Regulative Function Annotations

IG Logico further prescribes annotations of statements with institutional functions in regulative or constitutive form. While other features described above in practice focus on the refined coding of attributes, objects and conditions, institutional actions reflect the regulative function of a statement's action.

Being fined, for example, reflects the regulative function of "sanctioning", adhering to regulation reflects "compliance".

Institutional functions, and here specifically regulative functions, as identified in this specification include the following set, some of which operate complementary and are listed as comma-separated function pairs:

- Comply, Violate
- Reward, Sanction
- Monitor
- Detect compliance, Detect non-compliance
- Delegate

The Regulative Function Taxonomy is described in Section 5.5.

Taking the previously used statement as an example (including inferred components, but omitting any annotations), we can identify regulative functions associated with actions.

{*When [Program Manager] reveals any non-compliance [by the accrediting certifying agent] with the Act or regulations in this part*}, *[Program Manager] shall [send] a written notification of noncompliance to the certifying agent.*

Actions of institutional relevance include:

- Reveal (non-compliance)
- Send (notification)

In this context revealing non-compliance abstractly corresponds to the "*detection of a violation*", whereas sending a notification reflects a form of "*sanctioning*". In consequence, the statement can be annotated with these regulative functions, so as to enable inferences from a purely institutional perspective without concern for the specific operationalization of detecting compliance or sanctioning in a specific scenario.

Cac{*When* **A[anim=animate; role=experiencer]**(*[Program Manager])* **I[regfunc=detect violation]**(*reveals)* **Bdir,p**(*any)* **Bdir[anim=inanimate]**(*non-compliance)* *[by the* **Bind,p1**(*accrediting)* **Bind,p2**(*certifying)* **Bind[anim=animate; role=originator]**(*agent)]* **Cex**(*with the* (**[anim=inanimate; ref="policy"]***Act* **[OR]** (**[ref="section"]***regulations in this part)*}, **A**; **[anim=animate; role=originator]**(*[Program Manager)])* **D**(*shall)* **I[regfunc=sanction]**(*[send])* *a* **Bdir,p1**(*written)* **Bdir**; **[anim=inanimate]**(*notification)* **Bdir,p2**(*of noncompliance)* *to the* **Bind,p1**(*certifying)* **Bind[anim=animate; role=recipient]**(*agent).*

Regulative Function Annotations (ctd.)

The annotation follows the syntactic specification applies for other forms of annotations, i.e., appending a key-value pair to the component coding, where the key is "*regfunc*" and the value carries the corresponding regulative function specific to the annotated action.

Providing an additional simplified example, we can explore the use of further annotations:

The Program Manager may initiate revocation proceedings against a certified operation

{*When the Program Manager has reason to believe that a* {*certified operation has violated the Act*}

[OR]

When a certifying agent fails to take appropriate action to enforce the Act}.

. . .

The **A***(Program Manager)* **D***(may)* **I[regfunc=sanction]***(initiate)* **Bdir***(revocation proceedings)* **Bind***(against a certified operation)*

Cac{*When the* **A***(Program Manager)* **I[regfunc=evaluate]***(has reason to believe) that a* **Bdir**{**A***(certified operation)* **I[regfunc=violate]***(has violated) the* **Bdir[ref="policy"]***(Act)*}

[OR]

When a **A***(certifying agent)* **I[regfunc=violate]***(fails)* **Bdir***(to take appropriate action)* **Cex**,**[ctx=pur; ref="policy"]***(to enforce the Act)*}.

Regulative Function Annotations (ctd.)	The key actions include: • Initiate (revocation proceedings), corresponding to a sanction • "Has reason to believe" reflects an evaluation on the part of the actor • "Violate" reflects a violation • "Fail to take appropriate action" likewise represents a violation In the coding, these actions can thus be annotated with the corresponding regulative functions.
General IG Logico Instructions for Regulative Statements	
A central objective is to provide a consistent coding that reflects the most fine-granular level of encoding. While the order of encoding is loosely prescribed by the order of specification, in some cases a variation of the order may be indicated. This should be considered as part of the coding preparation. The coding may further require iterative review, specifically with respect to annotations and logical relationships. While implicit in the multi-pass coding implied for IG Logico, a dedicated review of embedded object hierarchies (encoded as part of IG Extended) and the explication of logical relationships between component elements (e.g., specific execution constraints) is of central concern in IG Logico.	

Table 9: Coding Guidance on Syntactic Elements for IG Logico as Level of Expressiveness (Regulative Statements)

4.3. Constitutive Statement Coding

The principal structure of constitutive statements visualized in Figure 10 (necessary components are held in solid boxes, sufficient components in dashed boxes) highlights the relevant syntactic components (as defined in Table 3) using the corresponding symbols specified in Table 6. Analogous to the introduction of regulative statements, the figure summarizes additional coding refinements introduced across levels of expressiveness, with variations largely relating to the structural decomposition of selected syntax elements and the semantic annotations applicable under IG Logico.

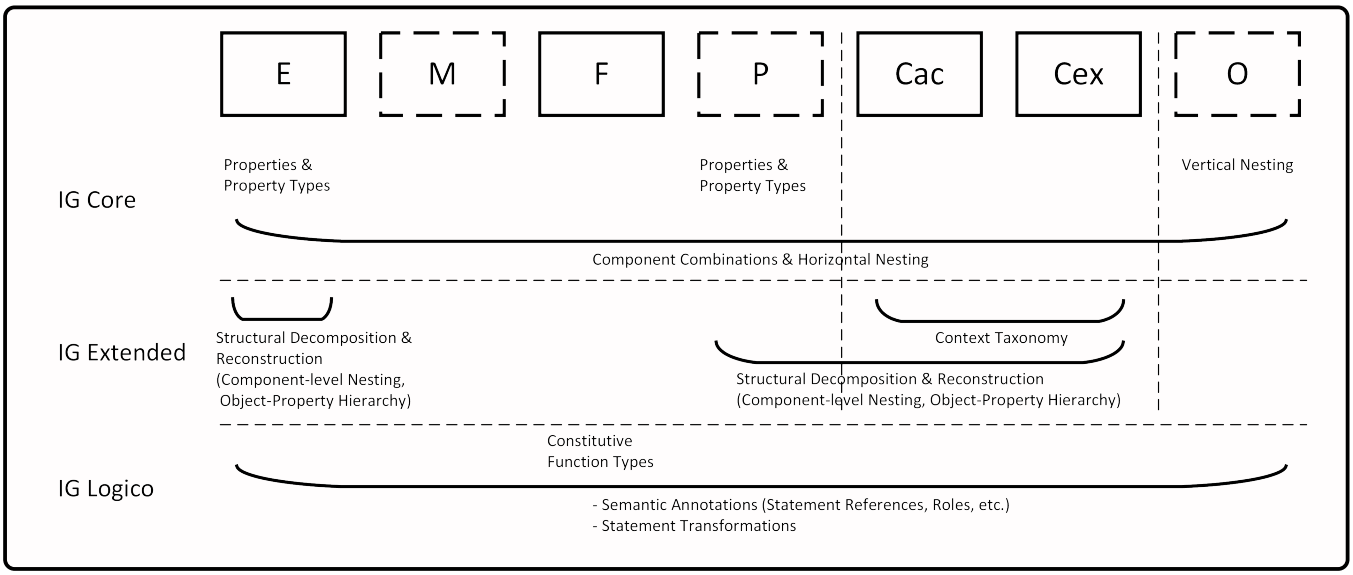

Figure 10: Syntax and Features of Constitutive Statements by Level of Expressiveness

Motivating the principal coding of constitutive statements (without further elaboration at this stage), a stylized atomic constitutive statement, coded in shorthand form, thus reads:

Cac*(In the context of organic farming),* **E**,**p***(certified)* **E***(farmers)* **F***(are)* **P***(farmers)* **P**,**p***(that have undergone a certification process)* **Cex***(following relevant procedural guidelines).*

Mirroring the introduction of coding guidelines for regulative statements, in Table 10 we provide the corresponding instructions for constitutive statements. Given the feature overlap between regulative and constitutive statements, we make reference to selected feature sets described in the context of regulative statements as part of the coding guidelines. As for regulative statements, symbols are color-coded to signal the association with features specific to IG Core, IG Extended or IG Logico. Symbols associated with IG Core features for constitutive statements are held in purple. As with regulative statements, symbols associated with IG Extended are held in green, and features associated with IG Logico are displayed in orange color.

4.3.1. IG Core Coding of Constitutive Statements

Level of Expressiveness: IG Core

IG Core enables basic, structural analysis of institutional statements. Encoding at this level is designed to be human readable and moderately comprehensive in the detail with which syntactic properties of institutional statements are captured.

Syntactic Component	**Treatment of Syntactic Components by Level of Encoding**	**Relevant Examples**	**Complete Syntactic Classification of Examples**
Constituted Entity	The encoding of the Constituted Entity, which reflects any entity created, modified or otherwise introduced into the institutional setting. Constituted entities can be of physical or virtual nature, reflect concrete or abstract concepts, typically including actors, roles, actions, and objects. Constituted entities can further be differentiated into entity and entity property. Note: The separation of Constituted Entity and the associated properties operates identical to the decomposition of Attributes and Attributes properties. See corresponding guidelines provided under the Attributes component specification (see here).	Example statement: *There is hereby established a public Food Security Advisory Board.* Entity = *Food Security Advisory Board* Entity property = *public*	*There is* **Cex***(hereby)* **F***(established) a* **E**,**p***(public)* **E***(Food Security Advisory Board).*
		Example statement: *No member of the Council shall be disqualified from holding any public office or employment.* While reflecting structural patterns of regulative statements, this statement parameterizes members with respect to rights in the context of the Council. Beyond the necessary components (**E**, **F** and implied Context), the substantive characteristics that do *NOT* apply (see negation applied) to the constituted entity are expressed as constituting properties.	**[NOT]***(No)* **E***(member) of the* **E**, **p***(Council)* **M***(shall)* **F***(be)* **P***(disqualified from holding any public office or employment).*
		Additional example: *Established in this Regulation subpart is the right to appeal to a revocation or certification.*	**F***(Established)* **Cex***(in this Regulation subpart) is the* **E***(right to appeal) to a* **E**,**p***(revocation or certification).*

Constituted Entity (ctd.)	Constitutive statements and Implied Attributes: In instances in which a coder is encountering ambiguity in discerning whether she is dealing with a constitutive or regulative statement, one shall consider the wider context of the statement (e.g., implied attribute, type of surrounding statements, etc.). A more detailed discussion can be found in Section 4.4.4.

Constitutive Function	The constitutive function characterizes the establishment, definition or introduction of a constituted entity into the institutional setting, and where constituting properties exist, functionally link constituted entity and constituting properties.	Example statement: *There is hereby established a public Food Security Advisory Board.* Constitutive Function: *[is] ... established* In this context the constitutive function signals the establishment of an entity. Example: *Commissioner of Agriculture and Markets shall be the Chairperson the Council.* Constitutive Function: *[serve as]* Here the constitutive function indicates a modified position (*Chairperson*) of a specific role (*Commissioner*) in a specific organizational context (*Council*). While diverse in nature, the constituting function can be sensibly organized along a set of patterns discussed in the context of IG Logico.	*There is* **Cex***(hereby)* **F***(established) a* **E**,**p***(public)* **E***(Food Security Advisory Board).* **P***(Commissioner of Agriculture and Markets)* **M***(shall)* **F***(be) the* **E***(Chairperson the Council).*

Constituting Properties	Constituting properties are optional components in constitutive institutional statements that capture elements functionally linked to the constituted entity by means of the constitutive function. Constituting properties may themselves have properties. Note: The separation of Constituting Properties and the associated nested properties operates identical to the decomposition of Attributes and Attributes properties. See corresponding guidelines provided under the Attributes component specification (see here).	Example: *The Committee shall consist of a President, Secretary, and Treasurer.* Constituting properties: *President, Secretary, and Treasurer* Here, the council is composed of the members as constituting properties. Example: *A majority of the members of the Council shall constitute a quorum.* Constituting properties: *majority of the members of the Council*	*The* **E***(Committee)* **M***(shall)* **F***(consist) of a* **P***(President, Secretary, and Treasurer).* *A* **P***(majority of the members of the Council)* **M***(shall)* **F***(constitute) a* **E***(quorum).*

Modal	The Modal signals the extent to which the instruction contained in the constitutive statement is required (necessary), or signals mere possibility, (effectively mirroring the prescriptive and discretionary notion on the regulative side). Linguistically, the expression can be ambiguous between the regulative and constitutive variant (e.g., must, may, must not), but the coder should make an attempt to classify the modal from an epistemic perspective (i.e., focusing on necessity and possibility of the activity expressed in the constitutive function). Note that in constitutive statements the use of the modal can follow legal traditions or be stylistic in kind. The coder should acquaint oneself with such specifics prior to the coding.	Example: *A majority of the members of the Council shall constitute a quorum.* Modal: *shall* Here *shall* signals the necessity that a quorum means "majority of the members of the Council" in the context of the policy.	*A* **P***(majority of the members of the Council)* **M***(shall)* **F***(constitute) a* **E***(quorum).*
		Example: *The Council may have an advisory committee.* Modal: *may* Here *may* signals the possibility (but not requirement) that the Council has an advisory committee.	*The* **E***(Council)* **M***(may)* **F***(have) an* **P***(advisory committee).*

Context	The encoding identifies the Context of the institutional statement. The encoding differentiates between "Activation Conditions," which are contextual clauses that specify preconditions under which the statement applies, and "Execution Constraints," which are contextual descriptors that qualify the constituting function by augmenting the statement with temporal, spatial, procedural, and/or other constraining parameters.	Example statement: *From 1st of January onward, food preparation guidelines must adhere to national standards, in addition to communal provisions.* Context clauses: *From 1st of January onward; in addition to communal provisions* Context encoding: Activation Condition: *From 1st of January onwards* Execution Constraint: *in addition to communal provisions* The activation condition signals an event that initiates a discretized setting in which the remaining statement holds. The execution constraint characterizes the constitutive function more explicitly.	**Cac***(From 1st of January onward)*, **E***(food preparation guidelines)* **M***(must)* **F***(adhere)* to **P***(national standards)*, **Cex***(in addition to communal provisions)*.

Or else

The encoding of Or else statements identifies consequences of non-fulfillment or violation of institutional statements, which, in the context of constitutive statements, can signal consequences addressing specific actors (e.g., consequences in regulative terms), or be of existential nature. Existential nature signals that an establishment or modification of an entity may simply not come about, or that alternative or linked constitutive statements are instantiated. The encoding captures these consequences generally in the form of institutional statements that nest on the leading monitored institutional statement, and can be expressed both in regulative or constitutive form (see Section 4.4 for conceptual details on mixed statement types).

Principles of horizontal and vertical nesting (see Section 2.2), as described in the regulative context (see Section 4.2), equally apply for constitutive statements, and within or else components.

Vertical nesting is applicable when there is at least one activity that is specified within a distinct institutional statement and activated as a consequence of a non-fulfilled action specified in the leading institutional statement. Horizontal nesting is applicable when there exist two or more activities that can be enacted in some logical combination as shown below.

Example:

In student recruitment plans, diversity must mean diversity in race, religion, sexual orientation and gender, or else plan is void.

Or else clause comprising statement: *or else plan is void*

The Or else signals an existential consequence for the constituted entity as expressed by the necessity that diversity is defined as prescribed (*"must mean"*).

Naturally, the consequence can also consist of multiple statements that are logically combined (horizontal nesting), as shown below.

Example:

In student recruitment plans, diversity must mean diversity in race, religion, sexual orientation and gender, or else plan is void and to be revised within 30 days.

Cac*(In student recruitment plans),*
E*(diversity)* **M***(must)* **F***(mean)*
P*(diversity in race, religion, sexual orientation and gender),*
or else
O{**E***(plan)* **F***(is)* **P***(void)*}.

Cac*(In student recruitment plans),*
E*(diversity)* **M***(must)* **F***(mean)*
P*(diversity in race, religion, sexual orientation and gender),*
or else
O{{**E***(plan)* **F***(is)* **P***(void)*}
[AND]
{*. . .* **E***(plan)* **F***(is . . . to be revised)*
Cex*(within 30 days)*}}.

In the previous example, horizontal nesting is signaled using braces around statements (as opposed to individual components), and vertical nesting is likewise expressed using braces ({ and }) scoped around the compound consequence.

Or else (ctd.)

These statement combinations can signal

- alternative exclusive action options – **[XOR]**s – (e.g., either initiate XOR expand the Council),
- inclusive action options – **[OR]**s – (e.g., rights may be assigned AND/OR responsibilities removed when an individual is added to the Council) or
- co occurring action options – **[AND]**s – (e.g., dissolving a Council AND compensating Council members)

Note:

1. Where component combinations exist, alternatives are combined (... are void and to be refined fine farmer or revoke ...), and are subsequently decomposed into separate logically-combined complete atomic institutional statements.
2. Ambiguities with respect to the linguistic use of logical operators (exclusive and inclusive or) are to be resolved as part of this process.

General IG Core Instructions for Constitutive Statements

Additional annotations for Constituted Entity, Constitutive Function, Constituting Property and Context (Note: This feature is analogous to the specification of additional annotations for Attribute, Object and Context components in the context of regulative statements.)	In addition to the identification of properties embedded in the original statements, components can further be annotated using additional annotation labels. Such labels can follow the categories listed in Section 5, or be specific to the project objectives. A systematic approach to labelling entities is discussed under *IG Logico Instructions*, Item *Cross-Component Semantic Annotations* in Table 9. This is particularly recommended if annotations are of strong relevance for the coding and of diverse nature.	Example: *The Committee shall consist of a President, Secretary, and Treasurer.* Subject to analytical necessity, additional annotations can for instance relate to the identification of aspects, such as the characterisation of encoded objects with respect to their animacy as either animate or inanimate – signified in brackets in the coded example. Where indicated, the annotation should be separated from the component specification by semicolon and have the structure "label=", followed by the annotation. Note that "*label*" is a characterizing prefix, either based on a specific taxonomy (see Section 5 for an overview), or a custom coder-defined category specification (e.g., defined as part of the project-specific guidelines). In this example, the Animacy Taxonomy has the prefix "anim" as defined in Section 5 and is used correspondingly. While exemplified here for constitutive statements, this equally applies to regulative statements.	*The* **E[anim=inanimate]***(Committee)* **M***(shall)* **F***(consist of)* *a* **P[anim=animate]***(President, Secretary, and Treasurer).*

Decomposition of component-level combinations

(Note: This applies to regulative and constitutive statements, and is discussed here with focus on the constitutive perspective.)

Where combinations of components (component-level combinations) are observed that are not explicitly decomposed as in the case of vertical nesting, these can be decomposed into logically-combined statements. Other than for constitutive functions, the decomposition is optional for IG Core.

Operationally, combinations of components are evidenced by the presence of multiple logically-combined tokens or clauses embedded in constituted entities, constitutive functions, constituting properties or context components.

Decomposition essentially entails constructing an individual statement to capture each of the unique components represented in multiples within institutional statements, noting the relation to the original statement in which multiple components are reflected. Information from component fields, other than that containing multiple components, is simply carried over to all related institutional statements.

Importantly, where decomposition actually changes the meaning of the original institutional statement containing multiple components within a particular syntactic field, the statement should not be decomposed. In such cases, multiples are typically intended to exist in coupled form. An example is provided in the next column.

Details are described in the *General IG Extended Instructions*, Item *Decomposition of component-level combinations* in Table 8.

Example (Multiple Properties):
The Committee shall consist of a President, Secretary, and Treasurer.

While expressed in condensed form as "*The* **E***(Committee)* **M***(shall)* **F***(consist of) a* **P***(***(***President* **[AND]** *Secretary* **[AND]** *Treasurer***)***)*" (note the parentheses),

it corresponds to the following statement composed of three atomic statements:

Statement 1: *The Committee shall consist of a President*

[AND]

Statement 2: *The Committee shall consist of a Secretary.*

[AND]

Statement 3: *The Committee shall consist of a Treasurer.*

Example (Multiple constitutive functions): *The form and function of the Council is hereby established.*

Condensed form:
The **E***(Committee)* **M***(shall)* **F***(consist of) a* **P***(***(***President* **[AND]** *Secretary* **[AND]** *Treasurer***)***).*

Expanded form:
{{*The* **E***(Committee)* **M***(shall)* **F***(consist of) a* **P***(President)*}
[AND]
{*The* **E***(Committee)* **M***(shall)* **F***(consist of) a* **P***(Secretary)*}
[AND]
{*The* **E***(Committee)* **M***(shall)* **F***(consist of) a* **P***(Treasurer)*}}.

{{**E***(Council form) is* **Cex***(hereby)* **F***(established)*}
[AND]
{**E***(Council function) is* **Cex***(hereby)* **F***(established).*}}

Decomposition of component-level combinations (ctd.)	Note: These guidelines highlight the motivation for the decomposition, and exemplify it explicitly. Depending on the use of annotation means and tool support, the decomposition may be partially automated, affording a mere annotation for such decomposition without requiring the user to perform statement duplication.

Table 10: Coding Guidance on Syntactic Elements for IG Core as Level of Expressiveness (Constitutive Statements)

4.3.2. IG Extended Coding of Constitutive Statements

Level of Expressiveness: IG Extended

Mirroring the progression on the regulative side, IG Extended enables more detailed structural analysis of institutional data than IG Core and accommodates computational application to aid in institutional coding and analysis. Encoding at this level is designed to be human readable, moderately computationally tractable, and moderately comprehensive in the detail with which syntactic properties of institutional statements are captured.

Coding institutional statements on this level enforces many of the features that have been optional in IG Core and affords a fine-grained decomposition of statements. This includes a richer context characterization based on predefined taxonomies, the expansion and combined attributes and aims that reconstruct atomic statements and their relationships, but also decomposes the hierarchical relationships amongst explicitly highlighted Constituted Entities, Constituting Properties and Constitutive Functions, alongside further refinements of contextual descriptors.

As a central feature IG Extended makes the use of component-level combinations explicit. This specifically facilitates the decomposition of the context component to express institutional content at a more nuanced level. In addition, structural refinements relate to the decomposition of relationships and properties of Constituted Entities and Constituting Properties.

For constitutive statements, IG Extended features correspond to the regulative side, with the essential difference for the application of refinements on Attributes and Objects, which, in the context of constitutive statements apply to Constituted Entities and Constituting Properties.

A specific consideration is the concept of Constitutive-Regulative Hybrids and syntactic polymorphs, both of which are of cross-cutting nature (i.e., affecting both constitutive and regulative statements) and thus discussed in a dedicated section. Their consideration, however, applies to IG Extended.

Syntactic Component	Treatment of Syntactic Components by Level of Encoding	Relevant Examples	Complete Syntactic Classification of Examples

Constituted Entity	In IG Extended encoding, Constituted Entities and their properties are decomposed hierarchically following the principles of the Object-Property Hierarchy (Section 2.3) and is applied analogous to "Attributes" in IG Extended for regulative statements (Table 8). The ensuing overview of constituted properties shows an extended set of examples applied analogously to constituted entities.	Example: *There is hereby established a standing public Food Security Advisory Board.*	*There is* **Cex[ctx=met]***(hereby)* **F***(established) a* **E,p1***(standing)* **E,p2***(public)* **E***(Food Security Advisory Board).*

Constituting Property	Analogous to the decomposition of object properties in the context of regulative statements, constituting properties are likewise decomposed following the principles of the Object-Property Hierarchy (as introduced in Section 2.3 and applied in the context of "Objects" in IG Extended for regulative statements in Table 8).	Example: *The Council consists of elected officials resident in the electorate.* In this example, the individual properties of the constituting property *officials*, namely *elected* and *resident in the electorate*, are uniquely identified as properties. Another feature is the richer hierarchical structure embedded in phrase expressing compound property characterizations. Example: *A majority of the members of the Council shall constitute a quorum.* In this example, the constituting property is captured in the phrase *A majority of the members of the Council*. While the entire phrase represents the constituting property (and is coded as such on IG Core), the embedded hierarchy, i.e., members are a property of the Council, and the majority is a property of the members, can be explicitly captured using hierarchical property annotations as shown on the right. Where properties are not functionally dependent on another property, they are signaled using unique identifiers (e.g., **Pa**, **Pb**) equivalent to "Object" decomposition highlighted in Table 8 and exemplified in the following.	*The* **E***(Council)* **F***(consists of)* **P**,**p1***(elected)* **P***(officials)* **P**,**p2***(resident in the electorate).* *A* **P**,**p1**,**p1***(majority) of the* **P**,**p1***(members) of the* **P***(Council)* **M***(shall)* **F***(constitute) a* **E***(quorum).*

Constituting Property (ctd.)	Collections of functionally independent entities are represented as a compound constituting property signaled by parentheses. Individual compound properties can be uniquely identified, alongside potential further properties shared across all embedded entities.[16] Example: *The Committee shall consist of a President, Secretary, and qualified Treasurer appointed by the public.* In this example, properties specific to an entity are called out with reference to the entity (*qualified*), whereas shared properties are associated with all entities (*appointed by the public*).	*The* **E***(Committee)* **M***(shall)* **F***(consist of) a* **P***(***(P1***(President)* **[AND]** **P2***(Secretary)* **[AND]** **P3**,**p1***(qualified)* **P3***(Treasurer))) P*,**p***(appointed by the public).*
Context	See "Context" in IG Extended for regulative statements (Table 8)	
General IG Extended Instructions	See "General IG Extended Instructions" in IG Extended for regulative statements (Table 8)	
Constitutive-regulative Hybrids	The introduction of constitutive statements as part of IG 2.0 (see Section 4.3) provides the basis for encoding statements that consist of structural elements both of regulative and constitutive statements. Details are discussed in Section 4.4.	

Table 11: Coding Guidance on Syntactic Elements for IG Extended as Level of Expressiveness (Constitutive Statements)

[16] Specific data structure patterns commonly found in institutional statements are revisited in Section 4.5.

4.3.3. IG Logico Coding of Constitutive Statements

Level of Expressiveness: IG Logico

IG Logico is designed to support semantic analysis of institutional statements wholly relying on computational tools. Encoding at this level is designed to be moderately human readable, computationally tractable and comprehensive in the detail with which syntactic properties of institutional statements are captured.

In contrast to IG Core and Extended that focus on the encoding of specific grammar components, IG Logico emphasises refinements across individual components and further establishes explicit references to related statements to establish computational tractability, as well as the ability to perform logical transformations on institutional statements.

While largely equivalent for regulative and constitutive statements, the only variant to the instructions provided in the context of regulative statements is the discussion of Constitutive Function taxonomies (as opposed to Institutional Functions in the context of regulative statements) as outlined below.

Syntactic Component	Treatment of Syntactic Components by Level of Encoding	Relevant Examples	Complete Syntactic Classification of Examples
Constitutive Function Annotations	Complementing the content characterization for other components, the constitutive function maintains the central role as a descriptor of constituted entities, and where constituting properties exist, links those to constituted entities. In an attempt to characterize the function of the constitutive statement as expressed in the constitutive function more generally, we propose a taxonomy capturing common relationships more generally. Doing so, we differentiate between statements that characterize the constituted entity as newly introduced into the institutional setting, and a commonly found alternative, that is, the characterization of the policy that contains the statements itself. Entities, such as novel actors, objects, roles or action, can be • defined explicitly ("is", "does"), • defined based on relationships, such as composition ("consists of"), organizational embedding ("is embedded in", "relates to"),	Example: *Starting January 1st, the Connecticut Food Policy Council shall be within the Department of Agriculture.* In this example the constitutive function signals the constituted entity (Connecticut Food Policy Council) as an organizational unit. Example: *The Committee shall consist of a President, Secretary, and Treasurer.* The constitutive function signals a composition of the constituted entity (*Committee*) based on constituting properties. Example: *The purpose of this Part is to establish standards for net metering in accordance with the requirements of Section 16-107.5 of the Act.* In this example, the constitutive function identifies the entity as a policy and signals the intent underlying the policy.	**Cac**(*Starting January 1st*), *the* **E**(*Connecticut Food Policy Council*) **M**(*shall*) **F**[**confunc=organization**](*be within*) *the* **Cex**(*Department of Agriculture*). *The* **E**(*Committee*) **M**(*shall*) **F**[**confunc=composition**](*consist of*) *a* **P**(**(***President* **[AND]** *Secretary* **[AND]** *Treasurer***)**).

Constitutive Function Annotations (ctd.)	• defined based on lifecycle stages ("established", "terminated"), and finally • characterized by the conferral of status in the form of rights, authority, or exertion of institutional power more generally. Policies as constituted entities in institutional statements, in contrast, are generally referred to with respect to the • lifecycle stage they are involved in ("come into force"), • relationship between and to other statements or policies ("amends", "substitutes"), • intent in the form of purpose of a specific policy, and appear as • information statements that offer information about the policy itself. Naturally, these characterizations are not exhaustive and can carry more specific forms. An overview of the different characterizations, alongside the labels used in this context is provided in Section 5.	Example: *In department's university plan, diverse population means diversity in religion, sexual orientation and race.* In this example, the constituted entity is defined intensionally, that is in terms of its underlying interpretations.	**Cac[ctx=dom]**(*In department's university plan*), **E**(*diverse population*) **F[confunc=definition]**(*means*) **P**(**(***diversity in religion* **[AND]** *sexual orientation* **[AND]** *race***)**).

Table 12: Coding Guidance on Syntactic Elements for IG Logico as Level of Expressiveness (Constitutive Statements)

4.4. Hybrid & Polymorphic Institutional Statements

In addition to the specific treatment of regulative and constitutive statements as part of the coding guidelines, an aspect that demands specific attention is the combined use of both statement types. While distinctively different in their function, regulative and constitutive statements, of course, share structural patterns as outlined in the context of the operational coding across varying levels of expressiveness.

However, an operational concern that links both statement types is the interleaved use in practice. In addition to the commonly found organization of primarily constitutive and regulative statements into distinctive sections of documents (e.g., constitutive statements as part of the preamble), in regulative statements we may encounter inline specifications of entities that are positioned in the institutional setting and are thus of relevance for subsequent statements. Conversely, in constitutive statements, we can potentially encounter embedded regulative elements that regulate behavior of the constituted entities. Linking the nested relationships of institutional statements across both types, we recognize the concept of *hybrid institutional statements*, in which the combined use of constitutive and regulative statements can either take constitutive-regulative form (where the overall statement is of constitutive nature) or be a regulative-constitutive hybrid (where the leading statement is of regulative nature). Where existing, their resolution is a central feature of IG Extended onwards (and optional for IG Core).

In contrast to statements that embed a combination of parameterizing and regulating features, in specific instances statements may be encodeable in both regulative and constitutive form. While such variable coding often reflects a concession to analytical objectives, potential coding inconsistencies can be resolved by a priori specification of the interpretational scope applied in the coding process (see Section 2.5). Where a generic coding of statements, agnostic of analytical uses, is desired, statements can be coded as *polymorphic institutional statements* that allow the concurrent encoding in different statement forms. This special form of institutional statements is discussed in Section 4.4.4.

Initially, however, we will exemplify both variants of statement hybrids.

4.4.1. Regulative-Constitutive Statements

A typical reflection of hybrids stems from the introduction of novel entities as part of a regulative statement, as shown in the following example (Figure 11):

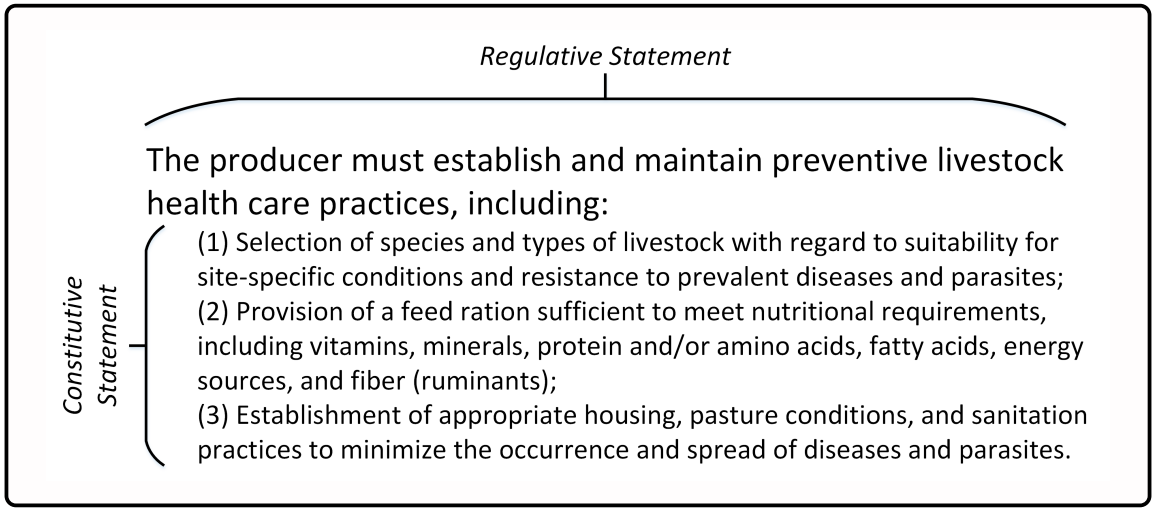

Figure 11: Regulative-Constitutive Hybrid Example

As signaled visually, this example highlights a regulative statement capturing an actor's obligations, with the latter defined in an embedded constitutive statement, reflecting a regulative-constitutive hybrid.

In this example, the constitutive statement is nested in a specific component of the regulative statement, such as the object as shown in the example in Figure 12. Note that the following figures use the same color-coding used in the preceding sections: Symbols associated with IG Core features for regulative statements are displayed in blue, whereas symbols signaling constitutive statements are held in purple.[17]

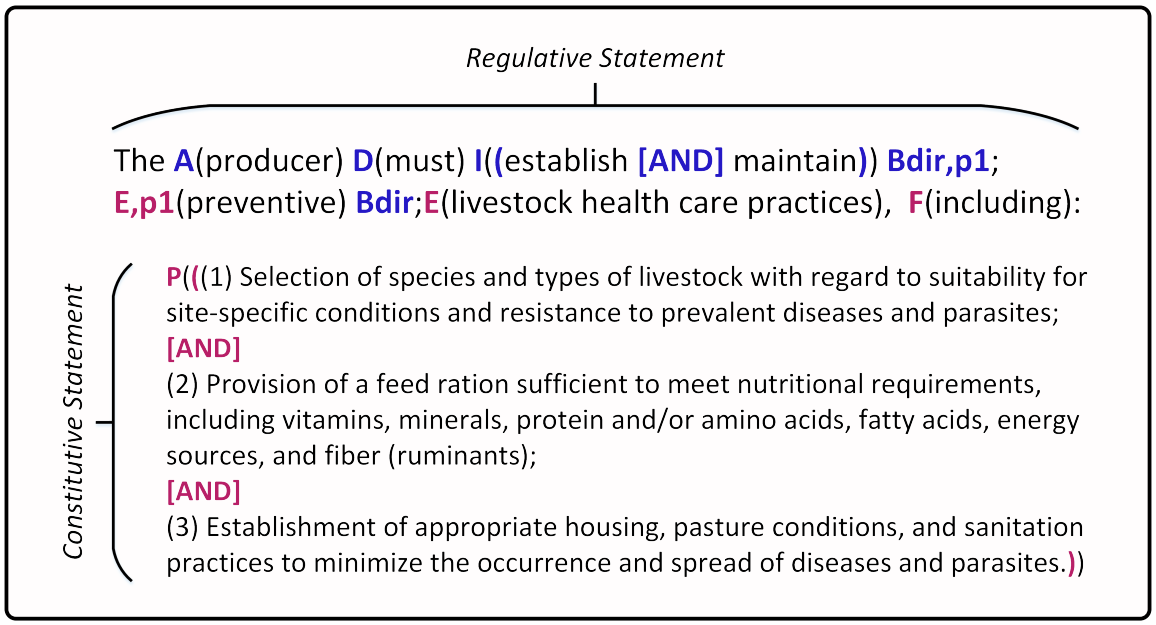

Figure 12: Coded Regulative-Constitutive Hybrid Example

In many instances (such as the one shown here), this interleaved representation affords a decomposition or transformation of hybrids into individual statements by separating the statements by syntactic components, and replication of overlapping components in the individual statements. This decomposition is exemplified in Figure 13.

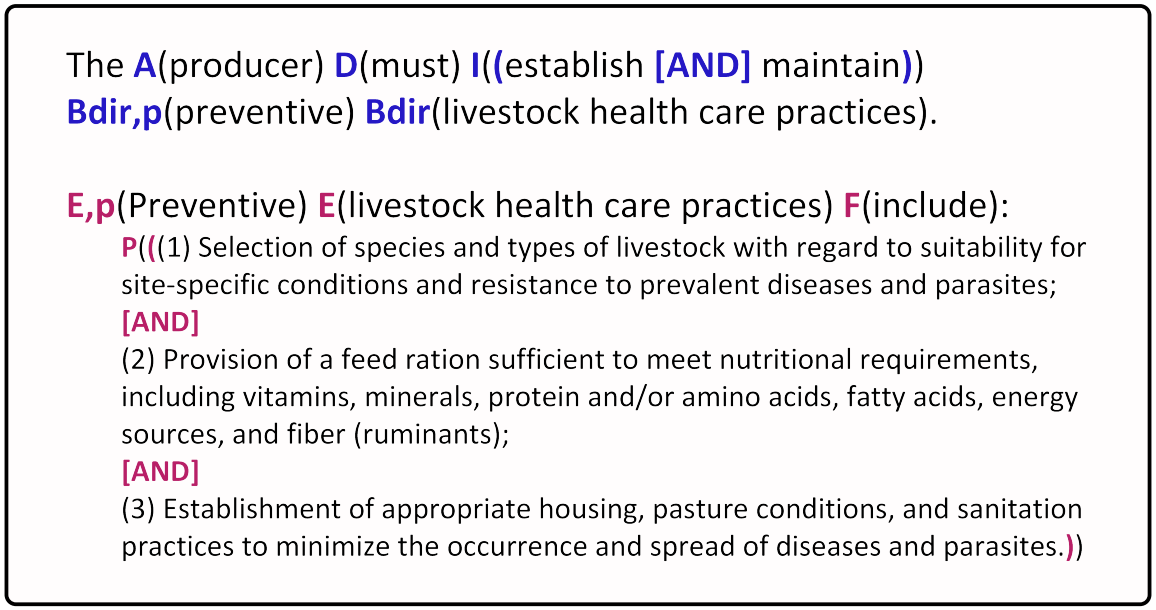

Figure 13: Decomposed Regulative-Constitutive Hybrid Example

[17] In the context of this section, parentheses and logical operators are color-coded to emphasize the association with the corresponding institutional statement type.

4.4.2. Constitutive-Regulative Statements

Contrasting the embedding of constitutive statements in regulative settings, we can likewise observe the embedding of regulative statements in constitutive ones. In the following example (Figure 14), the consequence of breaching a constitutive statement[18] is expressed in regulative terms, following the principles of statement-level nesting. While the leading statement is coded as a constitutive statement, the consequences are a combination of regulative statements.

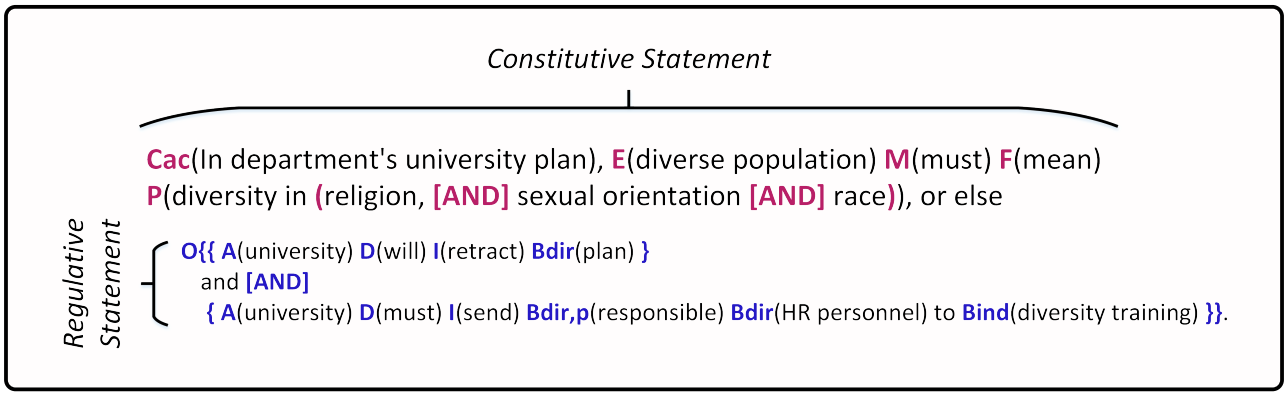

Figure 14: Constitutive-Regulative Hybrid Example

4.4.3. Second-Order Constitutive Statements

In addition to the combined characterisation of constitutive and regulative hybrids, we can further observe component-level nesting of constitutive statements as shown below. While, in principle, equally admissible for regulative statements, specifically the higher-order decomposition of constitutive statements is commonly found. Higher-order decomposition thereby implies the nesting of constitutive statements within individual components, such as property items. Naturally, as motivated in the earlier example, this can occur in conjunction with hybrid statements and independent of the regulative or constitutive nature of the leading institutional statement.

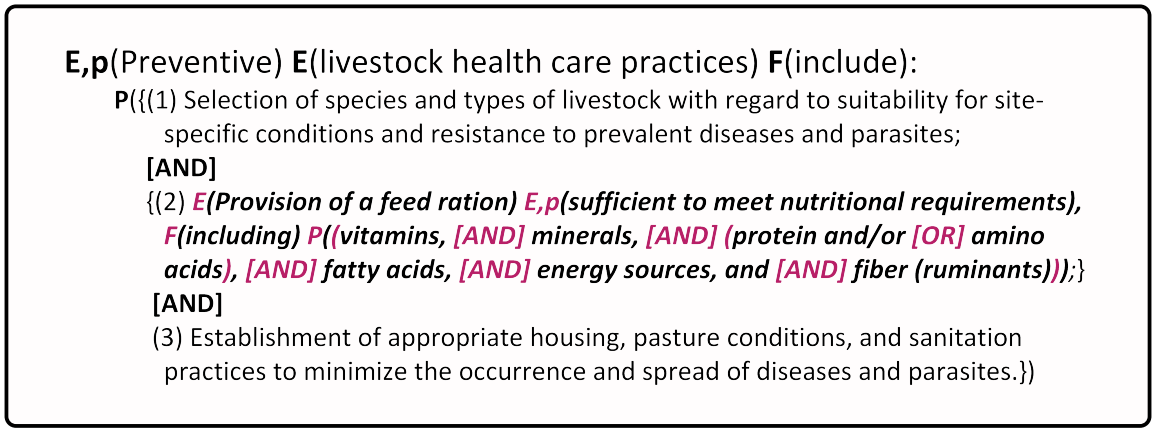

Figure 15: Second-order Constitutive Statement Example

Reviewing the example above (Figure 15, second-order statement in italicized bold font), we note the reference to feed rations as a property element of livestock health care practices that is defined in

[18] This statement signals the requirement that diversity is understood as specified.

terms of an embedded constitutive statement that in itself captures a complex set of properties that constitute a feed ration.

4.4.4. Polymorphic Institutional Statements

Another aspect of discussion of the combined nature of statements relates to the identification of constitutive and regulative statements. The function of either statement type is generally well defined based on the parameterizing role (e.g., introduction or modification of entities or endowment of rights/authority) of constitutive statements, and the behavioral regulation (specification of operational duties and constraints for given actors or actor interaction) captured by regulative statements. While the heuristics provided in Section 2.5 lend support for an unambiguous characterization of institutional statements as either constitutive or regulative, the characterization may not for all instances be non-contentious, or may make tacit analytical (or coding) preferences overt. While such preferences should inform the coding process by specifying the interpretational scope applied to individual institutional statements (see Section 2.5), where the data set is to be encoded in generic form (i.e., without concessions to specific analytical objectives), statements can be considered *polymorphic institutional statements*, or *syntactic polymorphs*. This means, they are encoded in both forms, and the analyst can draw on either shape depending on their desired analytical use.

To substantiate this approach with an example, let us use the statement

"The functions of the Board shall be: (a) give effect to the decisions and policies of the Health Assembly; (b) perform any other functions entrusted to it by the Health Assembly."

This statement can be read in constitutive terms, since it parameterizes the institutional system with respect to a specific entity, namely the characterization of the board in terms of its assigned functions and endowed responsibilities. The statement, however, can also be read in regulative terms, in which the functions of the board are expressed as obligations (and thus as regulated behavior) in operational terms. The same statement can thus be encoded as a polymorph by drawing on syntactic components of both statement types, as visualized in the following (including color coding to signal regulative (blue) and constitutive components (purple), respectively).

The **E***(functions) of the* **A***(Board)* **D**/**M***(shall)* **F***(be):*
P({
(a) **I***(give)* **Bdir***(effect) to the* **Bind***(decisions) of the* **Bind**,**p***(Health Assembly);*
[AND]
(b) **I***(perform)* **Bdir***(any other functions)* **Bdir**,**p***(entrusted to it by the Health Assembly)*
}**)**.

The coding shows overlap, but the central components for regulative and constitutive statements (namely constituted entity, attributes, constitutive function, aim, as well as constituting properties) showcase the varying focal aspects of different statement types. Subject to emphasis on either the actor (i.e., 'the Board') or its functions (i.e., 'the functions') can determine the coding, or even admit both approaches for analytical purposes.

A specific value of the polymorphic representation is to make the linkage between configurational aspects that affect the wider institutional setting, and behavioral regulation that specifically embeds actors in this institutional setting, both configurationally, as well as operationally, explicit.

While the notion of polymorphic structure is explicit in the previous example, the coding of statements in both forms can be more complex and may even afford reconstruction, as shown in the following example, with the same statement coded separately in constitutive and regulative forms: *"Members of the Council shall have the right to hold any public office or employment."*

In constitutive terms the statement reflects the Council members endowment with a right, namely the right to hold other public positions in addition to a Council membership, and is encoded as follows:

> **E***(Members) of the* **E**,**p***(Council)* **M***(shall)* **F***(have the right)* **P***(to hold any other public office or employment).*

Here, the Council member is at the center of the encoding. Depending on the analytical use (e.g., assumed actor perspective), constitutive statements reflect an abstract conception of a right or a property. Expressing this assurance in regulative terms by invoking the principle of *jural correlatives* (Hohfeld, 1913) requires the representation of behavioral constraints by introducing a conception of actorship in the form of a tacit (and in this case unidentified) actor whose behavior is regulated as corresponding duty, alongside further structural adaptations. Reconstructing the statement in regulative terms[19] thus produces the following coding:[20]

> **A***([Attribute])* **D***([must])* **[NOT]** **I***(disqualify)* **Bdir***(member of the council)* **Cex***(from holding any other public office or employment).*

Exemplifying the use of polymorphic structures and their practical implications further, the following statement constellation highlights interpretational challenges associated with statement references.

Assessing the set of statements shown below, we specifically turn our attention to the final statement. This statement is embedded within a paragraph of statements that regulate operational activity, but this statement specifically introduces further constraints that apply to all other statements.

Encoding this statement using a *narrow interpretational scope* (as described in Section 2.5), the statement can be interpreted as parameterizing the wider institutional setting by affecting a series of provisions ("Paragraphs in this section") and furthermore clearly signaling the requirement that the paragraphs do not apply (see *Deontic vs. Epistemic Modal* heuristic in Table 4 in Section 2.5). Consequently, the statement is coded as constitutive (as exemplified below).

> *(1) Organic farming operations must not utilize genetically-modified seeds.*
> *(2) Organic farming operations may not process crops other than the ones specified in Appendix A.*
> *. . .*
> *(10)* **E***(Paragraphs in this section)* **M***(do)* **[NOT]** **F***(apply)* **P***(to traces)* **P**,**p***(of genetically modified material).*

However, given the evident regulative nature of the referenced statements, coding such statement with *wide interpretational scope* in mind leads to a resolution of the semantic links to the referenced statements, and doing so, affords a reconstruction of the statement in regulative form – rendering the following encoding (simplified for illustrative purposes):

> **A***(Organic handling operations)* **D***(may)* **[NOT]** **I***(apply)* **Bdir***(paragraphs in this section) to* **Bind***(traces)* **Bind**,**p***(of genetically modified material).*

Similar to the previous case, the coding requires reformulation of the statement in regulative terms. However, unlike the previous case, the specification of the responsible actor is explicit in the referenced

[19] Explicit transformation rules associated with the reconstruction of statements of various types are subject to forthcoming work.

[20] Note that the following coded samples call out negations of deontics and modals explicitly for clarity.

institutional statements, thus affording a reliable encoding in a form that is aligned with the syntactic structure of the related institutional statements. As indicated for all cases above, the coding of such statement can selectively take regulative and constitutive form where interpretational scope favors narrow and wide resolution of semantic linkages. As showcased for the first example, where coding is independent from analytical use, the statement can be concurrently encoded in both forms.

Note: As discussed in the context of the interpretational scope applied by the coder (see Section 2.5), encoding statements in constitutive and/or regulative forms may invoke varying levels of complexity with respect to necessary adaptations of the statements, and, depending on analytical objectives, potential modifications of the statement semantics (in the given example, a concrete actorship for regulative statements is presumed, whereas the right more generally is expressed in constitutive terms). This can require a reformulation that could be challenged on methodological grounds and thus considered during design of the study. In addition to considering a dual annotation in the first place, the potential mischaracterization of a statement as either regulative or constitutive may be sensibly assessed in small-scale inter-coder reliability tests to resolve disagreements and misconceptions early in the encoding process.

4.5. Data Structure Patterns

Note that this section is under ongoing development and will likely experience significant extension in upcoming releases.

Another aspect related to the features introduced for both regulative and constitutive statements is the recognition of data structure patterns. A pattern commonly observed is the notion of collections, such as composition of committees, definition of practices in terms of underlying activities, delineation of goals or outcomes to be pursued, etc.

Collections: While dominant in objects and constituting properties, respectively, those can occur across other components (e.g., context). Central here is the identification of a collection descriptor and the corresponding elements, along with the explicit specification of logical operators that link the individual elements.

Example: **E***(Health care practices)* **F***(consist of)* **P***(***(***preventative measures [***[AND]***] appropriate nutrition [***[AND]***] rest***)***)*.

Complex elements: Where quantitative information is expressed, or listed (and thus potentially embedded in collections or referred to as a single item), the elements commonly follow a schematic structure specific to individual documents (e.g., based on style or disciplinary background), but follow general patterns, such as variations of the following: **[qualifier] [comparator] [quantity] [unit] [object property] [object]**.

Example: *significantly (qualifier) more than (comparator) 10 (quantity) tons (unit) high-quality (object property) building material (object)*

While those patterns can vary in extent and detail, their consideration in project-specific coding guidelines can be useful in as far as they are relevant for analytical purposes.

Comparisons/Equations: In many instances, statements, and more specifically, activation conditions, are expressed in terms of comparisons, such as "if gender distribution within department staff substantially differs from distribution within the wider organization, ...". Comparisons of such kind can be expressed in equation form, following the general pattern **[left-hand side] [comparison operator] [right-hand side]**, where left- and right-hand sides can be of simple form (e.g., numeric) or complex (e.g., consisting of complex elements as introduced above). Comparison operators include the following: $<$, $\leq$, $\lesssim$, $\approx$, $=$, $\gtrsim$, $\geq$, $>$, amongst others such as $\neq$. Oftentimes, comparators can further be complemented by additional qualification of the relationships.

Example: *if gender distribution (object) within department staff (object property) substantially (qualifier) differs (≠) from distribution (object) within the wider organization (object property)*

Naturally, generic mathematical expressions are likewise conceivable, but not subject to further elaboration at this stage. Where coding on finer levels of granularity (e.g., resolving comparative relationships) is relevant, level and specific form of decomposition should be considered as part of the study design.

4.6. IG Configurations (Institutional Grammar Profiles)

As discussed before, the various coding levels of IG 2.0 accommodate different analytical needs as well as complexity of the encoded documents. Commitment to a level, including all the associated features, may in some cases be too coarse-grained to accommodate analytical needs. This can include, for example, the omission of components entirely, as well as the selective considerations of features of higher coding levels. To capture the considered feature set, a coded dataset should be accompanied with the applied coding configuration.

For this purpose, we specify a fine-grained configuration syntax that allows the choice of features across levels of IG 2.0. The features for individual levels as specified in this document are captured in Table 13 for regulative statements, and Table 14 for constitutive statements, along with a symbol specification used for the definition of specific IG configurations, or profiles.

Coding Level	Feature	Symbol
IG Core	Attributes	A
IG Core	Object[1]	B
IG Core	Deontic	D
IG Core	Aim	I
IG Core	Context[2]	C
IG Core	Or else	O
IG Extended	Attributes refinements	A_{Ext}
IG Extended	Object refinements[1]	B_{Ext}
IG Extended	Context refinements[2]	C_{Ext}
IG Logico	Statement references	R
IG Logico	Logical relationship annotations	L
IG Logico	Semantic annotations	S
IG Logico	Regulative function annotations[3]	U_{reg}

[1] Where the configuration is specific to direct or indirect object component, those can be indicated using Bdir and Bind, respectively; where applied in combination with the specification of IG Extended features, '$_{Ext}$' is appended accordingly (e.g., $Bdir_{Ext}$).

[2] Where the configuration is specific to activation condition or execution constraint component, this can be indicated using C_{ac} and C_{ex}, respectively; where applied in combination with the specification of IG Extended features, '$_{Ext}$' is appended accordingly (e.g., $C_{ac,Ext}$).

[3] Where institutional functions are annotated for both regulative and constitutive statements, the symbol U (without subscript) is used.

Table 13: IG Feature Specifications for Regulative Statements

Using coding levels, along with − and + symbols in combination with specific features references as listed in the table, we can express specific coding configurations, or coding profiles, that allow the omission or inclusion of features across all levels, or the selective coding of specific components based on lower coding levels.

Abstractly specified, a configuration has the following structure (where < and > embeds symbol characterizing features omitted from or added to the baseline coding level):

<**Baseline level**>−<**omitted features from baseline level**>+<**additional features from higher level**>

Examples: To capture the commitment to IG Core, along with the Context coding from IG Extended (e.g., component-level nesting, use of taxonomies, is specified as the configuration **IG Core+C_{Ext}**, where the **+C_{Ext}** signals features from the next higher level (IG Extended).

Coding Level	Feature	Symbol
IG Core	Constituting Properties	P
IG Core	Modal	M
IG Core	Constituted Entity	E
IG Core	Constitutive Function	F
IG Core	Context[1]	C
IG Core	Or else	O
IG Extended	Constituting Properties refinements	P_{Ext}
IG Extended	Constituted Entity refinements	E_{Ext}
IG Extended	Context refinements[1]	C_{Ext}
IG Logico	Statement references	R
IG Logico	Logical relationship annotations	L
IG Logico	Semantic annotations	S
IG Logico	Constitutive function annotations[2]	U_{con}

[1] Where the configuration is specific to activation condition or execution constraint component, this can be indicated using C_{ac} and C_{ex}, respectively; where applied in combination with the specification of IG Extended features, '$_{Ext}$' is appended accordingly (e.g., $C_{ac,Ext}$).

[2] Where institutional functions are annotated for both regulative and constitutive statements, the symbol U (without subscript) is used.

Table 14: IG Feature Specifications for Constitutive Statements

Conversely, we can specify coding on IG Core level as the baseline, but without the consideration of Or else components as **IG Core–O**. Where multiple components are omitted, we can specify **IG Core–IO**, where features should be referred to in the order specified in the table (here: *Aim* before *Or else*). Selectively capturing features from IG Logico in IG Core-based coding, **IG Core+R** indicates the coding of statement relationships in addition to the base IG Core coding.

Finally, omission and extensions can be combined, with omissions specified first, followed by feature additions, such as **IG Extended-BC_{ex}+SU_{reg}**, to signal the coding of Object and execution constraints on IG Core level, while considering semantic annotations and institutional functions (here for regulative statements only) in addition to this (reduced) IG Extended baseline. Complementing this discussion for the highest level, **IG Logico-S** would imply complete coding on IG Logico level under omission of semantic annotations only.

Combining both omission and extension leverages complete flexibility with respect to the composition of features, and, where analytically useful, in principle even foregoing the inclusion of components defined necessary for institutional statements (i.e., A, I, and C component for regulative statements; F, E, and C for constitutive statements). For example, modeling the selective omission of components entirely, along with the inclusion of advanced features, **IG Core-AI+C_{Ext}SU** signals IG Core baseline encoding under omission of Attributes and Aim, while adding refined coding of Context (based on IG Extended), along with features from IG Logico, namely semantic annotations and institutional functions (here both for regulative and constitutive statements).

A common encoding level that offers the smallest possible extension to previous coding practice of institutional statements based on Crawford and Ostrom's original grammar is **IG Core+C_{Ext}**.

Custom refinements: Where coders seek more fine-granular refinements (e.g., applying a subset of the features of a given configuration (e.g., coding objects without properties), such modifications should be indicated alongside the specified configuration. Similarly, extensions (e.g., additional taxonomies, or extensions or substitution of existing ones) should be documented alongside the configuration.

5. Taxonomies

This section provides an overview of the taxonomies for the categorization of components, parts thereof, or annotation schemes including (but not limited to) the ones referred to from the coding guidelines in Section 4. The overview is largely summarizing, with essential specification of labels, but limited conceptual elaboration, which is provided in the corresponding guidelines (Section 4). The taxonomies further specify the *label prefixes* used to ensure unambiguous reference to the respective taxonomy/ies. Where only the circumstances taxonomy is used, the use of labels is optional.

The extension of existing taxonomies and introduction of additional taxonomies to draw seek epistemological linkage to the field of concern (e.g., to accommodate domain-specific characteristics or analytical necessities) is explicitly permitted. This includes the use of ontologies to support conceptually richer classification of entities. Any adjustment should be clearly indicated as part of the dataset specification (see Section A).

Note that the taxonomies in this section are subject to ongoing empirical evaluation and may experience refinement in the future.

5.1. Context Taxonomy

The Context Taxonomy captures contextual characterizations with respect to temporal, spatial and various further descriptors that capture institutional context more accurately. It is a systematic extension of the descriptors of the *Conditions* component as highlighted in the original grammar specification (Brady et al., 2018). Note that the listed categories include an embedded hierarchy, with more specific labels indented.

Specific considerations:

- Where possible (and analytically useful/specified in project-specific guidelines), the most specific annotation of a given category should be used (e.g., 'point in time', as opposed the more general 'temporal' category).
- Note that selected categories are more general in nature (e.g., the 'state' category, since it potentially captures temporal, spatial and domain characterizations). As before, the coder should attempt to classify context in more specific terms.
- Multiple annotations are explicitly suggested in cases where the characterizations overlap for an annotated element (e.g., 'temporal' and 'event'). Where possible, the coder may further consider the separation of distinctive activation conditions/execution constraints by category and annotate those correspondingly.

The suggested annotation label prefix – if applied – is *ctx*.

Categories:

Substantive Context

- Temporal (tmp) – Conditions/Constraints associated with time - the when
  - Point in time (tim) – References to specific points in time
    - Beginning (e.g., "from 1st January")
    - End (e.g., "until 31st January")
  - Time frame (tfr) – References to time frames

    * Frequency (frq) – References to frequencies (e.g., "annually")
- Spatial (spt) – Conditions/Constraints associated with spatial representations – the where
    * Location (loc) – References to specific locations
        · Beginning
        · End
    * Direction (dir) – References to directions, inclusion of intermediary locations (e.g., "via")
    * Path (pth) – References to pathways (e.g., "through the valley")
- Domanial (dom) – Conditions/Constraints applying to a specified activity, topical or existential realm.
    * Activity realm (e.g., "during decision-making")
    * Topical realm (e.g., "for drinking water")
    * Existential realm (e.g., "during childhood")

Procedural Context

- Order (prc) – Conditions/Constraints associated with explicit or implied execution (procedural) order (e.g., "...according to the following stages: ..."). Operationally, this can include expressions of input into the activity identified in the institutional statement (e.g., "with input from ..."). Procedural order can further include the explicit specification of required actions at any step in the procedural execution.
- Method (met) – Conditions/Constraints associated with means or method by which an action is performed, recognizing the following specializations:
    * Means – Action as method (e.g., "by handshake")
    * Instrument – Artifact as method (e.g., "by car")

  Note that the method specification is more general than the characterization of distinctive procedural steps under the Procedural 'Order' category.

Aspirational Context

- Purpose/Function (pur) – Conditions/Constraints describing the purpose or intent of an aim or constitutive function; generally output of action (e.g., "for the purpose of reducing pollution levels")

Situational Context

- State (ste) – References to a specific state environmental state (e.g., "when traffic light is red"); note that state characterization is general in kind and commonly associated with an entity that holds the referenced state (e.g., traffic light). Where possible, a more specific annotation should be chosen.
- Event (evt) – Conditions/Constraints referencing specific events (e.g., "on arrival at the airport ..."). In contrast to the state characterization, an events is instantaneous in nature, whereas a state can persist over longer time frames and is associated with an entity whose state is described.

5.2. Animacy Taxonomy

The Animacy Taxonomy captures differentiates between animate and inanimate entities, maintaining compatibility with annotation conventions commonly adopted in datasets coded according to the previous IG coding guidelines. The suggested annotation label prefix is *anim*.

- Animate – Living entities
- Inanimate – Non-living entities, both real and mental constructs

5.3. Metatype Taxonomy

The Metatype Taxonomy differentiates between concrete physical and abstract entities, e.g., mental constructs. This also applies to component-level nesting (see Section 2.2.3) for cases in which an institutional statement is nested in a statement component, such as an object, in which case the object is characterized as abstract in nature. The suggested annotation label prefix is *metatype*.

- Abstract – Abstract entities
- Concrete – Concrete entities

5.4. Role Taxonomy

The role taxonomy serves the annotation of attributes and objects with additional labels to capture their role within a statement structure with respect to the action, organized by relevant role of actor and effect in terms of affected subjects. The suggested annotation label prefix is *role*.

- Role Characterizations
  - Originator/Causer/Agent – Entity from which action originates
  - Recipient – Recipient of an artifact/sanction
  - Possessor – Owner of an object/entity (e.g., "house owner")
- Effect Characterizations
  - Experiencer – Indirectly affected actor (e.g., "observer of non-compliance")
  - Advantaged – Beneficiary distinctively advantaged by referenced activity/function; may not necessarily be recipient (e.g., "welfare recipient")
  - Disadvantaged – Maleficiary distinctively burdened by referenced activity/function; may not necessarily be recipient

5.5. Regulative Functions Taxonomy

Some types of syntactic annotations can aid the coder in discerning and capturing information that signals the broader function of institutional statements, as indicated by components of which they are comprised. Those are referred to as "institutional function", and more specifically "regulative functions" in as far as regulative statements are concerned (Constitutive functions are discussed in Section 5.6). Regulative functions facilitate the annotation of aims in order to capture the correspondence of aims to analytical functions of relevance through specific theoretical lenses. Exemplifying the use of the institutional grammar for the analysis from a regulatory compliance perspective, compliance and violation behaviour is of specific concern, whereas institutional life cycles may require the annotation of action verbs signalling the initiation of termination of institutional arrangements. Note that the offered taxonomy provides examples for regulative functions organized by categories (alongside potential specializations), but does not claim exhaustiveness. The suggested annotation label prefix is *regfunc*.

- Compliance action – action reflecting compliance behavior
  - Comply – action reflecting compliance
  - Violate – action reflecting violation
- Monitor – action reflecting the institutional function of monitoring
  - Detect compliance – action reflecting the detection of compliance
  - Detect violation – action reflecting the detection of violation
- Enforce – action reflecting enforcement acts
  - Reward – action reflecting rewarding behaviour (regulative-incentivizing)
  - Sanction – action reflecting sanctioning behaviour (regulative-punitive)
- Enforcement response – action reflecting responses to enforcement outcomes
  - Accept – action reflecting acceptance of enforcement outcome
  - Reject – action reflecting rejection of enforcement outcome
    - Appeal (specialization of reject) – action reflecting appeal against enforcement outcome

As noted above, the regulative functions concept is intended to draw the epistemological linkage to analytical frameworks and/or concepts associated with a given domain.

5.6. Constitutive Functions Taxonomy

The constitutive function taxonomy is complementary to the regulative functions taxonomy under the shared institutional functions umbrella. Constitutive function annotations offer a semantic characterization of specific constitutive functions in relation to the constituted entity and/or the linkage of constituted entity and constituting properties, respectively. The top-level distinction of constitutive functions is the identification of the constituted entity as either an *entity* established, modified or otherwise referenced in the context of the policy document, or the *policy* itself. On a more fine-grained level, the categories capture the role of the constituting function with respect to the constituted entity (i.e., a specific entity, or the policy). The suggested annotation label prefix is *confunc*.

The structure, alongside specific annotations, is visualized in Figure 16 and described in the following. Where entities are subject to the statement, constitutive functions can capture the definition of constituted entities as relevant for the parameterization of the institutional setting, including actors, object/artifacts, role specifications or actions, alongside status characterizations. This definition is often signaled intensionally (i.e., by explicit definition), or implied by ascription (e.g., implicit characterization of entity based on exhibited behavior, etc.).

In addition to the definition of entities, constitutive functions can capture relationships of different kinds, including organizational relationships in the form of compositions (e.g,. specification of committee membership) and further reflect hierarchical relationships (e.g., leadership structures, embedding positions within organizations).

Possible constitutive functions further include the initiation and termination of entity lifecycles (e.g., dates of appointment termination, etc.).

Finally, the explicit conferral of status (e.g., in the form of institutional power, such as authority or competence; rights; privileges, or liability) is a central application of constitutive statements.

Complementing the perspective on entity specification as part of constitutive statements, constitutive functions that characterize the policy or document itself, have a varying set of functions. Typical characterizations include the policy lifecycle (e.g., date of enactment), as well as its relationship to other policy (e.g., amending or superseding it). Further statements refer to the purpose or intent underlying

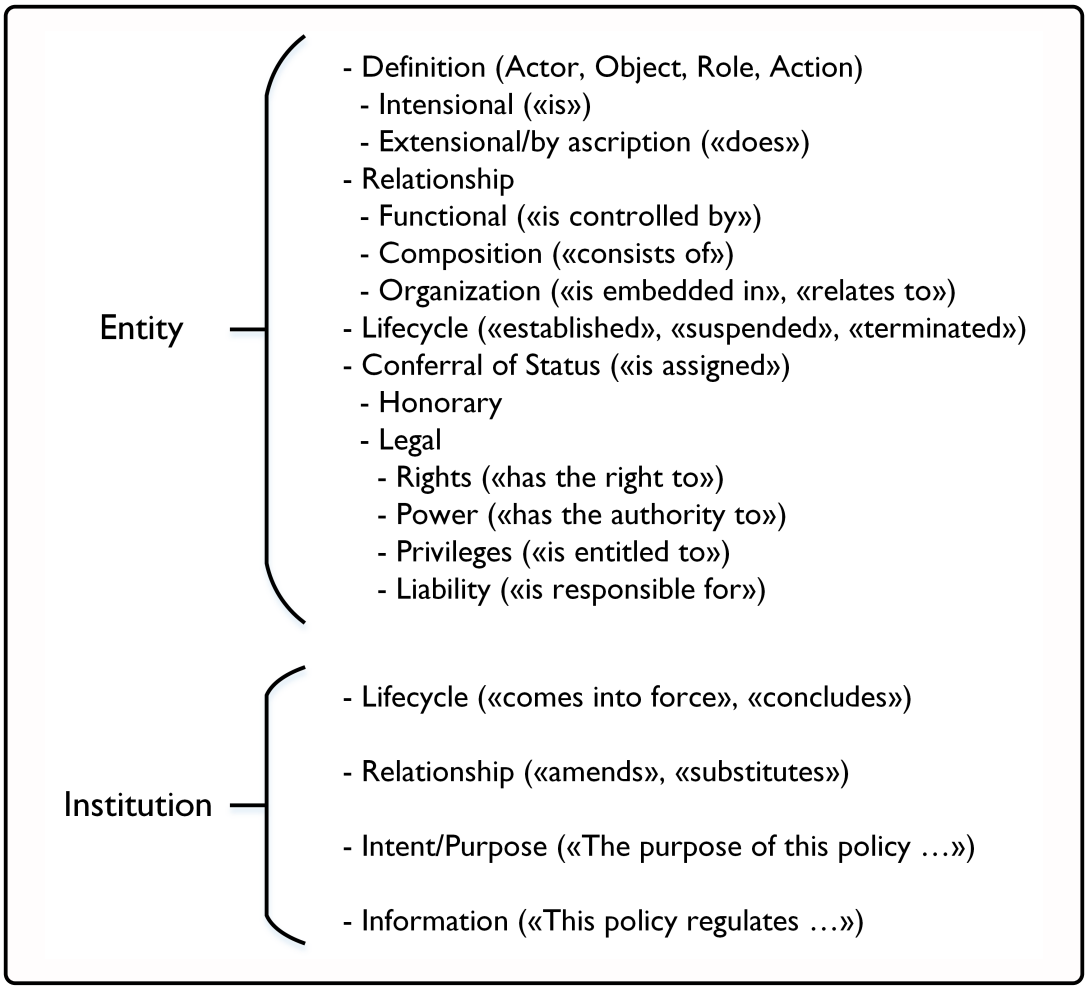

Figure 16: Constitutive Functions

a given policy. Informational statements offer supplementary information about the document, or state institutional facts contextualizing the policy or domain of concern. Note that the latter is often expressed in terms of statement of fact, rather than institutional statements.

As with the preceding taxonomies, the constitutive functions taxonomy is subject to further refinement based on ongoing empirical validation efforts.

6. Statement-Level Annotations

In addition to annotations pertaining to individual components or component types, annotations may also apply to statements as whole. Most notably, this includes the characterization of statements based on their function in the context of an interlinked institutional statement.

6.1. Governance Types (Vertical Nesting Annotations)

The governance annotation relates to the nature of the statement as either *monitored* or *consequential* to signal regulated, or monitored part of a statement (e.g., *"citizen must comply with the law, . . . "*), in contrast to the consequential statement that signals the associated consequence (e.g., *". . . or else official must sanction citizen."*). Augmenting this, a *monitoring* statement highlights the potentially dissociated function of a third-party observer (e.g., bystander, appointed monitor), who may or may not be the enforcer. As a consequence, statements can have multiple statement-level annotations.

6.2. Consequence Types

A second annotation relates to the nature of potential consequences. The main differentiation includes consequences of *existential*, and *non-existential* kind.

Consequences of the first kind are understood as ontologically existential, i.e., reflecting whether an entity does not come about, is invalidated, etc. This can further include the institution itself (e.g., lacking preconditions for its continuation), hence leveraging *configurational consequences*.

Social consequences, as common institutional consequences, are related to economic situation, status, or penalties. As far as the institution is concerned, these are considered non-existential.

As with other taxonomies, the analyst may introduce subdifferentiations that respond to specific analytical objectives, or establish epistemological linkages to the domain of concern.

The listing of taxonomies complements and concludes the supplementary coding guidelines for the Institutional Grammar 2.0.

7. Conclusion

This codebook provides operational guidance for the planning and execution of encoding institutions using the Institutional Grammar 2.0. This information is supplementary to the conceptual introduction provided in Frantz and Siddiki (2021, 2022).

Following a general introduction of the principles and motivations of the Institutional Grammar, the codebook initially outlines the theoretical concepts underlying IG 2.0, including the coding on multiple levels of expressiveness (IG Core, IG Extended, IG Logico) in Section 2, before supporting the navigation through the codebook based on the reader's objectives and background in the form of a Reader's Guide in Section 1.2. This is followed by the discussion of essential document preparation steps (pre-coding steps) in Section 3 that complement the broader design considerations outlined in Frantz and Siddiki (2022), central aspects of which are captured in the Study Design Checklist appended to this codebook (see Appendix A).

Following the general methodological orientation, the codebook provides detailed coding guidelines for regulative and constitutive statements across all IG 2.0 levels of expressiveness (Section 4). This is followed by advanced concepts, such as the discussion of encoding hybrid institutional statements (Section 4.4), i.e., institutional statements consisting of both regulative and constitutive components, polymorphic institutional statements (Section 4.4.4), and the discussion of structural patterns (Section 4.5).

A mechanism to document the parameterization of IG 2.0 studies based on the selective application of IG 2.0 components and features – in the form of IG 2.0 Profiles – is provided in Section 4.6. The listing of taxonomies to support the semantic annotation of institutional information in the previous section (Section 5) concludes the substantive part of the codebook.

It is important to note that the guidelines provided in this codebook are of general nature and emphasize the *operational use* of IG 2.0, with specific focus on the encoding of institutional information. Doing so, they may not capture specifics potentially relevant for a given project, let alone the detailed introduction of the underlying theoretical concepts. For an in-depth introduction into the Institutional Grammar, existing applications, challenges that motivate the IG 2.0 and detailed conceptual foundations, the reader is referred to companion literature (Frantz & Siddiki, 2021, 2022).

However, the codebook aims to support the development of supplementary project-specific guidelines that consider application context (e.g., domain, language, types of documents, legal traditions, etc.) and analytical objectives (i.e., evaluation of encoded statements) more explicitly, aspects that a general codebook cannot consider. As indicated above, the reader's guide in Section 1.2 intends to highlight relevant sections. It is furthermore important to note that the coding instructions provided here are intentionally tool-agnostic, and open to adaptation to arbitrary encoding means (e.g., spreadsheets,

text annotation tools, etc.), which themselves may be augmented with specific guidance.

For a complete overview of relevant resources, including a theoretical treatment of the underlying concepts and principles, as well as resources that support operational coding (e.g., cheat sheet, tool-specific guidance, software), please refer to https://newinstitutionalgrammar.org/resources. Given the ongoing development and application efforts, the website is updated whenever relevant new material becomes available.

Please further note that these guidelines will be continuously refined based on theoretical developments, feedback from users as well as ongoing empirical validation efforts. To retrace subsequent changes, please note the specific version and version history of these guidelines outlined at the beginning of this document. Irrespective of refinements, all revisions of the codebook will be retained for future reference.

References

Brady, U., Basurto, X., Bennett, A., Carter, D. P., Hanlon, J., Heikkila, T., Lien, A., Chonaiew, S. M., Olivier, T., Schlager, E., Siddiki, S., & Weible, C. (2018). *Institutional Analysis of Rules-In-Form Coding Guidelines* (tech. rep.). Center for Behavior, Institutions and the Environment. https://complexity.asu.edu/sites/default/files/papers/cbie_wp_2018-006_0.pdf

Crawford, S. E. S., & Ostrom, E. (1995). A Grammar of Institutions. *American Political Science Review*, *89*(3), 582–600. https://doi.org/10.2307/2082975

Crawford, S. E. S., & Ostrom, E. (2005). A Grammar of Institutions. In *Understanding institutional diversity* (pp. 137–174). Princeton University Press.

Frantz, C., Purvis, M. K., Nowostawski, M., & Savarimuthu, B. T. R. (2013). nADICO: A nested grammar of institutions. *Lecture Notes in Computer Science*, *8291 LNAI*, 429–436. https://doi.org/10.1007/978-3-642-44927-7_31

Frantz, C. K., & Siddiki, S. (2021). Institutional Grammar 2.0: A specification for encoding and analyzing institutional design. *Public Administration*. https://doi.org/10.1111/padm.12719

Frantz, C. K., & Siddiki, S. (2022). *Institutional Grammar*. Palgrave Macmillan. https://doi.org/10.1007/978-3-030-86372-2

Hohfeld, W. N. (1913). Some Fundamental Legal Conceptions as Applied in Judicial Reasoning. *The Yale Law Journal*, *23*(1). https://doi.org/10.2307/785533

Siddiki, S., Weible, C. M., Basurto, X., & Calanni, J. (2011). Dissecting Policy Designs: An Application of the Institutional Grammar Tool. *The Policy Studies Journal*, *39*(1), 79–103. https://doi.org/10.1111/j.1541-0072.2010.00397.x

Appendices

A. Study Design Checklist

This checklist summarizes considerations relevant for any study design using IG 2.0. The checklist is not prescriptive, but rather a collation of the methodological aspects introduced throughout the codebook, serving the development of a project-specific coding protocol. Reflecting on these aspects in the study design phase aims at ensuring high levels of reliability and validity of the coded institutional information.

Overarching the development of a project-specific codebook adaptation is the question as to whether you intend to develop a generically useful dataset (useful for later adoption using specific techniques or for comparative studies), or whether your coding is purpose-specific (i.e., aimed at addressing specific needs as well as relying on specific analytical techniques). This will inform choices with respect to statement types, encoding heuristics, as well as help establish the relevant IG feature set as well as the scope of encoding, both in terms of number of statements and interpretational scope (see below).

- Identify applicable statement types (regulative, constitutive, hybrids)
- Identify level of expressiveness (IG Core, IG Extended, IG Logico), or customized characterizations
  - See Section 4.6
- Decide on the interpretational scope of statement encoding (narrow vs. wide)
  - Note: This is of specific relevance when coding both constitutive and regulative statements, since it influences the decision heuristics.
  - See here (Section 2.5)
- Decide on applicable heuristics for Attributes/Attributes Properties, Objects/Object properties, Constituted Entity/Constituted Entity Properties, and Constituting Properties/Constituting Properties Properties respectively
  - See here (Section 4.2.1)
- Decide on specific pre-coding/processing steps based on the suggestions listed in Section 3
  - This is particularly relevant where potential reconstruction of institutional statements (conceptual reification) is considered prior to coding.[21]
  - Decide on aspects such has component scoping during encoding. This includes the decision whether prepositions ('by', 'above', 'beyond') and articles ('a', 'the') are included in the encoding. For instance, if employing modelling techniques, considering such information is counter-productive in the encoding process. An alternative vehicle is the use of semantic annotations to ensure unambiguous labelling of specific component values, especially where you experience divergence in expression (e.g., 'EU member states', 'member states', 'members').
  - Further consider project-specific preparation and coding conventions under consideration of domain-specific and epistemic considerations.
- Decide on applicable taxonomies (see Section 5), as well as possible extensions and/or additional taxonomies (derived from domain-specific theories or frameworks, for instance).

[21] For a detailed discussion of conceptual reification, see also Chapters 3 and 5 in Frantz and Siddiki (2022).