Institutional Grammar 2.0 Codebook

Christopher K. Frantz & Saba N. Siddiki

Version: 1.0 (15th August 2020)

Contents

1 Introduction 2

2 Syntactic Definitions, Institutional Statement Concepts, and Assumptions about Institutional Statement and Syntactic Component Nesting 3
  2.1 Syntactic Definitions ........................................... 3
  2.2 Institutional Statement Assumptions .......................... 3
  2.3 Syntactic Component-Level Nesting ............................. 6
  2.4 Attribute/Object-Property Hierarchy ............................ 7
  2.5 Relating Institutional Statements to Action Situations .......... 8
  2.6 IG Coding Levels ............................................... 14

3 Pre-Coding Steps 15
  3.1 General Steps .................................................. 16
  3.2 Pre-processing Guidelines for IG Core ........................ 18
  3.3 Pre-processing Guidelines for IG Extended ..................... 19
  3.4 Pre-processing Guidelines for IG Logico ........................ 19

4 Coding Guidelines 21
  4.1 Coding Syntax .................................................. 21
  4.2 Regulative Statement Coding ................................... 28
    4.2.1 IG Core Coding of Regulative Statements .................. 29
    4.2.2 IG Extended Coding of Regulative Statements .......... 38
    4.2.3 IG Logico Coding of Regulative Statements ............. 55
  4.3 Constitutive Statement Coding ................................ 68
    4.3.1 IG Core Coding of Constitutive Statements ............... 69
    4.3.2 IG Extended Coding of Constitutive Statements ....... 80
    4.3.3 IG Logico Coding of Constitutive Statements .......... 83
  4.4 Constitutive-Regulative Hybrids ................................ 86
    4.4.1 Regulative-constitutive Statements ........................ 86
    4.4.2 Constitutive-regulative Statements ........................ 88
    4.4.3 Second-order constitutive statements ..................... 88
    4.4.4 Institutional Statement Types & Polymorphic Institutional Statements 89
    4.4.5 Data Structure Patterns ................................... 91
  4.5 Coding Level Configurations (Institutional Grammar Profiles) 92

1 Introduction

The following instructions are intended to provide guidance for the coding of institutional statements, the focal unit of analysis in the Institutional Grammar (IG), according to the Institutional Grammar 2.0 (IG 2.0) specification.\(^1\) An institutional statement describes expected actions for actors within the presence or absence of particular constraints, or parameterizes features of an institutional system. 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, 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 constitute features of a system (e.g., actor positions and roles, processes, venues, etc). Regulative statements describe actions linked to specific actors 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. Regulative statements are composed of some or all of the following components with the corresponding syntactic labels: (i) an Actor, referred to as an Attribute; (ii) action associated with actor, referred to as an Aim; (iii) action 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) an incentive linked to 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 within a statement, referred to as a Constituted Entity; (ii) an action that constitutes the Constituted Entity, called the Constitutive Function; (iii) the constitution context, referred to as Context; (iv) properties that serve as input to the Constitutive Function, called Constituting Properties; (iv) A prescriptive operator that defines to what extent the action of an institutional statement is compelled, restrained, or discretionary, referred to as a Deontic; and (vi) an incentive linked to action, referred to as an Or else. The operational definition of an institutional statement is tied to the presence of certain syntactic components, or necessary

---

\(^1\)In addition to Crawford and Ostrom (1995, 2005), the specification draws on the original IG codebook (Brady et al., 2018), and integrates further specific refinements (Siddiki et al., 2011; Frantz et al., 2013).
components. To qualify as a complete regulative institutional statement, the statement must at least contain Attribute, Aim, and Context components. The Object, Deontic, and Or else components are deemed sufficient components. Institutional statements – regulative or constitutive statements – containing only respective necessary components are referred to as atomic statements.

In this codebook, in accordance with the IG 2.0 specification, coding 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 by the coding guidelines by syntactic component, and by level of expressiveness. Section 5 provides a concluding overview of taxonomies as referenced throughout the coding guidelines.

2 Syntactic Definitions, Institutional Statement Concepts, and Assumptions about Institutional Statement and Syntactic Component Nesting

IG 2.0 is premised on a set of syntactic definitions, conceptualizations of institutional statements, and assumptions regarding institutional statement and syntactic 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, and highlighting assumptions regarding institutional statement and component level nesting. Concluding this section, we organize the coding levels based on involved features, along with providing principal guidelines on the coding process.

2.1 Syntactic Definitions

The IG structure as referred to in this document relies on the elementary syntactic components of regulative and constitutive statements highlighted in Tables 1 and 2. Definitions of these components that hold across IG 2.0 encoding levels are provided along with each syntactic component. Components listed in both tables are ordered by their assumed necessity and sufficiency within institutional statements. Some components exist within both regulative and constitutive statements, and thus their definitions are presented in relation to each, however with additional information in accordance with different types of statements.

2.2 Institutional Statement Assumptions

IG 2.0 rests on the basic definition of institutional statements – linguistic constraints or opportunities. The construction of complex institutional statements as put forth in IG 2.0 builds on this basic definition to include the notion of nested institutional statements of different forms introduced in the following. Motivating the conception of nested institutional statements is the pragmatic observation that statements are fundamentally linked in institutions constituted of multiple institutional statements. Lack of specificity of these linkages undermines the coder’s ability to comprehensively, and accurately, capture institutional content.

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
<table>
<thead>
<tr>
<th>Syntactic Element</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute</strong></td>
<td>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 Attribute may also contain descriptors of the actor.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Deontic</strong></td>
<td>A prescriptive operator that defines to what extent the action of an institutional statement is compelled, restrained, or discretionary.</td>
</tr>
<tr>
<td><strong>Aim</strong></td>
<td>The goal or action of the statement assigned to the statement Attribute.</td>
</tr>
<tr>
<td><strong>Context</strong></td>
<td>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. 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.</td>
</tr>
<tr>
<td><strong>Or else</strong></td>
<td>A sanctioning provision associated with the action indicated in a particular institutional statement that can exist wholly within an institutional statement, or be represented in a nested institutional statement (as defined in the following discussion).</td>
</tr>
</tbody>
</table>

Table 1: Definitions of Syntactic Elements for Regulative Statements

that convey co-occurring or alternative actions. 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, 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 the two statements, such as “Organic farmers must commit to organic farming standards” AND “Organic farmers must accommodate regular reviews of their practices”, but allow for more complex constructs, such as
### Table 2: Definitions of Syntactic Elements for Constitutive Statements

<table>
<thead>
<tr>
<th>Syntactic Element</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constituting Properties</td>
<td>Delineate properties assigned to a Constitutive Entity.</td>
</tr>
<tr>
<td>Deontic</td>
<td>A prescriptive operator that defines to what extent the action of an institutional statement is compelled, restrained, or discretionary.</td>
</tr>
<tr>
<td>Constitutive Function</td>
<td>A verb that constitutes a Constituted Entity, or reflects the functional relationship between constituted entity and constituting properties.</td>
</tr>
<tr>
<td>Constituted Entity</td>
<td>The entity being constituted, reconstituted, modified or otherwise directly affected within an institutional statement.</td>
</tr>
<tr>
<td>Context</td>
<td>The context 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. 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.</td>
</tr>
</tbody>
</table>

| Or else | A sanctioning provision associated with the action indicated in a particular institutional statement that can exist wholly within an institutional statement, or be represented in a nested institutional statement (as defined in the following discussion). |

(“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”).

Note the use of parentheses to signal the precedence of individual statements. Possible logical operators 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”).

**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). 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 respectively. 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

(“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 (visually supported by corresponding formatting), 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 nesting approaches 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.

Figure 1: Nesting Principles

2.3 Syntactic Component-Level Nesting

In addition to the nesting of statements, IG 2.0 further assumes the possibility of nesting of individual components as introduced in the following.

Component-level nesting: Specific components may be substituted with entire institutional statements that characterize an individual with respect to its Attributes, or to express procedural order, i.e., statements whose fulfilment is a precondition for the application of a given statement. Substituting the conditions component in the IG syntax, we can exemplify this with “Organic farmers may sell their produce under the organic label under the condition that organic farmers apply for certification.”, where the statement embedded in braces reflects the precondition and may in itself be expressed in (a subset
of) syntactic components of institutional statements. The use case for such component-level nesting further includes the articulation of beliefs about individuals’ behaviours (e.g., an official sanctioning an individual if the official believes that the individual has performed a violation).

### 2.4 Attribute/Object-Property Hierarchy

**Attribute/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 `Attribute/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 centered around 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 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.

![Figure 2: Attribute/Object-Property Hierarchy for given Example](image-url)
2.5 Relating Institutional Statements to Action Situations

A concept central to the coding with IG 2.0 is the action situation (Crawford and 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 seven types of institutional statements, which can be regulative or constitutive in kind, with distinctive functional properties. These 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.

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.

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](image)

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. The example statement used here is regulative: “At 8am, farmers may begin selling their goods in the farmer’s market,” contains two context clauses (at 8am, in the farmer’s market), one of which is a conditions clause (at 8am) and one of which is a constraints clause (in the farmer’s market). 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.

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)
Fundamental Concepts

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)
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 of a basic syntactic structure. This level best accommodates relatively simple institutional statements that largely follow 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. 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 Attribute/Object-Property Hierarchies (Section 2.4), 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; where coding could: 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 in the research objectives (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 3. 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).

A high-level overview of the individual levels, along with the discussed objectives is captured in Figure 8. The principles and objectives of the individual codings are discussed in detail in Section 4.
2.6 IG Coding Levels

<table>
<thead>
<tr>
<th>Coding Level</th>
<th>Relevant Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>IG Core</td>
<td>• Horizontal and vertical nesting (Statement-level nesting)</td>
</tr>
<tr>
<td></td>
<td>• Activation conditions, execution constraints</td>
</tr>
<tr>
<td>IG Extended</td>
<td>• Component-level nesting</td>
</tr>
<tr>
<td></td>
<td>• Attributes/Object-Property Hierarchy in regulative statements; equally applies to Constituted Entities/Properties in constitutive statements (both of which may have properties on their own)</td>
</tr>
<tr>
<td></td>
<td>• Structural Decomposition Patterns (Constitutive Statements); discussed in Section 4.4</td>
</tr>
</tbody>
</table>

Table 3: Relevant concepts on different coding levels

![Figure 8: Overview of IG Coding Levels and associated objectives](image)

3 Pre-Coding Steps

Before discussing the coding of institutional statements in detail, we 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 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 candidate 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
3.1 General Steps

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.

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.**

NOT Institutional Statements:

**Organic farming promotes environmental and human health.**

**The Department of Agriculture is committed to marketing of agricultural products.**

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 that has been 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 pre-processing 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.

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
- 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:

  "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."

- 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 required 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 (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 an institutional 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 be offloaded to the pre-processing process, subject to the analytical objective, nature of the coded policy, as well as coder background and algorithmic treatment of 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 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.

<table>
<thead>
<tr>
<th>Example Statements with Referential Clauses</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
<tr>
<td>A production or handling operation that sells agricultural products as &quot;organic&quot; but whose gross agricultural income from organic sales totals $5,000 or less annually is exempt from certification under subpart E of this part.</td>
</tr>
<tr>
<td>Any agricultural product that is sold, labeled, or represented as “100 percent organic,” &quot;organic,&quot; or &quot;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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example Statements with Embedded Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A handler of organic products may use information provided by the certified operation to determine percentage of organic ingredients.</td>
</tr>
<tr>
<td>→ Embedded action: information provided by the certified operation (i.e., provision of information by certified operation)</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>→ Embedded action: approved by the certifying agent (i.e., approval of report by certifying agent)</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>→ Embedded action: accreditation is suspended by the Secretary (i.e., accreditation suspension by Secretary)</td>
</tr>
<tr>
<td>The Program Manager may initiate suspension proceedings against a certified operation, when a certifying agent fails to take appropriate action to enforce the Act.</td>
</tr>
<tr>
<td>→ 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.</td>
</tr>
</tbody>
</table>
4.1 Coding Syntax

Provided next, following a brief overview of conventions we rely on for syntactic notations, are the specific guidelines for encoding institutional statements at the IG Core, IG Extended, and IG Logico levels.

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 Syntax

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 4.

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 8 onwards). 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.

<table>
<thead>
<tr>
<th>Symbol/ Symbol Pairs</th>
<th>Coding Context</th>
<th>Lowest applicable Coding Level &amp; Statement Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>Component</td>
<td>IG Core, Regulative &amp; Constitutive Statements</td>
<td>Component classification: The characterization of an expression as a component type is signaled through parentheses that contain the component type. Where used, component annotations (e.g., animate, inanimate) can be appended to the component classification.</td>
<td>Certifier (A) . . . , where A identifies the certifier as an attribute in a given institutional statement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Certifier (A; label=animate) . . . , where A identifies the certifier as an attribute in a given institutional statement, and animate is an additional annotation.</td>
</tr>
</tbody>
</table>
### 4.1 Coding Syntax

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.

<table>
<thead>
<tr>
<th></th>
<th>Component</th>
<th>IG Core, Regulative &amp; Constitutive Statements</th>
<th>Tacit components: The explicit specification of implied components (e.g., actor(s)) is signaled with brackets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ]</td>
<td>Statement</td>
<td>IG Core, Regulative &amp; Constitutive Statements</td>
<td>Horizontally nested statements are represented using surrounding parentheses to emphasise the precedence of combined individual statements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Attendees must not (eat and drink) on the train.</td>
</tr>
<tr>
<td></td>
<td>[ ]</td>
<td>Statement</td>
<td>Vertically nested statements are represented using brackets that embedded the respective consequential statement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IG Core, Regulative &amp; Constitutive Statements</td>
<td>stmt1[stmt2], where stmt1 represents a monitored statement, and stmt2 the corresponding consequential statement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>They [farmers (A)] must comply with the certification regulation . . . , where [farmers (A)] characterizes the inferred actor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Attendees must not (eat and drink) on the train.</td>
</tr>
</tbody>
</table>
### 4.1 Coding Syntax

<table>
<thead>
<tr>
<th>Component</th>
<th>IG Extended, Regulative &amp; Constitutive Statements</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ }</td>
<td>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.</td>
<td>Certifier (A) believes (I) {farmer (A) violates (I) code of conduct (B)} (Cex)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Component IG Core, Regulative Statements</td>
<td>Identifies the preceding expression as Attribute component.[2]</td>
<td>Certifier (A)</td>
</tr>
<tr>
<td>I</td>
<td>Component IG Core, Regulative Statements</td>
<td>Identifies the preceding expression as aim component.</td>
<td>Certifier (A) monitors (I) farmers.</td>
</tr>
<tr>
<td>Bdir</td>
<td>Component IG Core, Regulative Statements</td>
<td>Identifies the preceding expression as direct object component.</td>
<td>Certifier (A) administers (I) certifications (B_dir).</td>
</tr>
<tr>
<td>Bind</td>
<td>Component IG Core, Regulative Statements</td>
<td>Identifies the preceding expression as indirect object component.</td>
<td>Certifier (A) registers (I) certification (B_dir) for organic farmer (B_ind).</td>
</tr>
<tr>
<td>B</td>
<td>Component IG Extended, Regulative Statements</td>
<td>Identifies objects that are neither direct nor indirect, but contained as functionally-independent objects in the institutional statement (see Attributes/Object-Properties Hierarchy in Section 2.4). First-order properties are identified by alphabetic identifiers (e.g., B_a, B_b, etc.); second-order properties are identified with numeric identifiers (e.g., B_1, B_2, etc.).</td>
<td>... notification (B_dir) of suspension (B_a,B_1) or revocation (B_a,B_2) of certification (B_a) ...</td>
</tr>
</tbody>
</table>

Here, the functionally independent object certification (B_a) is root of a property structure consisting of two objects as properties (suspension, revocation), both of which are annotated with reference to the certification.
<table>
<thead>
<tr>
<th>Component</th>
<th>IG Core, Regulative &amp; Constitutive Statements</th>
<th>Identifies the preceding expression as deontic component.</th>
<th>Regulative: Certifier (A) must (D) monitor (I) farmers (B_{dir}).</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D/D )</td>
<td>( D )</td>
<td></td>
<td>From 1st January onwards (C_{ac}), Council (E) shall (D) be responsible (F) for adherence with food production standards (C_{ex}).</td>
</tr>
<tr>
<td></td>
<td>( D )</td>
<td></td>
<td>Alternative example: From January 1st onward (C_{ac}), there shall (D) be (F) a National Organic Standards Advisory Council (E) within the Department of Agriculture (C_{ex}).</td>
</tr>
<tr>
<td>( C_{ac}/C_{ac} )</td>
<td>( C_{ac} )</td>
<td>Identifies the preceding expression as an activation condition component.</td>
<td>Regulative: Upon accreditation (C_{ac}) certifier (A) must (D) monitor (I) farmers (B_{dir}).</td>
</tr>
<tr>
<td></td>
<td>( C_{ac} )</td>
<td></td>
<td>Constitutive: From 1st January onwards (C_{ac}), Council (E) shall (D) include (F) organic farming representatives (P) to review chemical allowances within organic food production standards (C_{ex}).</td>
</tr>
<tr>
<td>( C_{ex}/C_{ex} )</td>
<td>( C_{ex} )</td>
<td>Identifies the preceding expression as an execution constraint component.</td>
<td>Regulative: Certifier (A) must (D) monitor (I) farmers (B_{dir}) at any time (C_{ex}).</td>
</tr>
<tr>
<td></td>
<td>( C_{ex} )</td>
<td></td>
<td>Constitutive: From 1st January onwards (C_{ac}), Council (E) shall (D) include (F) organic farming representatives (P) to review adherence with food production standards (C_{ex}).</td>
</tr>
<tr>
<td>Component</td>
<td>IG Core, Constitutive Statements</td>
<td>Constituted entity/constituting property/constitutive function</td>
<td>Statement</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------</td>
<td>-------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>E</td>
<td>Component IG Core, Constitutive Statements</td>
<td>Constituted entity</td>
<td>From 1st January onwards (C_{ac}), Council (E) shall (D) include (F) organic farming representatives (P) to review chemical allowances within organic food production standards (C_{ex}).</td>
</tr>
<tr>
<td>P</td>
<td>Component IG Core, Constitutive Statements</td>
<td>Constituting property</td>
<td>From 1st January onwards (C_{ac}), Council (E) shall (D) include (F) organic farming representatives (P) to review chemical allowances within organic food production standards (C_{ex}).</td>
</tr>
<tr>
<td>F</td>
<td>Component IG Core, Constitutive Statements</td>
<td>Constitutive function</td>
<td>From 1st January onwards (C_{ac}), Council (E) shall (D) include (F) organic farming representatives (P) to review allowances within organic food production standards (C_{ex}).</td>
</tr>
</tbody>
</table>
4.1 Coding Syntax

**prop**/prop

Attributes, Object, Entity and Property components
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 Core:**
Regulative:
*Certified organic (A, prop) farmers (A) must (D) respond (I) to formal (B<sub>dir,prop</sub>) certification requirements (B<sub>dir</sub>).*

Constitutive:
The Council (E) consists of (F) elected (P, prop) officials (P) resident in the electorate (P, prop).

**IG Extended:**
Regulative:
*Certified (A, prop1) organic (A, prop2) farmers (A) must (D) respond (I) to formal (B<sub>dir,prop1</sub>) certification requirements (B<sub>dir</sub>).*

Constitutive:
The Council (E) consists of (F) elected (P, prop1) officials (P) resident in the electorate (P, prop2).

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 object/properties they refer to (i.e., they are appended to the component specification). Further details on property coding are provided in Table 6 and illustrated below.
The example on the right highlights a complex object hierarchy structure previously discussed in the context of the Attributes/Object-Property Hierarchy (Section 2.4). As mentioned above, where multiple objects on a given hierarchy level exist, they are uniquely identified with a numeric index (e.g., \( B_1, B_2 \), etc.). Where multiple properties on a given hierarchy level exist, they are uniquely identified with a numeric index (e.g., \( \text{prop}_1, \text{prop}_2 \), etc.). Where a single property applies to multiple properties, references to both objects/properties are maintained on this property (separated by semicolon).

\[ \ldots \text{proposed } (B_a, B_1, \text{prop}_B, B_2, \text{prop}) \] suspension \((B_a, B_1)\) or revocation \((B_a, B_2)\) of certification \((B_a)\) \ldots \]

<table>
<thead>
<tr>
<th>AND, OR, XOR, NOT</th>
<th>Statement, Component IG Core, Regulative &amp; Constitutive Statements</th>
<th>The logical operators identify the relationship between statement and/or components as either conjunction (AND), inclusive disjunction (OR), or exclusive disjunction (XOR). Where negation is involved, the NOT operator is used (e.g., in deontics: must not; combination of exceptions: NOT option 1 AND option 2).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certifiers (must review applications and (AND) must not (NOT) approve applications) by offenders.</td>
<td>Table 4: Symbol Reference for IG Coding as applied in this document</td>
<td></td>
</tr>
</tbody>
</table>
4.2 Regulative Statement Coding

The base syntax of regulative statements (as shown in Figure 9) consist of necessary and sufficient components, and further involves further feature refinements across different levels of expressiveness.

![Figure 9: Syntax and Features of Regulative Statements by Level of Expressiveness](image)

Tables 5 to 7 provide detailed coding guidelines for regulative statements by level of expressiveness, starting with IG Core through IG Logico, 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).
- Recall that coding can 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).
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.

<table>
<thead>
<tr>
<th>Syntactic Component</th>
<th>Treatment of Syntactic Components by Level of Encoding</th>
<th>Relevant Examples</th>
<th>Complete Syntactic Classification of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>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 and actor property.</td>
<td>Example statement: Certified farmer must submit an organic system plan annually.</td>
<td>Certified ((A, \text{prop})) farmer ((A)) must ((D)) submit ((I)) an organic systems plan ((B_{\text{dir}})) annually ((C_{\text{ex}})).</td>
</tr>
</tbody>
</table>

Note that in all cases, named entities (e.g., United States Department of Agriculture) are not decomposed into actor properties.

A heuristic to decide as to whether entity names are decomposed relies on the institutional context. If, for example, a regulation differentiates between organic and non-organic farmers, the decomposition into descriptor and associated properties is useful. If, however, the regulation is exclusively concerned with organic farmers, the decomposition is of little analytical value at this level of encoding.
**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.

**Rules:**
- Identify object identifier
- Identify object properties

Note that in all cases, named entities (e.g., United States Department of Agriculture) are not decomposed into object properties. The heuristics for decomposition as outlined in the context of Attributes equally applies for objects.

**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.
Object (ctd.) 

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.

<table>
<thead>
<tr>
<th>Aim</th>
<th>The encoding identifies the focal action of the statement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organic <strong>certifier</strong> must send <strong>farmer</strong> notification of compliance.</td>
<td>Organic (<strong>A</strong>, prop) <strong>certifier</strong> (<strong>A</strong>) must (<strong>D</strong>) send (<strong>I</strong>) <strong>farmer</strong> (<strong>B_{ind}</strong>) notification of compliance (<strong>B_{dir}</strong>).</td>
</tr>
<tr>
<td><strong>Aim</strong> = send</td>
<td></td>
</tr>
</tbody>
</table>

---

*IG 2.0 Codebook Version: 1.0*
### 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

<table>
<thead>
<tr>
<th>Deontic</th>
<th>Organic (A, prop) certifier (A) must (D) send (I) farmer (Bind) notification of compliance (Bdir).</th>
</tr>
</thead>
</table>

### 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.
4.2 Regulative Statement Coding

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 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 brackets ([ and ]).
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:

Certified \(A, \text{prop1}\) organic \(A, \text{prop2}\) farmers \(A\) must not \(D\) apply \(I\) synthetic chemicals \(B_{\text{dir}}\) to crops \(B_{\text{ind}}\) at any time \(C_{\text{ex}}\) once organic certification is conferred \(C_{\text{ac}}\),

OR ELSE [(certifier \(A\) will \(D\) revoke \(I\) certification \(B_{\text{dir}}\) from farmer \(B_{\text{ind}}\)) XOR (certifier \(A\) will \(D\) fine \(I\) farmer \(B_{\text{dir}}\))].

Note:

• Where component-level combinations exist (... fine farmer or revoke ...), those have to signaled explicitly by parentheses, or be decomposed into separate logically-combined complete atomic institutional statements. Further details are provided under “General IG Extended Instructions” (Table 6), Item “Decomposition of component-level combinations”.

• 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 is equivalent to additional annotations for Constituted Entity, Constitutive Function and Constituting Properties 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” (Table 7), Item “Cross-component Semantic Annotations”. 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.

While exemplified here for regulative statements, this equally applies to constitutive statements.
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” (Table 6), Item “Decomposition of component-level combinations”; IG Core Examples can be found in the following.
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.
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 5: 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 and combined attributes and aims that reconstruct atomic statements and their relationships, but also decomposes the hierarchical relationships amongst explicitly highlighted actors, object and their respective properties.
<table>
<thead>
<tr>
<th>Syntactic Component</th>
<th>Treatment of Syntactic Components by Level of Encoding</th>
<th>Relevant Examples</th>
<th>Complete Syntactic Classification of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>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 attribute, property, parameter hierarchy, elements may be substituted by an institutional statement, in which case component-level nesting applies for the encoding. 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.</td>
<td>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.</td>
<td>A certified (A.prop1) farmer (A) {whose certification (B_dir) is suspended [suspends (I)] by the Secretary (A) under this section (C_ex)} (A.prop2) may (D) at any time (C_ac) submit (I) a recertification (B_dir, prop) request (B_dir).</td>
</tr>
</tbody>
</table>
Attributes (ctd.)

Where properties or parameters apply to multiple objects, the respective references are separated by semicolon (e.g., $A_a,prop1; A_b,prop1$). 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 $(A,prop1,prop1)$, 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.

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

Program managers $(A)$ who believe $(A,prop)$ that a certified operation $(A)$ has violated $(1)$ the Act $(B_{dir})$ $(A,prop,prop)$ may $(D)$ pursue $(I)$ revocation proceedings $(B_{dir})$. 
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 \((A, prop, prop)\), since the statement describes the content of the belief \((A, prop)\).
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 respective. 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.

Rules:

- The identification of the direct or indirect Object respectively.
- 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.

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 Program Manager (A) shall (D) send (I) a written (Bdir, prop) notification (Bdir) of proposed (Ba1,prop; Ba2,prop) suspension (Ba1) or revocation (Ba2) of certification (Ba) to certified (Bind,prop1) organic (Bind,prop2) farmer (Bind).
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., (B_a), (B_b), etc.

- Dependent objects = suspension, revocation

Dependent objects are coded with reference to the object they depend on and unique numeric index, e.g., (B_a,1), (B_a,2), 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., (B_a, prop), (B_a,2, prop), etc.

Where multiple properties exist for an object, they are, as in IG Core coding, uniquely identified by numeric index, e.g., (B_a, prop1), (B_a, prop2), etc.
Where higher-order properties exist, these are coded according as specified in the context of the Attributes component.

Object hierarchies outside the object component:

Taking the example “Certifier (A) must (D) send (I) notification (B_{dir}) to inspector (B_{ind}) to produce a written report of the assessment (C_{ex,pur})”, 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.

Certifier (A) must (D) send (I) notification (B_{dir}) to inspector (B_{ind}) to produce a written report (B_{a,1}) of the assessment (B_{a}) (C_{ex,pur}).

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

<table>
<thead>
<tr>
<th>Aim</th>
<th>The coding of the aim is identical to IG Core. See also Item “Decomposition of component-level combinations” in this table (Table 6) for associated decomposition rules.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deontic</td>
<td>The coding of the deontic is identical to IG Core.</td>
</tr>
</tbody>
</table>
In IG Extended, Activation conditions (\(C_{ac}\)) and Execution constraints (\(C_{ex}\)) (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 circumstantial taxonomy, an overview of which is provided in the following, along with corresponding coding:

Excerpt of Circumstance Taxonomy:
- Temporal (tmp)
- Spatial (loc)
- State (ste)
- Procedural (prc)
- Method (met)
- Purpose (pur)
- Observed state/outcomes (eff)

A fine-grained overview of subcategories (e.g., differentiation of temporal category into ‘point in time’ and ‘time frame’) is provided in Section 5, 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:

Organic certifier (\(A\)) must (\(D\)) inspect (\(I\)) farmer’s operation (\(B_{dir}\)) within 60 days (\(C_{ex,tfr}\)) under the condition \{that the organic farmer (\(A\)) enters (\(I\)) an agreement (\(B_{dir}\)) with the organic certifier (\(B_{ind}\)) to serve as his/her certifying agent (\(C_{ex,pur}\))(\(C_{ac,prc}\)).

Here the execution constraint for the initial inspection (\(C_{ex,tfr}\)) is activated by the preceding procedural activation condition (\(C_{ac,prc}\), expressed as a nested institutional statement, that the farmer enters an agreement, whose purpose is defined as an execution constraint (\(C_{ex,pur}\)).

Upon entrance (\(I\)) into agreement (\(B_{dir}\)) with organic farmer (\(A\)) to serve as his/her certifying agent (\(C_{ex,pur}\))(\(C_{ac,prc}\)), organic certifier (\(A\)) must (\(D\)) inspect (\(I\)) farmer’s operation (\(B_{dir}\)) within 60 days (\(C_{ex,tfr}\)).
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, ...). Where logical relationships between conditions or constraints exist, these are made explicit by annotating those as \textbf{AND}, \textbf{OR}, and \textbf{XOR} relationships. Where negation exists, this is to be likewise identified (\textbf{NOT}).

\textbf{Rules:}

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

\textbf{Example 2:}
\textit{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.

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

\begin{quote}
Top-level institutional statement = the Program Manager (A) shall (D) send (I) a written (B_{dir,prop}) notification (B_{dir}) of proposed (B_{a,1,prop}; B_{a,2,prop}) suspension (B_{a,1}) or revocation (B_{a,2}) of certification (B_{a}) to certified organic farmer (B_{ind}).
\end{quote}
4.2 Regulative Statement Coding

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 2.4), 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 [organic farmer (A)] [rebuts (I)] unsuccessfully (Cex,eff)

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 has not completed correction of noncompliance within the prescribed time period”.

IG 2.0 Codebook Version: 1.0
4.2 Regulative Statement Coding

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

"if certified (A,prop1) organic (A,prop2) certifier (A) has not (D) completed (I) correction (B_{dir},prop) of non-compliance (B_{dir}) within the prescribed time period (C_{ex,tfr})"

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.
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 non-compliance ...*  
(in which case the attribute is tacit).
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:

\[
\text{When } \text{[Program Manager } (A) \text{]} \text{ reveals} (I) \text{ any non-compliance} (B_{\text{dir}}) \text{ by the accrediting} (B_{\text{ind,prop1}}) \text{ certifying} (B_{\text{ind,prop2}}) \text{ agent} (B_{\text{ind}}) \text{ with the Act or regulations in this part} (C_{\text{ex,eff}}) \{ \text{[under the condition that] Program Manager } (A) \text{ [performs]} (I) \text{ inspection} (B_{\text{dir}}) \text{ of an accredited} (B_{\text{ind,prop1}}) \text{ certifying} (B_{\text{ind,prop2}}) \text{ agent} (B_{\text{ind}}), \ldots
\]

The final part of the statement (the main statement), likewise reformulated in active terms, implies that [Program Manager] shall [send] a written (B_{\text{dir,prop1}}) notification (B_{\text{dir}}) of non-compliance (B_{\text{dir,prop2}}) to the certifying (B_{\text{ind,prop1}}) agent (B_{\text{ind}}).

Structurally, in this statement we observe two levels of component-level nesting in the form \textit{ABDIC}\{\textit{ABDIC}\{\textit{ABDIC}\}\}, where the main statement’s (shall send notification; first \textit{ABDIC}) activation relies on revealing potential non-compliance (second \textit{ABDIC}), which in itself relies on the inspection in the first place (last \textit{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 6).
### 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 incongruences 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
- 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.

Certified \((A, \text{prop})\) operations \((A)\) must \((D)\) comply \((I)\) with organic farming \((B_{\text{dir}}, \text{prop})\) regulations \((B_{\text{dir}})\)

AND

Certified \((A, \text{prop})\) handlers \((A)\) must \((D)\) comply \((I)\) with organic farming \((B_{\text{dir}}, \text{prop})\) regulations \((B_{\text{dir}})\).

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).
Example Statement:
Certified operations or handlers must accept and comply with organic farming 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: (Operators (A) AND Certifiers (A)) must (D) comply (I) with (regulations (Bdir) AND best practices (Bdir)).

In expanded form (shown on the right), parentheses are then used to signal the association of the decomposed statements.

<table>
<thead>
<tr>
<th>Logical relationships among statement components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where statements make tacit reference to multiple actions, conditions or constraints, these are likewise resolved using the introductions provided in Table 7 (IG Logico), Item “Logical relationships among statement components”. For IG Extended this provision is recommended, but optional.</td>
</tr>
</tbody>
</table>

Grouping of statements using parentheses (in bold font):

(Certified (A, prop) operations (A) must (D) accept (I) organic farming (Bdir, prop) regulations (Bdir) AND Certified (A, prop) handlers (A) must (D) accept (I) organic farming (Bdir, prop) regulations (Bdir) AND Certified (A, prop) operations (A) must (D) comply (I) with organic farming (Bdir, prop) regulations (Bdir) AND Certified (A, prop) handlers (A) must (D) comply (I) with organic farming (Bdir, prop) regulations (Bdir).)
4.2 Regulative Statement Coding

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 6: 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.
## 4.2 Regulative Statement Coding

<table>
<thead>
<tr>
<th>Syntactic Component</th>
<th>Treatment of Syntactic Components by Level of Encoding</th>
<th>Relevant Examples</th>
<th>Complete Syntactic Classification of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation-centric Semantic Annotations</td>
<td>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.</td>
<td>Exploring this approach, we borrow the complex example statement previously coded in the context of IG Extended:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example Statement: <em>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.</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Coded form:</td>
<td></td>
</tr>
<tr>
<td>Rules:</td>
<td></td>
<td>${(When \text{ program manager (A) performs (I)}) \text{ an inspection (B}<em>{\text{dir}} \text{ of an accredited (B}</em>{\text{ind,prop1}} \text{ certifying (B}<em>{\text{ind,prop2}} \text{ agent (B}</em>{\text{ind}}) \text{ [AND] (Program Manager (A) reveals (I) any noncompliance (B}<em>{\text{dir}} \text{ with the Act or regulations in this part (C}</em>{\text{ex,eff}})) (C}<em>{\text{ac,prc}}), [Program Manager (A)] \text{ a written (B}</em>{\text{dir,prop1}} \text{ notification (B}<em>{\text{dir}} \text{ of noncompliance (B}</em>{\text{dir,prop2}} \text{ shall (D) be sent [send (I)]} \text{ to the certifying agent (B}_{\text{ind}}).} \right.$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>${(When \text{ program manager (A) performs (I)}) \text{ an inspection (B}<em>{\text{dir}} \text{ of an accredited (B}</em>{\text{ind,prop1}} \text{ certifying (B}<em>{\text{ind,prop2}} \text{ agent (B}</em>{\text{ind}}) \text{ [AND] (Program Manager (A) reveals (I) any noncompliance (B}<em>{\text{dir}} \text{ with the Act (ref=&quot;policy&quot;) or regulations in this part (ref=&quot;section&quot;) (C}</em>{\text{ex,eff}})) (C}<em>{\text{ac,prc}}), [Program Manager (A)] \text{ a written (B}</em>{\text{dir,prop1}} \text{ notification (B}<em>{\text{dir}} \text{ of noncompliance (B}</em>{\text{dir,prop2}} \text{ shall (D) be sent [send (I)]} \text{ to the certifying agent (B}_{\text{ind}}).}$</td>
<td></td>
</tr>
</tbody>
</table>
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., Act (Bdir,ref=“policy”)).

As stated before, while, in this specific example, cross-statement reference apply to constraints components, such references can likewise occur in other 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 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:

```
{When [Program Manager (A)] reveals (I) any non-compliance (Bdir) by the accrediting (Bind,prop1) certifying (Bind,prop2) agent (Bind)] with regulations in this part (ref="section") (Cex,eff)

{under the condition that} (Program Manager (A) [performs (I)] inspection (Bdir) of an accredited (Bind,prop1) certifying (Bind,prop2) agent (Bind))

OR

(Program Manager (A) [performs (I)] inspection (Bdir) of an accredited (Bind,prop1) certifying (Bind,prop2) agent (Bind))

OR

(Program Manager (A) [performs (I)] inspection (Bdir) of an accredited (Bind,prop1) certifying (Bind,prop2) agent (Bind))

} (Cac), [Program Manager (A)] shall (D) [send (I)] a written (Bdir,prop1) notification (Bdir) of noncompliance (Bdir,prop2) to the certifying (Bind,prop1) agent (Bind).
```
Logical relationships among statement components (ctd.)

… [Program Manager (A)] shall (D) [send (I)] a written (B_{dir}, prop1) notification (B_{dir}) of noncompliance (B_{dir}, prop2) to the certifying (B_{ind}, prop1) agent (B_{ind}).

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

\{[under the condition that] Program Manager (A) [performs (I)] inspection (B_{dir}) of an accredited (B_{ind}, prop1) certifying (B_{ind}, prop2) agent (B_{ind})\} (C_{ac})

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).
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]

\[
( ( \text{Program Manager (A) [performs (1)] inspection (B}_{\text{dir}}) \text{ of an accredited (B}_{\text{ind}, \text{prop}1} \text{ certifying (B}_{\text{ind}, \text{prop}2} \text{ agent (B}_{\text{ind}})) } \\
\text{OR} \\
( \text{Program Manager (A) [performs (1)] review (B}_{\text{dir}}) \text{ of an accredited (B}_{\text{ind}, \text{prop}1} \text{ certifying (B}_{\text{ind}, \text{prop}2} \text{ agent (B}_{\text{ind}})) } \\
\text{OR} \\
...}
Logical relationships among statement components (ctd.)

\[(Program\ Manager (A) \ \text{performs (1)})\ \text{investigation (B}_{\text{dir}})\ \text{of an accredited (B}_{\text{ind},\text{prop1}})\ \text{certifying (B}_{\text{ind},\text{prop2}})\ \text{agent (B}_{\text{ind}}))\]

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.
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 [Program Manager (A)] reveals (I) any non-compliance (B_dir) [by the accrediting (B_ind,prop1) certifying (B_ind,prop2) agent (B_ind)] with the Act (ref="policy") (C_ex,eff)

... (remainder of statements)...

OR

When [Program Manager (A)] reveals (I) any non-compliance (B_dir) [by the accrediting (B_ind,prop1) certifying (B_ind,prop2) agent (B_ind)] with regulations in this part (ref="section") (C_ex,eff)

... (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), including their animacy, actor roles as well as action themes of the coded statement. In contrast to the property annotations used in IG Core and Extended, the annotations introduced here apply across all component types, 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., physical descriptors, such as animate, can be combined with role descriptors, such as Recipient. By taxonomy, categories for such annotations include

- Physical type: animate, inanimate
- 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 the physical type and role taxonomies specified in Section 5.

\[
\begin{align*}
\text{When } & \text{[Program Manager (A; type=animate; role=experiencer)] reveals (I) any non-compliance (B_{dir}; type=inanimate) [by the accrediting (B_{ind},prop1) certifying (B_{ind},prop2) agent (B_{ind};type=animate; role=originator)] with the Act (type=inanimate, ref="policy") or regulations in this part (ref="section") (C_{ex,eff})(C_{ac}), [Program Manager (A; type=animate; role=originator)] shall (D) [send (I)] a written (B_{dir},prop1) notification (B_{dir}; type=inanimate) of noncompliance (B_{dir},prop2) to the certifying (B_{ind},prop1) agent (B_{ind}; type=animate; role=recipient).}
\end{align*}
\]
IG Logico further prescribes annotations of statements with institutional functions. While other features described above in practice focus on the refined coding of attributes, objects and conditions, institutional actions reflect the institutional function of a statement’s action.

Being fined, for example, reflects the institutional function of “sanctioning”, adhering to regulation reflects “compliance”.

Institutional 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

Taking the previously used statement as an example (including inferred components, but omitting any annotations), we can identify institutional 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 institutional 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.
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 “function” and the value carries the corresponding institutional 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 Program Manager (A) may (D) initiate (I; function=sanction) revocation proceedings (Bdir) against a certified operation (Bind)

{When the Program Manager (A) has reason to believe (I; function=evaluate) that a certified operation (A) has violated (I; function=violate) the Act (Bdir; ref="policy") (Bind)}

OR

When a certifying agent (A) fails (I; function=violate) to take appropriate action (Bdir) to enforce the Act (Cex; pur; ref="policy") (Cae).
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 institutional 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 7: Coding Guidance on Syntactic Elements for IG Logico as Level of Expressiveness (Regulative Statements)
4.3 Constitutive Statement Coding

Constitutive statements are treated analogous to regulative statements, offering selected syntactic correspondence and corresponding refinements across levels of expressiveness (visualized in Figure 10), with variations relating to the structural decomposition of selected syntax elements and the explicit treatment of statement-level decomposition for constitutive-regulative hybrids (see Section 4.4), as well as semantic annotations on IG Logico.

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

Mirroring the introduction of coding guidelines for regulative statements, in Table 8 we provide 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 in the previous tables, symbols associated with IG Extended are held in green, and features associated with IG Logico are displayed in orange.
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. | Example statement: 
There is hereby established a public Food Security Advisory Board. 
Entity = Food Security Advisory Board 
Entity property = public 
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. 
Additional example: 
Established in this Regulation subpart is the right to appeal to a revocation or certification. 
Established (F) in this Regulation subpart (C<sub>ex</sub>) is the right to appeal (E) to a revocation or certification (E, prop). | There is hereby (C<sub>ex</sub>) established (F) a public (E, prop) Food Security Advisory Board (E). 
No (NOT) member (E) of the Council (E, prop) shall (D) be (F) disqualified from holding any public office or employment (P). |
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.
### 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.

**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

---

### Deontic

The Deontic signals the extent to which the instruction contained in the constitutive statement is prescribed or signals discretion. In constitutive statements the use of the deontic can also be of conventional nature and relies on the contextual interpretation based on disciplinary (e.g., legal) traditions and stylistic conventions.

**Example:**
A majority of the members of the Council shall constitute a quorum.

Deontic: shall

**Example:**
The Council shall have an advisory committee.

Deontic: shall

**Example:**
A majority of the members of the Council shall constitute a quorum.

Deontic: shall

**Example:**
The Council shall have an advisory committee.

Deontic: shall
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 Policy Council reporting requirements apply for any communication between the Council and Regional Council 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.
The encoding of Or else statements identifies consequences (e.g., payoffs) of compliance/non-compliance or violation with institutional statements, which, in the context of constitutive statements, can be of consequential as well as existential nature. 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.

Principles of horizontal and vertical nesting, as described in the regulative context, equally apply for constitutive statements.

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:**

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.

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.

In student recruitment plans \((C_{ex})\), diversity \((E)\) must \((D)\) mean \((F)\) diversity in race, religion, sexual orientation and gender \((P)\), or else \((OR\ ELSE)\) \([\text{plan (E) is (F) void (P)}]\).

In the previous example, horizontal nesting is signaled using parentheses around statements (as opposed to individual components), and vertical nesting is expressed using brackets (\([\text{and }])\).
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)

**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

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” (Table 7), Item “Cross-component Semantic Annotations”. 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.

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

The Committee (E; label=inanimate) shall (D) consist (F) of a President, Secretary, and Treasurer (P; label=animate).
4.3 Constitutive Statement Coding

Decomposition of component-level combinations

(Note: This applies to regulative and constitutive statement, 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, 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 “General IG Extended Instructions” (Table 6), Item “Decomposition of component-level combinations”

Example (Multiple Properties):

The Committee shall consist of a President, Secretary, and Treasurer.

While expressed in condensed form as “The Committee (E) shall (D) consist (F) of a (President, Secretary, and Treasurer) (P)” (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 Committee (E) shall (D) consist of (F) a (President AND Secretary AND Treasurer) (P).

Expanded form:
(The Committee (E) shall (D) consist of (F) a President (P) AND The Committee (E) shall (D) consist of (F) a Secretary (P) AND The Committee (E) shall (D) consist of (F) a Treasurer (P)).

Council form (E) is hereby established (F)

AND

Council function (E) is hereby established (F).
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 8: 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 characterisation 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.

<table>
<thead>
<tr>
<th>Syntactic Component</th>
<th>Treatment of Syntactic Components by Level of Encoding</th>
<th>Relevant Examples Complete Syntactic Classification of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constituted Entity</td>
<td>In IG Extended encoding, Constituted Entities and their properties are decomposed hierarchically following the principles of the Attribute/Object-Property Hierarchy (Section 2.4) and is applied analogous to “Attributes” in IG Extended for regulative statements (Table 6).</td>
<td></td>
</tr>
</tbody>
</table>

IG 2.0 Codebook Version: 1.0
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 Attribute/Object-Property Hierarchy (as introduced in Section 2.4 and applied in the context of “Objects” in IG Extended for regulative statements in Table 6).

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., \( P_a, P_b \)) equivalent to “Object” decomposition highlighted in Table 6 and exemplified in the following.

The Council (\( E \)) consists of (\( F \)) elected (\( P,prop1 \)) officials (\( P \)) resident in the electorate (\( P,prop2 \)).

(A majority (\( P,prop1,prop1 \)) of the members (\( P,prop1 \)) of the Council (\( P \))(\( P \)) shall (\( D \)) constitute (\( F \)) a quorum (\( E \)).
Constitutive 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.\(^2\)

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).

<table>
<thead>
<tr>
<th>Context</th>
<th>See “Context” in IG Extended for regulative statements (Table 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General IG Extended Instructions</strong></td>
<td>See “General IG Extended Instructions” in IG Extended for regulative statements (Table 6)</td>
</tr>
<tr>
<td><strong>Constitutive-regulative Hybrids</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

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

\(^2\) Specific data structure patterns commonly found in institutional statements are revisited in Section 4.4.5.
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

**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"), and finally

<table>
<thead>
<tr>
<th>Syntactic Component</th>
<th>Treatment of Syntactic Components by Level of Encoding</th>
<th>Relevant Examples</th>
<th>Complete Syntactic Classification of Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting January 1st (<em>C</em>&lt;sub&gt;ac&lt;/sub&gt;), the Connecticut Food Policy Council (<em>E</em>) shall (<em>D</em>) be (<em>F</em>) within the Department of Agriculture (<em>C</em>&lt;sub&gt;ex&lt;/sub&gt;).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In this example the constitutive function signals the constituted entity (Connecticut Food Policy Council) as an organizational unit.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Committee shall consist of a President, Secretary, and Treasurer.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The constitutive function signals a composition of the constituted entity (Committee) based on constituting properties.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In this example, the constitutive function identifies the entity as a policy and signals the intent underlying the policy.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Committee (<em>E</em>) shall (<em>D</em>) consist of (<em>F</em>; <em>confunc=composition</em>) a (President AND Secretary AND Treasurer) (<em>P</em>).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The purpose of this Part (<em>E</em>) is (<em>F</em>; <em>confunc=intent</em>) to establish standards for net metering in accordance with the requirements of Section 16-107.5 (<em>ref=Section/16-107.5</em>) of the Act (<em>P</em>).</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Constitutive Function Annotations (ctd.)

- defined based on lifecycle stages ("established", "terminated"). A final further form of constitutive functions relates to their application in the context of conferral 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.

Table 10: Coding Guidance on Syntactic Elements for IG Logico as Level of Expressiveness (Constitutive Statements)
4.4 Constitutive-Regulative Hybrids

In addition to the specific treatment of regulative and constitutive statements as part of the coding guidelines, an aspect that demands specific attention in the coding guidelines is their combined use. 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 separation of constitutive and regulative statements into distinct sections (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 behaviour of the constituted entities. Linking the nested relationships of institutional statements across both types, we characterize the combined use of constitutive and regulative statements as constitutive-regulative hybrids (where the overall statement is of constitutive nature) or regulative-constitutive hybrids (where the leading statement is of regulative nature). Where existing, their resolution is a central feature of IG Extended (and optional for IG Core). In the following 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](image)

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.³

³In the context of this section, parentheses and logical operators are color-coded to emphasize the association with the corresponding institutional statement type.
4.4 Constitutive-Regulative Hybrids

Figure 12: Coded Regulative-Constitutive Hybrid Example

This interleaved representation can afford a decomposition of hybrids into individual statements by separating the statements by syntactic components, and replication of components where overlapping. This decomposition is exemplified in Figure 13.

Figure 13: Decomposed Regulative-Constitutive Hybrid Example

The producer (A) must (D) (establish and maintain) (I) preventive (B_{dir,prop}; E_{prop}) livestock health care practices (B_{dir}; E), including (F):

((1) Selection of species and types of livestock with regard to suitability for site-specific conditions and resistance to prevalent diseases and parasites; AND/AND
(2) Provision of a feed ration sufficient to meet nutritional requirements, including vitamins, minerals, protein and/or amino acids, fatty acids, energy sources, and fiber (ruminants); AND/AND
(3) Establishment of appropriate housing, pasture conditions, and sanitation practices to minimize the occurrence and spread of diseases and parasites.)(P)
4.4 Constitutive-Regulative Hybrids

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 violation of a constitutive statement can be 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](image)

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](image)

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 terms of an embedded constitutive statement that in itself captures a complex set of properties that constitute a feed ration.
Another final aspect of discussion relates to the identification of constitutive and regulative statements. The function of either statement type is generally well defined based on the introduction or modification of entities or endowment of rights/authority for constitutive statements, and the specification of operational duties and constraints for given actors (or actor interaction) for regulative statements. In practice, however, statements alone may offer limited clarity as to whether they are of constitutive or regulative nature in the first place.

Possible reasons for and approaches to resolution this ambiguity include the nature of the policy more generally, i.e., in how far the specific statement regulates central concerns of the policy, as opposed to parameterizing the scenario, or contextualizing the policy itself. Other, more specific indicators include the positioning of the statement within the document. If located in the preamble, for example, the intended interpretation as constitutive statement is likely. Another consideration is the immediate context of a statement, i.e., its surrounding statements. Those may offer a clearer indication as to whether the statement is of configurational nature (constitutive), or bears operational weight (regulative). A further heuristic is of stylistic nature. Reviewing the coded document, the coder may find specific terms characteristic for constitutive or regulative statements in the context of the policy and/or field (e.g., the use of “shall” as indicative for constitutive statements, if obligations in regulative statements are commonly expressed using more distinctive and specific deontics). Notwithstanding, a pragmatic approach to seek support for either position, is to code the statement using both syntactic forms, and identifying in how far the encoding affords extensive reformulation or reconstruction of a statement in order to arrive at a sensible coding outcome.

As an alternative to the focus on statement purpose, analytical objectives may drive a preference for either statement type in the form of a default strategy for the treatment of ambiguous statement types. For example, if the understanding of actor relationships across a given policy is of fundamental concern, but reconstruction of configurational aspects external to actorship are secondary, a potential default strategy for the coding of ambiguous statements (e.g., defined as part of the project-specific guidelines) could be their interpretation as regulative.

Where such ambiguity, and in consequence, flexibility exists, statements can be considered polymorphic institutional statements, or syntactic polymorphs. This means, they can take either shape based on the ambiguity they exhibit, but also the application they are subjected to.

To substantiate this approach with an example, we can use the statement “The functions of the Board shall be: (a) give effect to the decisions and policies of the Health Assembly; (b) act as the executive organ of the Health Assembly; (c) perform any other functions entrusted to it by the Health Assembly.”

This statement can be read in constitutive terms, i.e., the characterization of the board in terms of its 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 in operational terms. The same statement can thus be encoded in terms of both statement types, as visualized in the following (including color coding to signal regulative (blue) and constitutive components (purple), respectively.)

```
The functions (E) of the Board (A) shall (D/D) be (F):
   ((a) give (I) effect (Bdir) to the decisions and policies of the Health Assembly (Bind);
   AND
   (b) act (I) as the executive organ of the Health Assembly (Cex);
   AND
   (c) perform (I) any other functions (Bdir) entrusted to it by the Health Assembly
      (Bdir,prop))(P).
```
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.

While the notion of polymorphic structure is explicit in the previous example, the coding of statements in both terms can be more complex and may even afford reconstruction, as shown in the following example, in which the same statement is coded separately in constitutive and regulative terms:

“No member of the Council shall be disqualified from holding any public office or employment.”

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

No (NOT) member (E) of the Council (E, prop) shall (D) be (F) disqualified from holding any public office or employment (P).

Correspondingly, the Council member is at the center of the encoding. Depending on the analytical use, constitutive statements reflect an abstract conception of a right or a property. Expressing this assurance in regulative terms – offering a more concrete characterization of involved actors and actions – requires the operationalization of behavioural constraints by introducing a conception of actorship – a tacit actor whose behaviour is constrained, alongside further structural adaptations. Reconstructing the statement in regulative terms thus produces the following coding:

[Attribute (A)] [shall not (D)] disqualify (I) member of the council (B) from holding any office or employment (Cex).

We can recognize that in this case, the encoding of the statement in constitutive or regulative terms may invoke varying levels of complexity with respect to necessary adaptations of the statements, which, depending on analytical objectives, may potentially modify the semantics of a statement (in the given example, a concrete actorship for regulative statements is presumed, whereas the right is expressed more generally in constitutive terms) or afford a reformulation that may not be justifiable on methodological grounds. In addition to considering a dual annotation in the first place, the potential mischaracterization of a statement as either regulative or constitutive may sensibly be considered in inter-rater reliability tests, since it offers the opportunity to resolve disagreements and misconceptions early in the encoding process.
4.4.5 Data Structure Patterns

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

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.
4.5 Coding Level 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 11 for regulative statements, and Table 12 for constitutive statements, along with a symbol association used for the ensuing specification of configurations.

<table>
<thead>
<tr>
<th>Coding Level</th>
<th>Feature</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>IG Core</td>
<td>Attributes</td>
<td>A</td>
</tr>
<tr>
<td>IG Core</td>
<td>Object</td>
<td>B</td>
</tr>
<tr>
<td>IG Core</td>
<td>Deontic</td>
<td>D</td>
</tr>
<tr>
<td>IG Core</td>
<td>Aim</td>
<td>I</td>
</tr>
<tr>
<td>IG Core</td>
<td>Context</td>
<td>C</td>
</tr>
<tr>
<td>IG Core</td>
<td>Or else</td>
<td>O</td>
</tr>
<tr>
<td>IG Extended</td>
<td>Attributes refinements</td>
<td>A&lt;sub&gt;Ext&lt;/sub&gt;</td>
</tr>
<tr>
<td>IG Extended</td>
<td>Object refinements</td>
<td>B&lt;sub&gt;Ext&lt;/sub&gt;</td>
</tr>
<tr>
<td>IG Extended</td>
<td>Context refinements</td>
<td>C&lt;sub&gt;Ext&lt;/sub&gt;</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Statement references</td>
<td>R</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Logical relationship annotations</td>
<td>L</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Semantic annotations</td>
<td>S</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Institutional function annotations</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 11: 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 the description of the element content):

<Baseline coding level>–<omitted features from baseline coding 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<sub>Ext</sub>, where the +C<sub>Ext</sub> signals features from the next higher level (IG Extended).

Conversely, we can specify coding on IG Core level 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 as specified in the table (here: Aim before Or else). Selectively cap-
4.5 Coding Level Configurations (Institutional Grammar Profiles)

<table>
<thead>
<tr>
<th>Coding Level</th>
<th>Feature</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>IG Core</td>
<td>Constituting Properties</td>
<td>P</td>
</tr>
<tr>
<td>IG Core</td>
<td>Deontic</td>
<td>D</td>
</tr>
<tr>
<td>IG Core</td>
<td>Constituted Entity</td>
<td>E</td>
</tr>
<tr>
<td>IG Core</td>
<td>Constitutive Function</td>
<td>F</td>
</tr>
<tr>
<td>IG Core</td>
<td>Context</td>
<td>C</td>
</tr>
<tr>
<td>IG Core</td>
<td>Or else</td>
<td>O</td>
</tr>
<tr>
<td>IG Extended</td>
<td>Constituting Properties refinements</td>
<td>P_{Ext}</td>
</tr>
<tr>
<td>IG Extended</td>
<td>Constituted Entity refinements</td>
<td>E_{Ext}</td>
</tr>
<tr>
<td>IG Extended</td>
<td>Context refinements</td>
<td>C_{Ext}</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Statement references</td>
<td>R</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Logical relationship annotations</td>
<td>L</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Semantic annotations</td>
<td>S</td>
</tr>
<tr>
<td>IG Logico</td>
<td>Constitutive function annotations</td>
<td>U</td>
</tr>
</tbody>
</table>

Table 12: IG Feature Specifications for Constitutive Statements

turing 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+SF**, to signal the coding of Object and Context on IG Core level, while considering semantic annotations and institutional functions 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.

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}SF** signals IG Core baseline encoding under omission of Attributes and Aim, while adding refined coding of Context (based on IG Extended), along with semantic annotations and institutional functions (from IG Logico).

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 of existing ones) should likewise 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 (e.g., to accommodate domain-specific characteristics or analytical necessities) is explicitly permitted. Both cases should be clearly indicated and defined or referred to in conjunction with the coded dataset.

5.1 Circumstances Taxonomy

The circumstances 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. Note that the listed categories include an embedded hierarchy, with more specific labels noted indented. Where possible (and analytically useful/specfied in project-specific guidelines), the more specific annotation should be used. Note that the state category is of highest abstractions and includes temporal, spatial and other categories. Where other categories do not apply, the coder can test for the more general notion of state. The suggested annotation label prefix – if applied – is ctx.

- 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”)

- State (ste) – References to a specific state (e.g., “during childhood”); note that state is more general than temporal and spatial specification. Where possible, a more specific annotation should be chosen.
  - Beginning
  - End

- State transition (tra) – References to a change in state (e.g., “when traffic light switches from red to green”)

IG 2.0 Codebook Version: 1.0
• Procedural order (prc) – Conditions/Constraints associated with explicit or implied execution order. 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 required actions, de/activation of other statements or compliance with or violations of statements, respectively.

• Method (met) – Conditions/Constraints associated with means or method by which an action is performed
  – Means – Action as method (e.g., “by handshake”)
  – Instrument – Artefact as method (e.g., “by car”)

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

• Observed state/Outcome/Effect (eff) – Conditions/Constraints describing the outcome or effect of an aim by an actor involved in the action situation – a change in the environment emanating from the observed actor(s); observation of compliance/non-compliance (e.g., “if an non-compliant action is observed”; “if participant fails to meet certification requirements”). This characterization is commonly invoked when coding discretionary actions of observers (e.g., beliefs, suspicions, evaluations).

5.2 Physical Type Taxonomy

The physical taxonomy 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 type.

• Animate – Living entities

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

• Goal – Goal is a specialization of an inanimate mental construct

5.3 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. The suggested annotation label prefix is role.

• Originator/Causer/Agent – Entity from which action originates

• Recipient – recipient of an artefact/sanction

• Possessor – owner of an object/entity (e.g., “house owner”)

• Experiencer – observer of action (e.g., “observer of non-compliance”)

• Beneficiary – beneficiary of action; may not necessarily be action/artefact recipient (e.g., “welfare recipient”)

• Position – organisation or institutional role assumed by involved actor
5.4 Institutional 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, referred to as “institutional function.” Institutional 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 institutional functions organized by categories (alongside potential specializations), but does not claim exhaustiveness. The suggested annotation label prefix is function.

- 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
- Process – Life cycle
  - Initiate
  - Interrupt
  - Resume
  - Conclude
- Transaction – action reflecting a request and corresponding response
  - Request – action reflecting a request.
  - Response – action reflecting a response to a request. Central difference to enforcement responses is lack of a regulatory compliance function.
    * Accept
    * Reject
- Decide – action reflecting a decision/discretion
- Inform – action reflecting information dissemination
5 Taxonomies

- Declare – action reflecting change in role, position, environment, institutional fact
- Assign – action reflecting the assignment of responsibilities to other actors
  - Delegate – action reflecting the delegation of functions/tasks, generally downwards in an organisational structure (e.g., delegation to subordinate)
  - Elevate – action reflecting the elevation of functions/tasks, generally upwards in an organisational structure (e.g., elevation to supervisor)

5.5 Constitutive Functions Taxonomy

While institutional function characterizations map the diverse operational expression of institutionally-relevant, the constitutive function annotations emphasize the specific role a constitutive function entertains with respect to the constituted entity and/or the linkage of constituted entity and constituting properties. Generally, the top-level distinction is the identification of the constituted entity as either an entity established or 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.

![Constitutive Functions Diagram](image)

**Figure 16: Constitutive Functions**

Where entities subject to the policy (or introduced by policy) are of concern, constitutive functions can capture the definition of constituted entities as relevant for the parameterization of the institutional setting, including actors, object/artefacts, role specifications or actions. This definition is often signaled in the form of intensional definitions (e.g., explicit definition), or implied by ascription (e.g., implicit characterization of entity based on behavior).

In addition to the definition of entities, constitutive functions can capture composition relationships (e.g., specifying the composition of committees) and further reflect hierarchical relationships (e.g., leadership structures, embedding positions within organizations).
Further characterizations include the initiation and termination of entity lifecycles (e.g., dates of termination), and the explicit conferral of institutional power, such as authority or rights to entities (e.g., authority to enforce, the right to vote).

Complementing the perspective on entity specification as part of constitutive statements, we further identify constitutive functions that characterize the policy or document itself, as opposed to focusing on the entities central to the representation of the institutional setting. 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 a given policy. Informational statements offer supplementary information about the document, or state institutional facts contextualizing the policy or domain of concern.

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

6 Conclusion

The listing of taxonomies complements and concludes the coding guidelines for the Institutional Grammar 2.0. 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, the choice of which is subject to analytical objectives. Following the discussion of essential document preparation steps (pre-coding steps) in Section 3, detailed coding guidelines are provided for regulative and constitutive statements across all IG 2.0 levels (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 4.4.5). The listing of taxonomies 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. Doing so, they may not capture specifics potentially relevant for a given project. Instead, many aspects of the coding guidelines are of suggestive nature to inform the development of 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. The instructions provided here are further tool-agnostic, and open to adaptation for arbitrary encoding means (e.g., Excel sheets, text annotation tools, etc.).

For supplementary information, both including a theoretical treatment of the underlying concepts and principles, as well as resources that support operational coding (e.g., videos, tool-specific guidance, software), please refer to https://institutionalgrammar.org/resources/.

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


