

GLYDE-II – GLYcan structural Data Exchange using Connection Tables

William S. York, Krzysztof J. Kochut, John A. Miller, Satya Sahoo, Christopher Thomas, and Cory Henson

1. Overview.

GLYDE-II is a standard for the representation of the chemical structures of complex glycans that is based on a connection table formalism using XML syntax. The GLYDE-II standard can be divided into two conceptually distinct parts, syntax and implementation. The syntax of a GLYDE-II document is fully defined by its schema (either a DTD or XML schema, Section 2), which provides a framework we call PARCHMENT (PARtonomy of CHEMical ENTities). PARCHMENT allows the complete structure of biological molecules (including complex glycans) to be completely and unambiguously specified at several levels of granularity. That is, PARCHMENT is a modular approach to specify molecular structure in terms of the parts that make up the whole. It provides a very general, machine readable format (XML) for their representation. The implementation of the GLYDE-II standard also includes a set of rules, naming conventions for the parts, and enumeration of chemical entities that are acceptable parts at various levels of granularity. These implementation rules are absolutely required for representational consistency and disambiguation. However, purely syntactic enforcement of these rules (e.g., solely by the GLYDE-II schema) would be very difficult and would probably result in an unstable standard. This document describes the XML syntax and implementation rules for GLYDE-II.

1.1. Partonomy and Granularity. The fundamental relationship between objects in the GLYDE-II formalism is partonomy (also known as mereology – [Casati and Varzi, 1999]). That is, larger structures are defined by their parts. For example, a *molecule*¹ is a complex entity that consists of parts that are connected to each other. A part can be a *moiety* (such as a glycosyl moiety), a *residue* (such as a glycosyl residue or an amino acid residue) or a *bound_atom* (such as a carbon atom that is covalently linked to another atom). Two parts of a *molecule* can be connected by a *link*. It is important to emphasize that one *molecule* cannot be connected to another *molecule* by a *link*. Thus, a *molecule* is an independent entity, unlike its parts, which are linked together. Another independent entity is a *free_atom*, which is an atom that is not bound to any other atom. The third independent entity is an *aggregate*, which is composed of independent entities (*molecules*, *free_atoms*, and/or other *aggregates*), which are not linked to each other.

An atom (either a *free_atom* or a *bound_atom*), which has no parts in the GLYDE-II formalism, is defined in the common chemical sense and includes entities such as “Oxygen_atom” and “Carbon_atom”. Larger structures are built up from smaller parts in a hierarchical manner, using independent entities (*molecule* or *free_atom*) as archetypes to specify the parts of larger entities. For example, a *free_atom* (“C”, “H”, “N”, “O”, etc.) can be referenced (used as an archetype) to specify a *bound_atom* within a *molecule*, such as a monosaccharide. This monosaccharide, in turn, can be referenced (used as a archetype) to specify a glycosyl *residue* that is a part of a

¹ In this document, specific GLYDE-II elements and attributes are *italicized*, definitions of the XML language itself are *italicized and boldface*, GLYDE-II code is in `Courier` font, and monosaccharideDB terms are in **boldface**. For a description of XML, see <http://www.w3.org/XML/> and <http://www.w3schools.com/xml/default.asp>.

glycan *molecule*. This glycan *molecule*, in turn, can be referenced (used as an archetype) to specify a glycosyl *moiety* that is part of a glycoconjugate *molecule*. At a larger scale, a *molecule*, which by definition is not covalently attached to any other entity, can be referenced to specify a part of an *aggregate*. Thus, the monomeric form of avidin is a glycoprotein *molecule*, while the native form of avidin is an *aggregate* composed of four avidin *molecules*.

In common biochemical language, a residue is often defined as a structural subunit of a biological molecule that is released by a hydrolytic reaction. Thus, a GLYDE-II *residue* may correspond to a glycosyl residue (such as β -D-Glcp and α -L-Fucp), an amino acid residue (such as L-Gly and L-Asn) or a lipid residue (such as oleic acid). A GLYDE-II *moiety*, such as a glycosyl moiety or peptide moiety within a glycopeptide, is composed of at least one *residue*.

The GLYDE-II hierarchy allows structures to be represented at several different levels of granularity. For example, a software application may only require information specifying that the molecule of interest contains carbohydrate *moiety X* and peptide *moiety Y*, and may not depend on the molecular details of these structures. In this case, a very coarse granularity will suffice, and it will not be necessary to parse the GLYDE-II representation to the atomic level. In contrast, a fully atomistic representation that does not provide for abstraction of larger substructures would not be appropriate for such a case, as this would require the larger substructures (e.g., moieties) to be identified and abstracted by the software application.

2. The GLYDE-II syntax - DTD.

The syntactic aspects of structure abstraction in GLYDE-II are defined by the Document Type Definition (DTD - <http://www.w3schools.com/dtd/default.asp>). The DTD defines XML *elements* (http://www.w3schools.com/dtd/dtd_elements.asp) that make up the GLYDE-II file. For example, the DTD specifies that a GLYDE-II document contains XML *elements* called *molecule* and *residue*. Each XML *element* can have several *attributes* (http://www.w3schools.com/dtd/dtd_attributes.asp). The most recent version of the GLYDE-II DTD (at the time of this writing) can be found at "http://glycomics.crc.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD". Previous versions of the DTD for GLYDE-II should not be used, because these were created solely for illustrative purposes. Note that GLYDE-II will eventually be implemented using an XML schema (<http://www.w3schools.com/schema/default.asp>). The GLYDE-II DTD is listed below.

```
<!--+++++
      DTD Schema Specification for GlydeII: GLYDE-II_v0.7.DTD

      The free atom, molecule and aggregate ELEMENTS can stand alone.  These
      ELEMENTS are the parts of a GlydeII document, with GlydeII being the ROOT
      ELEMENT.  Note, an aggregate can contain these same elements.
-->

<!ELEMENT GlydeII (free_atom*, molecule*, aggregate*)>

<!--+++++
      An AGGREGATE can contain a mixture of different ELEMENTS, which are not
      attached to each other (i.e., there are no links).

      A model_ref is an ELEMENT that allows reference to an arbitrary external
      description of the ELEMENT in which it is used.  Only applies to top level
      elements (free atom, molecule, or aggregate).
```

```

-->
<!ELEMENT aggregate (free_atom*, molecule*, aggregate*, model_ref?)>

<!--+++++
A MOLECULE can be composed of uniform ELEMENTS selected from the following
set: (bound_atom, residue, moiety) and corresponding links (atom_link,
residue_link or moiety_link). A molecule cannot contain a mixture of
diffent kinds of ELEMENTS, e.g., moieties and residues cannot be ELEMENTS
of the same molecule. Also, the links in a molecule must correspond to the
molecule's ELEMENT types. For example, a molecule that contains residues
can only contain residue_links.

A molecule can be one of several subtypes.
A molecule that has an unpaired electron is a radical.
An ion has a non-zero charge.
-->

<!ELEMENT molecule (((bound_atom*, atom_link*) |
                    (residue*, residue_link*) |
                    (moiety*, moiety_link*)),
                    combination*, repeat_block*, model_ref?)>

<!ATTLIST molecule
  id          CDATA #REQUIRED
  subtype     (glycoprotein | glycopeptide | glycolipid | peptidoglycan | glycan |
              protein | peptide | lipid | monosaccharide | amino_acid) #IMPLIED
  name        CDATA #IMPLIED
  radical     (yes | no) "no"
  charge      CDATA "0">

<!--+++++
A FREE_ATOM is not linked to any other structure.
-->

<!ELEMENT free_atom (model_ref?)>

<!ATTLIST free_atom
  id          CDATA #REQUIRED
  name        CDATA #IMPLIED
  radical     (yes | no) "no"
  charge      CDATA "0">

<!--+++++
The moiety, residue, and bound_atom ELEMENTS are parts of molecules. Each
of these has a ref ATTRIBUTE whose value references a previously defined
archetype for the part (hence the EMPTY specification of subelements).
The ref ATTRIBUTE can take the following forms:

1. archetype defined in the same file
ref="#molecule_3"

2. archetype dynamically generated by external database using http get method
ref="http://www.monosaccharidedb.org/GLYDE-II.jsp?G=a-dglc-hex-1:5"

3. archetype defined in a separate, static file, accessed as a web address:
ref="http://glycomics.ccr.c.uga.edu/GLYDE-II/atoms.xml#C"

Molecules serve as the archetypes for moieies and residues, while free_atoms
serve as the archetypes for bound_atoms.
-->

<!--+++++
A MOIETY consists of residues, but is defined by reference to a molecule
that serves as its archetype. This archetype molecule must be composed
of residues.
-->

```

```

<!ELEMENT moiety EMPTY>

<!ATTLIST moiety
  partid    CDATA #REQUIRED
  subtype   (glycan | protein | peptide | lipid) #IMPLIED
  ref       CDATA #REQUIRED
  name      CDATA #IMPLIED>

<!--+++++
A RESIDUE consists of bound atoms, but is defined by reference to a molecule
that serves as its archetype.  This archetype molecule must be composed of
atoms.

A residue can have several subtypes.  Some classically defined residues
(such as GlcNAc) are defined using more than one GLYDE-CT residue.
Carbohydrate residues are composed of a base_type residue and zero or more
substituent residues.  For example, a-D-GlcNAc is composed of the base_type
residue "a-dglc-hex-1:5" and the substituent residue "n-acetyl".  The
"n-acetyl" substituent is defined by reference to a previously defined
"acetamide" archetype molecule.
-->

<!ELEMENT residue EMPTY>

<!ATTLIST residue
  partid    CDATA #REQUIRED
  subtype   (lipid | amino_acid | base_type | substituent) #IMPLIED
  ref       CDATA #REQUIRED
  name      CDATA #IMPLIED>

<!--+++++
A BOUND_ATOM is linked to other bound_atoms within a molecule.  It is
defined by reference to a free_atom that serves as its archetype.  The
parity of an atom defines its stereochemistry, using the InChI notation.
The InChIatom ATTRIBUTE specifies the InChI numbering of the atom in the
context of the molecule that contains it.
-->

<!ELEMENT bound_atom EMPTY>

<!ATTLIST bound_atom
  partid    CDATA #REQUIRED
  ref       CDATA #REQUIRED
  name      CDATA #IMPLIED
  parity    (-1 | 1) #IMPLIED
  InChIatom CDATA #IMPLIED>

<!--+++++
A link (specifically a moiety_link, residue_link or atom_link) represents a
connection between structures.
-->

<!--+++++
A MOIETY_LINK connects two moieties and wraps residue_links.

The stat ATTRIBUTE specifies the probability that the link exists in the
molecule.  For example, a peptide moiety may bear an N-glycan moiety 30% of
the time.  In this case, the stat ATTRIBUTE of the link connecting the
N-glycan to the peptide would be 0.3.
-->

<!ELEMENT moiety_link (residue_link*)>

<!ATTLIST moiety_link
  from    CDATA #REQUIRED
  to      CDATA #REQUIRED

```

```

stat CDATA #IMPLIED>

<!--+++++
A RESIDUE_LINK connects two residues and wraps atom_links.
-->

<!ELEMENT residue_link (atom_link*)>

<!ATTLIST residue_link
  from CDATA #REQUIRED
  to CDATA #REQUIRED
  stat CDATA #IMPLIED>

<!--+++++
An ATOM_LINK connects two bound_atoms. An atom_link not only specifies
connections between atoms, it may also specify the replacement of atoms that
accompanies formation of the link. For example, formation of a glycosidic
bond involves the liberation of a water molecule. In this case, the
anomeric oxygen of the monosaccharide archetype of the glycosyl residue is
lost (i.e., replaced by an oxygen atom on the aglycon.)
For example, the following values might be specified for a glycosyl linkage

<residue_link from="residue_A" to="residue_B">
  <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
</residue_link>

The assertion < to_replaces="O1" > specifies that formation of the bond
from C1 of residue_A to O4 of residue_B results in the replacement of O1 of
residue_A by O4 of residue_B. That is, O4 of residue_B (i.e., the
"to" atom) replaces O1 of residue_A. In this case, < from_replaces > is
not defined, as no atom that is defined in residue_B is replaced. (Hydrogen
atoms are not specified in GLYDE-II.) The replaced atom (O1 of residue_A)
is released as part of a water molecule. Replacement of hydrogen atoms
(which form the rest of the water molecule) are not specified.
The bond_order is the number of bonds between the two atoms. Bond parity
is specified using the InChI bond parity notation to, for example,
differentiate cis and trans double bond structures.
-->

<!ELEMENT atom_link EMPTY>

<!ATTLIST atom_link
  from CDATA #REQUIRED
  to CDATA #REQUIRED
  from_replaces CDATA #IMPLIED
  to_replaces CDATA #IMPLIED
  bond_order CDATA #IMPLIED
  parity (-1 | 1) #IMPLIED>

<!--+++++
A COMBINATION concisely represents several possible structural variations
for an ELEMENT by listing possible links between parts that are defined in
the molecule containing the combination. The links listed in a combination
can be combined programatically to generate all possible structures while
ruling out mutually exclusive links.

-->

<!ELEMENT combination (moiety_link+ | residue_link+ | atom_link+)>

<!--+++++
A REPEAT_BLOCK specifies the (existing) parts (residues, moieties, or
bound_atoms) to be repeated and the tandemly repeated link between copies of
the repeat block. The parts and the links within the repeat block are
specified outside the repeat_block. The only link specified within the
repeat_block is the tandemly repeated link between copies of the
repeat_block. The repeat_number indicates the number times the

```

```

    repeat_block is repeated.
-->

<!ELEMENT repeat_block (repeat_part+, (moiety_link | residue_link | atom_link) )>

<!--ATTLIST repeat_block
    repeat_number CDATA #REQUIRED-->

<!--+++++
    A REPEAT_PART is a component of a repeat_block. The ref for each
    repeat_part must be a partid that identifies a part (residue, moiety, or
    bound_atom) that is included in the molecule containing the repeat_block.
-->

<!ELEMENT repeat_part EMPTY>

<!--ATTLIST repeat_part
    ref CDATA #REQUIRED-->

<!--+++++
    A MODEL_REF is a reference to an external source that defines the ELEMENT.
-->

<!ELEMENT model_ref (#PCDATA)>

<!--ATTLIST model_ref
    type CDATA #IMPLIED-->

```

While, the major features of the DTD are documented internally, additional notes are included below.

- The **root element** is called *GlydeII* and contains **elements** called *free_atom*, *molecule*, and *aggregate*. Each of these is an independent object (not covalently linked to other objects). Each can be present zero or more times.
- The **aggregate element** can contain the same **elements** as the **root element** along with a **model_ref element** that refers to an external description of the aggregate. An aggregate is simply a way to collect independent objects into a set. An example of an aggregate would be a multimeric glycoprotein.
- The **molecule element** is composed of parts and links that connect them. The DTD enforces the restriction that a *molecule* must be made entirely of *moieties*, *residues*, or *bound_atoms*, and these cannot be mixed together. Furthermore, the links in the *molecule* must correspond to the parts. For example, a *molecule* composed of *residues* must only contain *residue_links*. It is important to note that *molecules* always serve as archetypes for the polyatomic parts called *moiety* and *residue* (see below).
- The **free_atom element** has no parts, and contains a **model_ref element** that defines its structure.
- The **moiety**, **residue**, and **bound_atom elements** are building blocks from which *molecules* are constructed. Two **moiety elements** can only be connected by a *moiety_link*, which can wrap a *residue_link* that connects the two *residues* (one in each *moiety*) involved in the *moiety_link*. Similarly, the *residue_link* can wrap an *atom_link* that connects the two *bound_atoms* (one in

each *residue*) that are involved in the *residue_link*. The DTD enforces syntax that help to maintain this hierarchy and assure that each link will be appropriate and symmetrical (i.e., *moiety* to *moiety*, *residue* to *residue*, or *bound_atom* to *bound_atom*).

- Instantiation of a part of a *molecule* (i.e., a *moiety*, *residue*, or *bound_atom*) involves the specification of a *partid attribute*, which is of type *CDATA*. A *partid* can thus be any text, and need not be unique. This makes it possible to reuse the same *partid* for different *parts* in the same GLYDE-II file. For example, two different monosaccharide *molecules* (say “b-D-Glcp” and “b-D-Galp”) in the same GLYDE-II file may both contain a *bound_atom* with a *partid* whose value is “C1”, so this value would not be unique within the GLYDE-II file. One can explicitly refer to “C1” of the b-D-Glcp *molecule* (e.g., using x-path) because it is identified as the *part* with *partid*="C1" that is found within the *molecule* having *id*="b-dglc-hex-1:5". (See section 3.2 for conventions for *id attributes* of carbohydrate residues.)
- A *combination element* is a collection of links that can be combined in different ways to generate several mutually exclusive chemical entities. The combination element is used when there is ambiguity regarding the location of a part or group of parts within a *molecule*, as described in Section 2.5.4.
- The *model_ref element* specifies an alternative representation of the structure, an external database instance of the structure, or a semantic model of the structure. One type of *model_ref* is the fully atomistic InChI representation (<http://www.iupac.org/inchi/>) of the structure, which can be specified when this is practical.

3. Implementation rules for GLYDE-II structures.

The smallest possible structures are atoms (*free_atoms* or *bound_atoms*), as illustrated in the following example.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccruc.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD">
<GlydeII>

  <free_atom name="Carbon-atom" id="C">
    <model_ref>http://www.chemsoc.org/viselements/pages/data/carbon_data.html</model_ref>
  </free_atom>

  <free_atom name="Hydrogen-atom" id="H">
    <model_ref>http://www.chemsoc.org/viselements/pages/data/hydrogen_data.html</model_ref>
  </free_atom>

  <free_atom name="Nitrogen-atom" id="N">
    <model_ref>http://www.chemsoc.org/viselements/pages/data/nitrogen_data.html</model_ref>
  </free_atom>

  <free_atom name="Oxygen-atom" id="O">
    <model_ref>http://www.chemsoc.org/viselements/pages/data/oxygen_data.html</model_ref>
  </free_atom>

  <free_atom name="Carbon-13-atom" id="13C">
    <model_ref>http://www.chemsoc.org/viselements/pages/data/carbon_data.html</model_ref>
  </free_atom>

</GlydeII>
```

Note that the URL of the DTD for GLYDE-II v0.7 is indicated in the <!DOCTYPE> tag. This particular example contains only a few *free_atom elements*. The physical properties of each atom can be found by reference to the *model_ref*.

3.1. Implementation rules for atomic structures and parts. Rules (outside the DTD specification) are defined to enforce vocabulary control for atomic *structures* and *parts*.

Rule 1: The values of the *id attribute* of *free_atom elements* in GLYDE-II are limited to the standard elemental or isotopic string representations, such as “H”, “C”, “13C”, etc.

Rule 2: Atomic *ids* cannot be assigned to non-atomic *structures* in a GLYDE-II document, as this could result in degeneracy of the *id*, which is not allowed.

Rule 3: The values of the *partid attribute* for *bound_atom elements* follow specific guidelines described below. The reason for this rule is that it allows an application to generate GLYDE-II representations of a glycan without explicitly looking up the *partids* of the constituent atoms of each residue in order to assign an inter-residue link between two atoms. For example, a monosaccharide archetype for a *residue* is composed of *bound_atoms*, with *partids* “C1”, “O1”, etc. According to this rule, “1-4” (O-glycosidic) linkage from an aldosyl residue and another monosaccharide residue is always specified by declaring: *from* = “C1” and *to* = “O4”.

The *partids* for *bound_atoms* in a carbohydrate *residue* (where *type*=“base_type”) are assigned using the numbering system sanctioned by the IUPAC-IUB Joint Commission on Biochemical Nomenclature (JCBN), as described in “Symbols for Specifying the Conformation of Polysaccharide Chains” (<http://www.chem.qmul.ac.uk/iupac/misc/psac.html#130>). Quoting that document: “Atoms are thus designated C3, O2, H4, etc. The hydrogen atoms of a methylene group may be distinguished by an additional number, e.g. H61 and H62 where the lower number is selected for the pro-S atom.” This IUPAC-IUB document also indicates that oxygens within a furanose or pyranose ring are named using a number rather than the letter “R”. Thus, the oxygen within the ring of Glcp is identified as “O5” rather than “OR”. Exchangeable hydrogens of hydroxyl groups are not explicitly named in GLYDE-II. (The IUPAC-IUB document uses names such as “O2H” for the hydrogen attached to O2.)

The atomic *partids* for amino-acids are assigned using the numbering system sanctioned by the IUPAC-IUB Joint Commission on Biochemical Nomenclature (JCBN), as described in “Nomenclature and Symbolism for Amino Acids and Peptides (3AA-2)” (<http://www.chem.qmul.ac.uk/iupac/AminoAcid/AA1n2.html#AA22>).

3.2. Implementation rules for monosaccharide molecules and residues. The scope and identities of monosaccharide *molecules* and corresponding monosaccharide *residues* are defined according to the Glyco-CT specification (www.eurocarbdb.org/recommendations/encoding). This includes the following definitions:

base-type: a description of a stereo-chemically defined structures from the chemical class of polyhydroxyaldehydes or ketones, without any substituent. On this level, acidic functions,

double bonds, deoxygenations, sp²-hybridisation, reductions of the anomeric carbon, and additional carbonyls (e.g., keto groups) are encoded. Thus, base-types are polyhydroxy structures composed entirely of C, H and O atoms. This includes many common simple sugars such as glucose, mannose, and fucose. Valid base-types that contain up to four stereo-centres are named using IUPAC nomenclature (*IUPAC Nomenclature of carbohydrates*, <http://www.chem.qmul.ac.uk/iupac/2carb/02.html#0222>) with a single configuration specifier. For example, “dglc” signifies the D-*gluco* configuration. Base-types with more than four stereo-centres also follow established IUPAC naming convention, which results in composite names, such as “dgro-dgal” for D-*glycero*-D-*galacto* **base-types**. Trivial names for such structures are deprecated. The anomeric configuration of the **base-type** is also specified.

Substituent: A non-base-type entity with linkage(s) to a base-type. A substituent is typically a small chemical entity, which is encoded in a list of substituents. The monosaccharideDB (<http://www.monosaccharideDB.org>) manages this list.

Monosaccharide: Every distinct entity of residues connected via glycosidic linkages to other entities – typically a **base-type** with **substitutions**.

In GLYDE-II, *residues* are specified by reference to *molecules*. Glycosyl residues are specified by reference to the corresponding free monosaccharides. However, a **monosaccharide** is composed of its **base-type** and **substituents**. Thus, the “b-D-Glcp” **monosaccharide residue** corresponds to a single **base-type** and is specified by reference to the “b-D-Glcp” *molecule* as its archetype. In contrast, the “b-D-GlcpN” **monosaccharide** cannot be represented as a single *residue*, as it consists of a “b-D-Glcp” *residue* (type=“base_type”) and an “N” *residue* (type=“substituent”). That is, two separate *residues* are required to specify this **monosaccharide** in GLYDE-II. (In GLYDE-II, the “N” *residue* is specified by reference to ammonia [NH₃] as the archetypal *molecule*.) The “b-D-GlcpNAc” **monosaccharide** consists of a “b-D-Glcp” *residue* (type=“base_type”) and an “n-acetyl” *residue* (type=“substituent”). (In GLYDE-II, the “NAc” *part* is specified by reference to acetamide [CH₃CONH₂] as the archetypal *molecule*.)

Rule 4. In GLYDE-II, composite **monosaccharide molecules** that contain both a **base-type residue** and **substituent residues** can be defined by combining these *residues*. However, such composite **monosaccharide molecules** can never be referenced to specify a **monosaccharide residue**. That is, a composite **monosaccharide residue** must be explicitly defined by declaration of its component parts (*i.e.*, **base-type** and **substituents**).

Rule 5. The valid names for all **monosaccharides** (including composite **monosaccharides** consisting of both a **base-type** and **substituents**) are managed by the monosaccharideDB (<http://www.monosaccharideDB.org>), which also manages the naming of **base-types** and **substituents** as well as the naming of atoms in these structures. MonosaccharideDB provides several services. For example, it allows a **monosaccharide** to be identified from its **base-type** and **substituents**, and provides alternate names (IUPAC, trivial) of the **monosaccharide**.

Implementation Note. To maintain vocabulary consistency, GLYDE-II representations of monosaccharide *molecules* that are used as archetypes for carbohydrate *residues* will be dynamically generated by services provided by monosaccharideDB. Thus, full specification of the *ref* attribute of a *part* will include the monosaccharideDB URL and the Glyco-CT name of the *part*. For example,

```
ref=http://www.monosaccharideDB.org/GLYDE-II.jsp?G=a-dman-hex-1:5
```

The GLYDE-II representation can be made more concise by using a DTD *entity* (http://www.w3schools.com/dtd/dtd_entities.asp) to define the URL of the service provided by monosaccharideDB. For example, the GLYDE-II file could contain the following code:

```
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccr.c.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [  
  <!ENTITY mDBget "http://www.monosaccharideDB.org/GLYDE-II.jsp?G">  
>
```

Then, reference to a **base-type** or **substituent** in monosaccharideDB can be succinctly made using the code like the following example:

```
ref="&mDBget;a-dman-hex-1:5"
```

Subsequent examples in this document will use this succinct representation. In the future, the *entity* specifying the URL of the monosaccharideDB service used in this context will be included in the DTD or XML schema itself, to insure that monosaccharideDB is used as the authority for defining residue *partids*.

As indicated above, monosaccharideDB will provide GLYDE-II representations of **base-type** and **substituent** *molecules* that are referenced for use as *residue* archetypes. This representation can be of two forms: “atomistic” or “model reference”. An atomistic GLYDE-II representation contains all of the component atoms and their mutual links. A model reference representation simply contains a reference to the monosaccharideDB record that fully describes the **base-type** or **substituent**. For example, invoking the following URL

```
ref=http://www.monosaccharideDB.org/GLYDE-II.jsp?G=a-dman-hex-1:5
```

will initially provide the following fully compliant GLYDE-II representation of the “ α -D-Manp” *molecule* by using the model reference approach:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccr.c.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [  
  <!ENTITY mDBget "http://monosaccharidedb.org/GLYDE-II.jsp?G">  
>  
<GlydeII>  
  <molecule subtype="monosaccharide" id="a-dman-hex-1:5" name="a-D-Manp">  
    <model_ref>  
      http://www.monosaccharideDB.org/GLYDE-II.jsp?r=a-dman-hex-1:5  
    </model_ref>  
  </molecule>  
</GlydeII>
```

Note that the URL in the *model_ref* is slightly different than the URL that is used to get this GLYDE-II representation. When the parameter “G” is requested via the HTTP GET method, the service returns a fully compliant GLYDE-II representation of the structure. When the parameter “r” is requested via the HTTP GET method, the service returns information about the structure that can be in any format. The difference is based on the fact that the *ref attribute* of a *residue* requires a pointer to a fully compliant GLYDE-II representation, while the *model_ref element* can serve to access a structural representation in some arbitrary format.

As monosaccharideDB matures, it will support fully atomistic representations of **base-types** and **substituents**. Then, invocation of

```
ref=http://www.monosaccharideDB.org/GLYDE-II.jsp?G=a-dman-hex-1:5
```

will provide a fully atomistic GLYDE-II representation of the “ α -D-Manp” structure. This will allow direct parsing of the GLYDE-II representation of the glycan to make a complete atomistic model.

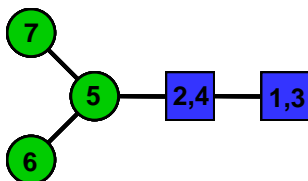
3.3. Implementation rules for the direction of links. The GLYDE-II standard includes a partitioning of links. (See Section 1.1.) That is, a *residue_link* between two *residues* (e.g., specifying linkage of residues “A” and “B”) contains an *atom_link* with finer granularity (e.g., specifying linkage of “C1” of residue “A” and “O4” of residue “B”). This is implemented using the *from* and *to attributes* of each link to specify an ordered pair. Thus, the object specified by the *from attribute* of a child link corresponds to a part of the object specified by the *from attribute* of the parent *link*. A similar relationship applies to the *to attribute*. In order to maintain consistency in the specification of structures, rules are required to constrain the ordering of this pair (i.e., which of two parts is specified by the *from attribute* and which is specified by the *to attribute*). In general, one should be able to trace at least one pathway between any two parts of the structure without reversal of link direction. (However, it is recognized that one could find structures for which no set of specific rules can be specified that maintain this general rule.)

Rule 6. Direction of *atom_links*. The *atom_links* between two atoms in a monosaccharide all point toward the anomeric carbon, (e.g., from C2 to C1, from O2 to C2, from O1 to C1, etc). The *atom_link* between the anomeric carbon and the ring oxygen is an exception. That is, the *atom_links* around the ring all point in the same direction, so the *atom_link* connecting C1 and O5 in glucopyranose is from C1 to O5. The *atom_links* between two atoms in an amino acid all point toward C1. The *link* between an atom in one residue and an atom in another residue follow from **Rule 7.**

Rule 7. Direction of *residue_links*. The *residue_links* between glycosyl residues point toward the reducing end of a glycan. The *residue_links* between amino acid residues point toward the carboxy terminus. The *residue_links* between a monosaccharide residue and an amino acid (as in a glycopeptide) point toward the amino acid if the link is via a glycosidic bond and toward the glycosyl residue if the link is via an ester involving the carboxyl of the amino acid and a hydroxyl of the monosaccharide residue. That is, ester and amide *residue_links* generally point from the acid-containing *residue* to the alcohol- or amine-containing *residue*, respectively.

Rule 8. All links must connect two parts that are of the same *type*. The *from* and *attributes* of the child link point to structures that have a granularity that is exactly one level finer than those in the parent link. That is, an *atom_link* is always the child of a *residue_link*, which is always the child of a *moiety_link*.

These rules are illustrated in the following example.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccr.c.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "http://www.monosaccharideDB.org/GLYDE-II.jsp?G">
]>
<GlydeII>

  <molecule subtype="glycan" id="M3N2" name="pentaglycoside">
    <residue subtype="substituent" partid="1" ref="&mDBget;=n-acetyl"/>
    <residue subtype="substituent" partid="2" ref="&mDBget;=n-acetyl"/>
    <residue subtype="base_type" partid="3" ref="&mDBget;=b-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="4" ref="&mDBget;=b-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="5" ref="&mDBget;=b-dman-hex-1:5"/>
    <residue subtype="base_type" partid="6" ref="&mDBget;=a-dman-hex-1:5"/>
    <residue subtype="base_type" partid="7" ref="&mDBget;=a-dman-hex-1:5"/>

    <residue_link from="1" to="3">
      <atom_link from="N1" to="C2" from_replaces="O2" bond_order="1"/>
    </residue_link>
    <residue_link from="2" to="4">
      <atom_link from="N1" to="C2" from_replaces="O2" bond_order="1"/>
    </residue_link>
    <residue_link from="4" to="3">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="5" to="4">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="6" to="5">
      <atom_link from="C1" to="O3" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="7" to="6">
      <atom_link from="C1" to="O6" to_replaces="O1" bond_order="1"/>
    </residue_link>
  </molecule>
</GlydeII>
```

Application Note. In this example, evaluation of the *entity* “&mDBget;” is followed by an equal sign to instruct monosaccharideDB to use the HTTP GET method (<http://www.w3.org/2001/tag/doc/whenToUseGet.html>) to identify the record to use when dynamically generating the GLYDE-II representations of an archetypal *molecule*. However, it is possible that some *molecules* (e.g., oligosaccharides that are used as archetypes for the carbohydrate *moieties* of glyconjugates) or *free_atoms* (that are used as archetypes for *bound_atoms*) will be found in static XML files. In other cases, the archetypal *molecules* or *free_atoms* may reside in the same XML file as the parts that reference them. In any case, GLYDE-II parser must know where to look for the referenced *molecule* or *free_atom* within the GLYDE-II code. This is accomplished by dividing the

string specified by the *ref attribute* into two substrings separated by a delimiter. For static representations “#” is used as the delimiter. For dynamically generated representations, “=” is used as the delimiter. The substring following the delimiter is the *id attribute* of the *molecule* or *free_atom* that is being references as an archetype for the part.

In summary, if the structure is found in the same XML file as the part that references it, “#” is the first character in the string specified by the *ref attribute* of the *part*, as follows.

```
<moiety subtype="glycan" partid="moiety_1" ref="#M3N2" />
```

If the structure is found in a different XML file, the “#” delimiter is used as follows.

```
<moiety subtype="glycan" partid="moiety_1"
  ref="http://glycomics.ccr.c.uga.edu/GLYDE-II/M3N2.xml#M3N2" />
```

If the structure is found in a dynamically generated GLYDE-II representation, the “=” delimiter is used as follows.

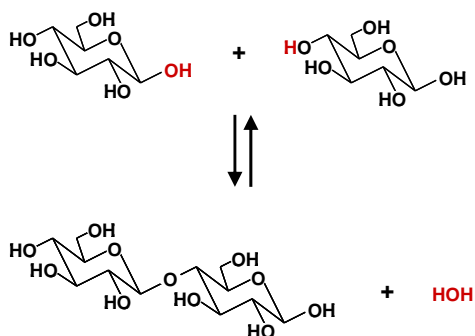
```
<residue subtype="base_type" partid="3" ref="&MDBget;=b-dglc-hex-1:5" />
```

It is important to note that when the archetype *molecule* with `id=" b-dglc-hex-1:5"`, for example, is used as a part in a larger *molecule*, it is recast as a *residue*. That is, the original *molecule* is an independent entity, the monosaccharide “ β -D-Glcp”. However, within the context of a larger *molecule*, this part is not independent (it is glycosidically linked), and therefore it is no longer an independent monosaccharide *molecule*, but a *residue* within a glycan *molecule*.

Each of the links in this pentaglycoside is specified at two levels of granularity. The coarsest level (a *residue_link*) just specifies that two residues are connected to each other. Each of the *residue_links* has a child *atom_link* that embodies a finer granularity. For example, the following code snippet

```
<residue_link from="4" to="3">
  <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
</residue_link>
```

specifies that there is a *link* from the residue with `partid="4"` to the residue with `partid="3"`, and that this link is actually a covalent bond from “C1” of the residue with `partid="4"` to “O4” of the residue with `partid="3"`. Furthermore, the *attribute* `to_replaces="O1"` specifies that the atom to which the bond extends (i.e., “O4” of the residue with `partid="3"`) **replaces** “O1” of the residue **from** which the bond extends (i.e, the one with `partid="4"`). This reflects the chemistry of glycosidic-bond formation, which transforms the monosaccharide *molecule* into a *residue*. As illustrated in the Figure below, bond formation is a dehydrating condensation that results in the liberation of a molecule of water. In the example above, the oxygen of this liberated water molecule is derived from “O1” of the *residue* with `partid="4"`, and this oxygen is replaced by “O4” of the *residue* with `partid="3"` when the bond is made. (The two hydrogen atoms of the liberated water are not specified in the definition of the residues, as they are exchangeable hydrogens that are only transiently attached to oxygen atoms.)



Glycosidic bond formation is a dehydrating condensation

One might think of designing a carbohydrate *residue* archetype that does not contain “O1”, making it unnecessary to specify that this oxygen is replaced upon glycosidic bond formation. However, this approach would have several key disadvantages. (1) This presumes that all bond formations in which the residue participates involve the loss of O1, which may not be true. (2) Failure to include O1 in the definition of the carbohydrate residue makes it difficult, if not impossible, to specify the stereochemistry (anomeric configuration - α or β) of C1 using a formalism (such as InChI) that specifies the parity of the anomeric carbon in the archetype. (3) Leaving out O1 in the definition of the carbohydrate *residue* forces one to add an oxygen atom (O1) in order to use this structure as an archetype to represent a reducing *residue*. In summary, it is critical to have the ability to specify that bond formation by chemical condensation results in the replacement of a specific atom(s) in the newly linked residue(s). This is also true when specifying links between amino acids in peptides, where bond formation results in the loss of one of the carboxylate oxygens of the amino-acid *residue* from which the bond extends. Finally, this approach is consistent with atom replacement formalism of Glyco-CT (www.eurocarbdb.org/recommendations/encoding).

When a single residue (such as a β -D-GlcpNAc that is linked to a serine residue in a cytoplasmic protein), **Rule 8** requires that the β -D-GlcpNAc must be “wrapped” in the *moiety* that contains it. The *molecule* used as the archetype this *moiety* must consist of a glycan *moiety* that itself consists of a single β -D-GlcpNAc *residue*. Directly linking the β -D-GlcpNAc *residue* to the peptide *moiety* would break **Rule 8**, making it difficult to implement code to parse the connection between objects. In other words, a three-level link is required in this case: (i) from the monosaccharide *moiety* to the peptide *moiety*; (ii) from the β -D-GlcpNAc *residue* to the serine *residue*; and (iii) from C1 (a *bound_atom*) of the β -D-GlcpNAc residue to O3 (a *bound_atom*) of the serine residue. This is illustrated in the code below.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccr.c.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "http://www.monosaccharideDB.org/GLYDE-II.jsp?G">
]>
<GlydeII>
  <molecule subtype="monosaccharide" id="monosaccharide_1" name="GlcNac">
    <residue subtype="substituent" partid="residue_1" ref="&mDBget;n-acetyl"/>
    <residue subtype="base_type" partid="residue_2" ref="&mDBget;b-dglc-hex-1:5"/>
    <residue_link from="residue_1" to="residue_2">
      <atom_link from="N1" to="C2" from_replaces="O2" bond_order="1"/>
    </residue_link>
  </molecule>
```

```

<molecule subtype="peptide" id="peptide_1" name="some peptide">
  <residue subtype="amino_acid" partid="ser_5" ref="URI_for_asparagine"/>
</molecule>

<molecule id="gp1" name="GlcNAcylated protein">
  <moiety subtype="glycan" partid="moiety_3" ref="#monosaccharide_1"/>
  <moiety subtype="peptide" partid="moiety_4" ref="#peptide_1"/>
  <moiety_link from="moiety_3" to="moiety_4">
    <residue_link from="residue_2" to="ser_5">
      <atom_link from="C1" to="O3" to_replaces="O1"/>
    </residue_link>
  </moiety_link>
</molecule>
</GlydeII>

```

3.4. Atypical structures. GLYDE-II is capable of representing atypical structures, such as cyclic glycans and large glycans with repeating structures.

3.4.1. Macrocytic structures. The connection-table format of GLYDE-II makes representation of cyclic structures trivial, as illustrated in the following example. Simply adding a *residue_link* from residue “1” to residue “6” cyclizes the molecule.

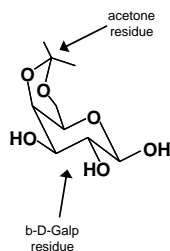
```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.cccr.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "http://www.monosaccharideDB.org/GLYDE-II.jsp?G">
]>
<GlydeII>
  <molecule subtype="glycan" id="a-cd" name="alpha-cyclodextrin">
    <residue subtype="base_type" partid="1" ref="&mDBget;a-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="2" ref="&mDBget;a-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="3" ref="&mDBget;a-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="4" ref="&mDBget;a-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="5" ref="&mDBget;a-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="6" ref="&mDBget;a-dglc-hex-1:5"/>

    <residue_link from="2" to="1">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="3" to="2">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="4" to="3">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="5" to="4">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="6" to="5">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <!-- This link closes the macrocycle -->
    <residue_link from="1" to="6">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
  </molecule>
</GlydeII>

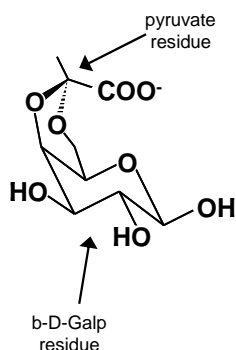
```

3.4.2. Other cyclic structures. Another type of cyclic structure involves the connection of two residues by two distinct bonds, such as the 4,6-acetonide illustrated below.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccr.c.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "ref=http://www.monosaccharideDB.org/GLYDE-II.jsp?G">
]>
<GlydeII>
  <molecule id="molecule_1" name="acetonide">
    <residue subtype="base_type" partid="1" ref="&mDBget;b-dgal-hex-1:5"/>
    <residue subtype="substituent" partid="2" ref="&mDBget;=acetone" name="acetone"/>
    <residue_link from="2" to="1">
      <atom_link from="C2" to="O4" to_replaces="O21" bond_order="1"/>
      <atom_link from="C2" to="O6" to_replaces="O22" bond_order="1"/>
    </residue_link>
  </molecule>
</GlydeII>
```

Note that the *residues* in this *molecule* include one **base-type** and one **substituent**. The **substituent** is identified by reference to the structure “acetone”, which corresponds to a hydrated acetone molecule ($C_3H_6O + H_2O = C_3H_8O_2$). The link between the *residues* embodies **two** covalent bonds. In this case, “O4” and “O6” of the “b-D-Galp” residue replace “O21” and “O22” of the hydrated acetone residue, respectively. The fact that two different oxygens are attached to C2 of the acetonide residue makes it necessary to use a *molecule* that also has two oxygens attached to C2 as an archetype for this *residue*. Therefore, the archetype molecule is actually “acetone_hydrate”. This acetonide is achiral (as is acetone hydrate), so “O21” and “O22” are equivalent. However, this is not the case for all acetals of this general type. For example, pyruvate can be linked to a carbohydrate residue via a chiral ketal linkage, as illustrated next.



In this case, C2 of the pyruvate is chiral, but this is not the case in free pyruvate (or even in hydrated pyruvate, where there are two different prochiral oxygen atoms, O21 and O22, at C2.) The GLYDE-II representation of this molecule is shown next.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccr.c.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "ref=http://www.monosaccharideDB.org/GLYDE-II.jsp?G">
]>
```

```

<GlydeII>
  <molecule id="molecule_1" name="pyruvate ketal">
    <residue subtype="base_type" partid="1" ref="&mDBget;b-dgal-hex-1:5"/>
    <residue subtype="substituent" partid="2" ref="&mDBget;=pyruvate" name="pyruvate-hydrate"/>
    <residue_link from="2" to="1">
      <atom_link from="C2" to="O4" to_replaces="O21" bond_order="1"/>
      <atom_link from="C2" to="O6" to_replaces="O22" bond_order="1"/>
    </residue_link>
  </molecule >
</GlydeII>

```

The archetype *molecule* for the pyruvate residue is defined as a hydrated pyruvate (identified in monosaccharideDB as the “pyruvate” substituent). Two non-equivalent oxygen atoms are attached to the prochiral C2 of the archetype *structure* for this substituent. Unless these two oxygens can be distinguished, the pyruvate structure is achiral. However, the two oxygen atoms can be distinguished, as the *atom_links* in the above example specify that O4 of the β -D-Galp residue replaces O21 of the pyruvate and O6 of the β -D-Galp residue replaces O22 of the pyruvate, so the chirality of the pyruvate is fully specified.

According to **Rule 3**, “The hydrogen atoms of a methylene group may be distinguished by an additional number, e.g. H61 and H62 where the lower number is selected for the pro-S atom.” This is applied to the prochiral oxygen atoms of hydrated pyruvate. That is, O21 is the pro-S oxygen because replacing the hydroxyl hydrogen on O21 with an atom having a higher atomic number would result in the S-configuration at C2. (See, for example, <http://en.wikipedia.org/wiki/Prochiral>.) Similarly, O22 is the pro-R oxygen because replacement of the hydroxyl hydrogen on O22 would result in the R-configuration at C2. Thus, these two atoms can be distinguished, and the above GLYDE-II representation, along with the extension of **Rule 3** to systematically name oxygen atoms of the hydrated pyruvate archetype *molecule*, unambiguously defines the stereochemistry of this pyruvate ketal.

3.4.3. Repeating structures. As illustrated in the next example (γ -cyclodextrin), a *residue* can constitute a repeat unit. More generally, a repeating block might contain substructures that are internally linked and the block itself can be linked to other structures. Within the repeat block, one must distinguish the “head-to-tail” link that connects the tandemly-arranged copies of the block to each other from any links that are internal to the block and from links to structures that are external to the block. The *repeat_block element*, which is designed to implement these requirements, contains *repeat_part elements*, each of which embodies a reference to a component of the *repeat_block*. These components are defined (once) outside of the *repeat_block*. The links between these components are also defined outside the *repeat_block*. However, as copies of the *repeat_block* are tandemly arranged, the link between each of these tandem repeats must be specified. The head-to-tail link that connects tandem copies of the *repeat_block* is specified within the *repeat_block*. The links connecting the ends of the *repeat_block* to other components of the molecule are specified outside the *repeat_block*. The *repeat_block* has an attribute called *repeat_number*, which indicates how many times the *repeat_block* is tandemly repeated. This is illustrated in the following code.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccruc.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "http://monosaccharidedb.org/GLYDE-II.jsp?G">
]>
<GlydeII>
  <molecule id="molecule_1" name="gamma-cyclodextrin">
    <residue subtype="base_type" partid="residue_1" ref="&mDBget;=a-dglc-hex-1:5"/>

```

```

<residue subtype="base_type" partid="residue_2" ref="&mDBget;a-dglc-hex-1:5"/>
<residue subtype="base_type" partid="residue_3" ref="&mDBget;a-dglc-hex-1:5"/>
<residue_link from="residue_2" to="residue_1">
  <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
</residue_link>
<residue_link from="3" to="2">
  <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
</residue_link>
<residue_link from="residue_1" to="residue_3">
  <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
</residue_link>
<repeat_block repeat_number="6">
  <repeat_part ref="residue_2"/>
  <!-- This link is tandemly repeated (i.e., from residue_2 to a copy of residue_2) -->
  <residue_link from="residue_2" to="residue_2">
    <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
  </residue_link>
</repeat_block>
</molecule>
</GlydeII>

```

A slightly more complex example is hyaluronan (<http://www.chem.qmul.ac.uk/iupac/2carb/39.html#398>), in which the repeating block structure is the disaccharide β -D-GlcA-(1-3)- β -D-GlcpNAc. The disaccharides are linked together via β -(1-4) linkages from the β -D-GlcpNAc residue of one repeat unit to the β -D-GlcpA of the next. This is illustrated in the following code.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccr.c.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "http://www.monosaccharideDB.org/GLYDE-II.jsp?G">
]>
<GlydeII>
  <molecule id="molecule_1" name="hyaluronan">
    <residue subtype="substituent" partid="residue_1" ref="&mDBget;=nac"/>
    <residue subtype="base_type" partid="residue_2" ref="&mDBget;=b-dglc-hex-1:5"/>
    <residue subtype="base_type" partid="residue_3"
      ref="&mDBget;=b-dglc-hex-1:5,6:a" name="b-D-GlcpA"/>
    <!-- connect the base-type and substituent of the GlcNAc -->
    <residue_link from="residue_1" to="residue_2">
      <atom_link from="N1" to="C2" from_replaces="O2" bond_order="1"/>
    </residue_link>
    <!-- specify that the GlcA is linked to the GlcNAc -->
    <residue_link from="residue_3" to="residue_2">
      <!-- via a 1-3 bond -->
      <atom_link from="C1" to="O3" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <!-- The repeat_block specifies which residues are repeated
      and the link between repeat_blocks -->
    <repeat_block repeat_number="n">
      <repeat_part ref="residue_1"/>
      <repeat_part ref="residue_2"/>
      <repeat_part ref="residue_3"/>
      <!-- This tandem link is head-to-tail for the end residues of the repeat_block -->
      <!-- Specify that GlcNAc is linked to GlcA to make the tandem repeat -->
      <residue_link from="residue_2" to="residue_3">
        <!-- via a 1-4 bond -->
        <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
      </residue_link>
    </repeat_block>
  </molecule>
</GlydeII>

```

The hyaluronan polysaccharide consists of tandemly repeated copies of a disaccharide *repeat_unit*, which is specified by listing its *residue* parts and the tandemly repeated *residue_link* that connects copies of the *repeat_unit*. The disaccharide *repeat_unit* contains three internal

parts (residues 1, 2 and 3) and two internal *residue_links*. The link between adjacent copies of the disaccharide is explicitly specified as a link from “C1” of the β -D-GlcpNAc residue in one repeat unit to “O4” of the β -D-GlcpA in the next. Thus, no relationship between the internal and tandem linkages is assumed and the internal and tandem linkages are fully specified.

3.4.4. Incomplete or statistically known structures. An example of statistically known structural information is chondroitin sulfate ([http://en.wikipedia.org/wiki/Chondroitin sulphate](http://en.wikipedia.org/wiki/Chondroitin_sulphate)), which is related to hyaluronan. Chondroitin sulfate is composed of partially sulfated β -D-GalNAc and β -D-GlcA residues. The sulfate groups are specified as *parts* of *type*="residue" (by reference to the H₂SO₄ structure, which corresponds to the “so₄” substituent in Glyco-CT.) The partial presence of sulfate substituents in the chondroitin sulfate repeat unit is specified using an *attribute* of the *link element* called *stat*, as illustrated in the following code.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE GlydeII SYSTEM "http://glycomics.ccruc.uga.edu/GLYDE-II/GLYDE-II_v0.7.DTD" [
  <!ENTITY mDBget "http://www.monosaccharidedb.org/GLYDE-II.jsp?G">
]>
<GlydeII>
  <molecule id="molecule_1" name="chondroitin sulfate">
    <!-- The next parts make up the repeating disaccharide -->
    <residue subtype="substituent" partid="1" ref="&mDBget;n-acetyl"/>
    <residue subtype="base_type" partid="2" ref="&mDBget;b-dgal-hex-1:5"/>
    <residue subtype="base_type" partid="3"
      ref="&mDBget;b-dglc-hex-1:5,6:a" name="b-D-GlcpA"/>
    <residue subtype="substituent" partid="4"
      ref="&mDBget;so4" name="sulfate"/>
    <residue subtype="substituent" partid="5"
      ref="&mDBget;so4" name="sulfate"/>

    <!-- The next parts make up the non-repeating core tetrasaccharide -->
    <residue subtype="base_type" partid="6"
      ref="&mDBget;b-dxyl-pen-1:5" name="b-D-Xylp"/>
    <residue subtype="base_type" partid="7"
      ref="&mDBget;b-dgal-hex-1:5" name="b-D-Galp"/>
    <residue subtype="base_type" partid="8"
      ref="&mDBget;b-dgal-hex-1:5" name="b-D-Galp"/>
    <residue subtype="base_type" partid="9"
      ref="&mDBget;b-dglc-hex-1:5,6:a" name="b-D-GlcpA"/>

    <!-- connect the base-type and substituent of the GalNAc -->
    <residue_link from="1" to="2">
      <atom_link from="N1" to="C2" from_replaces="O2" bond_order="1"/>
    </residue_link>
    <!-- specify that, within the disaccharide, the GlcA is linked to the GalNAc -->
    <residue_link from="3" to="2">
      <!-- via a 1-3 bond -->
      <atom_link from="C1" to="O3" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <!-- 90% of the time, GalNAc's base-type (part 2) has a sulfate (part 4) at O6 -->
    <residue_link from="4" to="2" stat="0.9">
      <atom_link from="S1" to="O6" to_replaces="O11" bond_order="1"/>
    </residue_link>
    <!-- 5% of the time, GalNAc's base-type (part 2) has a sulfate (part 5) at O4 -->
    <residue_link from="5" to="2" stat="0.05">
      <atom_link from="S1" to="O4" to_replaces="O11" bond_order="1"/>
    </residue_link>

    <!-- The next links connect the residues within the core tetrasaccharide -->
    <residue_link from="7" to="6">
      <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
    </residue_link>
    <residue_link from="8" to="7">
      <atom_link from="C1" to="O3" to_replaces="O1" bond_order="1"/>
    </residue_link>
```

```

<residue_link from="9" to="8">
  <atom_link from="C1" to="O3" to_replaces="O1" bond_order="1"/>
</residue_link>
<!-- The GalNAc in the disaccharide repeat is linked to GlcA
      in the core tetrasaccharide -->
<residue_link from="2" to="9">
  <!-- via a 1-4 bond -->
  <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
</residue_link>

<!-- The repeating nature of the disaccharide is specified next -->
<repeat_block repeat_number="n">
  <repeat_part ref="1"/>
  <repeat_part ref="2"/>
  <repeat_part ref="3"/>
  <repeat_part ref="4"/>
  <repeat_part ref="5"/>
  <!-- This tandem link is head-to-tail for the end residues of the disaccharide -->
  <!-- The tandem repeat is made by linking the GalNAc to GlcA -->
  <residue_link from="2" to="3">
    <!-- via a 1-4 bond -->
    <atom_link from="C1" to="O4" to_replaces="O1" bond_order="1"/>
  </residue_link>
</repeat_block>
</molecule>
</GlydeII>

```

This code illustrates several different features of GLYDE-II. Note that two different sulfate residues (“residue_4” and “residue_5”) are defined, and the *stat attribute* of their *links* to the GalpNAc residue specify that the sulfate at O6 is present 90% of the time and the sulfate at O4 is present 5% of the time. (Specifying two different sulfate residues distinguishes this situation from the mutually exclusive case, wherein a single sulfate is present, but it may be in one of two different locations.) This formalism is closely related to that used by Glyco-CT (www.eurocarbdb.org/recommendations/encoding) for specifying statistically defined structures of this kind.

Chondroitin sulfate also has a “core tetrasaccharide composed of a β -D-GlcA, two β -D-Gal, and one β -D-Xyl residue. The tandemly repeating, partially sulfated disaccharide is linked to the core tetrasaccharide β -D-GlcA.

Sometimes a part is known to be attached to another specific part, but its attachment site is unknown. In this case, the link with coarser granularity is fully specified, but the child link (finer granularity) is specified as a choice. The syntax is similar to that used in Glyco-CT (www.eurocarbdb.org/recommendations/encoding) for this situation, using the vertical bar “|”, signifying “or”. This is illustrated in the code snippet listed below.

```

<residue_link from="residue_6" to="residue_4">
  <atom_link from="C1" to="O6|O2" to_replaces="O1" bond_order="1"/>
</residue_link>

```

This code specifies that the linkage site is partially known (i.e., O6 or O2, but not O3). In general, what is implicitly known should be specified. If it is known, for example, that the attachment is by an O-glycosidic linkage *to* an aldohexopyranose, then one should specify `to="O2|O3|O4|O6"` (i.e., it cannot be to O5 of the pyranose.) As in Glyco-CT (www.eurocarbdb.org/recommendations/encoding), the “?” is not allowed for the values of the *to* and *from attributes* of a *link*.

There exists another type of uncertainty, where a *residue* or group of *residues* are known to be present, but their precise location is unknown. This type of uncertainty is indicated by the “PRO section” of Glyco-CT (www.eurocarbdb.org/recommendations/encoding). The proposed GLYDE-II formalism is similar but uses an *element* called *combination*. Consider, for example, a N-linked glycan that contains a single terminal β -D-GlcpNAc *residue*, which may be attached at any one of the four following sites: (i) O6 of the α -D-Manp on the 6-arm, (ii) O2 of the α -D-Manp on the 6-linked arm, (iii) O4 of the α -D-Manp on the 3-linked arm, and (iv) O2 of the α -D-Manp on the 3-linked arm. Thus, both the residue-level site and the atomic-level site are unknown, but not completely unknown. That is, it is known that the β -D-GlcpNAc *residue* (e.g, with `partid="6"`) is attached to O6 or O2 of one α -D-Manp residue (e.g, with `partid="4"`) or O4 or O2 of another α -D-Manp residue (e.g, with `partid="5"`). This uncertainty is represented by the following code snippet.

```
<combination>
  <residue_link from="6" to="4">
    <atom_link from="C1" to="O6|O2" to_replaces="O1" bond_order="1"/>
  </residue_link>
  <residue_link from="6" to="5">
    <atom_link from="C1" to="O4|O2" to_replaces="O1" bond_order="1"/>
  </residue_link>
</combination>
```

The *element* called *combination* is a collection of different links that represent all of the possible partially known attachment sites. In this example, the *combination* specifies the uncertainty of locating a single part (i.e., the β -D-GlcpNAc *residue* with `partid="6"`) that can be attached by any of the linkages enclosed in the tag. The *combination* must contain two different *residue_links*, as the *residue_link* to `residue_4` is via O6 or O2, while the *residue_link* to `residue_5` is via O4 or O2. Since this *combination* has only one *part*, only one of the possible linkage sites can be occupied. As with the “PRO section” of the Glyco-CT formalism (www.eurocarbdb.org/recommendations/encoding), a combination can only be used to represent homogenic structures (where only one structure exists, but its precise structure is unknown). Mixtures should be represented by a collection of crisp representations.

Alternatively, two β -D-GlcpNAc residues (one with `partid="6"` and one with `partid="7"`) may be present at the same set of four possible sites described above, but their precise locations are not known. Obviously, they cannot both be present at the same location, and this is where the *combination element* is most useful. This is illustrated in the code snippet below.

```
<!-- this specifies the possible sites for residue 6 and residue 7 -->
<!-- Only 1 combination is true -->
<combination>
  <residue_link from="6|7" to="4">
    <atom_link from="C1" to="O6|O2" to_replaces="O1" bond_order="1"/>
  </residue_link>
  <residue_link from="6|7" to="5">
    <atom_link from="C1" to="O4|O2" to_replaces="O1" bond_order="1"/>
  </residue_link>
</combination>
```

This is the case that is most similar to the case used to describe the “PRO section” of Glyco-CT (www.eurocarbdb.org/recommendations/encoding). A parsing algorithm could easily enumerate all of the possible combinations implicit in this representation, excluding those where the same site is occupied by two different structures.

This formalism would also be very useful in describing the structure of fragment ions observed in MS/MS spectra. For example, a fully methylated oligosaccharide ion is fragmented and it is known that a particular fragment ion contains an α -D-Manp residue that has a β -D-GlcpNAc *residue* attached at O2, O3, O4, or O6 and methyl substituents (which are formally *residues*) attached to the oxygens that do not bear the β -D-GlcpNAc residue. That is, the attachment site of the β -D-GlcpNAc residue in the original structure of the parent ion is not known, but the attachment sites of the β -D-GlcpNAc and methyl residues are mutually exclusive. Then, a *combination* whose parts are 3 methyl residues and the β -D-GlcpNAc residue can be specified, and this combination would contain the possible (mutually exclusive) links to the α -D-Manp residue. The code would look like the following:

```
<residue subtype="substituent" partid="3" ref="&mDBget;=nac"/>
<residue subtype="base_type" id="4" ref="&mDBget;=b-dman-hex-1:5"/>
<residue subtype="base_type" id="5" ref="&mDBget;=b-dglc-hex-1:5"/>
<residue subtype="substituent" id="6" ref="&mDBget;=methyl"/>
<residue subtype="substituent" id="7" ref="&mDBget;=methyl "/>
<residue subtype="substituent" id="8" ref="&mDBget;=methyl "/>
.....
<combination parts="5|6|7|8">
  <!-- this specifies the possible sites for residues 5-8 -->
  <!-- only one combination is true -->
  <link from="5|6|7|8" to="4">
    <link from="C1" to="O2|O3|O4|O6" to_replaces="O1" bond_order="1"/>
  </link>
</combination>
```

In this case, the methyl groups are based on a methanol *molecule* and O1 of the methanol is replaced by O2 or O3 or O4 or O6 of the α -D-Manp residue when the *links* are instantiated. One disadvantage of this approach is that, for combinations that include chemically identical structures, several degenerate combinations of mutually exclusive links are possible. That is, the combination in which (methyl) residue 6 is at O2 and (methyl) residue 7 is at O3 is formally distinct from the converse combination in which (methyl) residue 6 is at O3 and (methyl) residue 7 is at O2, even though these two combinations actually have the same chemical structure. This degeneracy is also a characteristic for the above example with two different β -D-GlcpNAc residues. It also appears to be a characteristic of Glyco-CT (www.eurocarbdb.org/recommendations/encoding) when ambiguity is represented using the "PRO section".

One might imagine that collecting these links within a *combination element* is unnecessary. However, it is possible that a single molecule can have more than one set of structures with mutually exclusive attachment sites, and these sets must be logically separated by specifying more than one *combination*.

Literature Cited

Casati, Roberto. Varzi, Achille C. , Parts and Places: The Structures of Spatial Representation. Cambridge, Mass. MIT Press, 1999.