OBOL

Open Bio-Ontology Language

GO Meeting
Stanford
Jan 2004

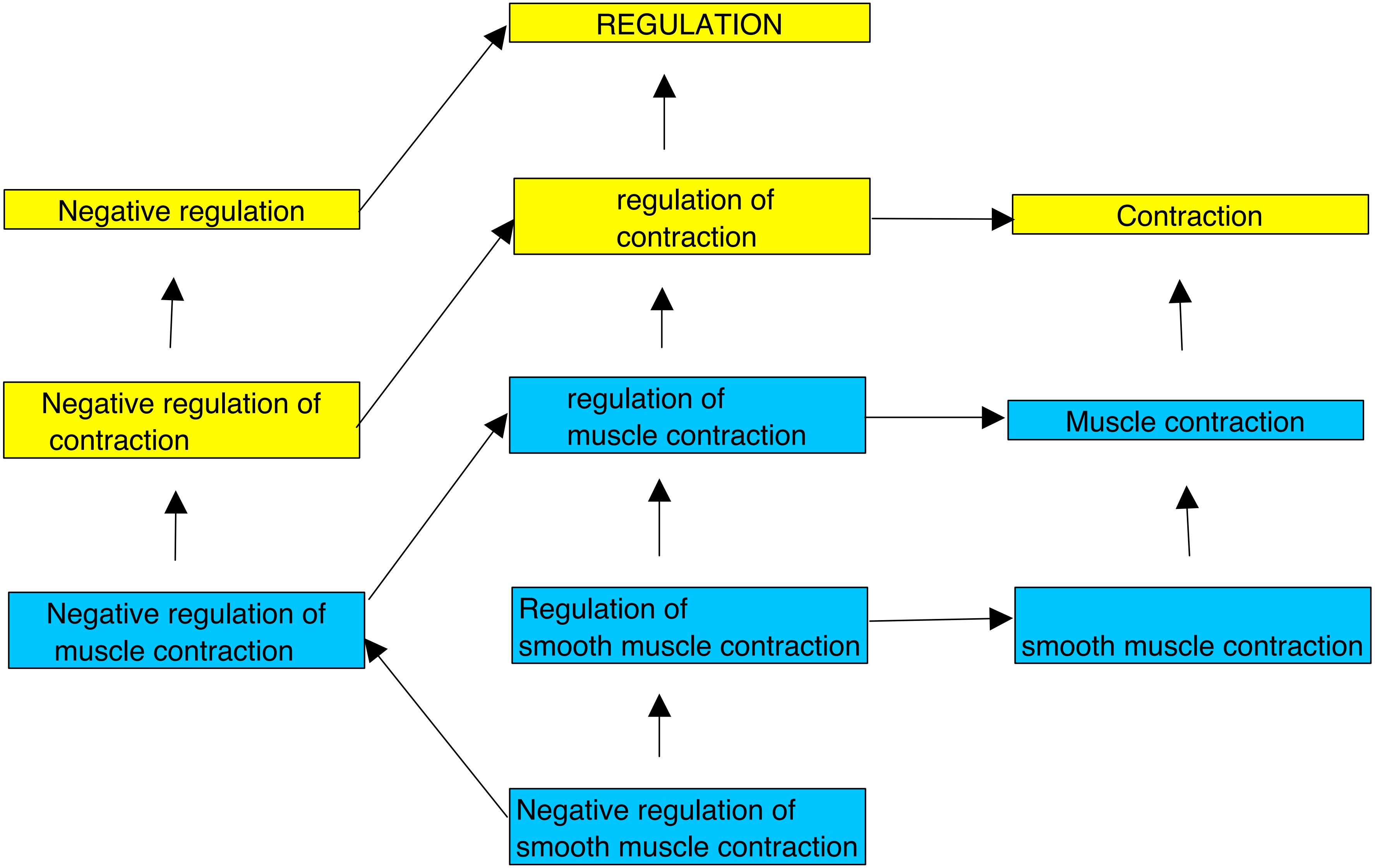
Outline

- Curation and annotation issues that OBOL addresses
- *Grammatical* composition and decomposition
- Making logical class definitions via slots
- Rules for logical inference
- DAG-Edit Plugin
- Results so far
- How it will work

How OBOL will help

- Slot-based annotations (eg interleukin-18 binding)
- Cross-products and composite terms
  - decomposing existing terms
  - new composite terms
- Consistency Checking
- New term creation
- Research (DLs, text-mining)

A Typical Fiendishly Hard Lattice


REGULATION

Negative regulation

regulation of contraction

Contraction

Negative regulation of contraction

regulation of muscle contraction

Muscle contraction

Negative regulation of muscle contraction

Regulation of smooth muscle contraction

smooth muscle contraction

Negative regulation of smooth muscle contraction

An Observation

- GO Term“sentences” often follow consistent implicit rules
  - Ogden et al
- Hidden knowledge is embedded in the text strings
- It should be possible to *decompose/parse* GO terms into **logical definitions** (and also to *compose/generate* new GO terms from new **logical definitions**)
- We can do this with a **Grammar**

Formal Grammars

- A rule system for parsing (decomposing) and generating (composing) sequences of symbols (eg sentences, NA or AA sequences)
- Invented by Chomsky in the 50s
- Used all over computer science (e.g. Compilers)
- Also in bioinformatics – eg RNA structure analysis

Grammars

- Terminal and non-terminal symbols
- We write non-terminals in upper case by convention
- Production Rules   X --> Y, Z
  - Specify means of making LHS by composing RHS
  - (or decomposing RHS into LHS)
  - Recursive

A Simple English Language Grammar

- Sentence --> NounPhrase, VerbPhrase
- VerbPhrase --> Verb, NounPhrase
- NounPhrase --> Det, Noun
- Det --> a | the
- Noun --> cat | mouse | house
- Verb --> scares | hates | eats | kisses
- Eg “the cat scares a mouse”

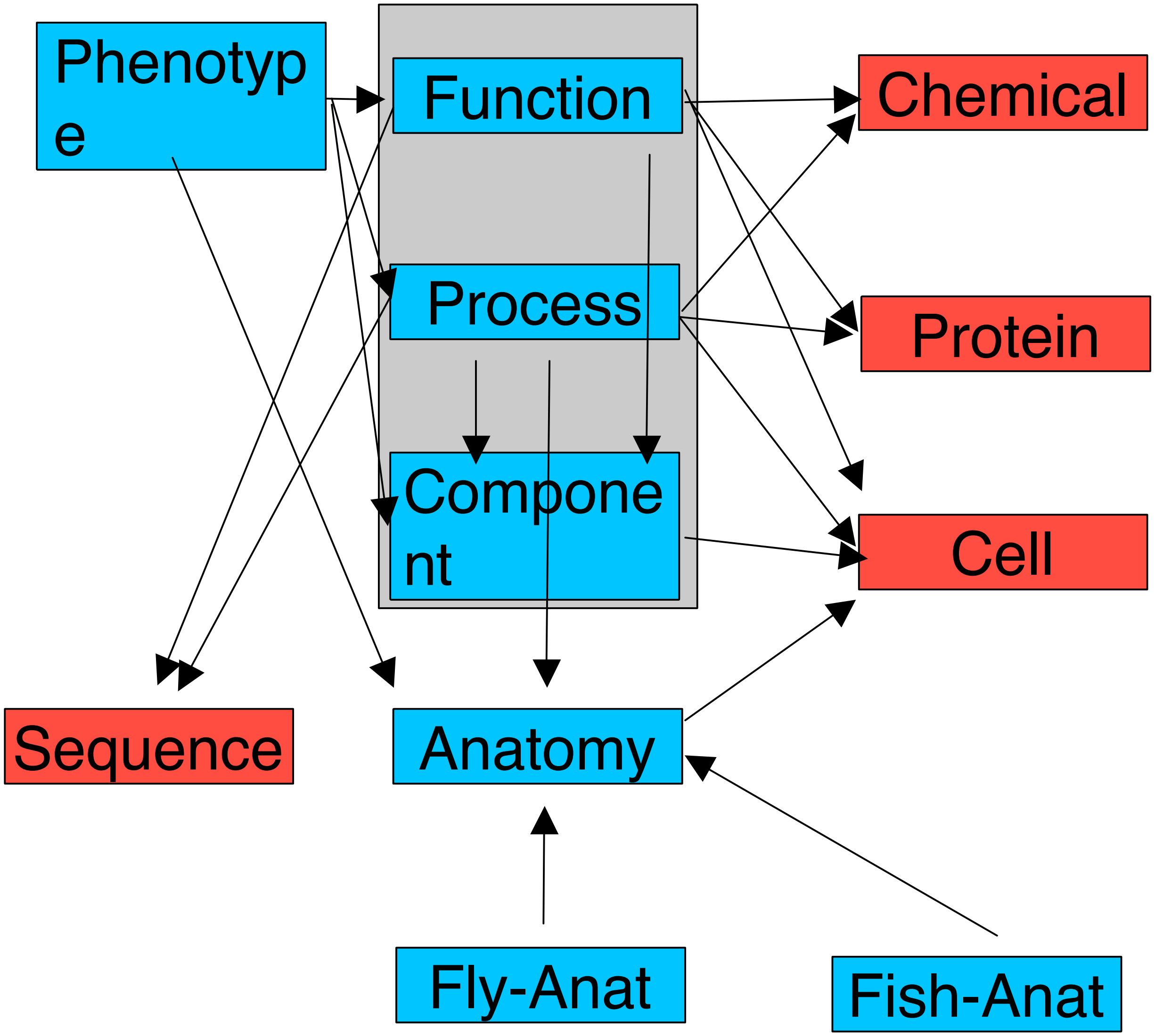
A grammar for OBO terms

- All(?) OBO terms are NOUN-PHRASES
- A NOUN-PHRASE is (recursively) made from
  - a noun
  - an adjective followed by a NOUN-PHRASE
  - a NOUN-PHRASE preceeded by a NOUN-PHRASE acting as adj
  - a NOUN-PHRASE then preposition then NOUN-PHRASE
  - an (optional) NOUN-PHRASE then a relational adjective then a NOUN-PHRASE
- Precedence rules
- Implemented in Prolog (one page of code!)

OBO WordLists

- Partitioned by ontology
- Types:
  - nouns
  - adjectives
  - prepositions
  - relational adjectives (cytosol*ic*, coat*ed*)
- Incomplete information
  - orphan nouns

The OBO Universe (partial)



An example

- negative regulation of smooth muscle contraction

np = adj+np

- negative regulation of smooth muscle contraction
- (negative **regulation**) (smooth **muscle**)

np = np+np

- negative regulation of smooth muscle contraction
- (negative **regulation**) ((smooth **muscle)** **contraction)**
- **(negative regulation) (smooth muscle)**

np = np+p+np

- negative regulation of smooth muscle contraction
- ((negative **regulation**) ((smooth **muscle) contraction))**
- (negative **regulation**) ((smooth **muscle) contraction)**
- **(negative regulation) (smooth muscle)**

alternate parses

- smooth muscle contraction
- (smooth (muscle **contraction**)) <-- wrong!!
- (muscle **contraction**)

Making Logical Class Definitions from Parse Trees

- Tokenize OBO/GO term strings
- Make ParseTree using Prolog Grammar
- Transform tree to class definition using slot definitions
- *reversible*
- *Classdefs can be represented in OBO format or OWL format*

Making classes via slots

- slot: regulates
  - domain (subject): regulation
  - range (object): biological_process
  - grammar-context: preposition(of)
- slot: qualifier
  - domain (subject): regulation
  - range (object): **negative OR positive**
  - grammar-context: adjective

A Recursively Defined Class

- Regulation (**biological_process** class)
  - *type*: negative (**general** class)
  - *regulates*:
    - Contraction (**biological_process** class)
      - *affects_cell*:
        - Muscle (**cell** class)
        - *type*: smooth (**general** class)

COPII-coated vesicle membrane

- membrane (**cellular_component** class)
  - *part_of*
    - vesicle (**cellular_component** class)
      - *has_part*
        - coat (**cellular_component** class)
        - *made_from*
        - COPII (**complex** class)

Reasoning over class definitions

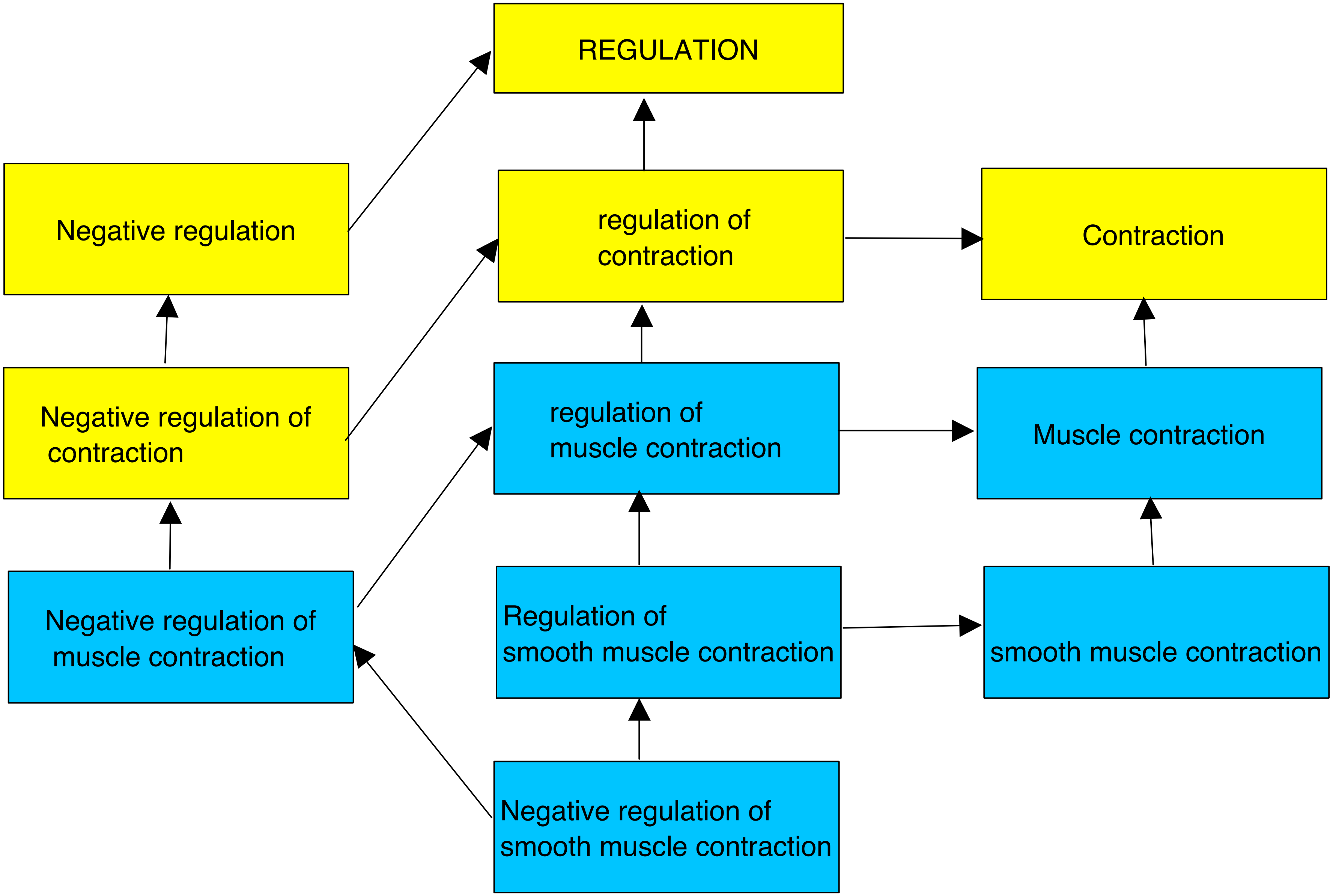
- Use inference over classdefs to
  - place new terms in the correct place in the DAG
  - check for missing relationships in the DAG
  - find inconsistencies within the DAG
  - other kinds of reasoning?
- Method:
  - Inference rules implemented in Prolog

Inferring intermediate terms and ISAs

regulation(process_regulated:R,qual:Q) isa
regulation(process_regulated:R',qual:Q)
<=>
R isa R'

IFF the stem-class is the same AND all the slot-values in the restriction-list are identical EXCEPT for one slot, where the slot-values are linked by an isa, then the classdefs are linked
by an isa

Inference of terms and rels


REGULATION
Negative regulation
regulation of contraction
Contraction
Negative regulation of contraction
regulation of muscle contraction
Muscle contraction
Negative regulation of muscle contraction
Regulation of smooth muscle contraction
smooth muscle contraction
Negative regulation of smooth muscle contraction

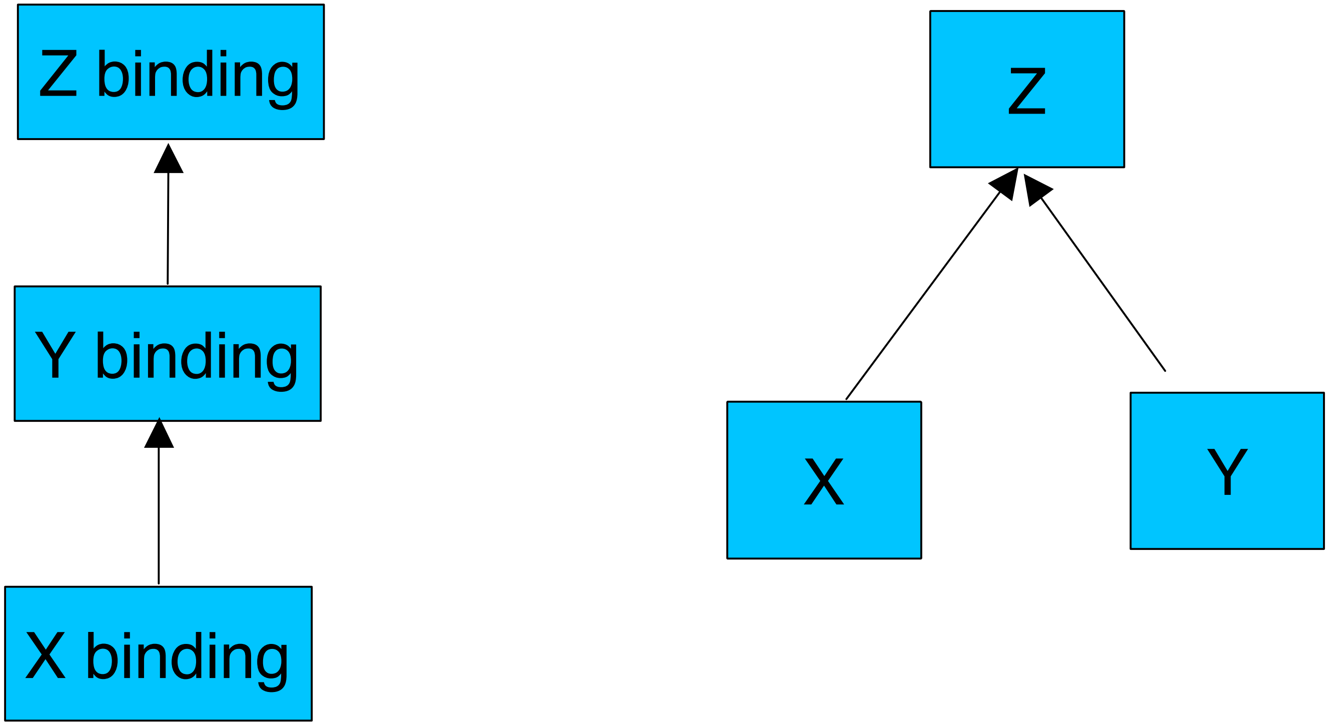
Inference in Prolog is easy

- **% DATABASE OF FACTS**
- isa(carb_binding, binding).
- isa(polysac_binding, carb_binding).
- isa(chitin_binding, polysac_binding)
- isa(cellulose_binding, polysac_binding).

- **% INFERENCE RULES**
- isaT(X,Y):- isa(X, Y).
- isaT(X,Y):-isa(X,Z),
  isaT(Z,Y).

- ?- isaT(chitin_binding, binding).
- **YES**
- ?-isaT(X, polysac_binding).
- **X=carb_binding.**
- **X=chitin_binding.**
- **X=cellulose_binding.**
- ?-isaT(chitin_binding, cellulose_binding).
- **NO**
- **?-isaT(X, Y).** ***% returns all paths***

Detecting inconsistencies



OBOL: Combining Grammars and Reasoning

- Allows complex logical definitions to be maintained as linear text strings
- Maintains facade of narrative approach, whilst implementing a combinatorial approach behind the scenes
- Description Logic Reasoners can be used
  - Racer, FACT – OR use OBOL prolog rules
- Speeds up term creation?
- Flexibility for annotators

Results so far

- Main corpus of logical rules implemented
- wordlists and slot definitions incomplete
- Unique classdefs:
  - function: 2133
  - process: 3240
  - component: 430
- Missing relationships (UNVALIDATED): 130
- Existing relationships that can be infered: ?

Problems to address

- Multiple parses
- Difficulties with nascent biochemical ontology
- No cross-species anatomy ontology (as yet)
- No protein/complex ontology
- Can we use OBOL to help build these other ontologies?

Gradual Introduction of OBOL

- now: periodic generation of **editlists**
- **soon: DAG-Edit plugin**
- **hopefully soon: anatomy and biochem onts**
- **on request: autogeneration of crossproducts**
- **?: maintenance of classdefs in OBO files**
- **?: regular releases in OWL format**
- **?: slot-based annotation**

What next?

- Grammar for Text-Definitions
- Extend inference rules
  - e.g. Non-monotonic reasoning (cell HAS-PART nucleus EXCEPT erythroctye)
- Getting it to work as a DAG-Edit plugin
- Wait for, or help create good chemical, protein and “meta-anatomy” ontologies

Conclusions

- Decomposing GO terms is useful, and achievable
- Reasoning over the resulting logical definitions is possible, and can help maintenance
- Combination of grammar & reasoning is powerful – rigor + ease of use
- A new way of thinking about GO/OBO?
- Useful in all realms of complex biological data modeling

Acknowledgements

- Suzi
- John
- Brad
- David & Joel
- Michael
- GO Curators
- Chris Wroe
- Robert Stevens