

# Obol: Open Bio-Ontology Language

*Using grammars to extract and use implicit  
knowledge in the GO and OBO*

Chris Mungall

*Berkeley Drosophila Genome Project /  
GO Consortium*

# Obol

- Obol is a system for discovering and reasoning over hidden knowledge in ontologies
- Obol is useful for helping maintain cross-products in the Gene Ontology
- Obol works by parsing syntax and semantics from GO and OBO terms

# Motivation: Ontology Maintenance

- GO: 3 ontologies, 16k terms, 23k relationships
- OBO: cell, biochemical, sequence and multiple anatomical ontologies
- Many GO terms are combinatorial (cross-products)
  - “regulation of neutrophil differentiation”
- No explicit links between ontologies
- Difficult to maintain manually

# Some Sample GO terms

‘regulation of neutrophil differentiation.’

‘neutrophil differentiation.’

‘granulocyte differentiation.’

‘smooth muscle contraction.’

‘nucleolar chromatin.’

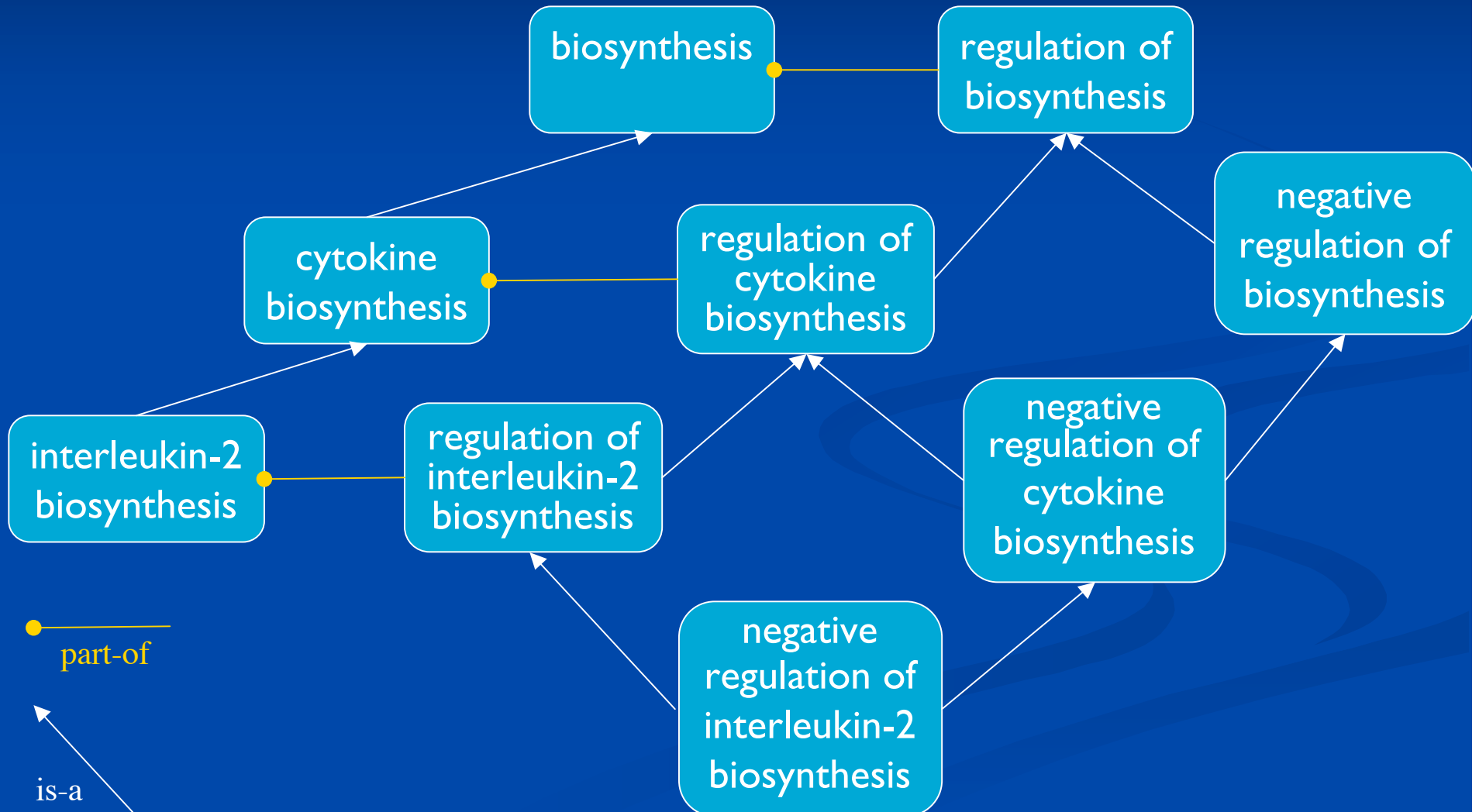
‘nucleolus.’

‘oxygen transport.’

‘negative regulation of interleukin-2 biosynthesis.’

‘oxidoreductase activity, acting on paired donors,  
with incorporation or reduction of molecular oxygen,  
reduced iron-sulfur protein as one donor,  
and incorporation of one atom of oxygen.’

# Graph complexity



# Automatic inference of relationships

- Some relationships can be derived computationally...
- ...provided we have complete logical definitions

regulation ^  
(*regtype:negative*) ^  
(*regprocess:biosynthesis* ^ (*makes:interleukin-2*))

Tools exist for reasoning over these logical definitions, but...

# Generating logical definitions

- Generating and maintaining logical definitions for GO/OBO is non-trivial
- Obol exploits the highly regular grammatical structure of GO term names
  - “regulation of X”, never “X regulation”
  - “Y biosynthesis”, never “biosynthesis of Y”
  - no stemming required
- Obol derives candidate class definitions from term names, and performs basic reasoning over them

# Obol: parsing and reasoning

GO/OBO Term

*Lexical string*



Class Definition(s)

*may involve relationships to other OBO terms*



Inferences

*using definitions and existing ontologies*

“interleukin-2 biosynthesis”



biosynthesis^(makes:interleukin-2)



“interleukin-2 biosynthesis”

*is\_a*

“cytokine biosynthesis”

*inferred from: interleukin-2 is\_a cytokine*

# How Obol Works

- term names are broken into lexical tokens (words) using a tokeniser
- tokens are parsed using a *grammar*, generating parse trees
- parse trees are turned into class definitions using *transformation rules* and *property definitions*
- transformation is reversible
- class definitions are reasoned over
- implemented in XSB Prolog

# Word tokens

- Obol uses an atomic vocabulary of word tokens
- tokens are partitioned by ontology domain
  - cell, anatomy, biological process, etc
- tokens have a grammatical type
  - adj, noun, prep, relational adj, special
- vocabularies need not be correct or complete

# Computational Grammars

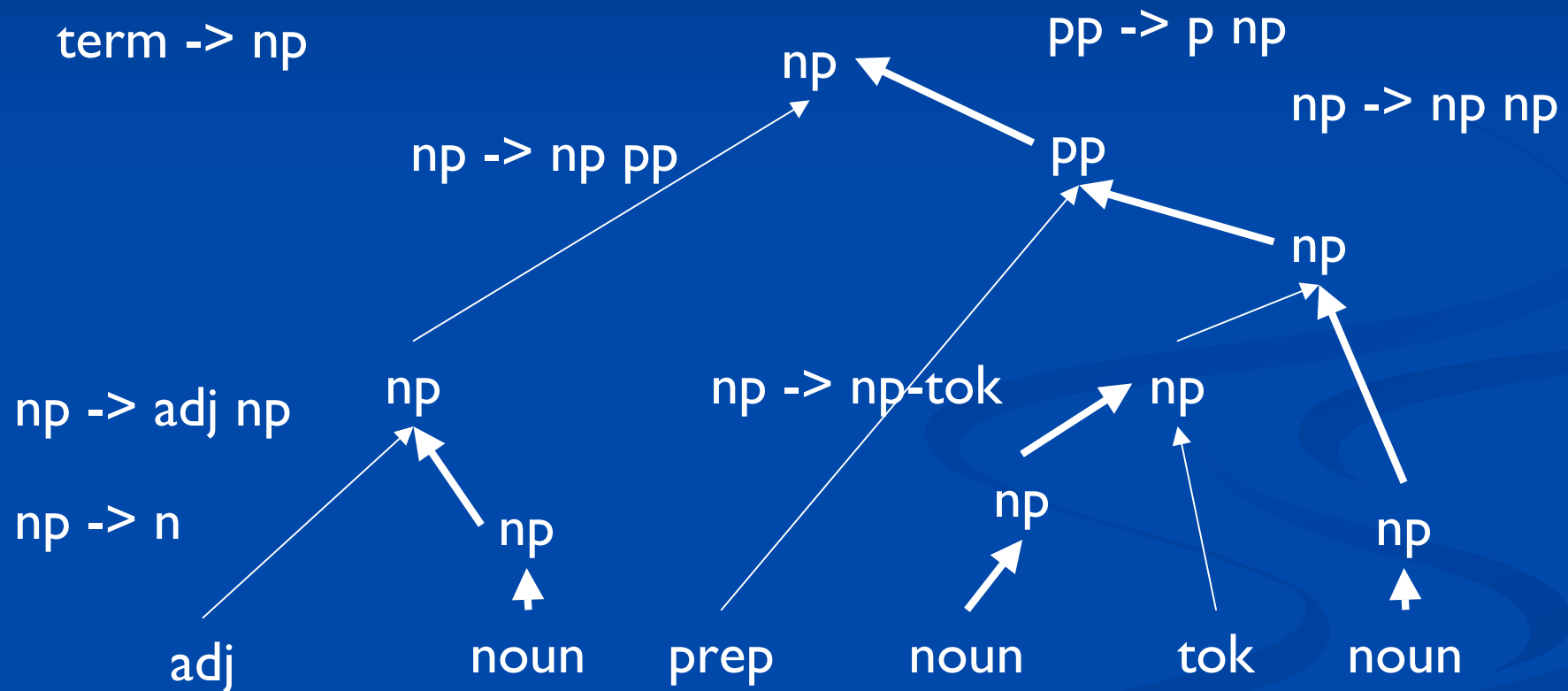
- formal grammars can elucidate sentence structure
- grammars transform token lists into parse trees
- multiple parses may be possible
- parses are reversible
- a grammar is a collection of transformation rules

# A simple OBO term grammar

*(subset of the whole OBO grammar)*

Term	--> NP	<i>e.g. negative regulation of interleukin-2 biosynthesis</i>
NP	--> NP PP	<i>e.g. negative regulation of interleukin-2 biosynthesis</i>
NP	--> NOUN	<i>e.g. interleukin;; regulation;; biosynthesis</i>
NP	--> NP-TOK	<i>e.g. interleukin-2</i>
NP	--> ADJ NP	<i>e.g. negative regulation</i>
NP	--> NP NP	<i>e.g. interleukin-2 biosynthesis</i>
PP	--> PREP NP	<i>e.g. of interleukin-2 biosynthesis</i>

# Applying grammar rules



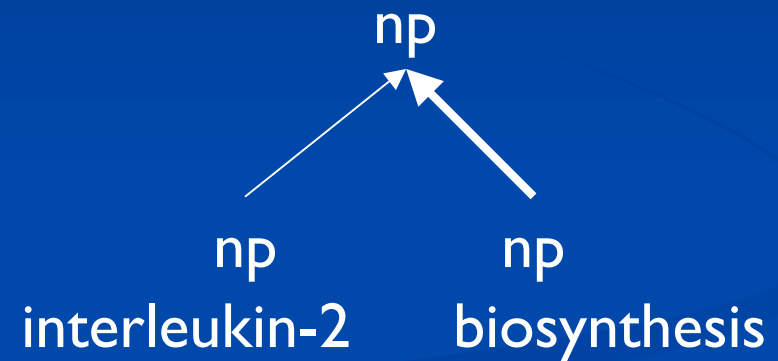
negative regulation of interleukin-2 biosynthesis

# Generating Class Definitions

- A parse tree shows the *syntax* structure of a term
- A class definition is a description of the *meaning* of a term
- An Obol classdef is a cross product (intersection) of necessary and sufficient conditions
- Classdefs are generated from parse trees using tree transform rules and property descriptions
- Classdefs can be exported using obo or OWL format

# Property definitions guide class construction

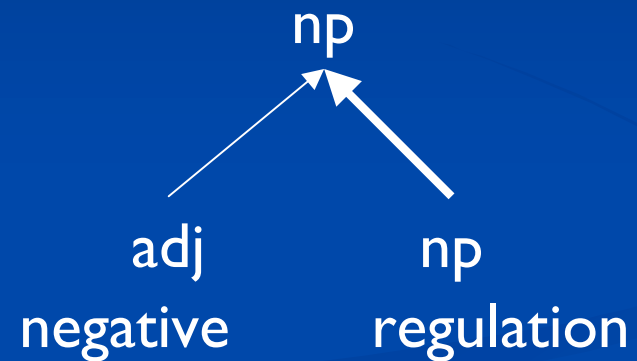
Property name: *makes*  
domain: biosynthesis  
range: substance  
grammar: np\_modifier



$\text{biosynthesis}^{\wedge}(\text{makes:interleukin-2})$

# Property definitions guide class construction

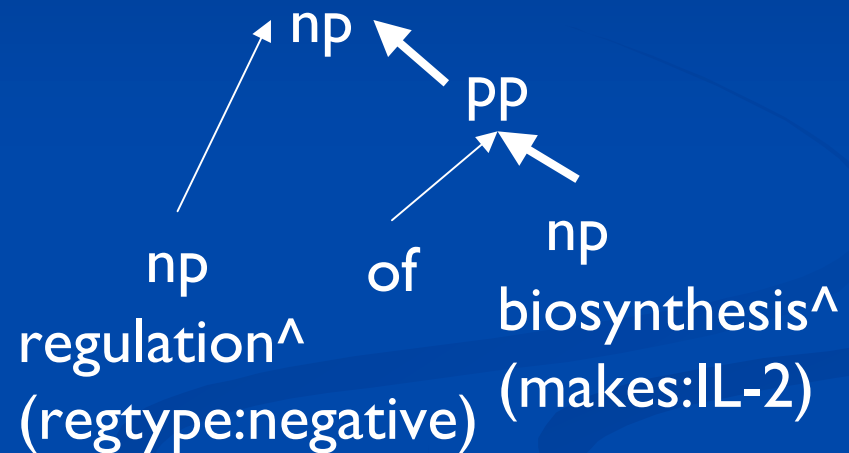
Property name: **regtype**  
domain: regulation  
range: neg/pos  
grammar: np\_modifier



regulation^(regtype:negative)

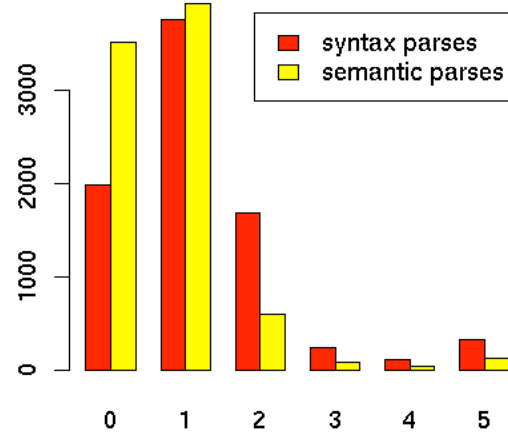
# Property definitions guide class construction

Property name: **regprocess**  
domain: regulation  
range: biological\_process  
grammar: prep(of)

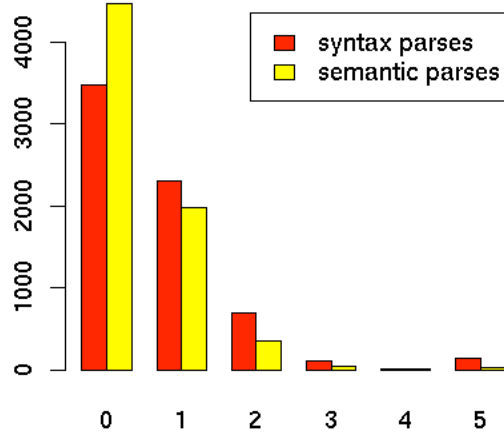


regulation^  
(regtype:negative)^  
(regprocess:biosynthesis^(makes:IL-2))

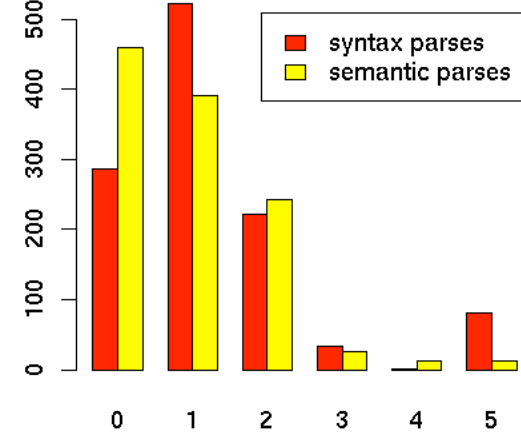
# Unparseable terms and multi-parse terms



biological  
process



molecular  
function



cellular  
component

*single-token terms excluded from this analysis*

# Reasoning over class definitions

- Using class definitions, we can:
  - autcreate parentage for new terms
  - check for missing relationships
  - find inconsistencies between ontologies
  - generate implicit orthogonal ontologies
- Method:
  - Use native OBOL rules (via prolog or DAG-Edit)
  - *OR* use external reasoner; eg RACER, FaCT

# Finding missing relationships

- Obol is run periodically on GO to check for missing IS A and PART OF relationships
- Multiple parses produce false-positives
- 223 missing relationships added to GO
- ToDo: increase specificity by improving vocabularies and property definitions

# Obol sample report

nucleolar chromatin PART OF nucleus

clathrin-coated vesicle HAS PART clathrin coat

chromoplast membrane IS A plastid membrane

nuclear microtubule PART OF nucleus

vitamin E biosynthesis IS A vitamin E metabolism

uracil permease activity IS A permease activity

chloroplast envelope IS A plastid envelope

negative regulation of lipid biosynthesis

IS A negative regulation of lipid metabolism

ketone body metabolism IS A ketone metabolism

dense nuclear body IS A nuclear body

inverse

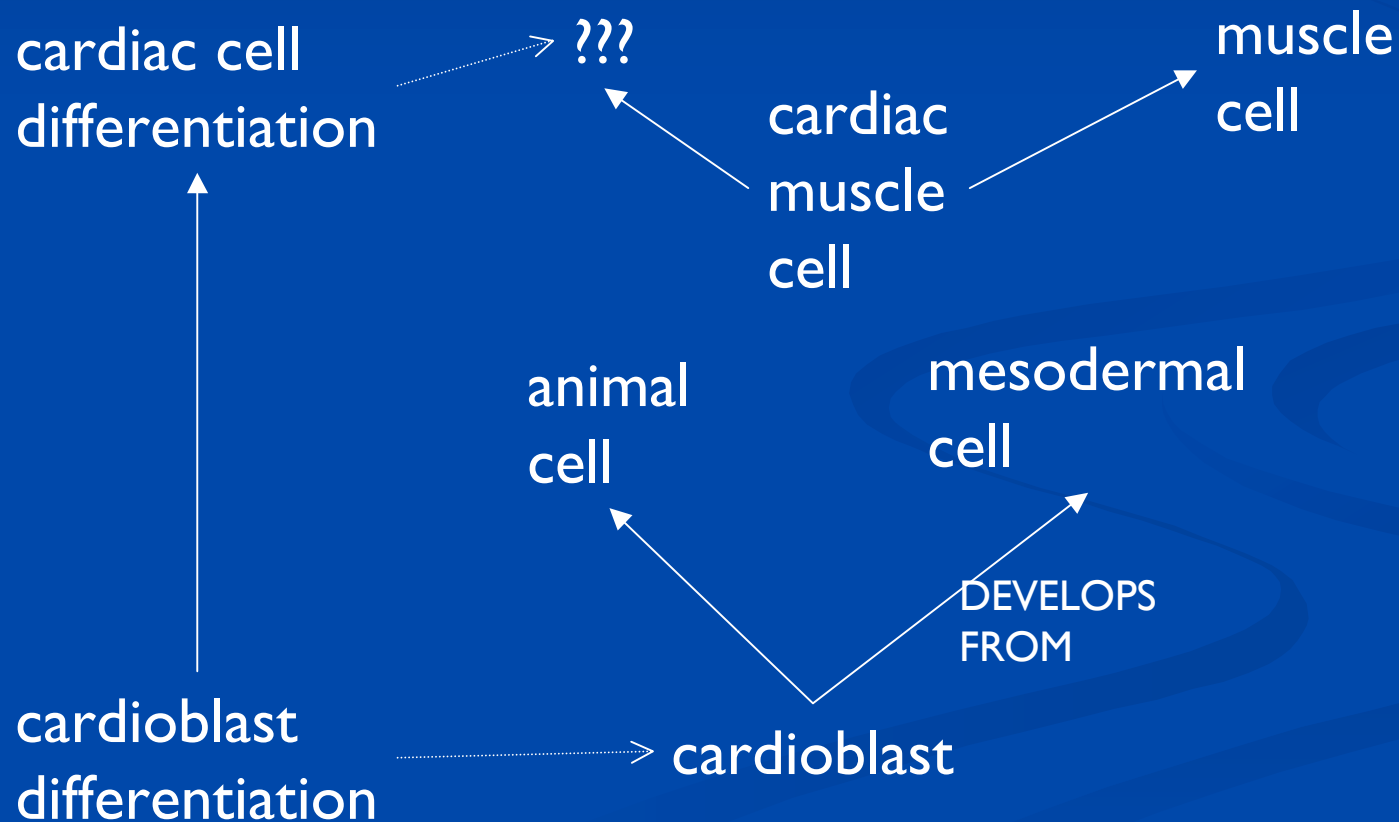
present

false

positive!

# Aligning to the OBO cell ontology

most differentiation terms align precisely; some don't...



# Deriving existing GO relationships

	Function	Process	Component
all relationships	8002	13613	1691
obol-derivable relationships	479	3055	346
non-trivially derivable relationships	0	1089	63

# Obol as an ontology curation tool

- Obol can be used by GO curators in a variety of ways
- “Behind the scenes”
  - Iterative
    - GO curator receives periodic suggestion reports
  - Continuous
    - GO curator uses OBOL interactively via DAG-Edit plugin
- To help the transition to a fully specified ontology
  - GO curators then maintain class definitions
- Obol as a search tool?

# Problems to address

- Integration with curation process
- Memory usage
- Syntax parsing
  - chemical terms, long terms
- Dealing with **and**, **or** and **not**
- Generating text definitions
- Word list maintenance
  - solution: integrate with ontology maintenance
- Ontology dependencies
  - protein and generic anatomy ontologies needed
  - Obol can be used to help generate these

# Conclusions

- Obol is useful for maintainng large GO-style ontologies
- combination of semantic parsing with reasoning is powerful
  - benefits of both GO-style ontology development and formal reasoning

# Acknowledgements

## ■ Berkeley/GO

- John Richter
- Brad Marshall
- Karen Eilbeck
- Suzanna Lewis
- Gerry Rubin

## ■ Jackson Labs/GO

- David Hill
- Joel Richardson
- Judith Blake

## GO Curators

Midori Harris  
Jennifer Clark  
Amelia Ireland  
Jane Lomax

## Manchester

Chris Wroe  
Robert Stevens  
Phillip Lord  
J Michael Cherry  
Michael Ashburner  
all the GO Consortium