

## Using Cyc Content and Services.

Michael Witbrock, for PhD Open.

### *Part 1: Accessing and Assembling content.*

The vocabulary of Cyc is made available at <http://sw.opencyc.org/> and [1] describes how to assemble Cyc concepts into sentences in CyCL, a syntactic variation on predicate calculus.

Question 1a: What are the Cyc terms for the following concepts, both as external ids, un-versioned URIs and en concept names: Poland; the land occupied by Poland; moving from one place to another; Łódź, the city; Warsaw; Nicolas Copernicus

Question 1b: How would one express, in CyCL, the apparently counterfactual sentence “Nicolas Copernicus lives in Warsaw.” How about “Copernicus is an expert in Astronomy”. Optional: May need research into microtheories to answer: How does one express the sentence “Nicolas Copernicus lived in Krakow from 1491 to 1495”, assuming he didn’t leave for more than a month at a time.

Question 1c: How would one express, in CyCL, the sentence “Nicolas Copernicus went from Lodz to Warsaw on Jan 14<sup>th</sup> 1490.” How about “Nicolas Copernicus took a trip from Lodz to Warsaw and onto Krakow in 1490.”

[1] Cynthia Matuszek, J. Cabral, M. Witbrock, J. DeOliveira.

An Introduction to the Syntax and Content of Cyc

In Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Stanford, CA, March 2006. ([http://www.cyc.com/doc/white\\_papers/AAAI06SS-SyntaxAndContentOfCyc.pdf](http://www.cyc.com/doc/white_papers/AAAI06SS-SyntaxAndContentOfCyc.pdf) )

### *Part 2: Using content; Access and mash-ups.*

Cyc web services are available at <http://ws.cyc.com/> Use external IDs in your code.

Question 2a: In any convenient language, write code to look up the terms in Question 1a. Explain the success or failure of looking up each particular term.

Question 2b: In any convenient language, write code to find the classes that Nicolas Copernicus is a member of. And for Warsaw. And for Peking University in China.

Question 2c: In any convenient language, write code to interestingly combine knowledge from an Cyc web service with information from at least one other source of semantic web content and at least one other non-cyc, non SW web service. Document what you want to do, and why doing so will be interesting, lightly document the access methods used for each included knowledge source or web service, and explain the design and implementation philosophy of your code (all of this documentation should take less than three A4 pages in a 11 point of larger type face).

Show some screenshots illustrating your mash-up’s operation, and comment on how successfully it met your design goals. Interesting design goals and partial success count for more than completeness or robustness here.