2: Abstraction and Definition in Conceptual Graphs

CG Discussion 02 – April 02, 2010

Benjamin Chu Min Xian

Abstraction & Definition

Definition:

A **definition** consists of the **genus** (the family) of things to which the defined thing belongs, and the **differentia** (the distinguishing feature which marks it off from other members of the same family).

What is a triangle?

Triangle is defined as a **plane figure** bounded by **three straight sides** .

Genus

Differentia

2 ways to specify a type:

- By stating the necessary and sufficient conditions of the type

- Giving examples and say everything similar to these belong to this type

Why need type definition?

To **expand** and **contract** a concept from a graph.

Abstraction & Definition

Recap: What is a canonical graph?

```
[elephant:Dumbo]<-(agnt)<-[pink]<-(attr)<-[fly]->{
                                                    (org)->[asian-cities:kl];
                                                    (dest)->[european-cities:london];
                                                 }.
```

#Not canonical- Canonical graphs are those graphs which represent a real-world situation either by observation, derivation, or insight.

In type definition, some concept is chosen to be the genus and a canonical graph becomes the differentia.

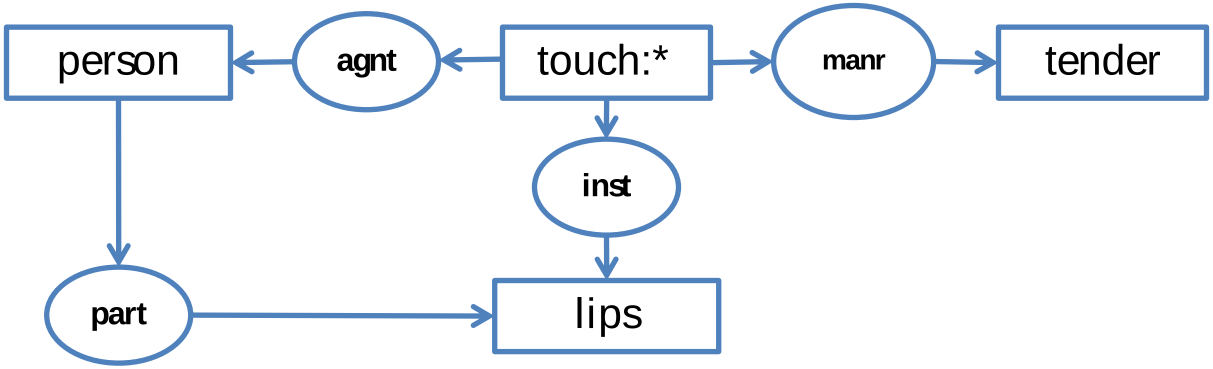
The syntax for type definition is shown below:

Type *type-name* (x) is
canonical-graph.

Abstraction & Definition

Example (1):

type *kiss* (x) is:



person
agnt
touch:*
manr
tender
inst
part
lips

Defines **kiss** with genus **touch** and with a differentia graph that says that the touching is done by a person's lips in a tender manner.

Example (2):

Type Person(x) is

```
[Animate-Being: *x]-{
                 (attr) -> [leg: *] ;
                 (attr) -> [arm: *] ;
                 (attr) -> [eye: *];
               }.
```

The argument x is a way of identifying unique instances of type person

* Denotes any number of legs, arms or eyes

Abstraction & Definition

Type definition is used for type contraction and expansion:

Type contraction deletes subgraphs that can be recovered from information in the differentia.

Type expansion replaces a concept type with its definition. The type label of the genus replaces the defined type label, and the graph for the differentia is joined to the concept.

Consider the type definition graph for **person** shown below:

Type Person(x) is

```
               [Animate-Being: *x]-{
                                   (attr) -> [leg: 2] ;
                                   (attr) -> [arm: 2] ;
                                   (attr) -> [eye: 2];
                                   (attr)->[birthday]-
                                                 {
                                                   (attr)->[month];
                                                   (attr)->[day];
                                                   (attr)->[year];
                                                 }
                                   (attr)->[height:@inches];
                                   (attr)->[age:*];
                             }.
```

Abstraction & Definition

Example(3):

Consider the graph “A person travels from London to Paris via Lille by Eurostar".

```
[person:*]<-(agnt)<-[travel]-{
                                  (org)->[european-cities:london];
                                  (dest)->[european-cities:paris];
                                  (path)->[european-cities:lille];
                                  (inst)->[train:eurostar];
                            }.
```

The type expansion of the graph based on the concept type person is shown below:

```
[person: *x]-{
               (attr)->[leg: 2] ;
               (attr)->[arm: 2] ;
               (attr)->[eye: 2];
               (attr)->[birthday]-{
                                     (attr)->[month];
                                     (attr)->[day];
                                     (attr)->[year];
                                  }
              (attr)->[height:@inches];
              (attr)->[age:*];
             }<-(agnt)<-[travel]-{
                                    (org)->[european-cities:london];
                                    (dest)->[european-cities:paris];
                                    (path)->[european-cities:lille];
                                    (inst)->[train:eurostar];
                                 }.
```

Abstraction & Definition

Type definition is used for type contraction and expansion:

Type contraction deletes subgraphs that can be recovered from information in the differentia.

Type expansion replaces a concept type with its definition. The type label of the genus replaces the defined type label, and the graph for the differentia is joined to the concept.

Consider the type definition graph for **person** shown below:

Individual person(felix) is

```
        [person: felix]-{
                   (attr) -> [leg: 2] ;
                   (attr) -> [arm: 2] ;
                   (attr) -> [eye: 2];
                   (attr)->[birthday]-
                          {
                           (attr)->[september];
                           (attr)->[27];
                           (attr)->[1987];
                          }
                   (attr)->[height:@172inches];
                   (attr)->[age:27];
                };
```

Abstraction & Definition

Example(3):

Consider the graph “Felix travels from London to Paris via Lille by Eurostar".

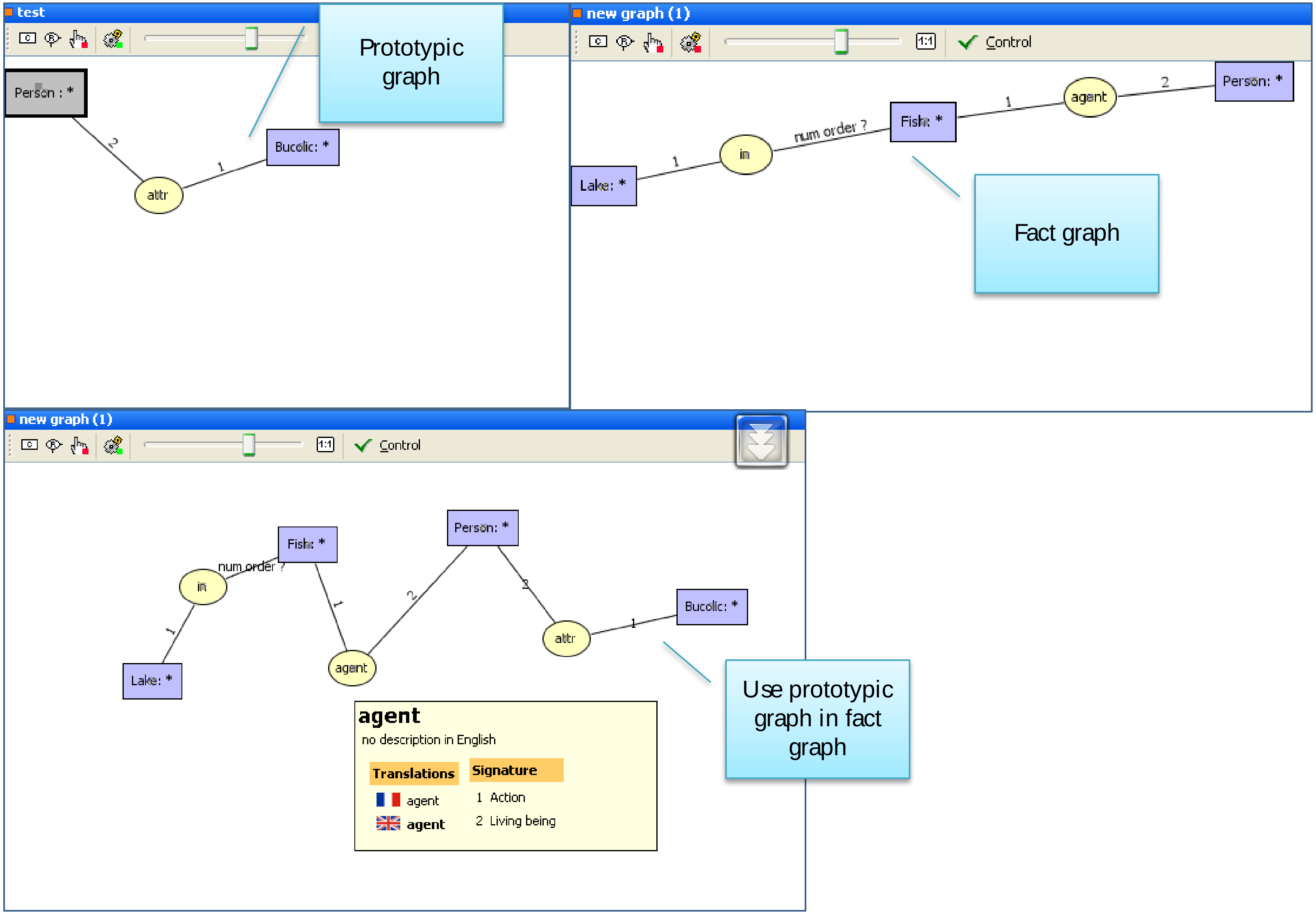
```
[person:felix]<-(agnt)<-[travel]-{
                                   (org)->[european-cities:london];
                                   (dest)->[european-cities:paris];
                                   (path)->[european-cities:lille];
                                   (inst)->[train:eurostar];
                             }.
```

The type expansion of the graph based on the concept type person is shown below:

```
[person: felix]-{
            (attr)->[leg: 2] ;
            (attr)->[arm: 2] ;
            (attr)->[eye: 2];
            (attr)->[birthday]-{
                                (attr)->[september];
                                (attr)->[27];
                                (attr)->[1987];
                             }
           (attr)->[height:@172inches];
           (attr)->[age:27];
          }<-(agnt)<-[travel]-{
                                (org)->[european-cities:london];
                                (dest)->[european-cities:paris];
                                (path)->[european-cities:lille];
                                (inst)->[train:eurostar];
                             }.
```

Abstraction & Definition

Example(4): Using prototypic graph in COGUI



Prototypic graph

Fact graph

Use prototypic graph in fact graph

agent

no description in English

Translations	Signature
agent	1 Action
agent	2 Living being