

# Contextual Vocabulary Acquisition of the Verb ‘Fold’

Masashi Sato

Fall 2003

CSE499: Independent Study

## Abstract

This paper describes the results of my research on contextual vocabulary acquisition. In order to improve current verb algorithm, I represented a sentence involving one unknown verb “fold”, applied the algorithm, and added ability of recognizing properties and synonyms of the verb to it.

## 1. Introduction

Contextual Vocabulary Acquisition (CVA) is defined as “the active, deliberate acquisition of a meaning for a word in a text by reasoning from textual clues and prior knowledge, including language knowledge and hypotheses developed from prior encounters with the word, but without external sources of help such as dictionaries or people” (Rapaport & Ehrlich 2000). One of the current CVA research projects at UB focuses on testing and developing computational algorithms for clarifying this process of the vocabulary acquisition. For this purpose, we represent our knowledge base in the Semantic Network Processing System (SNePS), which is a computational semantic network reasoning system. More informations about SNePS can be found at its main project website <http://www.cse.buffalo.edu/sneps/>. After constructing a sufficient amount of representations for the unknown word in SNePS, we apply our algorithms. In other words, we test and develop our algorithm by using a computational mechanism in order to derive the meaning of an unknown verb. Throughout the summer and fall of 2003, I concentrated on verb algorithm by using a sentence containing the unknown verb “fold”. In this report, I would like to describe my representation and reasoning for the unknown verb “fold”, the improvements to the verb algorithm, the final results of this project, and some research points I would suggest for future CVA researchers to consider.

## 2. Representation

The passage that I was assigned for this project was: “During the early 1930's, many small businesses **folded** (**fold**: to go out of business) or were bought up by large corporations” (Dulin 1970). In this context, the phrase “fold (fold: to go out of business)” is assumed to be unknown. I took two steps, analyzing its critical knowledge and expressing it in SNePS, to represent this main passage and its background knowledge.

First, I clarified two core concepts and their implications. There are two key concepts in the main passage. One was “1930's” and the other was that “small businesses were bought up by large corporations”. 1930's was the period of the Great Depression, in which a number of businesses faced serious economic slumps throughout the world. This first concept already implies the second key concept that some business organizations bought others, because one can buy others in economic situations when a power balance is broken, such as in a depression. The subconcept of the scale of large and small is also crucial in this context, since

large businesses usually sustain larger funds than small ones do, and, as a result, small corporations cannot survive in those circumstances. From the viewpoint of the sentence structure of the main passage, I could derive its underlying implications. The structure of the sentence saying that during some time period event A or B happens, implies that events A and B possibly share the same property. This is because they are undergoing the identical time period and social circumstances. In other words, the first event, in which small businesses folded, has possibly the same property to the agent as the second event, in which they were acquired by large corporations. Since the event in which large corporations acquire small business organizations is destructive for small businesses, the event in which small businesses fold is possibly destructive as well. Thus, I derived that the unknown verb “fold” is also destructive to businesses by analyzing these concepts and this underlying sentence structure.

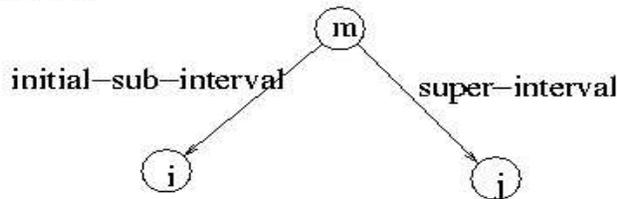
Second, I expressed these critical concepts and implications together with additional background knowledge in SNePS. I will describe my representations with network diagrams.

Main passage representation:

In order to represent the initial part of the time interval or duration in SNePS, two new case frame was developed in addition to the standard case frames. The INITIAL-SUB-INTERVAL/SUPER-INTERVAL and ETIME/EAFFAIR case frames were suggested from Almeida's detailed work for time representation (1995, p.185).

**INITIAL-SUB-INTERVAL/SUPER-INTERVAL**

**SYNTAX**

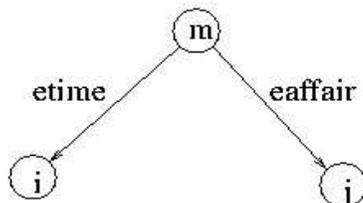


**SEMANTICS**

[[m]] is the proposition that [[i]] is the initial part of the interval [[j]]

**ETIME/EAFFAIR**

**SYNTAX**

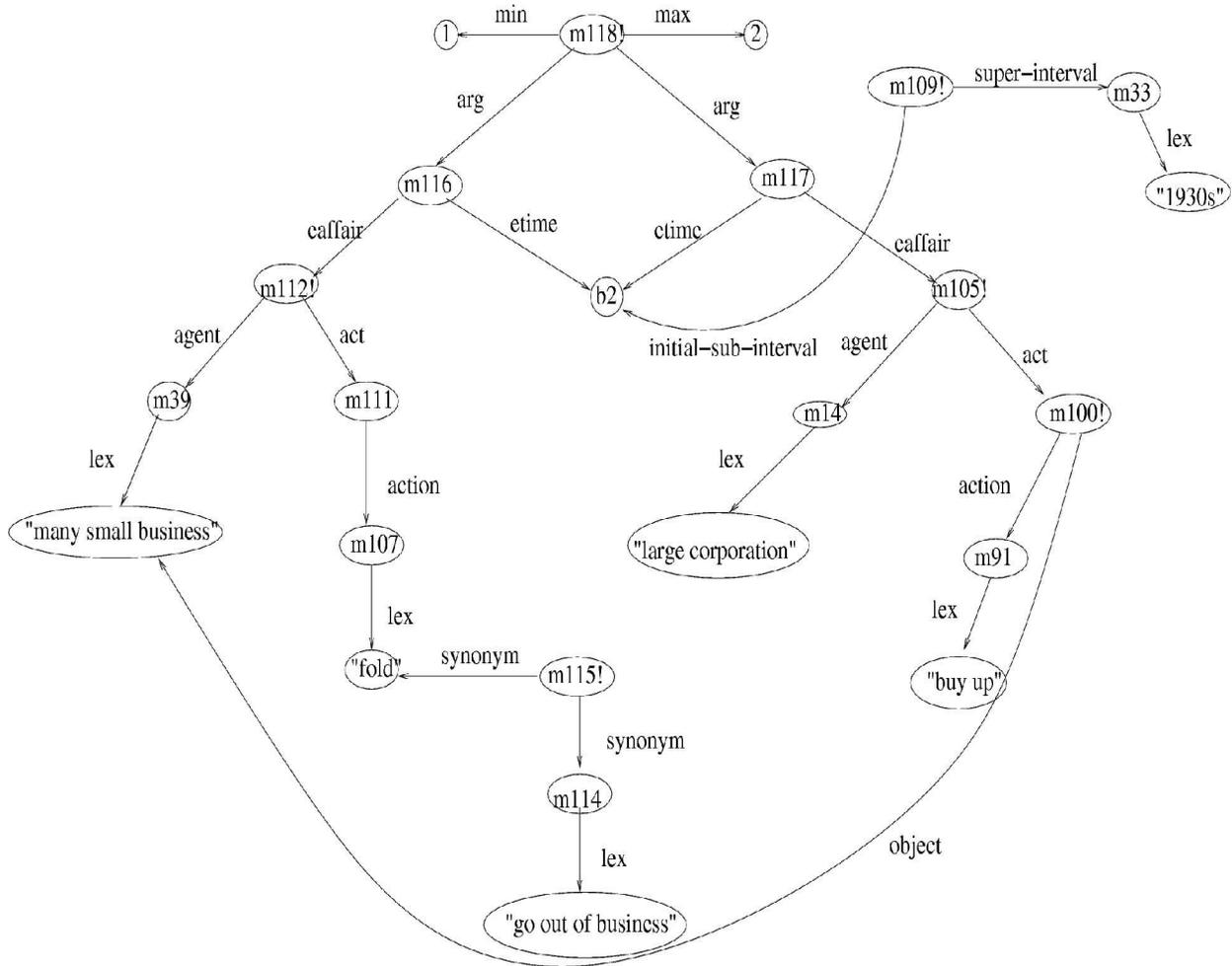


**SEMANTICS**

[[m]] is the proposition that [[i]] is the time period during which [[j]] occurs

I modified the second part of the main sentence from passive to active tense since there is no significant difference between them. Thus, the main sentence became: “During the early 1930's, many small businesses folded (fold: to go out of business) or large corporations bought them up”. As an important point, I inserted the notion that “fold” is a synonym of “go out of business” into my representation.

During the early 1930s, many small business folded (fold: to go out of business), or large corporation bought them up.

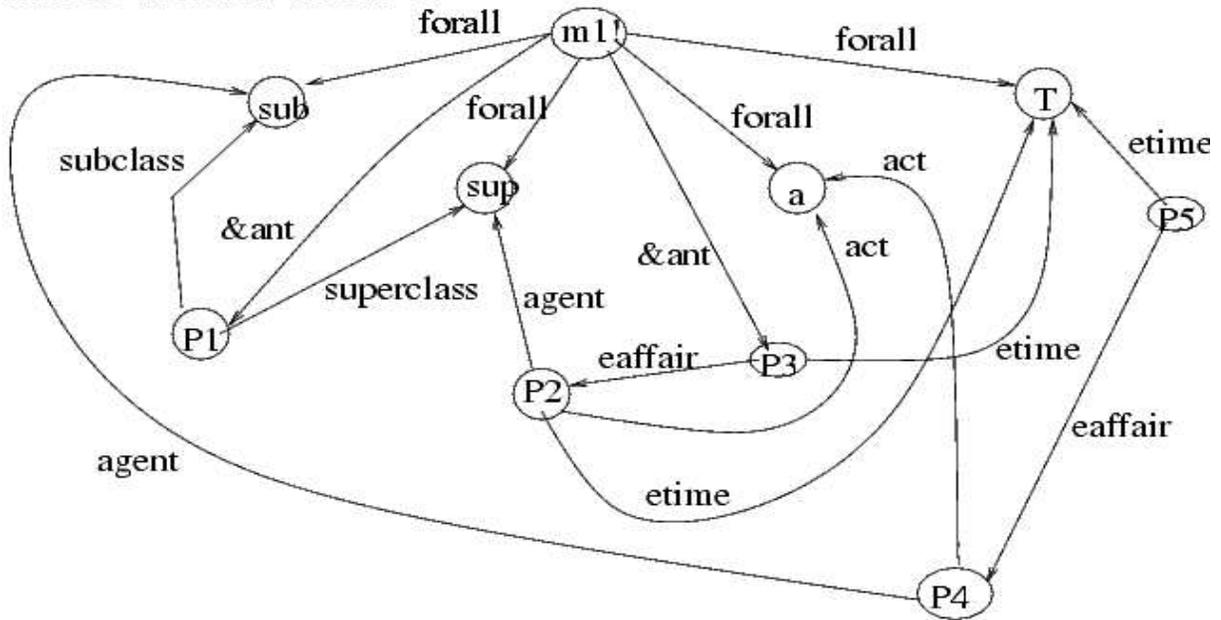


**Background knowledge representation:**

**RULES**

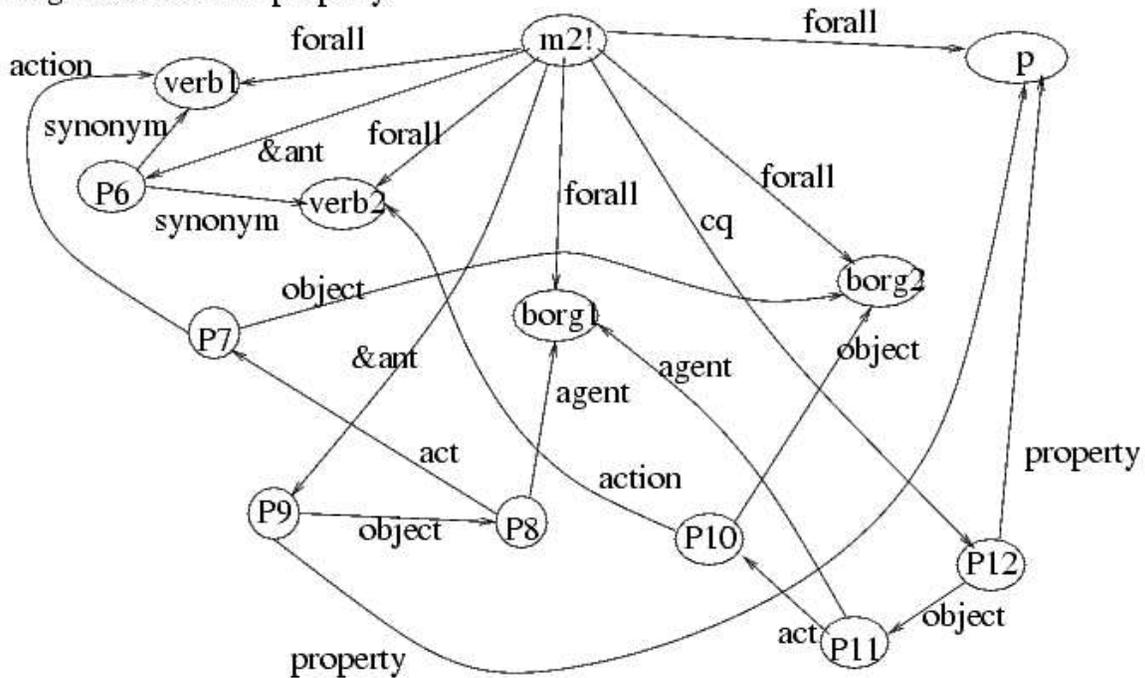
Subclass/superclass relation background

1. For all sub, sup, a, and T, if sub is a subclass of sup, and sup does act a at T, then sub also does act a at T.



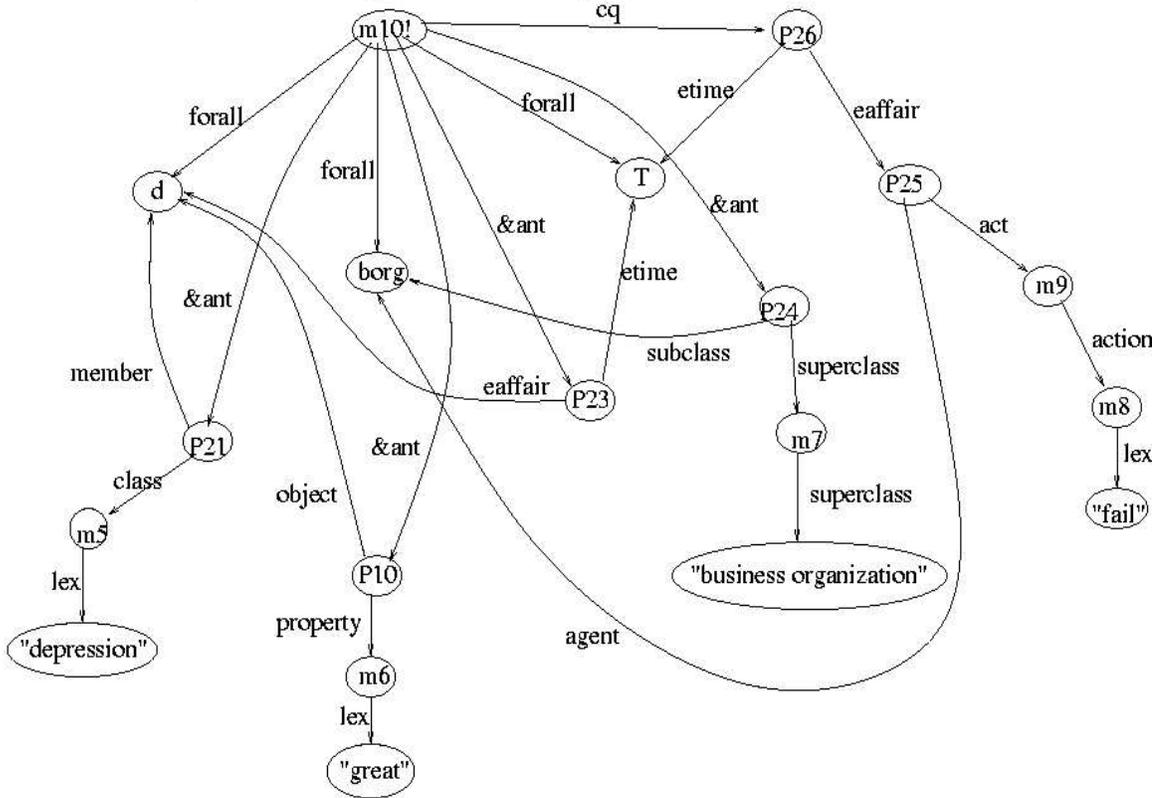
**Synonym rule**

For all verb1, verb2, borg1, borg2, and p, if verb1 and verb2 are synonyms, the fact that agent borg1 does verb1 on borg2 has property p, then the fact that borg1 does verb2 on borg2 has the same property.



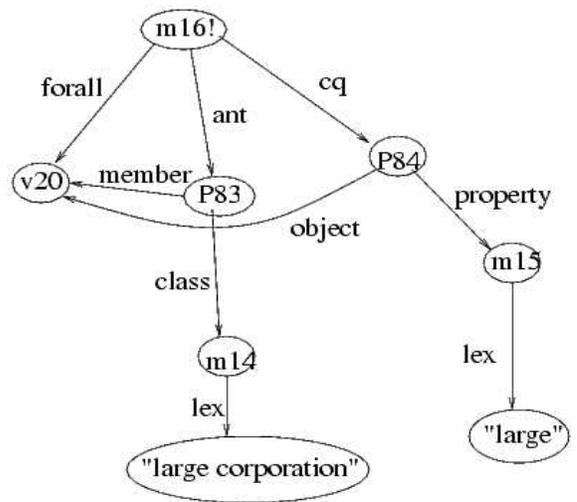
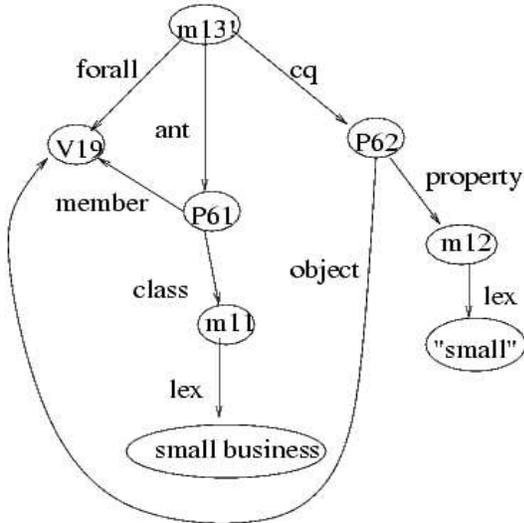
Depression rule

For all d, borg, and T, if d is a member of depression with a property of being great, and depression occurs throughout a time interval T, then business organizations fail throughout T.



Business organization rules 1

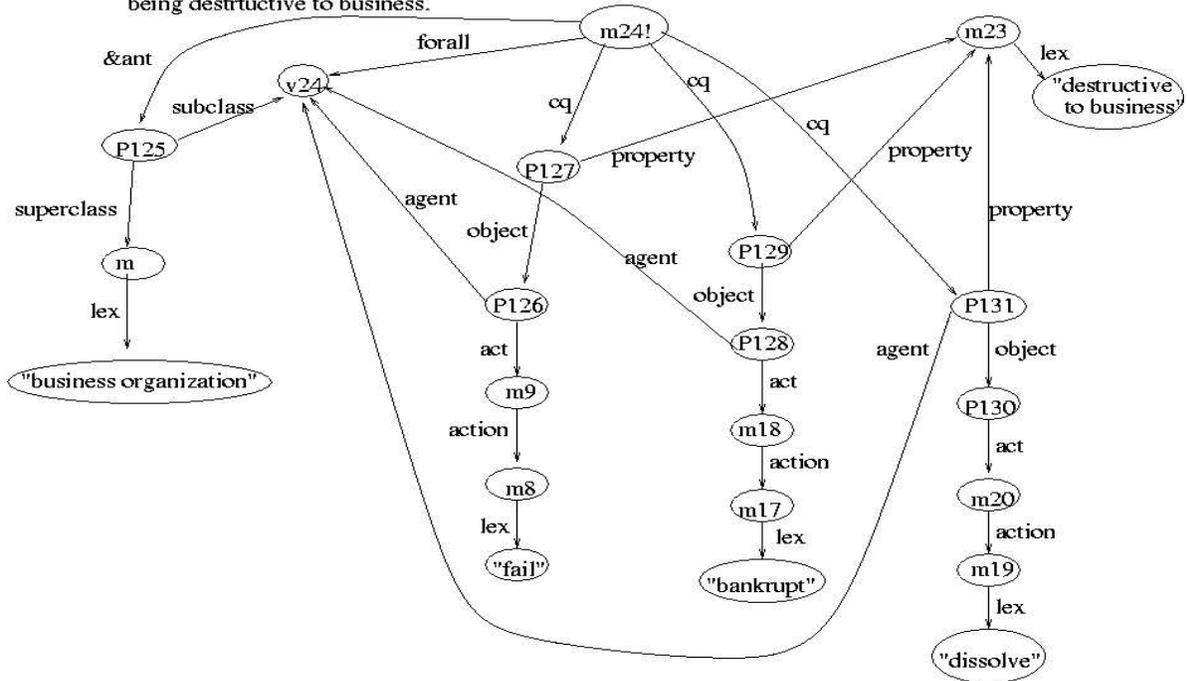
1. Each member of a small business class has a property of being small
2. Each member of the large corporation has the property of being large.





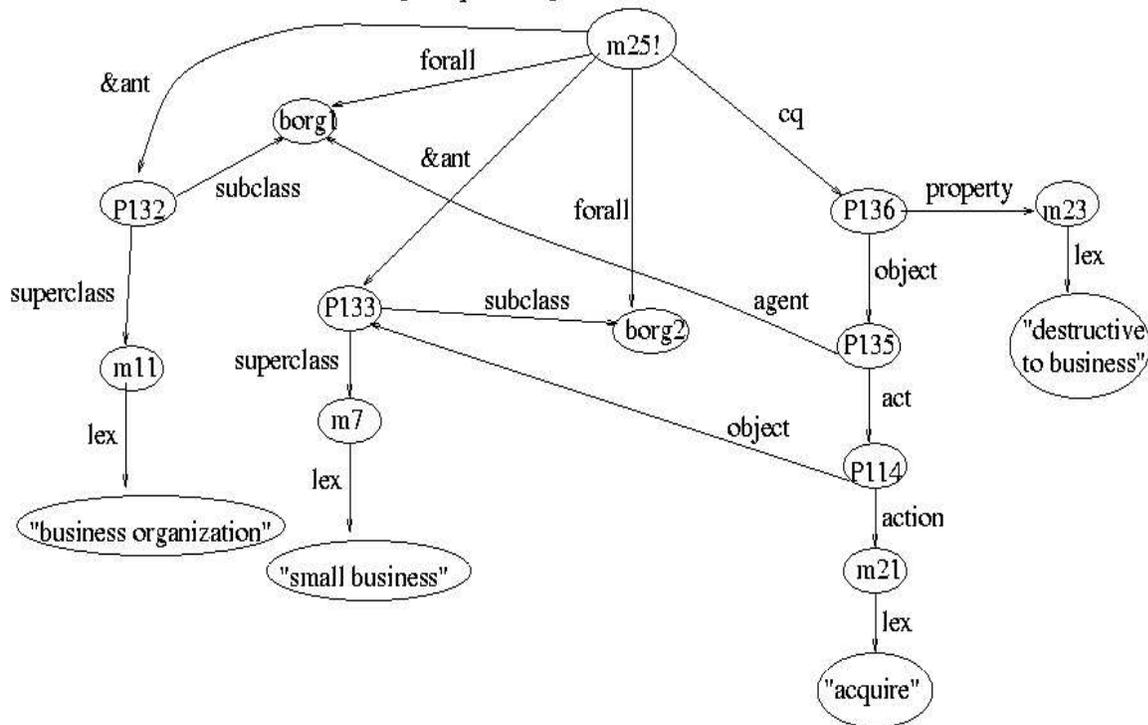
Event affair property rule1

1. Event affair such that any business organization, fails, bankrupts, and dissolves have the property of being destructive to business.



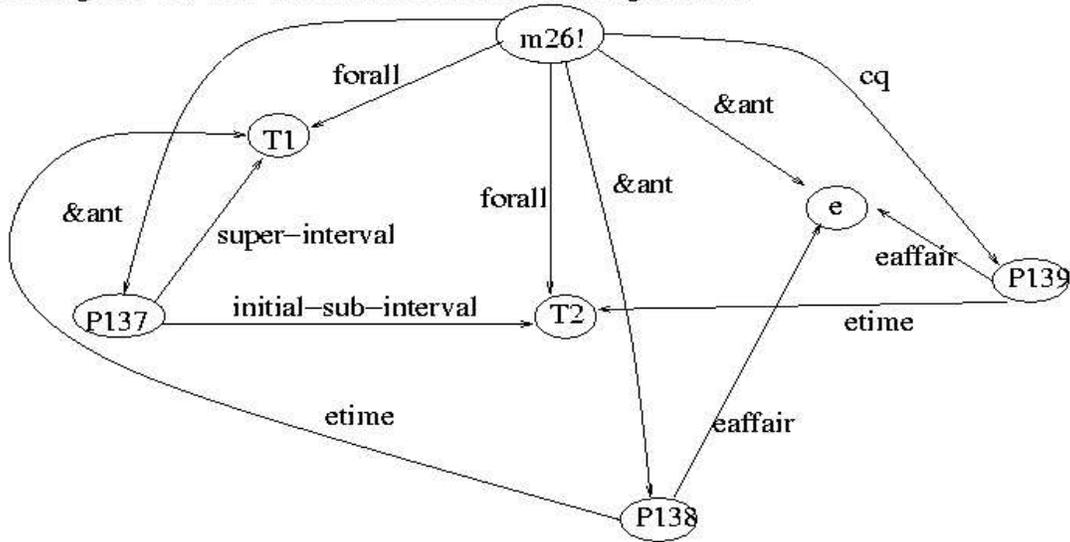
Event affair rule2

Forall borg1, and borg2, if borg1 is a subclass of business organization, and borg is a subclass of small busin then the event in which borg1 acquire borg2 is destructive to business.



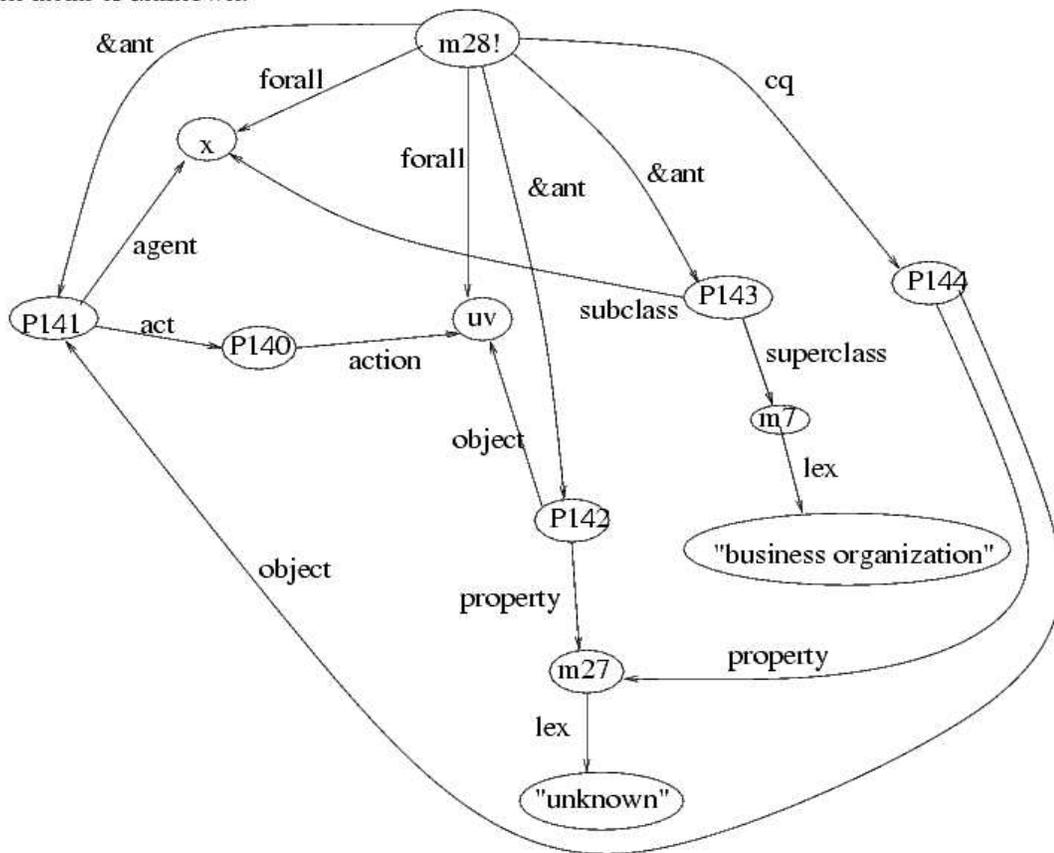
Event time interval rule

1. For all T1, T2, and e, if T1 is an super-interval of the initial-sub-interval T2, and e is an event at occurs throughout T1, then e is also an event affair throughout T2.



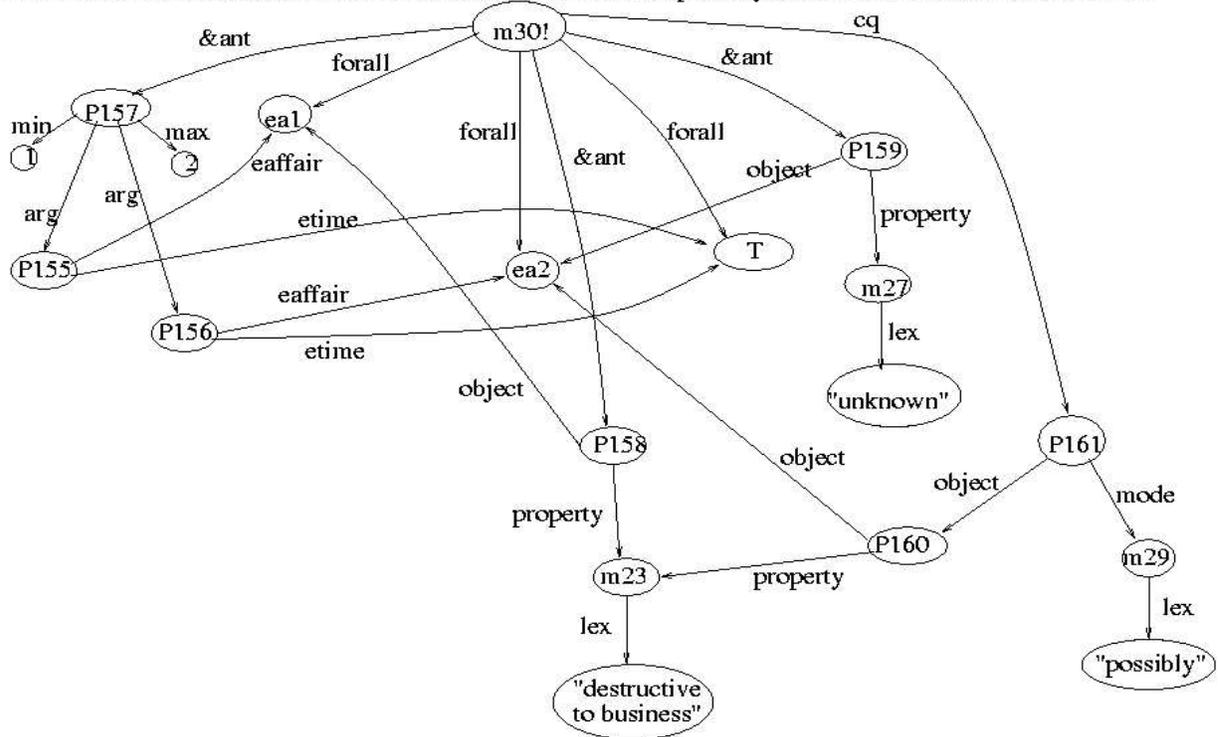
Unknownverb rule

For all x and uv, if agent x does uv action, uv is unknown, and x is a subclass of business organization then its event affair is unknown.



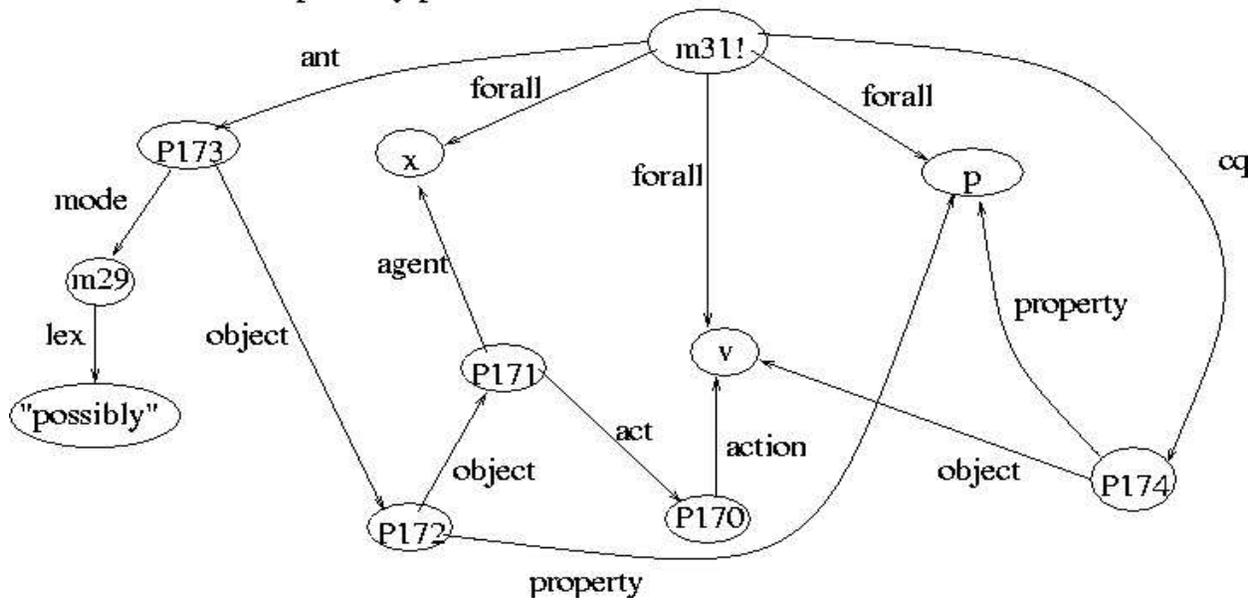
First key rule

For all ea1, ea2, and T, if ea1 or ea2 happens throughout a time interval T, ea1 is destructive to business, and ea2 is an unknown event, then possibly, ea2 is also destructive to business.



Second key rule

For all x, v, and p, if agent x does act v action, and this event is possibly p, then v action is also possibly p.



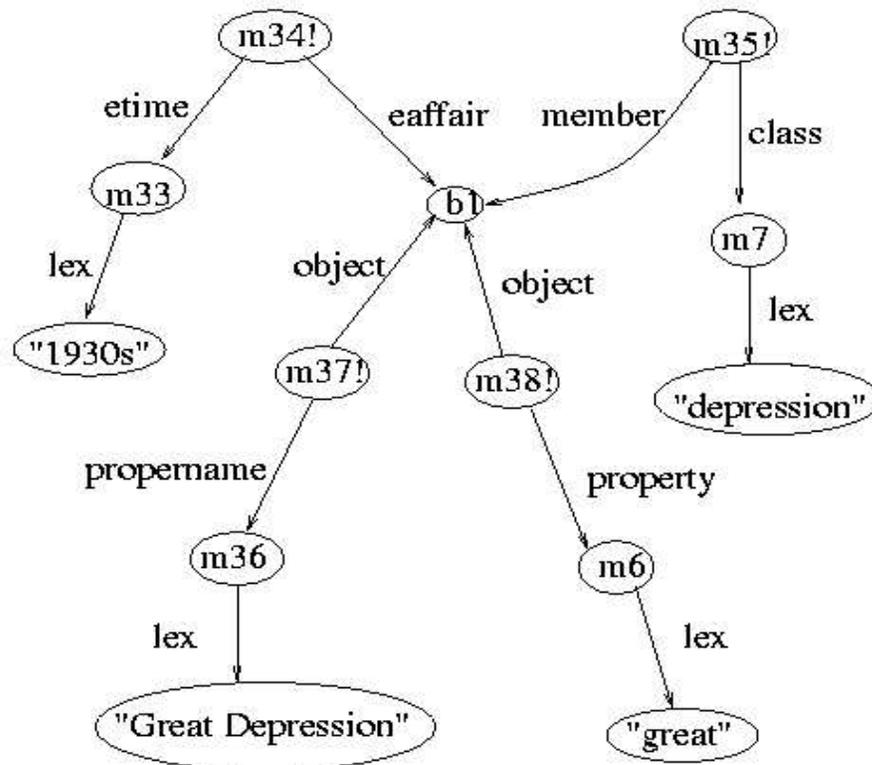
1930s' background

1. Something took place during 1930s

2. That was a member of depression.

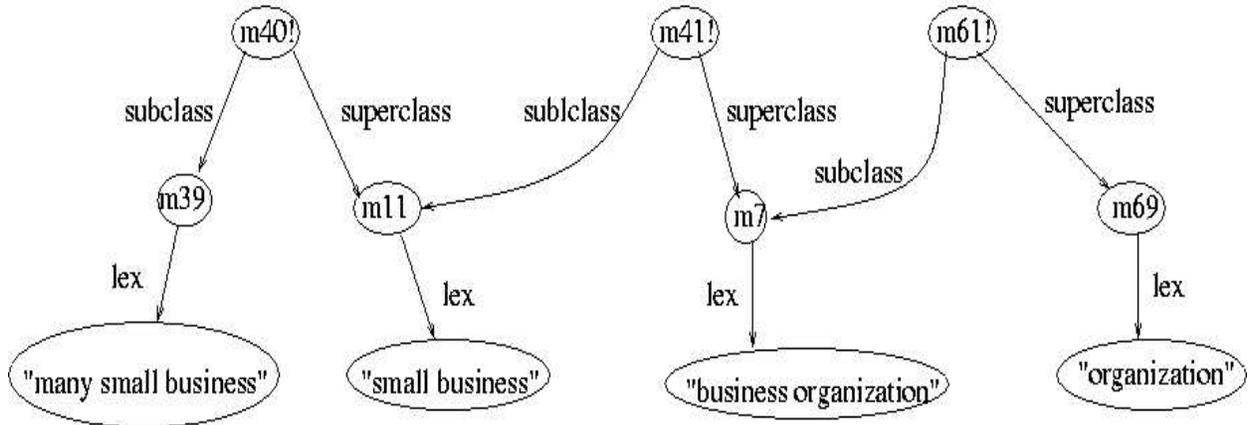
3. Its propername was "Great Depression"

4. Great Depression has the property of being great



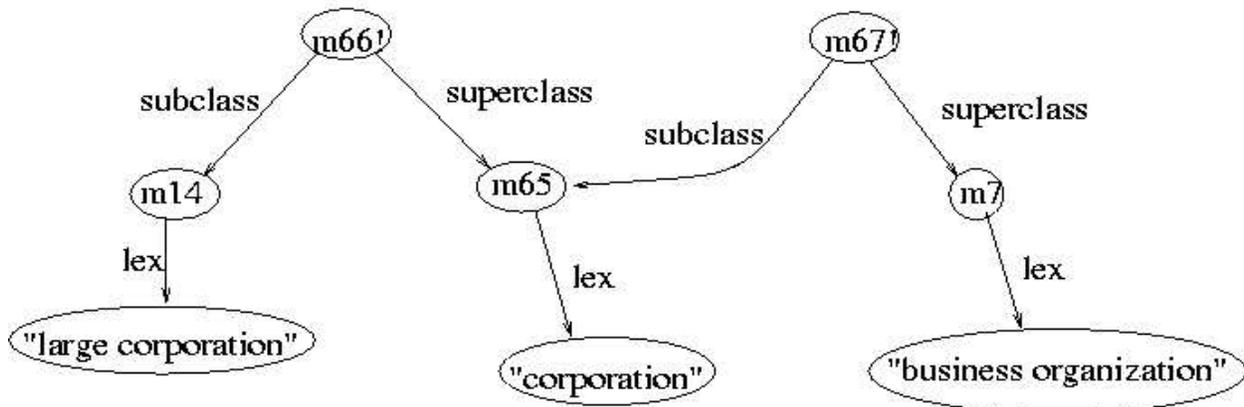
### Small businesses background

1. Many small business is a subclass of small business.
2. Small business is a subclass of business organization.
3. Business organization is a subclass of organization.



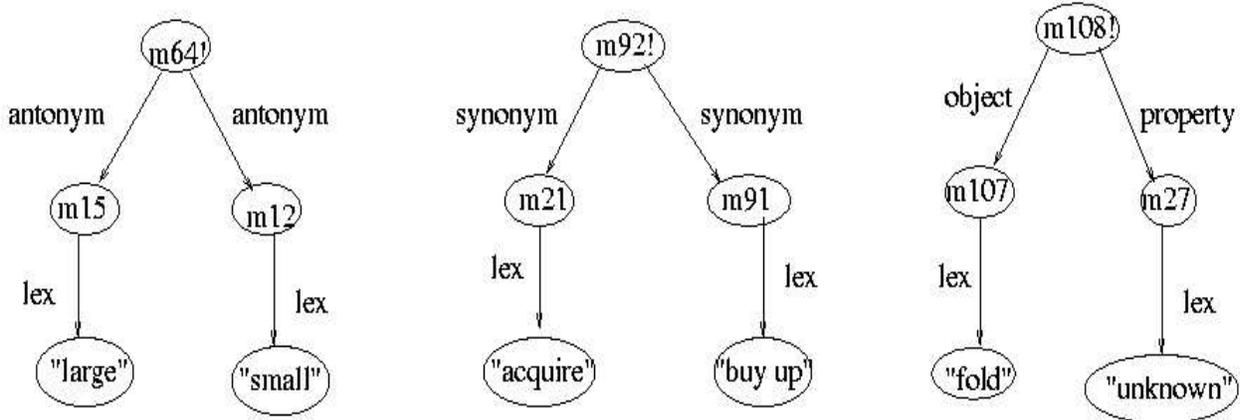
### Large corporation background

1. Large corporation is a subclass of corporation.
2. Corporation is a subclass of business organization.



Words background

1. Small and large are antonyms.
2. "Acquire" and "buy up" are synonyms.
3. "fold" is unknown.



### 3. Improvements on the verb algorithm

Throughout the CVA research, at least four verb algorithms have been developed already. When I applied those verb algorithms to my representations above, no algorithm could produce useful information about the unknown verb “fold” even though the inference package of SNePS could succeed in producing a node indicating that the verb “fold” is destructive to business. I chose one algorithm and added some improvements so that it could derive the valuable information about the verb.

I chose the verb algorithm developed by Del Vecchio during his summer research in 2002. The actual file can be found at <http://www.cse.buffalo.edu/~jmdv/summer2002-verbalgorithm-jmdv/0/2.html>. I chose this algorithm because it is the only algorithm, which succeeded in deriving that most uses of the verb “fold” is intransitive (Del Vecchio 2002). The main improvements were adding two functions to the algorithm. The first function is for implying the properties of the verb, and second one is for indicating the synonym of the verb;

#### •PrintVerbPropertyString function

```
(defun printVerbPropertyString
  "List the properties of the unknown verb"
  (setf verbPropertyString "Properties of the verb are: ")
  (setf verbNodes (first (list #3! ((find (lex ) ~verb )))))
  (setf propertyNodes nil)
  (dolist (v verbNodes)
    (setf propertyNodes (first (list #3! (( find (property- object) ~v))))))

  (cond ((eql propertyNodes nil)
        (setf verbPropertyString "No properties were found for this verb.")
        )
  )
```

```

(t
  (dolist (property propertyNodes)
    (setf verbProperty (first (list #3! ((find lex- ~property))))))
    (setf name nil)
    (setf name (first verbProperty))
    (setf name (node-to-lisp-object name))
    (setf name (symbol-name name))
    (setf verbPropertyString (concatenate 'string verbPropertyString name))
    (setf verbPropertyString (concatenate 'string verbPropertyString " : "))
  )
)
)
)
(setf verbPropertyString (concatenate 'string verbPropertyString " ~2% "))
verbPropertyString
)

```

### •PrintVerbSynonymString function

```

(defun printVerbSynonymString
  "List the synonyms of the unknown verb"
  (setf verbSynonymString "Synonyms of the verb are: ")
  (setf verbNodes1 (first (list #3! ((find (lex ) ~verb )))))
  (setf synonymNodes nil)

; (setf synonymNodes (findSynonyms ~verb))

  (dolist (v verbNodes1)
    (setf synonymNodes (remove v (first (list #3! (( find (synonym- synonym)
~v)))))))
;;This also returns the original verb node as well.

  (cond ((eql synonymNodes nil)
    (setf verbSynonymString "No properties were found for this verb.")
  )
  (t
    (dolist (synonym synonymNodes)
      (setf verbSynonym (first (list #3! ((find lex- ~synonym))))))
      (setf name nil)
      (setf name (first verbSynonym))
      (setf name (node-to-lisp-object name))
      (setf name (symbol-name name))
      (setf verbSynonymString (concatenate 'string verbSynonymString name))
      (setf verbSynonymString (concatenate 'string verbSynonymString " : "))
    )
  )
)

```

```

    )
    )
    )
    (setf verbSynonymString (concatenate 'string verbSynonymString " ~2% "))
    verbSynonymString
  )

```

#### 4. Final results and conclusion of my research project

As final results, the modified verb algorithm achieved to derive from my representations that the most common type of the sentence involving “fold” is intransitive, its properties are unknown and “destructive to business”, and its synonym is “go out of business”.

#### 5. Suggestions for future works

Even though I could improve the verb algorithm, there were several points which researchers should take into account in the future.

First, the algorithm failed to specify the agent, or the subject of the verb “fold”. The most obvious agent was “many small businesses” in my representation. Because the algorithm searches for the member arc of the MEMBER/CLASS case frame pointing to the agent, it failed to find “many small businesses”, which points to the subclass arc of SUBCLASS/SUPERCLASS case frame. This suggests two possible improvements. First possibility is further modification of the verb algorithm so that it searches for both member and subclass arcs. This can be done by more detailed examinations of the list code for the algorithms. The second possibility, which is beyond of the scope of the CVA project, is to develop representations for generalized quantifiers such as “most” and “many” in SNePS. However, we need more research into this topic in logic itself before it can be done (for more details, see Barwise & Cooper 1980).

Second, the algorithm also failed to imply possible synonym words from the representations. Even though my representations indicated that “fail”, “bankrupt”, and “dissolve” are also results of the Great Depression like “fold”, it could not derive any possible synonyms from these facts. This suggest to create a way of connecting the unknown verb with the other verbs belonging to the same category of effects.

#### 6. Bibliography

- Rapaport, William J., & Ehrlich, Karen (2000), “A Computational Theory of Vocabulary Acquisition”, in Lucja M. Iwanska & Stuart C. Shapiro(eds.), *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language* (Menlo Park, CA/Cambridge, MA: AAAI Press/MIT Press): 347-375.
- Dulin, Kenneth L. (1970), "Using Context Clues in Word Recognition and Comprehension", *The Reading Teacher* 23(5):440-445, 469.

Michael J. Almeida. [Time in narratives](#). In Judith F. Duchan, Gail A. Bruder, and Lynne E. Hewitt, editors, *Deixis in Narrative: A Cognitive Science Perspective*, pages 159-189. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1995.

Del Vecchio, Justin M. (2002b), "A Contextual Vocabulary Acquisition Algorithm for Verbs, Implemented in SNePS".

J. Barwise and R. Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4, 1980.