Journal of Mathematical Sciences 2 (2015) 185-189
ISSN 2372-5214
Betty Jones & Sisters Publishing
http://www.bettyjonespub.com

Contrastings on Textual Entailmentness and Algorithms of Syllogistic Logics

Selçuk TOPAL[1], Tahsin ÖNER[2]

1. Department of Mathematics, Bitlis Eren University, Bitlis 13000, Turkey

2. *Department of Mathematics, Ege University, İzmir 35040, Turkey*

Abstract: In this paper, we are interested in comparisons two different semantic interpretations of Aristotelian syllogism systems which are Corcoran's and Moss's. These comparisons shall be depended on expressive power and time complexity of the interpretations.

Keywords: Logic of natural languages, algorithms, analysis of algorithms and problem complexity.

1. Introduction

Aristotle presented syllogism notion by helping approach of categories [1] and [2]. Aristotelian Syllogism was begun to evaluate as an issue of formal logic by Łukasiewicz [3]. Corcoran gave a completeness theorem for Aristotelian syllogism as named "Completeness of an ancient logic" [4].The interpretation of syllogism of Corcoran was made reassessment by Andrade and Becerra [5]. Parsons introduced the square of opposition with the intention of illustrating Classical Aristotelian Syllogism [6] and modern one by Westersthal [7]. Some complexity results of syllogistic sentences of English, completeness results of some syllogistic logics and algorithms and completeness results of some relational syllogistic logics were given in order of by [8], [9] and [10].

2. Syllogistic Logic and Algorithms from Corcoran

This language starts with a set of $p, q, \ldots$ plural nouns as $variables$ in set of of $\mathcal{P}$. A sort of Apq, Spq, Npq, $\$pq$ as modern usage in order of $All\ p's\ are\ q's$, $Some\ p's\ are\ q's$, $No\ p's\ are\ q's$, $Some\ p's\ are\ not\ q's$. The semantics is built on $finite$ set of sentences and $finite$ set U. For all $p \in \mathcal{P}$, $[[p]] \subseteq U$ where $[[\,]]$ is an $interpretation\ function$ from $\mathcal{P}$ to subsets of U. A model $\mathcal{M} = (U, [[\,]], \mathcal{P})$ has the following truth properties between model and sentences:

$$\mathcal{M} \vDash All\ p\ are\ q :\Leftrightarrow [[p]] \subseteq [[q]]$$
$$\mathcal{M} \vDash Some\ are\ q :\Leftrightarrow [[p]] \cap [[q]] \neq \emptyset$$
$$\mathcal{M} \vDash No\ p\ are\ q :\Leftrightarrow [[p]] \cap [[q]] = \emptyset$$
$$\mathcal{M} \vDash Some\ p\ are\ not\ q :\Leftrightarrow [[p]] \nsubseteq [[q]]$$

We shall use CSL instead of "Corcoran's syllogistic logic" for abbreviation during this article.

Table 1 Rules for CSL.

$\dfrac{All\ p\ are\ q}{Some\ p\ are\ q}\ \star$	
$\dfrac{Some\ p\ are\ q}{Some\ q\ are\ p}\ \spadesuit$	$\dfrac{No\ p\ are\ q}{No\ q\ are\ p}$
$\dfrac{All\ q\ are\ n\quad Some\ p\ are\ q}{Some\ p\ are\ n}\ \clubsuit$	$\dfrac{All\ p\ are\ n\quad No\ n\ are\ q}{No\ p\ are\ q}$
$\dfrac{No\ m\ are\ p\quad Some\ s\ are\ p}{Some\ m\ are\ not\ s}$	$\dfrac{All\ p\ are\ n\quad All\ n\ are\ q}{All\ p\ are\ q}$

Syllogistic sentences *Every S is P*, *Some S is P*, *Some S is not P* and *No S is P* in Figure 1 are lexically

---

Corresponding author: Selçuk TOPAL, Ph.D., research fields: applied logic and computer science. E-mail: selcuk.topal@yandex.com; s.topal@beu.edu.tr.

equivalent in order of sentences *All S are P*, *Some S are P*, *Some S are not P* and *No S are P*.

2.1. Derivation of All in CSL

Whether semantics of a model accept or does not empty set, *All p are p* is true in every model since every set is subset of itself. It is possible to add $All\, p\, are\, p$ to the set of rules asan axiom.

Definition 2.1. Let Γ be a finite set of All sentences. We say $p \overset{All}{\rightarrow} q :\Leftrightarrow \Gamma \vdash All\, p\, are\, q$. In other saying, there is a path from node p to node q if taking variables as nodes of graph obtained from Γ and $p \overset{All}{\rightarrow} q$ is a directed edge from p to p of the graph [9] .

Definition 2.2. Reachability Problem: Given a graph $G = (V, E)$ and a vertex v in G which other vertices can be reached by a path starting from v? [11]

Theorem 2.3. Reachability problem in the worst case requires $O(|V|^3)$ time complexity [11].

Definition 2.4. Set Element Problem: Given a graph $G = (V, E)$ and a vertex x which x in G or not ?

Theorem 2.5. Set element problem requires $O(|V|)$ in the worst case.

Proof 2.5. It is easy to see that checking at one of all elements in $|V|$ is equal to x requires maximum $|V|$ steps.

Observation 2.6. Let Γ be a set of sentences and φ be a sentence in CSL. Then Checking whether φ is in Γ requires maximum $O(|\Gamma|)$ from theorem 2.5.

Observation 2.7. Let Γ be a set of sentences and be $\mathcal{P}$ set of variables of Γ in CSL. Then Checking $\Gamma \vdash Allpareq$ requires maximum $O(|\mathcal{P}|^3)$ from theorem 2.3.

Remark 2.8. All following algorithm steps shall be given with its time complexity.

Algorithm 2.9. Let Γ be a set of sentences in CSL. The following steps are controlled whether $\Gamma \vdash All\, p\, are\, q$:

1) If $p = q$ then $\Gamma \vdash All\, p\, are\, q$ (always), $O(n)$;
2) $All\, p\, are\, q \in \Gamma$, $O(n)$;
3) $p \overset{All}{\rightarrow} q$ (see definition 2.1), $O(n^3)$;
4) Inconsistent Γ, $O(25n) + O(21n^3)$.

2.2. Derivation of Some in CSL

There are two kinds of forms for $Some$ sentences. They are *Some p are q* and *Some p are not q*. It is clear to see that there is no noun p such that $[[p]] = \emptyset$ in the system since the rule $\star$. In the other words, for all $p \in \mathcal{P}$, *Some p are p*. This means that there could be an axiom of $\overline{Some\ p\ are\ p}$ or it is possible to let same variable to substitute to all variables in rules $\star$, $\spadesuit$ and uit in Table 1.

Algorithm 2.10. Let Γ be a set of sentences in CSL. The following steps are controlled whether $\Gamma \vdash Some\, p\, are\, q$:

1) If $p = q$ *then* $\Gamma \vdash Some\, p\, are\, q$ *(always),* $O(n)$;
2) $Some\, p\, are\, q \in \Gamma$ *or* $Some\, q\, are\, p \in \Gamma$, $O(2n)$;
3) $All\, p\, are\, q \in \Gamma$ *or* $\Gamma \vdash All\, p\, are\, q$, $O(n^3)$;
4) $All\, q\, are\, p \in \Gamma$ *or* $\Gamma \vdash All\, q\, are\, p$, $O(n^3)$;
5) { $All\, n\, are\, p \in \Gamma$ *or* $\Gamma \vdash All\, n\, are$ } *and* { $\Gamma \vdash All\, m\, are\, q$ *or* $All\, m\, are\, q \in \Gamma$ } *and* { $Some\, m\, are\, n \in \Gamma$ *or* $Some\, n\, are\, m \in \Gamma$ }, $O(2n^3) + O(2n^3) + O(2n)$;
6) Inconsistent Γ, $O(25n) + O(21n^3)$.

Algorithm 2.11. Let Γ be a set of sentences in CSL. The following steps are controlled whether $\Gamma \vdash Some\, p\, are\, not\, q$:

1) 1) If $p = q$ then Γ is inconsistent, $O(n)$;
2) 2) $Some\, p\, are\, not\, q \in \Gamma$, $O(n)$;
3) Check Algorithm 2.9 for $\Gamma \vdash Nomarep$ and Algorithm 2.9 for $\Gamma \vdash Somesarep$ except inconsistent Γ steps, $O(4n^3) + O(5n) + O(6n^3) + O(5n)$;
4) Inconsistent Γ, $O(25n) + O(21n^3)$.

2.3. Derivation of No in CSL

Although $No\ p\ are\ p$ is convenient for using in syntax of the language, it impossible to derive it from a consistent set of premises. If it is in a set Γ of premises, then Γ is inconsistent automatically since there is no $[[q]] = \emptyset$ that is q in CSL.

Algorithm 2.12. Let Γ be a set of sentences in CSL. The following steps are controlled whether $\Gamma \vdash No\ p\ are\ q$:

1) If $p = q$ *then* Γ *is inconsistent,* $O(n)$;
2) $No\ p\ are\ q \in \Gamma$ *or* $No\ q\ are\ p \in \Gamma$, $O(2n)$;
3) $\{All\ p\ are\ n \in \Gamma\ or\ \Gamma \vdash All\ p\ are\ n\}$ *and* $\{\Gamma \vdash All\ q\ are\ m\ or\ All\ q\ are\ m \in \Gamma\}$ *and* $\{No\ m\ are\ n \in \Gamma\ or\ No\ n\ are\ m \in \Gamma\}$, $O(2n^3) + O(2n^3) + O(2n)$;
4) Inconsistent Γ, $O(25n) + O(21n^3)$.

2.4 Derivation of Inconsistency in CSL

Definition 2.13. A set Γ is *inconsistent* if $\Gamma \vdash \varphi$ for all φ. Otherwise, Γ is *consistent* [9].

Algorithm 2.14. Let Γ be a set of sentences in CSL. The following steps are controlled whether $\Gamma \vdash All\ p\ are\ q$ and $\Gamma \vdash Some\ p\ are\ not\ q$:

For $\Gamma \vdash Some\ p\ are\ not\ q$:

1) If $p = q$ then Γ inconsistent and $\Gamma \vdash Someparenotq$ and $\Gamma \vdash Allpareq$, $O(n)$,
2) Check Algorithm 2.8 for $\Gamma \vdash Someparenotq$ except inconsistent Γ steps, $O(12n) + O(10n^3)$,
3) Skip other possible inconsistencies for avoiding infinite loops.

For $\Gamma \vdash Allpareq$:

1) Check Algorithm 2.6 for $\Gamma \vdash Allpareq$ except inconsistent Γ steps, $O(2n) + O(n^3)$,
2) Skip other possible inconsistencies for avoiding infinite loops.
3) Skip other possible inconsistencies for avoiding infinite loops.

Algorithm 2.15. Let Γ be a set of sentences in CSL. The following steps are controlled whether $\Gamma \vdash No\ p\ are\ q$ and $\Gamma \vdash Some\ p\ are\ q$:

1) Check Algorithm 2.7 for $\Gamma \vdash Somepareq$ and Algorithm 2.9 for $\Gamma \vdash Nopareq$ except inconsistent Γ steps, $O(5n) + O(6n^3) + O(5n) + O(4n^3)$,
2) Skip other possible inconsistencies for avoiding infinite loops.

3 S Logic and Algorithms from Moss

S language begins by set $\mathcal{P}$ with $p, q, r, \ldots$ variables (plural nouns) and a finite universe M. For all $p \in \mathcal{P}$, $[[p]] \subseteq M$ where $[[\ ]]$ is an $interpretation\ function$ from $\mathcal{P}$ to subsets of M. A model $\mathcal{M} = (M, [[\ ]], \mathcal{P})$ has the following truth properties between model and sentences:

$$\mathcal{M} \vDash All\ p\ are\ q :\Leftrightarrow [[p]] \subseteq [[q]]$$
$$\mathcal{M} \vDash Some\ are\ q :\Leftrightarrow [[p]] \cap [[q]] \neq \emptyset$$
$$\mathcal{M} \vDash No\ p\ are\ q :\Leftrightarrow [[p]] \cap [[q]] = \emptyset$$

Table 2 Rules for S.

$\overline{All\ x\ are\ x}$	
$\dfrac{No\ x\ are\ x}{All\ x\ are\ y}$ ♦	$\dfrac{All\ x\ are\ y \quad All\ y\ are\ z}{All\ x\ are\ z}$
$\dfrac{No\ x\ are\ y}{No\ y\ are\ x}$	$\dfrac{All\ x\ are\ y \quad No\ y\ are\ z}{No\ x\ are\ z}$
$\dfrac{Some\ x\ are\ y}{Some\ y\ are\ x}$	$\dfrac{All\ x\ are\ y \quad Some\ x\ are\ z}{Some\ y\ are\ z}$
$\dfrac{Some\ x\ are\ y \quad No\ x\ are\ y}{C}\ \mathcal{X}$	

We shall use MSL instead of "Moss's syllogistic logic" for abbreviation during this article.

3.1 Derivation of All in MSL

Whether semantics of a model accept or does not empty set, *All p are p* is true in every model since every set is subset of itself. It is possible to add $All\ p\ are\ p$ to the set of rules asan axiom.

Algorithm 3.1. Let Γ be a set of sentences in MSL. The following steps are controlled whether $\Gamma \vdash All\ p\ are\ q$:

1) If $p = q$ then $\Gamma \vdash All\ p\ are\ q$ always (axiom), $O(n)$,
2) $All\ p\ are\ q \in \Gamma$, $O(n)$,
3) $p \overset{All}{\rightarrow} q$ (see Definition 2.1), $O(n^3)$,
4) Check Algorithm 3.3 for $\Gamma \vdash Noparep$, $O(2n) + O(2n^3)$;
5) Inconsistent Γ, $O(4n) + O(4n^3)$.

3.2 Derivation of Some in MSL

Algorithm 3.2. Let Γ be a set of sentences in MSL. The following steps are controlled whether $\Gamma \vdash Some\ p\ are\ q$:

1) $Some\ p\ are\ q \in \Gamma$ or $Some\ q\ are\ p \in \Gamma$, $O(2n)$;
2) { $All\ n\ are\ p \in \Gamma$ or $\Gamma \vdash All\ n\ are\ p$ } and { $\Gamma \vdash All\ m\ are\ q$ or $All\ m\ are\ q \in \Gamma$ } and { $Some\ m\ are\ n \in \Gamma$ or $Some\ n\ are\ m \in \Gamma$ } [10], $O(n^3) + O(n^3) + O(2n)$;
3) Inconsistent Γ, $O(4n) + O(4n^3)$.

3.3 Derivation of No in MSL

Algorithm 3.3. Let Γ be a set of sentences in MSL. The following steps are controlled whether $\Gamma \vdash No\ p\ are\ q$:

1) $No\ p\ are\ q \in \Gamma$ or $No\ q\ are\ p \in \Gamma$, $O(2n)$,
2) { $All\ p\ are\ n \in \Gamma$ or $\Gamma \vdash All\ p\ are\ n$ } and { $\Gamma \vdash All\ q\ are\ m$ or $All\ q\ are\ m \in \Gamma$ } and { $No\ m\ are\ n \in \Gamma$ or $No\ n\ are\ m \in \Gamma$ } [10], $O(n^3) + O(n^3) + O(2n)$,
3) Inconsistent Γ, $O(4n) + O(4n^3)$.

3.4 Derivation of Inconsistency in MSL

Algorithm 3.4. Let Γ be a set of sentences in MSL. The following steps are controlled whether $\Gamma \vdash Some\ p\ are\ q$ and $\Gamma \vdash No\ p\ are\ q$:

1) Check Algorithm 3.2 for $\Gamma \vdash Some\ p\ are\ q$ and Algorithm 3.3 for $\Gamma \vdash No\ p\ are\ q$ except inconsistent Γ steps and $\Gamma \vdash Noparep$, $O(4n) + O(4n^3)$.
2) Skip other possible inconsistencies for avoiding infinite loops.

4. Conclusion

Table 3 gives comparisons of time complexity of derivations in two logics. It is pretty clear to see that algorithms of Moss's logic is more efficient than Corcoran's logic.

Some p are not q is not expressible in the language of MSL. On the other hand, although *No p are p* gives rise to inconsistency in CLS, this sentence increases possibilities derivations of variety and numbers of *All* sentences in MSL since the rule ♦ in Table 2.

Table 3 Time complexity comparisons ($\mathbf{n} = |\mathcal{P}|$).

Sentence φ	φ in MSL	φ in CSL	Time Complexity of $\Gamma_{MSL} \vdash \varphi$	Time Complexity of $\Gamma_{CSL} \vdash \varphi$
All p are q	✓	✓	$O(7n^3) + O(8n)$	$O(22n^3) + O(27n)$
Some p are q	✓	✓	$O(6n^3) + O(8n)$	$O(27n^3) + O(30n)$
No p are q	✓	✓	$O(6n^3) + O(8n)$	$O(25n^3) + O(30n)$
Some p are are not q	×	✓	×	$O(25n^3) + O(39n)$

References

[1] Aristotle, Categories, in: J. Barnes (Ed.), The Complete Works of Aristotle, Princeton University Press, 1984, vol. 1, p. 1256.

[2] R. Smith, Categories, Aristotle's Prior Analytics, Hackett Publishing Company, 1989, p. 320.

[3] J. Łukasiewicz, Aristotle's Syllogistic from the Standpoint of Modern Formal Logic, Clarendon Press, Oxford, 1951.

[4] J. Corcoran, Completeness of an ancient logic, Journal of Symbolic Logic 37 (1972) 696-702.

[5] E.J. Andrade, E. Becerra, Corcoran's Aristotelian syllogistic as a subsystem of first-order logic, Revista Colombiana de Matemáticas 41 (1) (2007) 67-80.

[6] T. Parsons, Things that are right with the traditional square of opposition, Logica Universalis 2 (1) (2008) 3-11.

[7] D. Westersthå˚J Classical vs. modern squares of opposition, and beyond, in: New Perspectives on the Square of Opposition, Proceedings of the First International Conference on the Square of Oppositions, Montreux, Switzerland, 2007, pp. 1-3.

[8] I. Pratt-Hartmann, Fragments of language, Journal of Logic, Language and Information 13 (2) (2004) 207-223.

[9] L.S. Moss, Completeness theorems for syllogistic fragments, in: F. Hamm and S. Kepser (Eds.), Logics for linguistic structures, Walter de Gruyter 201 (2008) 143-175.

[10] I. Pratt-Hartmann, L.S. Moss, Logics for the relational syllogistic, The Review of Symbolic Logic 2 (4) (2009) 647-683.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, The MIT Press and McGraw-Hill Book Company, 1989, p. 597.