An Extended Syllogistic Logic for Automated Reasoning

Ersin Çine
Department of Computer Engineering
İzmir Institute of Technology
İzmir, Turkey
ersincine@gmail.com

Bora İ. Kumova
Institute for Web Science and Technologies
University of Koblenz and Landau
Koblenz, Germany
borakumova@gmail.com

***Abstract*—In this work, we generalise the categorical syllogistic logic in several dimensions to a relatively expressive logic that is sufficiently powerful to encompass a wider range of linguistic semantics. The generalisation is necessary in order to eliminate the existential ambiguity of the quantifiers and to increase expressiveness, practicality, and adaptivity of the syllogisms. The extended semantics is expressed in an extended syntax such that an algorithmic solution of the extended syllogisms can be processed. Our algorithmic approach for deduction in this logic allows for automated reasoning directly with quantified propositions, without reduction of quantifiers.**

***Keywords*—Automated deduction, automated reasoning, knowledge representation, syllogistic reasoning.**

I. Introduction

A categorical syllogism is a logic schema for reasoning with quantified propositions. It consists of two premising propositions that have a sort of transitive relationship, out of which a concluding proposition can be drawn. The advantage over modern logics is that syllogisms allow reasoning directly with the quantifiers, without reducing them.

Although Aristotle's syllogisms have been superseded by modern logics [6], syllogistic reasoning is still an active research topic in cognitive sciences that intersect with philosophy [14], linguistics [9] and psychology [5], [8], [4]. As a result of these extensive analyses of the syllogisms, computational models that use syllogisms as a primary logic of human reasoning have been developed recently. For instance, a reasoner has been modeled based on experimental data that was collected from test persons for 64 syllogistic moods [10]. An algorithmic calculation of validity ratios for the classical syllogistic system of 256 syllogistic forms which can be used as a reasoner over ontologies [11] has been proposed [12].

In categorical syllogisms, propositions consist of a quantifier, subject, and predicate. There are four kinds of propositions:

1) $\mathtt{A(X,Y)}$: "All X are Y"
2) $\mathtt{E(X,Y)}$: "No X are Y"
3) $\mathtt{I(X,Y)}$: "Some X are Y"
4) $\mathtt{O(X,Y)}$: "Some X are not Y"

Following propositions draw a sample syllogistic form: "some students are not successful"; "all students are smart"; "some smart are not successful" where the last proposition is concluded from the previous two. This syllogistic form is known as $\mathtt{OAO}-3$. It is structurally valid. However, the semantics of any loaded example is usually arguable. For instance, "all students are smart" is certainly arguable. This syllogistic form is a complicated syllogism to follow for humans, as they can hardly reason by considering all possible syllogistic cases of inference.

The logic of a syllogistic form is represented set-theoretically with the syllogistic cases of the form. Every one of the 256 forms has a unique combination of true or false syllogistic cases [12]. Consequently, a deduction with a form can be calculated based on its syllogistic cases. However, this combination can change for every form and ultimately change its validity depending on which quantifier possibilities were set in the calculation. Previous calculations used only those possibility configurations that resulted in the well known 24 valid forms or 25 or even only 11 valid syllogisms [12], [11], [17]. Based on various configurations of quantifier possibilities that are discussed in the literature, further different sets of valid syllogisms are presented, for instance 15 and 24 [15], 19 [16], 22 [7], 14 [3].

A unifying model for quantifiers that covers all possibilities would allow calculating them as the ambiguities of an automated syllogistic deducer. Besides decidable ambiguity, further properties should be considered in the design of an automated syllogistic deducer, such as expressiveness, practicality and adaptability. Expressiveness can be increased by allowing for complex terms as discussed in [13], [1], [2]. Practicality can be increased by allowing more than just two premises, which is discussed in the literature as polysyllogisms. Adaptivity can be provided by allowing asserting individuals.

II. An Extended Syllogistic Logic

We generalise traditional categorical syllogisms in several dimensions. Extended syllogistic logic covers existential configurations, arbitrary number of premises, set-theoretical complex terms, and individuality assertions.

1) Existential Configurations: In traditional syllogisms, it is not clear whether $\mathtt{A}(X,Y)$ implies $\mathtt{I}(X,Y)$ or not. The implication depends on the personal interpretation of the

978-1-5386-0930-9/17/$31.00 © 2017 IEEE

978-1-5386-0930-9/17/$31.00 ©2017 IEEE

quantifiers. In extended syllogisms, existential configuration of the quantifiers is used to solve the "existential import" problem, e.g. $\mathtt{A}_{\emptyset}(X,Y) \not\models \mathtt{I}_{\{\upsilon\}}(X,Y)$ but $\mathtt{A}_{\{\upsilon\}}(X,Y) \models \mathtt{I}_{\{\upsilon\}}(X,Y)$.

A quantifier is configured using an index which is a subset of $\{\upsilon, \kappa\}$. For example, $\mathtt{A}_{\emptyset}$, $\mathtt{A}_{\{\upsilon\}}$ and $\mathtt{O}_{\{\upsilon,\kappa\}}$ are configured quantifiers. Here, υ indicates the existence of the subject term and κ indicates the existence of the predicate term. For example, both X and Y may be non-existing terms even if $\mathtt{A}_{\emptyset}(X,Y)$ holds, but X is an existing term if $\mathtt{A}_{\{\upsilon\}}(X,Y)$ holds.

A 4-tuple $\mathtt{config} \in \{(\mathtt{a},\mathtt{e},\mathtt{i},\mathtt{o}) \mid \mathtt{a},\mathtt{e},\mathtt{i},\mathtt{o} \subseteq \{\kappa,\upsilon\}\}$ is a configuration that describes the existential indices for the quantifiers A, E, I, O respectively. When a configuration is fixed, a quantifier without existential index can be used as they will no longer be ambiguous. Our default configuration is $(\emptyset, \emptyset, \{\upsilon\}, \{\upsilon\})$ as in predicate logic (i.e. we will treat A, E, I, O as $\mathtt{A}_{\emptyset}$, $\mathtt{E}_{\emptyset}$, $\mathtt{I}_{\{\upsilon\}}$, $\mathtt{O}_{\{\upsilon\}}$ respectively).

2) Arbitrary Number of Premises: In traditional syllogisms, typically 2 premises are used. As in polysyllogisms, in extended syllogisms, any (natural) number of premises can be used. For example, $\models \mathtt{A}(X,X)$ is an extended syllogism with 0 premise, and $\mathtt{A}(Z,T), \mathtt{A}(Y,Z), \mathtt{A}(X,Y) \models \mathtt{A}(X,T)$ is an extended syllogism with 3 premises.

3) Complex Terms: In traditional syllogisms, all terms are atomic. For example, an atomic term can be $\mathit{red\ things}$, $\mathit{expensive\ things}$, cars or $\mathit{motorcycles}$. An atomic term can also be a complex-looking term such as $\mathit{red\ cars}$. However, it is not possible to conclude $\mathtt{I}(\mathit{car}, \mathit{expensive})$ or $\mathtt{I}(\mathit{car}, \mathit{red})$ from $\mathtt{I}(\mathit{red\ car}, \mathit{expensive})$ if $\mathit{red\ car}$ is an atomic term.

In extended syllogisms, new terms are produced from the atomic terms using a production rule. For example, a complex term can be $\mathit{expensive} \sqcap \mathit{motorcycle}$ ("expensive motorcycles"), $\neg \mathit{expensive} \sqcap \mathit{red} \sqcap \mathit{car}$ ("non-expensive red cars"), or $\mathit{car} \sqcup \mathit{motorcycle}$ ("cars and motorcycles"). It is possible to conclude $\mathtt{I}(\mathit{car}, \mathit{expensive})$ and $\mathtt{I}(\mathit{car}, \mathit{red})$ from $\mathtt{I}(\mathit{red} \sqcap \mathit{car}, \mathit{expensive})$.

Complex terms implicitly covers indefinite terms (such as $\neg \mathit{cat}$ which means "non-cats" or everything that is not a cat), indirect syllogisms (i.e. subject and predicate terms of conclusion are switched as in $\mathtt{A}(Y,Z), \mathtt{I}(X,Y) \models \mathtt{I}(Z,X)$), and non-standard figures (such as the figure in $\mathtt{A}(X,Z), \mathtt{I}(Y,X) \models \mathtt{I}(X,Z)$ which contains the middle term in the conclusion).

4) Individuals: Propositions about individuals do not fit into the standard notation for propositions of traditional syllogisms (e.g. "Socrates is human"). In extended syllogisms, individuals are treated as singleton sets in order to prevent quantification problems.

Asserting individuality of a term can change the validity of a syllogism. For example $\mathtt{I}(\mathit{Socrates}, \mathit{human}) \not\models \mathtt{A}(\mathit{Socrates}, \mathit{human})$ and $\mathtt{I}(\mathit{Socrates}, \mathit{human}), \mathtt{E}(\mathit{human}, \mathit{car}) \not\models \mathtt{E}(\mathit{Socrates}, \mathit{car})$. However, if it is known that there is only one $\mathit{Socrates}$, we can conclude $\mathtt{A}(\mathit{Socrates}, \mathit{human})$ and $\mathtt{E}(\mathit{Socrates}, \mathit{car})$.

A. Syntax

Atomic terms are simple term names such as cat (represents the set of cats in our domain), smart (represents the set of smart things in our domain), or $\mathit{Socrates}$ (represents the singleton set that contains the person Socrates).

1) Complex Terms: Let X and Y be complex terms and a be an atomic term. The production rule for complex terms is defined as:

$$X, Y ::= a \mid \top \mid \bot \mid \delta \mid \neg X \mid X \sqcap Y \mid X \sqcup Y$$

For example, $\neg \mathit{smart} \sqcap \mathit{cat}$ is a complex term assuming smart and cat are also complex (or atomic) terms.

2) Quantified Propositions: Let X and Y be complex terms, $Q \in \{\mathtt{A},\mathtt{E},\mathtt{I},\mathtt{O}\}$ be a quantifier, and $\Upsilon \in \{\emptyset, \{\upsilon\}, \{\kappa\}, \{\upsilon,\kappa\}\}$ be an existential index. A quantified proposition P is defined as:

$$P ::= Q_{\Upsilon}(X,Y)$$

For example, $\mathtt{I}_{\{\upsilon\}}(\neg \mathit{expensive} \sqcap (\mathit{car} \sqcup \mathit{motorcycle}), \top)$ is a quantified proposition assuming $\mathit{expensive}$, car and $\mathit{motorcycle}$ are complex terms.

3) Individuality Assertions: Let X be a complex term. An individuality assertion A is defined as:

$$A ::= \mathtt{IND}(X)$$

For example, $\mathtt{IND}(\mathit{Socrates})$ is an individuality assertion assuming $\mathit{Socrates}$ is a complex term.

4) Propositions: A proposition is either a quantified proposition or an individuality assertion.

B. Semantics

Let T_A be the set of atomic terms. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a set of individuals $\Delta^{\mathcal{I}}$ called the domain and an interpretation function $\cdot^{\mathcal{I}}$ that maps a complex term X to a set $X^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ such that

1) $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$,
2) $\bot^{\mathcal{I}} = \emptyset$,
3) $(\neg X)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus X^{\mathcal{I}}$,
4) $(X \sqcap Y)^{\mathcal{I}} = X^{\mathcal{I}} \cap Y^{\mathcal{I}}$,
5) $(X \sqcup Y)^{\mathcal{I}} = X^{\mathcal{I}} \cup Y^{\mathcal{I}}$,
6) $\delta^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus (\bigsqcup_{X \in T_A} X)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \bigcup_{X \in T_A} X^{\mathcal{I}}$.

A non-existing term corresponds to an empty set and an existing term corresponds to a non-empty set.

Table I shows all possible types of propositions and their corresponding conditions. A proposition holds if and only if its corresponding condition is met. For example, $\mathtt{A}_{\emptyset}(X,Y)$ holds if and only if $X^{\mathcal{I}} \setminus Y^{\mathcal{I}} = \emptyset$ but $\mathtt{A}_{\{\upsilon\}}(X,Y)$ holds if and only if $X^{\mathcal{I}} \setminus Y^{\mathcal{I}} = \emptyset$ and $X^{\mathcal{I}} \neq \emptyset$.

$\mathtt{A}_{\{\upsilon\}}(X,\top)$ can be used to claim that X is an existing term and $\mathtt{A}_{\emptyset}(X,\bot)$ can be used to claim that X is a non-existing term. X and Y are equal terms if and only if $\mathtt{A}_{\emptyset}((X \sqcap \neg Y) \sqcup (Y \sqcap \neg X), \bot)$ holds. X and Y are unequal terms if and only if $\mathtt{A}_{\{\upsilon\}}((X \sqcap \neg Y) \sqcup (Y \sqcap \neg X), \top)$ holds. X and Y are disjoint terms if and only if $\mathtt{E}_{\emptyset}(X,Y)$ holds. X and Y are intersecting terms if and only if $\mathtt{I}_{\{\upsilon\}}(X,Y)$ holds.

TABLE I
PROPOSITIONS AND CORRESPONDING CONDITIONS

Proposition	Condition	
$\mathcal{I} \vDash \mathtt{A}_{\Upsilon}(X,Y)$	$X^{\mathcal{I}} \setminus Y^{\mathcal{I}} = \emptyset$	$\wedge\ (\upsilon \in \Upsilon \Rightarrow X^{\mathcal{I}} \neq \emptyset)$ $\wedge\ (\kappa \in \Upsilon \Rightarrow Y^{\mathcal{I}} \neq \emptyset)$
$\mathcal{I} \vDash \mathtt{E}_{\Upsilon}(X,Y)$	$X^{\mathcal{I}} \cap Y^{\mathcal{I}} = \emptyset$	
$\mathcal{I} \vDash \mathtt{I}_{\Upsilon}(X,Y)$	$X^{\mathcal{I}} \neq \emptyset \Rightarrow X^{\mathcal{I}} \cap Y^{\mathcal{I}} \neq \emptyset$	
$\mathcal{I} \vDash \mathtt{O}_{\Upsilon}(X,Y)$	$X^{\mathcal{I}} \neq \emptyset \Rightarrow X^{\mathcal{I}} \setminus Y^{\mathcal{I}} \neq \emptyset$	
$\mathcal{I} \vDash \mathtt{IND}(X)$	$\lvert X^{\mathcal{I}}\rvert = 1$	

III. ALGORITHMS

A. Terms

Algorithm 1 calculates an array of set of region numbers for a given number of atomic terms. For example, if we call the function as in `T = AtomicTerms(3)`, then $\mathtt{T}_1 = \{1,3,5,7\}$, $\mathtt{T}_2 = \{2,3,6,7\}$, $\mathtt{T}_3 = \{4,5,6,7\}$, and $\mathtt{T.length} = 3$. Hereafter, `t` will represent the number of atomic terms.

Algorithm 1 Calculation of region sets

```
1: function ATOMICTERMS(t)    ▷ Number of atomic terms
2:     for i ← 1, t do
3:         T(i) ← { Σ_{x∈X∪{i}} 2^{x−1} | X ⊆ {1, 2, ..., t} }
4:     return T
```

The intuition behind this function is simple: For every $\mathtt{T_i}$, there is a dedicated number $2^{\mathtt{i}-1}$. For example, $2^{3-1} = 4$ is dedicated to $\mathtt{T}_3$. Any region is numbered the sum of the dedicated numbers of those atomic terms that contains this region. For example, when $\mathtt{t} = 3$, $(\mathtt{T}_1 \cap \mathtt{T}_3) \setminus \mathtt{T}_2$ defines a single region that is contained by only $\mathtt{T}_1$ and $\mathtt{T}_3$. Therefore, that region will be numbered $2^{1-1} + 2^{3-1} = 1 + 4 = 5$.

Figure 1 shows a visualisation of atomic terms and region numbers. It is useful to replace terms with meaningful identifiers (e.g. $\mathtt{human} = \mathtt{T}_1$).

The bottom term $\bot$ defines a term that represents the empty set. $\bot$ is defined as a global constant: $\mathtt{VOID} = \{\ \}$

The top term $\top$ defines a term that represents all the domain. $\top$ is defined as a global variable: $\mathtt{UNIVERSE} = \{0, 1, ..., 2^{\mathtt{t}} - 1\}$

Value of `UNIVERSE` does not change unless the value of t is changed. If the value of `t` is changed, then the atomic terms should be also updated.

The surplus term δ defines a term that represents everything in the domain other than the atomic terms. δ is defined as a global constant: $\mathtt{EXTRAS} = \{0\}$

Atomic terms and the surplus term are the building blocks of complex terms. The production of complex terms is done by the set operations such as intersection, union, difference, and complement. For example, if $\mathtt{t} = 3$ then $\mathtt{T}_1 \cup \overline{\mathtt{T}_2} = \{0,1,3,4,5,7\}$. It is true that $\mathtt{UNIVERSE} = \bigcup_{\mathtt{i}=1}^{\mathtt{t}} \mathtt{T_i} \cup \mathtt{EXTRAS}$. An indefinite term is a basic example of complex terms that is produced by taking complement of an atomic term with respect to `UNIVERSE`. Each complex term is a subset of `UNIVERSE`. Therefore, there can be only $\lvert\mathcal{P}(\mathtt{UNIVERSE})\rvert = 2^{2^{\mathtt{t}}}$ different complex terms.

For all atomic terms $\mathtt{T_i}$, $\lvert\mathtt{T_i}\rvert = 2^{\mathtt{t}-1}$. For all complex terms $\mathtt{C}$, $0 \leq \lvert\mathtt{C}\rvert \leq 2^{\mathtt{t}}$. For example, $\lvert\mathtt{UNIVERSE}\rvert = 2^{\mathtt{t}}$, $\lvert\mathtt{VOID}\rvert = 0$, $\lvert\mathtt{EXTRAS}\rvert = 1$. Furthermore, $\lvert\overline{\mathtt{T_i}}\rvert = 2^{\mathtt{t}} - 1$, $\lvert\mathtt{T_i} \cap \mathtt{T_j}\rvert = 2^{\mathtt{t}-2}$, $\lvert\mathtt{T_i} \cup \mathtt{T_j}\rvert = 2^{\mathtt{t}} - 2^{\mathtt{t}-2}$ where $\mathtt{T_i}$ and $\mathtt{T_j}$ are atomic terms and $\mathtt{T_i} \neq \mathtt{T_j}$.

B. Propositions

A quantified proposition is an object with 5 properties. It consists of a quantifier, two boolean properties called `subjectShouldExist` and `predicateShouldExist` which are determined by the configuration of the quantifier and two complex terms which are called `subject` and `predicate`.

An individuality assertion consists of a complex term.

A world is a subset of `UNIVERSE`. It contains the numbers of all the existing regions in `UNIVERSE`. Figure 2 shows a visualisation of a possible world.

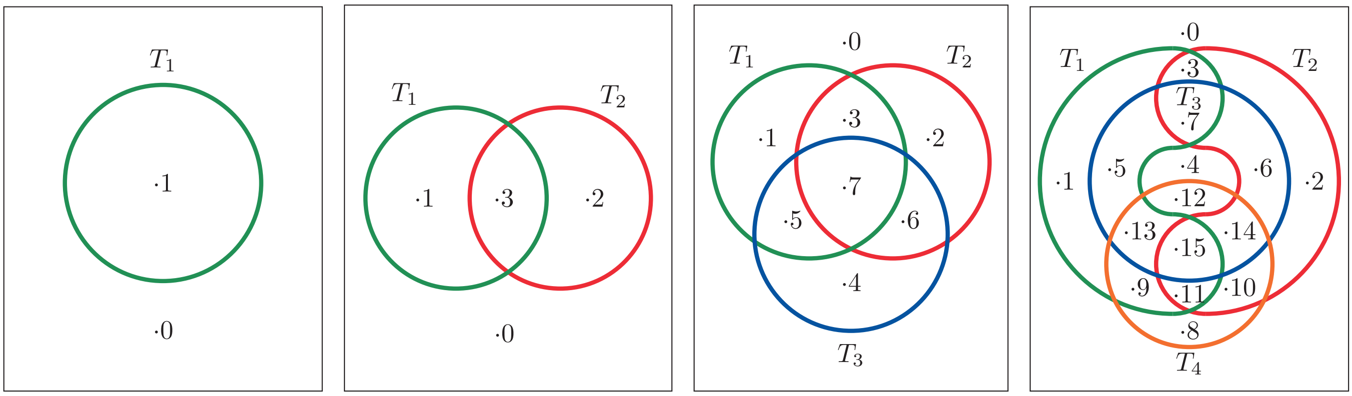
Algorithm 2 decides whether a proposition is true or not in a world. All the quantifiers are only related to existence or non-existence of the regions. When the proposition is a quantified proposition, if a region exists, it does not matter how many corresponding elements there are. However, single existing region does not imply single corresponding element in the interpretation. As a result, any model can be represented by a world from syllogistic point of view if and only if the conclusion is a quantified proposition. Therefore, the first part of the algorithm is unsound. However, this will not be a problem as the deductive machinery is designed accordingly. When the conclusion is a quantified proposition, the existing regions of the subject and the predicate are the intersections of those terms with the world. If there is an existential error, the proposition is false. Otherwise, the truth value is decided according to the quantifier of the proposition.

C. Satisfiability and Validity Checking

Algorithm 3 decides whether a set of propositions is consistent. A set of propositions is consistent if and only if at least one possible world satisfies all the propositions.

Algorithm 4 decides whether an extended syllogism is valid, i.e. conclusion necessarily follows from the premises. This algorithm takes a possibly empty set of premises and a conclusion. Each of the premises and the conclusion is a proposition.

The first part of the algorithm 4 searches an upper bound 1 for the number of corresponding elements in the interpretation of the regions of interest. If it finds an upper bound, it searches a lower bound 1, i.e. existence of the regions of interest. Otherwise, the validity requires an inconsistency among the premises. The correctness of the second part of the algorithm 4 relies on the iteration of all possible worlds that works like a truth table for propositional logic. A syllogism is invalid if and only if there is at least one world that satisfies all the premises but does not satisfy the conclusion. In the corner case, if no



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 4$

Fig. 1. Universes with different number of atomic terms.

Algorithm 2 Does a given proposition hold in a given world?

```
function HOLDS(p, w)                     ▷ p is a proposition, w is a world
    if type(p) == IND then               ▷ complete & unsound
        return |w ∩ p.term| == 1
    else                                 ▷ complete & sound
        subj ← w ∩ p.subject
        pred ← w ∩ p.predicate
        if p.subjectShouldExist and subj == ∅ then
            return False
        if p.predicateShouldExist and pred == ∅ then
            return False
        if p.quantifier == A then
            return subj \ pred == ∅
        if p.quantifier == E then
            return subj ∩ pred == ∅
        if p.quantifier == I then
            return subj == ∅ or subj ∩ pred ≠ ∅
        if p.quantifier == O then
            return subj == ∅ or subj \ pred ≠ ∅
```

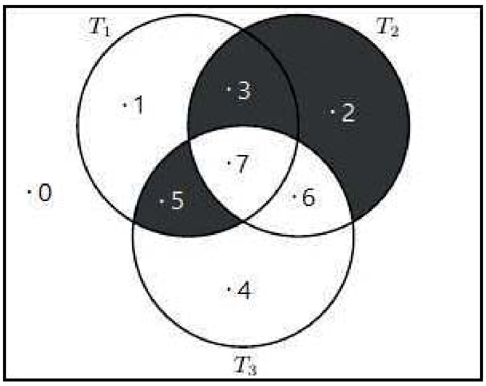


Fig. 2. Visualisation of a possible world $\texttt{w} = \{0, 1, 4, 6, 7\}$ in a universe with 3 atomic terms. Shaded regions corresponds to the empty regions in the interpretation. w satisfies $\texttt{IND}(\texttt{T}_1 \sqcap \texttt{T}_2)$ but does not satisfy $\texttt{IND}(\texttt{T}_1)$. In w, $\texttt{I}_{\{v\}}(\neg T_1, T_2 \sqcap T_3)$ holds but $\texttt{I}_{\{v\}}(T_3 \sqcap \neg T_2, T_1)$ does not hold.

Algorithm 3 Inconsistency detection & satisfiability checking

```
function ISCONSISTENT(propositions)
    loop: for all world ⊆ UNIVERSE do
        for all proposition ∈ propositions do
            if not Holds(proposition, world) then
                continue loop
        return True
    return False
```

Algorithm 4 Validity checking

```
function ISVALID(pre, con)               ▷ premises, conclusion
    if type(con) == IND then
        for all p ∈ pre do
            if type(p) ≠ IND then
                continue
            if Holds(A_∅(con.term, p.term), UNIVERSE) then
                return IsValid(pre, I_{v}(con.term, UNIVERSE))
        return not IsConsistent(pre)
    else
        loop: for all world ⊆ UNIVERSE do
            for all premise ∈ pre do
                if not Holds(premise, world) then
                    continue loop
            if not Holds(con, world) then
                return False
        return True
```

world satisfies all the premises, any conclusion is valid because of the principle of explosion, i.e. from contradiction, anything follows.

An advantage of these algorithms for satisfiability and validity checking is that they are very flexible in terms of responding to needs (i.e. it is possible to add more dimensions beyond our recommendation). For example, other types of constraints can easily be defined.

IV. Discussion

A. Theoretical Properties

Decidability of satisfiability and validity for this extended syllogistic logic was shown via algorithmic solution.

The proof system that was presented is seemingly complete (i.e. every true formula is provable) and sound (i.e. a formula that is provable is true).

B. Implementation

Let $\mathtt{t}$ be the number of atomic terms. Regions are numbered from $\mathtt{0}$ to $\mathtt{2^t - 1}$ inclusively. Any complex term is represented as sets of these region numbers. For efficiency, these sets are implemented as unsigned integers. A set $\{\mathtt{x_1, x_2, ..., x_k}\}$ implemented as the result of the summation $\sum_{\mathtt{i=1}}^{\mathtt{k}} \mathtt{2^{x_i}}$ called a representative number. For example, $\{1, 3, 5, 7\}$ is implemented as $\mathtt{170}$, $\{0\}$ is implemented as $\mathtt{1}$, and $\{\}$ is implemented as $\mathtt{0}$. Using binary numeral system, it can be easily seen that this mapping is injective.

This approach has several consequences:

- Implementation usually becomes easier. For example, $\mathtt{for\ all\ world} \subseteq \mathtt{UNIVERSE}$ becomes $\mathtt{for}\ (\mathtt{world} = 0;\ \mathtt{world} <= \mathtt{UNIVERSE};\ ++\mathtt{world})$ and $\mathtt{VOID} = \emptyset$ becomes $\mathtt{VOID} = 0$.
- A region number $\mathtt{x} \in \{0, 1, ..., 2^{\mathtt{t}} - 1\}$ can be stored using $\mathtt{t}$ bits in a computer memory. Maximum number of regions in a complex term is $|\mathtt{UNIVERSE}| = 2^{\mathtt{t}}$. Therefore, number of bits to store this term is roughly $\mathtt{t} \times 2^{\mathtt{t}}$. However, with the efficient implementation of region sets, the maximum number of bits to store a term is only $\lfloor \log_2 2^{2^{\mathtt{t}}-1} \rfloor + 1 = 2^{\mathtt{t}}$ as the universal term $\mathtt{UNIVERSE} = \{0, 1, ..., 2^{\mathtt{t}} - 1\}$ is implemented as $\sum_{\mathtt{x}=0}^{2^{\mathtt{t}}-1} 2^{\mathtt{x}} = 2^{2^{\mathtt{t}}} - 1$. Space complexity reduces in the worst case. The average case depends on probability distribution over the number of regions in a complex term.
- Efficient bitwise operations replace the set operations that require using a loop over the elements: $\mathtt{bitwise\ AND}$ replaces $\mathtt{intersection}$, $\mathtt{bitwise\ OR}$ replaces $\mathtt{union}$, and $\mathtt{bitwise\ NOT}$ replaces $\mathtt{complement}$. For example, $\mathtt{X} \setminus \mathtt{Y}$ is implemented as $\mathtt{X}\ \&\ \sim \mathtt{Y}$. Nonetheless, when 2^t is greater than word size, multi-word arithmetic is required. Therefore, space complexity of a set and time complexity of a set operation are linear in terms of the number of regions (i.e. $\Theta(2^t)$ where t is the number of atomic terms, and therefore 2^t is the number of regions).

C. Feasibility

The time complexity of checking validity or satisfiability for a single world is $\Theta(2^t \cdot p)$ where $\mathtt{t}$ is the number of atomic terms and $\mathtt{p}$ is the number of premises. This process is done for $2^{2^{\mathtt{t}}}$ worlds independently. Independence of many instances of the process allows us to run the program parallel and distributed. Nonetheless, the algorithms are seemingly not scalable in terms of the number of atomic terms as the number of possible worlds grows very fast. Despite theoretical time complexity, empirical results are promising:

1) invalid syllogisms are very dense in syllogisms that are generated uniformly at random,
2) a world contradicting with the conclusion in a consistent invalid syllogism is usually found at the very early stage of the search.

V. Conclusion

The categorical syllogistic logic was generalised to a more expressive logic. Syntax and semantics of this logic were defined. Sample reasoning algorithms were presented.

Proposed logic and its implementation provides a reasoning tool for cognitive science research.

For several atomic terms, occurrence of superhuman performance in terms of correctness and speed of reasoning is indisputable. Therefore, intelligence amplification is one of the candidate areas for application.

As future work, analysing statistical results of the extended syllogisms is interesting. Extending the generic quantifier model to fuzzy quantifications such as few, most, etc. and numeric quantifications such as “at least 1”, “all but 3”, etc. as well as extending the copula model to other possibly transitive roles such as “be friend of”, “trust”, etc. can also be done. There is a need for a scalable solution in order to make the computations feasible for large number of atomic terms.

Acknowledgment

This research was partially funded by the grant project TÜBİTAK EEEAG-115E761.

References

[1] Alvarez, E; Correia, M; 2012; “Syllogistic with Indefinite Terms”; History and Philosophy of Logic 33
[2] Alvarez-Fontecilla, E; 2016; “Canonical Syllogistic Moods in Traditional Aristotelian Logic”, Logica Universalis 10
[3] Başkent, C; 2008; “Hypothetical Syllogism in Aristotle and Boethius”; canbaskent.net/logic/early/syllogism.pdf
[4] Chater, N; Oaksford, M; 1999; “The probability heuristics model of syllogistic reasoning”; Cognitive Psychology, 38
[5] Dickstein, LS; 1981; “Conversion and possibility in syllogistic reasoning”; Bulletin of the Psychonomic Society, 18
[6] Frege, LGF; 1879; “Begriffsschrift, eine der Arithmetischen Nachgebildete Formalsprache des Reinen Denkens”; Verlag von Louis Nebert
[7] Gabbay, DM; Woods, J; 2011; “Inductive Logic”; Hartmann, S; vol 10 in Handbook of the History of Logic; North Holland
[8] Johnson-Laird, PN; Bara, BG; 1984; “Syllogistic inference”; Cognition 16
[9] Keenan, D; Westerstahl, D; 2011; “Generalized Quantifiers in Linguistics and Logic”; Elsevier; Handbook of Logic and Language
[10] Khemlani, S; Johnson-Laird; PN; 2013; “The processes of inference”; Argument & Computation 4
[11] Kumova, Bİ; 2016; “Fuzzy-Syllogistic Systems: A Generic Model for Approximate Reasoning”; Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE); LNCS; Springer
[12] Kumova, Bİ; Çakır, H; 2010; “Algorithmic Decision of Syllogisms”; Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE’10); Springer LNAI
[13] Moss, L; 2011; “Syllogistic Logic with Complements”; Games, Norms and Reasons; 353 of the series Synthese Library
[14] Mostowki, A; 1957; “On a generalization of quantifiers”; Fundamenta Mathematicae, 44/1
[15] Russinoff, IS; 1999; “The Syllogism’s Final Solution”; The Bulletin of Symbolic Logic 5/4, pp 451-469; Association for Symbolic Logic
[16] Schonbeck, T; 2000; “Facts About Syllogisms”; math.fau.edu/schonbek/mfla/mfla1f01syl.html
[17] Zarechnev, M; Kumova, Bİ; 2015; “Truth Ratios of Syllogistic Moods”; IEEE International Conference on Fuzzy Systems (FUZZ-IEEE); IEEE