System Description : JGXYZ
An ATP System for Gap and Glut Logics

Geoff Sutcliffe[1][0000−0001−9120−3927] and
Francis Jeffry Pelletier[2][0000−0002−3399−2718]

[1] University of Miami, Miami, USA
http://www.cs.miami.edu/~geoff
[2] University of Alberta, Edmonton, Canada
https://sites.ualberta.ca/~francisp

Abstract. This paper describes an ATP system, named JGXYZ, for some gap and glut logics. JGXYZ is based on an equi-provable translation to FOL, followed by use of an existing ATP system for FOL. A key feature of JGXYZ is that the translation to FOL is data-driven, in the sense that it requires only the addition of a new logic's truth tables for the unary and binary connectives in order to produce an ATP system for the logic. Experimental results from JGXYZ illustrate the differences between the logics and translated problems, both technically and in terms of a quasi-real-world use case.

Keywords: Multi-valued logic · Gap logic · Glut logic · ATP system

1 Gap and Glut Logics

Logic "is a subject concerned with the most general laws of truth, and is now generally held to consist of the systematic study of the form of valid inference", and "A valid inference is one where there is a specific relation of logical support between the assumptions of the inference and its conclusion." [21]. Classical first-order logic (FOL), with its truth values **T**rue and **F**alse has "been the logic suggested as the ideal for guiding reasoning" and "For this reason, classical logic has often been called the one right logic." [18]. Despite this view, in 1920 Łukasiewicz noted that future contingent statements like "There will be a sea-battle tomorrow" are not true (now), nor are they false (now). To reason about such statements Łukasiewicz invented a new truth-value, **N**either[1], to form the logic Ł3 [12]. Łukasiewicz basically wanted to use classical logic, except to allow **N** to be "in the *gap* between" **T** and **F**. He kept the usual connectives of $\neg$, $\vee$, $\wedge$, and $\rightarrow$, but found it necessary to change the definition of the conditional connective.

In contrast to statements that appear to have no truth value, paradoxical statements such as the Liar Paradox "This sentence is false." provide motivation for dialetheic logics [17] that allow statements to be one of **T**rue, **F**alse, or "have the *glut* of" **B**oth true and false. Dialetheic logics are paraconsistent, so that a contradiction in the input does not lead to logical omniscience. The most famous – and persistent – advocate of

[1] Actually, Łukasiewicz called it (the Polish equivalent for) **I**ndeterminate, but to keep things consistent with other works, we use **N**either.

dialetheism is Graham Priest, who developed the Logic of Paradox [16], which provides a foundation for the dialetheic logics RM3 [19] and a logic that we call A3, after [1]. As with the gap logic Ł3, these *glut* logics require particular conditional connectives in order to retain useful reasoning properties.

In 1977 Nuel Belnap published two articles, "How a Computer Should Think" and "A Useful Four-Valued Logic" [5]. One of the leading ideas was of a then-futuristic knowledge based system that would not only retrieve explicitly stored data, but would also reason and deduce consequences of the stored data. A further idea was that such a knowledge base might be given contradictory data to store, and that there might be topics for which no data is stored. This led to the development of the FDE logic [6, 4], which merges the ideas of gap and glut logics by including all four truth values: **T**, **B**, **N**, and **F**. Belnap envisaged the four truth values of FDE in a lattice, the "Truth Diamond" shown in Figure 1. The Truth Diamond represents the amount of truth in the four truth values, with **T** having the most (only truth) and **F** the least (no truth). **B** and **N** are between the two extremes of **T** and **F**, with different ways of balancing their true and false parts, and therefore have incomparable amounts of truth. Again, the choice of conditional connective for FDE is important, with different choices leading to different theories [20, 9].

Fig. 1. The Truth Diamond

$$\begin{tikzcd} & \mathbf{T} \arrow[dl] \arrow[dr] & \\ \mathbf{B} \arrow[dr] & & \mathbf{N} \arrow[dl] \\ & \mathbf{F} & \end{tikzcd}$$

This work deals with the development of an ATP system called JGXYZ[2] for these and other first-order logics. The system is "data-driven", in the sense that it requires only the addition of a new logic's truth tables for the unary and binary connectives in order to produce an ATP system for the logic. The data-driven approach is also taken in MUltlog [2], leading to the specification of a logic and deduction systems, but no actual running ATP system like JGXYZ. An implemented ATP system for multi-valued logics was $_{3}T^{A}P$ [3], but it is no longer supported. A survey of work done around the end of the last century is provided by [8]. Note that the input language for gap/glut logics is the same as for FOL – it is the semantics and reasoning that changes when a gap/glut logic is adopted (and consequently it does not not make sense to compare an ATP system's reasoning in gap/glut logics with the reasoning of a FOL ATP system).

2 A Motivating Example

As a quasi-real-world use case, consider the situation faced by script writers for a TV series that features "undead" characters [11]. In such shows there are characters who are alive, characters who are not alive, and undead characters who are both alive and not alive. Additionally, there will be (in future episodes) new characters whose liveliness is yet unknown. All characters that have ever appeared in an episode are either alive or have been buried. Each week the script writer must provide the necessary dialogue and placement of the characters who appear in the episode.[3] Characters who are alive

[2] Named after the authors **J**eff and **G**eoff, for any logic **XYZ**.

[3] Computer geeks ... think of the characters as UNIX processes, which can be alive, not alive, or zombies. Burial corresponds to reaping the process from the process table. FDE can thus be used to reason about UNIX processes. (Thanks to Josef Urban for this interpretation.)

need words and placement. Characters who are not alive need no words but still need placement. For now, let there be four characters: Alan, who is alive; Désirée, who is not alive and has been buried; Umberto, who is undead (i.e., both alive and not alive); and Nigel, who has not yet appeared in the script. The kinds of questions the script write might ask include:

– Does Désirée need words?
– Does Nigel need placement?
– Is Umberto both alive and not alive?
– Is Nigel alive or (inclusively) not alive?
– Has Umberto been buried?
– Was Désirée buried because she is not alive?

If such a scenario is to be formalized so that the questions can be correctly (logically!) answered, the possibility of characters being both alive and not alive requires a glut logic that supports the truth value **B**oth, and the possibility of new characters whose liveliness is unknown requires a gap logic that supports the truth value **N**either. The gap and glut logics discussed in Section 1 are appropriate, and the JGXYZ ATP system can provide the necessary reasoning.

The formalization in TPTP syntax is as follows:

```
%----Axioms of the undead
fof(alive_or_buried,axiom,! [X] : ( alive(X) | buried(X) )).
fof(alive_scripting,axiom,
    ! [X] : ( alive(X) => ( script(X,words) & script(X,placement) ))).
fof(not_alive_scripting,axiom,
    ! [X] : ( ~alive(X) => ( ~script(X,words) & script(X,placement) ))).

%----Current characters
fof(alan_alive,axiom,     alive(alan) ).
fof(desiree_dead,axiom,   ~alive(desiree) ).
fof(desiree_buried,axiom, buried(desiree)).
fof(umberto_alive,axiom,  alive(umberto) ).
fof(umberto_dead,axiom,   ~alive(umberto) ).

%----Queries
fof(desiree_needs_words,conjecture,   script(desiree,words) ).
fof(nigel_needs_placement,conjecture, script(nigel,placement)).
fof(umberto_alive_and_not,conjecture, alive(umberto) & ~alive(umberto)).
fof(nigel_alive_or_not,conjecture,    alive(nigel) | ~alive(nigel)).
fof(umberto_buried,conjecture,        buried(umberto)).
fof(not_alive_buried,conjecture,     ~alive(desiree) => buried(desiree)).
```

The answers to these queries, for each of the logics that are presented in Section 3, are presented in Section 5.

3 Truth Values and Conditional Connectives

Section 1 briefly introduced four gap/glut logics: Ł3, RM3, A3, and FDE. These differ in terms of the truth values they support, and the conditional connective that they use.

This section provides further details of these logics, and examines their conditional connectives.

Section 1 provided motivation for having the four truth values used by gap and glut logics: **T**, **B**, **N**, and **F**. As usual, the truth values are divided into those that are *designated* – the values that "true" statements should have (like being **T** in classical logic), and those that are *undesignated*. Logical truths are formulae that are always designated regardless of the truth values of their atomic components, and are the formulae that a reasoning tool should be able to prove. The truth tables for negation, disjunction, and conjunction over the four truth values are given in Table 1. The truth value of a conjunction (disjunction) is the meet (join) of its conjuncts (disjuncts) in the truth diamond, and negation inverts the order in the diamond.

Table 1. Truth Tables for Negation, Disjunction, and Conjunction

$\neg$	
T	**F**
B	**B**
N	**N**
F	**T**

$\vee$	**T**	**B**	**N**	**F**
T	**T**	**T**	**T**	**T**
B	**T**	**B**	**T**	**B**
N	**T**	**T**	**N**	**N**
F	**T**	**B**	**N**	**F**

$\wedge$	**T**	**B**	**N**	**F**
T	**T**	**B**	**N**	**F**
B	**B**	**B**	**F**	**F**
N	**N**	**F**	**N**	**F**
F	**F**	**F**	**F**	**F**

In this work, two conditional connectives are used:

- Classical Material Implication $\rightarrow_{cmi}$ [1, 20, 9]. This conditional was proposed in response to the observation that modus ponens (MP) fails in FDE if the classical FOL conditional $\rightarrow_{cls}$ defined in terms of $\vee$ and $\neg$, $(\varphi \rightarrow_{cls} \psi) =_{df} (\neg\varphi \vee \psi)$, is used [20]. $\rightarrow_{cmi}$ does however emphasize the classical aspects of a conditional. In the cases when the antecedent is designated, the value of the consequent is assigned to the conditional. In the cases when the antecedent is undesignated, **T** is assigned to the conditional.
- The "Łukasiewicz" conditional $\rightarrow_{Łuk}$ [13]. One of the features missing from $\rightarrow_{cmi}$ is contraposition with respect to negation, i.e., $(\varphi \rightarrow_{cmi} \psi) \neq (\neg\psi \rightarrow_{cmi} \neg\varphi)$. Contraposition can be added by taking a conjunction of $\rightarrow_{cmi}$ and its contraposed form, $(\varphi \rightarrow_{Łuk} \psi) =_{df} ((\varphi \rightarrow_{cmi} \psi) \wedge (\neg\psi \rightarrow_{cmi} \neg\varphi))$. This can be seen as a generalization of Łukasiewicz' implication from Ł3, hence the name "Łukasiewicz".

Table 2 shows the definitions of $\rightarrow_{Łuk}$ and $\rightarrow_{cmi}$. It is clear that they are very similar, differing only in the values of **T** $\rightarrow$ **B**, **N** $\rightarrow$ **B**, and **N** $\rightarrow$ **F**. The biconditional connective is understood to be the conjunction of the conditional and its converse, hence the differences between the two conditionals are propagated to the bi-conditionals. These differences are enough to produce some quite different theorems between the logics that use them, as can be seen in the experimental results presented in Section 5.

Given the choices of truth values and conditional connectives, five logics are considered:

- Ł3: The truth values are **T**, **N**, and **F**, with **T** designated. The conditional is $\rightarrow_{Łuk}$, restricted to the three truth values.
- RM3: The truth values are **T**, **B**, and **F**, with **T** and **B** designated. The conditional is $\rightarrow_{Łuk}$, restricted to the three truth values.

Table 2. Truth Tables for $\rightarrow_{Łuk}$ and $\rightarrow_{cmi}$

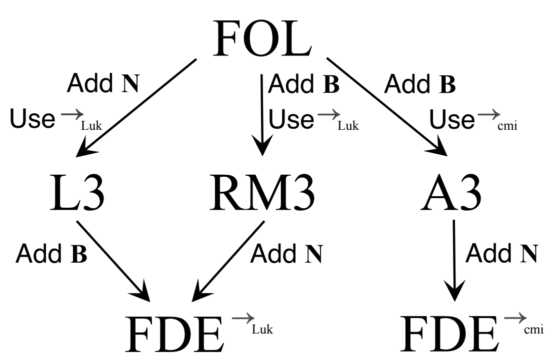
$\rightarrow_{Łuk}$	**T**	**B**	**N**	**F**
T	**T**	**F**	**N**	**F**
B	**T**	**B**	**N**	**F**
N	**T**	**N**	**T**	**N**
F	**T**	**T**	**T**	**T**

$\rightarrow_{cmi}$	**T**	**B**	**N**	**F**
T	**T**	**B**	**N**	**F**
B	**T**	**B**	**N**	**F**
N	**T**	**T**	**T**	**T**
F	**T**	**T**	**T**	**T**

- A3: The truth values are **T**, **B**, and **F**, with **T** and **B** designated. The conditional is $\rightarrow_{cmi}$, restricted to the three truth values.
- $\text{FDE}^{\rightarrow_{Łuk}}$: The truth values are **T**, **B**, **N**, and **F**, with **T** and **B** designated. The conditional is $\rightarrow_{Łuk}$.
- $\text{FDE}^{\rightarrow_{cmi}}$: The truth values are **T**, **B**, **N**, and **F**, with **T** and **B** designated. The conditional is $\rightarrow_{cmi}$.

The relationship between FOL and these logics is shown in Figure 2.

Fig. 2. The Relationships between the Logics



4 System Architecture and Implementation

JGXYZ proves theorems in the gap/glut (and other) logics by translating the problem to an equi-provable FOL problem, then using a FOL ATP system to find a proof (or countermodel) for the FOL problem. In [15] two translations from RM3 to FOL were presented, and in [20] the "truth evaluation" translation was extended to $\text{FDE}^{\rightarrow_{cmi}}$. The truth evaluation translation function *trs* takes a target formula (e.g., in $\text{FDE}^{\rightarrow_{cmi}}$) and a target truth value (e.g., one of **T**, **B**, **N**, or **F**) as arguments, and translates the target formula, either directly for atoms, or recursively on the subformulae for non-atoms, to produce a FOL formula. Intuitively, *trs* captures the necessary and sufficient conditions for the target formula to have the target truth value. Prior implementations of JGXYZ (called JGRM3 in [15], and later JGXYZ 0.1 in [20]) encoded trs directly. This meant that extending the translation to a new logic required significant effort. The new implementation of JGXYZ (version 0.2) is the same as for version 0.1 for quantified and atomic formulae, but makes the translation data-driven for formulae under a unary or binary connective. For each logic, its truth values and the designated subset of them are specified, and the truth tables for the logic's negation, disjunction, conjunction, and conditional connectives are provided.

Universally qualified formulae are treated as a conjunction of their ground instances, requiring that there exists an instance that has the target truth value, and that there do not exist any instances that have a truth value lower in the truth diamond. For example, for a universally quantified formula in FDE$^{\rightarrow_{cmi}}$ and the target truth value **B**, the translation requires that there exists an instance of the formula whose translation is **B**, and that there do not exist any instances whose translation is **F**, i.e., $trs(\forall x\ \varphi, \mathbf{B}) \Rightarrow \exists x\ trs(\varphi, \mathbf{B}) \wedge \neg\exists x\ trs(\varphi, \mathbf{F})$. Existentially quantified formulae are treated as a disjunction of their ground instances, requiring that there exists an instance that has the target truth value, and that there do not exist any instances that have a truth value higher in the truth diamond. For example, for an existentially quantified formula in FDE$^{\rightarrow_{cmi}}$ and the target truth value **B**, the translation requires that there exists an instance of the formula whose translation is **B**, and that there do not exist any instances whose translation is **T**, i.e., $trs(\exists x\ \varphi, \mathbf{B}) \Rightarrow \exists x\ trs(\varphi, \mathbf{B}) \wedge \neg\exists x\ trs(\varphi, \mathbf{T})$.

For formulae under a unary or binary connective, the appropriate truth table is consulted to find tuples of truth values such that the value of the connective for those inputs is the target truth value. The tuple elements are then the target truth values for the arguments of the connective in the formula. The translation is the disjunction (one disjunct for each tuple) of conjunctions (one conjunct for element of the tuple), applied to the translations of the arguments of the connective. For the n-ary connective $\oplus$ and the target truth value TTV:

$$trs(\oplus\varphi, TTV) \Rightarrow \bigvee_{i=1}^{k} \bigwedge_{j=1}^{n} trs(\varphi_j, inputs_{i,j}(\oplus, TTV))$$

where $inputs_{i,j}(\oplus, TTV)$ is the j^{th} element element of the i^{th} tuple of the k tuples from the truth table for $\oplus$ such that the value of $\oplus$ for those inputs is TTV, and φ_j is the j^{th} argument of φ (n is 1 for a unary connective and 2 for a binary connective, etc.). For example, for the FDE$^{\rightarrow_{cmi}}$ conditional formula $\varphi \rightarrow_{cmi} \psi$ and the target truth value **B**, k is 2 and the input tuples are [**T**, **B**] and [**B**, **B**]. Then:

$$trs(\varphi \rightarrow_{cmi} \psi, \mathbf{B}) \Rightarrow ((trs(\varphi, \mathbf{T}) \wedge trs(\psi, \mathbf{B})) \vee (trs(\varphi, \mathbf{B}) \wedge trs(\psi, \mathbf{B}))).$$

Atoms are translated to FOL atoms that capture what it means for the atom to have the target truth value. Equality atoms are treated classically[4], so that for a target truth value of **T** an equality atom is unchanged, for a target truth value of **F** an equality atom is negated, and for a target truth value of **B** or **N** an equality atom is translated to the FOL truth value **F**. A non-equality atom Φ that has predicate symbol $\mathcal{P}$ and arity n is translated to a FOL atom with predicate symbol $\mathcal{P}^{TTV}$ and arity n, where TTV is the

[4] The classical interpretation of equality is due to the classical interpretation of terms. Since a term is interpreted as an element of the domain, if two terms are interpreted as the same element then their equality is **T**rue, and if they are interpreted as different elements then their equality is **F**alse. There is no middle ground (**B**oth or **N**either). [7, 14]

target truth value. The FOL atom has the same term arguments as $\mathcal{P}$ in Φ. *Definition axioms* are added to relate each predicate symbol $\mathcal{P}^{LTV}$ to atoms that correspond to the two FOL truth values **T** and **F**, where LTV is each of the truth values used by the logic. The axioms introduce two new predicate symbols, $\mathcal{P}^{cT}$ and $\mathcal{P}^{cF}$ (for **c**lassical **T**rue and **F**alse) for each predicate symbol $\mathcal{P}$ in the input problem. The axioms are:

$$\forall\overline{x}\ (\mathcal{P}^{T}(\overline{x}) \leftrightarrow (\mathcal{P}^{cT}(\overline{x}) \land \neg\mathcal{P}^{cF}(\overline{x}))) \qquad \forall\overline{x}\ (\mathcal{P}^{B}(\overline{x}) \leftrightarrow (\mathcal{P}^{cT}(\overline{x}) \land \mathcal{P}^{cF}(\overline{x})))$$
$$\forall\overline{x}\ (\mathcal{P}^{N}(\overline{x}) \leftrightarrow (\neg\mathcal{P}^{cT}(\overline{x}) \land \neg\mathcal{P}^{cF}(\overline{x}))) \qquad \forall\overline{x}\ (\mathcal{P}^{F}(\overline{x}) \leftrightarrow (\neg\mathcal{P}^{cT}(\overline{x}) \land \mathcal{P}^{cF}(\overline{x})))$$

Finally, *exhaustion axioms* are added to enforce that each of the FOL atoms takes on exactly one of the truth values of the logic. By example, the axioms for $\mathrm{FDE}^{\rightarrow_{cmi}}$ are:

$$\forall\overline{x}\ (\mathcal{P}^{T}(\overline{x}) \lor \mathcal{P}^{B}(\overline{x}) \lor \mathcal{P}^{N}(\overline{x}) \lor \mathcal{P}^{F}(\overline{x}))$$

(The exclusive disjunction of the disjuncts, so that each of the FOL atoms takes on only one of the truth values, is a logical consequence of the exhaustion and definition axioms.)

For a set of formulae ϕ, let $def(\phi)$ be the set of definition axioms and $exh(\phi)$ the set of exhaustion axioms, for the predicate symbols that occur in ϕ. Define

$$des(\phi) =_{df} \bigvee_{i=1}^{n} trs(\phi, DTV_i)$$

where DTV is the set of designated truth values of the logic. For a problem $\phi \vDash \psi$ define

$$trans(\phi) \ = \ des(\phi) \cup exh(\phi \cup \{\psi\}) \cup def(\phi \cup \{\psi\})$$

Then $\phi \vDash_{logic} \psi$ iff $trans(\phi) \vDash_{FOL} des(\psi)$. A theorem prover for the logic is simply implemented by submitting $trans(\phi) \vDash_{FOL} des(\psi)$ to a FOL ATP system.

The JGXYZ translation is implemented in Prolog, and the full ATP system uses some scriptin' magic to connect the translation to a FOL ATP system. By default, 80% of the CPU time is allocated to searching for a proof, and if no proof is found the remaining 20% is used to search for a countermodel. Currently Vampire 4.2.2 [10] is used for the FOL reasoning, in CASC mode for proving, and in finite model finding mode for finding countermodels. JGXYZ for FOL,[5] Ł3, A3, RM3, $\mathrm{FDE}^{\rightarrow_{Łuk}}$, and $\mathrm{FDE}^{\rightarrow_{cmi}}$ are available through the SystemOnTPTP interface at http://www.tptp.org/cgi-bin/SystemOnTPTP.

5 Experimental Results

The implementation has been tested for all the logics encoded, on a set of problems, taken from [20]. All the problems are valid in FOL. Testing was done on an Intel(R) Xeon(R) CPU E5-2609 v2 @ 2.50GHz, with a CPU time limit of 600s per problem. Note that the time taken to translate a problem is negligible, so that almost all of the time is available to the FOL ATP system. The test problems in TPTP format are available at http://www.tptp.org/JGXYZ, and they can be run through SystemOnTPTP.

Table 3 gives the results of the testing, using the default JGXYZ settings described in Section 4. The results with a CPU time were proved, countermodels were found for those marked CSA, and no result was obtained within the CPU time limit for those marked GUP. The results marked GUP^{+} are known (from previous experiments [15,

[5] This can be used to empirically check that the translation does produce equi-provable problems. For more fun, it is possible to repeatedly apply the translation to a FOL problem to produce a new FOL problem, to produce a sequence of ever more difficult FOL problems.

20]) to be theorems for that logic, and the results marked GUP$^-$ are known to have countermodels for that logic.

As is expected, there are differences in the results between the various logics because of their different truth values and also their different conditional connectives. Problem 1 shows how purely glut logics such as RM3 and A3 can prove FOL tautologies, while logics that include the gap truth value **N** cannot. In contrast, Problems 8 and 14 show that there are some theorems of purely gap logics that are not theorems of glut logics. Problem 8 in particular shows that Ł3 is not paraconsistent. Problems 2 and 12 illustrate a difference between $\rightarrow_{Łuk}$ and $\rightarrow_{cmi}$: e.g., for Problem 2, in RM3 and FDE$^{\rightarrow_{Łuk}}$ with q set to **B** and p set to **T**, the conjecture is **F** and hence not a theorem, while in A3 and FDE$^{\rightarrow_{cmi}}$ the conjecture is **B**. In contrast, Problem 11 shows how this difference can work the other way. Problem 7 shows the difference between FOL and the gap/glut logics with their extra truth values and extended conditional connectives. In Ł3 and FDE$^{\rightarrow_{Łuk}}$ with p to **N** and q to **F** the conjecture is **N**; in A3 and FDE$^{\rightarrow_{cmi}}$ with p to **B** and q to **F** the conjecture is **F**; and in RM3 with p to **B** and q to **F** the conjecture is **N**. In contrast, Problem 6 is a theorem for all the logics, despite the extra truth values and extended conditional connectives.

Problems 16-20 are interesting both from a historical and also a contemporary point of view of the foundations of mathematics. They represent some of the motivating claims that drove the modern development of axiomatic set theory and mathematics. Read the relation $E(x, y)$ as saying that *x is an element of the set y*. Then each formula represents a crucial part of the various paradoxes of set theory. For example, Russell's paradox is in part captured by Problem 16, which says that there cannot be a set (y) all of whose members (x) are not members of themselves. See [20] for a more detailed discussion of these problems.

Problem 20, which is a theorem for Ł3, RM3, A3, and FDE$^{\rightarrow_{Łuk}}$, is quite hard for JGXYZ. An examination of syntactic characteristics of the translated problems illustrates how the translation blows up the problem. Table 4 provides some measures of the original and translated problems.[6] The translation blows up the problem significantly, with the effect being least for the purely gap logic Ł3, greater for the purely glut logics RM3 and A3, and most for the gap/glut logics FDE$^{\rightarrow_{Łuk}}$ and FDE$^{\rightarrow_{cmi}}$. The use of $\rightarrow_{Łuk}$ by RM3 and FDE$^{\rightarrow_{Łuk}}$ apparently has a greater effect than the use of $\rightarrow_{cmi}$ by A3 and FDE$^{\rightarrow_{cmi}}$. As RM3 and A3 are both purely glut logics, this difference is attributed to the different values for $\mathbf{T} \rightarrow \mathbf{B}$. The different values for $\mathbf{N} \rightarrow \mathbf{B}$ and $\mathbf{N} \rightarrow \mathbf{F}$ further contribute to the differences between the translations for FDE$^{\rightarrow_{Łuk}}$ and FDE$^{\rightarrow_{cmi}}$. The different blow ups naturally contribute correspondingly to the difficulty of the translated problems for the FOL ATP system.

For the motivating example of Section 2, the different logics again produce interestingly different results, as shown in Table 5. A proof is a positive answer to the query, while a CSA result is a negative answer. For FOL and Ł3, the axioms are contradictory (Umberto is alive and not alive), thus all the conjectures are theorems. For interest, the axiom stating that Umberto is alive was removed to make the axioms consistent in FOL and Ł3, then Vampire and JGXYZ were run on the resulting problems - these results

[6] Thanks to Giles Reger for providing a special version of Vampire that normalises the formulae into comparable forms.

Table 3. Example Axiom-Conjecture Pairs and their Provability

#	Axioms	$\models$	Conjecture	Ł3	RM3	A3	FDE$^{\rightarrow}$ Łuk	FDE$^{\rightarrow}$ cmi
1		$\models$	$p \vee \neg p$	CSA	0.1	0.1	CSA	CSA
2	q	$\models$	$p \rightarrow q$	0.1	CSA	0.1	CSA	0.1
3	$\neg p$	$\models$	$p \rightarrow q$	0.1	CSA	CSA	CSA	CSA
4	$\neg(p \rightarrow q)$	$\models$	p	0.1	0.1	0.1	0.1	0.1
5		$\models$	$p \rightarrow (q \vee \neg q)$	CSA	CSA	0.1	CSA	CSA
6		$\models$	$p \rightarrow (p \vee \neg p)$	0.1	0.1	0.1	0.1	0.1
7		$\models$	$(p \wedge \neg p) \rightarrow q$	CSA	CSA	CSA	CSA	CSA
8	$p, \neg p$	$\models$	q	0.1	CSA	CSA	CSA	CSA
9	$p \vee q, \neg p$	$\models$	q	0.1	CSA	CSA	CSA	CSA
10		$\models$	$(\neg p \vee q) \leftrightarrow (p \rightarrow q)$	CSA	CSA	CSA	CSA	CSA
11		$\models$	$((p \rightarrow q) \wedge (q \rightarrow p)) \rightarrow$ $(p \vee q \vee \neg(p \rightarrow q) \vee \neg(q \rightarrow p) \vee$ $((\neg p \rightarrow \neg q) \wedge (\neg q \rightarrow \neg p)))$	3.7	3.8	4.4	39.9	CSA
12	$H(a)$	$\models$	$\exists x\, G(x) \rightarrow H(a)$	0.1	CSA	0.1	CSA	0.1
13		$\models$	$\exists x\, (G(x) \wedge \neg G(x)) \rightarrow H(b)$	CSA	CSA	CSA	CSA	CSA
14	$\exists x\, (G(x) \vee H(x)), \neg\exists y\, G(y)$	$\models$	$\exists z\, H(z)$	0.1	CSA	CSA	CSA	CSA
15	$H(a)$	$\models$	$\forall x\, (H(x) \rightarrow G(x)) \leftrightarrow$ $\forall x\, ((H(x) \wedge G(x)) \vee (\neg H(x) \wedge G(a)))$	CSA	CSA	CSA	CSA	CSA
16		$\models$	$\neg\exists y \forall x\, (E(x,y) \leftrightarrow \neg E(x,x))$	CSA	0.1	0.1	CSA	CSA
17		$\models$	$\forall z \exists y \forall x\, (E(x,y) \leftrightarrow (E(x,z) \wedge \neg E(x,x)))$ $\rightarrow \neg\exists w \forall u\, E(u,w)$	CSA	GUP$^-$	GUP$^-$	GUP	CSA
18		$\models$	$\neg\exists y \forall x\, (E(x,y) \leftrightarrow$ $\neg\exists z\, (E(x,z) \wedge E(z,x))$	CSA	162.1	37.6	GUP	CSA
19		$\models$	$\exists y \forall x\, (E(x,y) \leftrightarrow E(x,x)) \rightarrow$ $\neg\forall x \exists y \forall z\, (E(z,y) \leftrightarrow \neg E(z,x))$	CSA	430.8	GUP$^+$	GUP	CSA
20	$\forall y \exists z \forall x\, (E(x,z) \leftrightarrow x = y)$	$\models$	$\neg\exists w \forall x\, (E(x,w) \leftrightarrow \forall u\, (E(x,u) \rightarrow$ $\exists y\, (E(y,u) \wedge \neg\exists z\, (E(z,u) \wedge E(z,y)))))$	73.3	263.1	GUP$^+$	412.3	CSA

Table 4. Syntactic Measures for Problem 20

Measure	FOL	Ł3	RM3	A3	FDE$^{\rightarrow}$ Łuk	FDE$^{\rightarrow}$ cmi
Number of formulae	2	6	6	6	7	7
Number of atoms	7	2080	9912	8356	45496	41130
Maximal formula depth	12	29	33	31	43	45
Number of connectives	7	2182	10418	8780	47920	43364
Number of predicates	2	6	6	6	7	7
Number of variables	8	460	2173	1847	7835	7211

are shown in the columns marked FOL′ and Ł3′. Problem 1 shows that all the logics understand that Désirée does not need words, because she is not known to be alive. Problem 2 should have a negative answer, because Nigel is not known to be alive nor is he known to be not alive. However, FOL′ assumes that he is either alive or not alive. RM3 and A3 do not escape from this conclusion because the only other possibility they offer is that he is both alive and not alive. In contrast, Ł3′, $\mathrm{FDE}^{\rightarrow_{Łuk}}$, and $\mathrm{FDE}^{\rightarrow_{cmi}}$ allow Nigel to be neither alive nor not alive. Problem 3 extends Problem 2, so that FOL′, RM3, and A3 conclude that Nigel needs placement, while Ł3′, $\mathrm{FDE}^{\rightarrow_{Łuk}}$, and $\mathrm{FDE}^{\rightarrow_{cmi}}$ do not. Problem 4 should have a positive answer, which all the logics (taking the original FOL and Ł3) support. However, for FOL and Ł3 with the contradictory axioms the positive answer might be for the wrong reason, depending on how the ATP system uses the axioms. Problem 5 is answered positively by FOL′ and Ł3′ as it is known that Umberto is not alive (recall, the axiom stating that Umberto is alive is removed). In contrast, for the other logics it is known that Umberto is alive and hence has not necessarily been buried. Problem 6 illustrates the difference between $\rightarrow_{Łuk}$ and $\rightarrow_{cmi}$. For RM3 and $\mathrm{FDE}^{\rightarrow_{Łuk}}$, which both use $\rightarrow_{Łuk}$, it is possible that Désirée is definitely (**T**) not alive, but has been both (**B**) buried and not buried. Under $\rightarrow_{Łuk}$ the implication is false (**F**) and hence a negative answer is returned. For A3 and $\mathrm{FDE}^{\rightarrow_{cmi}}$, which both use $\rightarrow_{cmi}$, the conditional would be both (**B**) true and false, and thus a positive answer is returned. The only way for the implication to not be a theorem in A3 and $\mathrm{FDE}^{\rightarrow_{cmi}}$ would be for Désirée to have been neither (**N**) buried nor not buried, or definitely not buried, which is not the case because it's an axiom that she has been buried.

Table 5. Provability of Queries about the Undead

						$\mathrm{FDE}^{\rightarrow}$	
#	Query	FOL′	Ł3′	RM3	A3	Łuk	cmi
1	Does Désirée need words?	CSA	CSA	CSA	CSA	CSA	CSA
2	Is Nigel alive or not alive?	0.1	CSA	0.2	0.2	CSA	CSA
3	Does Nigel need placement?	0.1	CSA	0.1	0.2	CSA	CSA
4	Is Umberto both alive and not alive?	CSA	CSA	0.3	0.2	0.2	0.2
5	Has Umberto been buried?	0.1	0.1	CSA	CSA	CSA	CSA
6	Was Désirée buried because she is not alive?	0.1	0.2	CSA	0.1	CSA	0.1

6 Conclusion

This paper has described an ATP system, named JGXYZ, for some gap and glut logics. JGXYZ is based on an equi-provable translation to FOL, followed by use of an existing ATP system for FOL. A key feature of JGXYZ is that the translation to FOL is data-driven, in the sense that it requires only the addition of a new logic's truth tables for the unary and binary connectives in order to produce an ATP system for the logic. Experimental results from JGXYZ have illustrated the differences between the logics and translated problems, both technically and in terms of a quasi-real-world use case.

Future work includes a more comprehensive investigation of gap and glut logics, their implementation in JGXYZ, and full experimental evaluation.

References

1. Avron, A.: Natural 3-Valued Logics: Characterization and Proof Theory. Journal of Symbolic Logic **56**(1), 276–294 (1991)
2. Baaz, M., Fermúller, C., Salzer, G., Zach, R.: The Design of the CADE-13 ATP System Competition. In: McRobbie, M., Slaney, J. (eds.) Proceedings of the 13th International Conference on Automated Deduction. pp. 226–230. No. 1104 in Lecture Notes in Artificial Intelligence, Springer-Verlag (1996)
3. Beckert, B., Hähnle, R., Oel, P., Sulzmann, M.: The Tableau-based Theorem Prover $_3T^AP$ Version 4.0. In: McRobbie, M., Slaney, J. (eds.) Proceedings of the 13th International Conference on Automated Deduction. pp. 302–307. No. 1104 in Lecture Notes in Artificial Intelligence, Springer-Verlag (1996)
4. Belnap, N.: A Useful Four-Valued Logic. In: Dunn, J., Epstein, G. (eds.) Modern Uses of Multiple-Valued Logic, pp. 5–37. Springer Netherlands (1977)
5. Belnap, N.: A Useful Four-valued Logic: How a Computer Should Think. In: Anderson, A., Belnap, N., Dunn, J. (eds.) Entailment: The Logic of Relevance and Necessity, Volume II, pp. 506–541. Princeton UP (1992)
6. Dunn, J.: Intuitive Semantics for First Degree Entailment and Coupled Trees. Philosophical Studies **29**, 149–168 (1976)
7. Evans, G.: Can There Be Vague Objects? Analysis **38**, 208 (1978)
8. Hähnle, R.: Advanced Many-Valued Logics. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, pp. 297–395. Springer Netherlands (2001)
9. Hazen, A., Pelletier, F.: K3, Ł3, LP, RM3, A3, FDE, M: How to Make Many-Valued Logics Work for You. In: Omori, H., Wansing, H. (eds.) New Essays on Belnap-Dunn Logic, p. To appear. Synthese Library, Springer Verlag (1980)
10. Kovacs, L., Voronkov, A.: First-Order Theorem Proving and Vampire. In: Sharygina, N., Veith, H. (eds.) Proceedings of the 25th International Conference on Computer Aided Verification. pp. 1–35. No. 8044 in Lecture Notes in Artificial Intelligence, Springer-Verlag (2013)
11. Levine, E., Parks, L.: Undead TV. Duke University Press (2007)
12. Łukasiewicz, J.: On Three-Valued Logic. Ruch Filozoficny **5**, 170–171 (1920)
13. Omori, H., Wansing, H.: 40 Years of FDE: An Introductory Overview. Studia Logica **105**(6), 1021–1049 (2017)
14. Pelletier, F.: Another Argument against Vague Identity. Journal of Philosophy **86**, 481–492 (1989)
15. Pelletier, F., Sutcliffe, G., Hazen, A.: Automated Reasoning for the Dialetheic Logic RM3. In: Rus, V., Markov, Z. (eds.) Proceedings of the 30th International FLAIRS Conference. pp. 110–115 (2017)
16. Priest, G.: The Logic of Paradox. Journal of Philosophical Logic **8**, 219–241 (1979)
17. Priest, G., Routley, R., Norman, J.: Paraconsistent Logic: Essays on the Inconsistent. Philosophia Verlag (1989)
18. Shapiro, S., Kouri Kissel, T.: Classical logic. In: Zalta, E. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University (2018), https://plato.stanford.edu/archives/spr2018/entries/logic-classical/
19. Sobociński, B.: Axiomatization of a Partial System of Three-valued Calculus of Propositions‘. Journal of Computing Systems **1**, 23–55 (1952)
20. Sutcliffe, G., Pelletier, F., Hazen, A.: Making Belnap's "Useful Four-Valued Logic" Useful. In: Brawner, K., Rus, V. (eds.) Proceedings of the 31st International FLAIRS Conference. pp. 116–121 (2018)
21. Wikipedia contributors: Logic - Wikipedia, The Free Encyclopedia (2018), https://en.wikipedia.org/w/index.php?title=Logic&oldid=875884116