Diplomarbeitspräsentationen der Fakultät für Informatik

Dialogue Games for Fuzzy Logics

Masterstudium
Computational Intelligence

Christoph Roschger

Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Theoretische Informatik und Logik
Betreuer: Ao.Prof. Dipl.Ing. Dr.techn. Christian G. Fermüller

Giles's Game

Overview & Motivation

- dialogue game introduced by Robin Giles in the 1970s
- models reasoning in physical theories
- asserting a proposition means committing oneself to pay a certain amount of money if the associated experiment(s) fail(s)
- seperates evaluation of atomic formulas from decomposing compound formulas

Betting for Positive Results:

- each atomic proposition $\mathfrak{a}$ is associated with a binary (yes/no) experiment $\mathtt{E}_\mathfrak{a}$
- experiments may be probabilistic, i.e. show dispersion
- for each assertion of an atomic proposition an experiment is made
- each player places bets on positive outcomes of experiments corresponding to his claims

Decomposing Compound Formulas:

- arguments about complex formulas are systematically reduced to arguments about less complex formulas
- dialogue rules have already been introduced by Lorenzen for Intuitionistic Logic
- these rules characterize the *meaning* of logical connectives, independently of the underlying betting scheme

Rules

Atomic Evaluation: *Let $\mathfrak{a}$ be an atomic proposition. He who asserts $\mathfrak{a}$ agrees to pay his opponent €1 if a trial of the experiment associated with $\mathfrak{a}$ yields the outcome "no".*

Implication: *He who asserts $\mathtt{A} \to \mathtt{B}$ agrees to assert $\mathtt{B}$ if his opponent will assert $\mathtt{A}$.*

Negation: *He who asserts $\neg\mathtt{A}$ agrees to assert $\bot$ if his opponent will assert $\mathtt{A}$* where *$\bot$ is associated with an experiment that always evaluates to "no".*

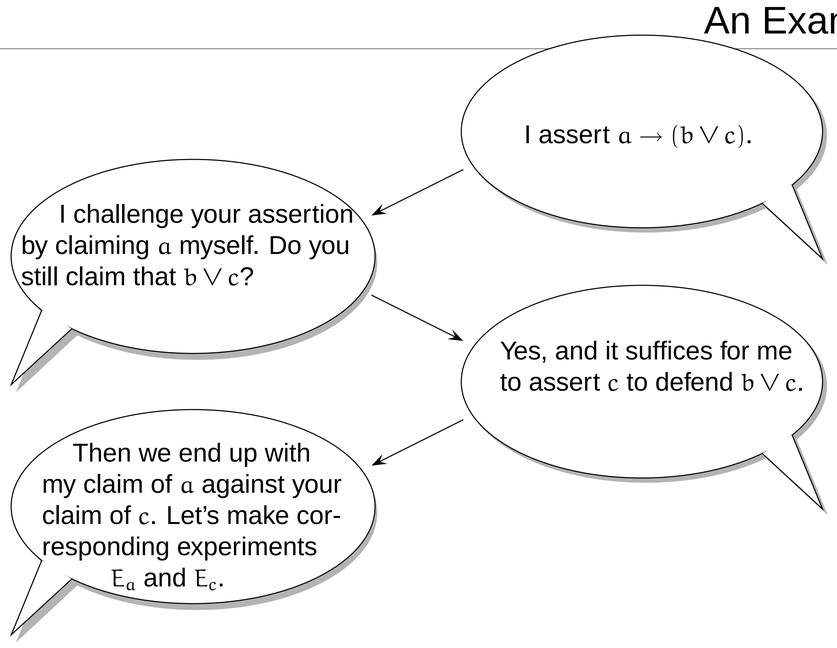
Disjunction: *He who asserts $\mathtt{A} \vee \mathtt{B}$ commits himself to assert either $\mathtt{A}$ or $\mathtt{B}$ at his own choice.*

Conjunction: *He who asserts $\mathtt{A} \wedge \mathtt{B}$ commits himself to assert either $\mathtt{A}$ or $\mathtt{B}$ at his opponent's choice.*

Strong conjunction: *He who asserts $\mathtt{A}\&\mathtt{B}$ commits himself either to assert both $\mathtt{A}$ and $\mathtt{B}$ or to admit falsity by asserting $\bot$.*

After being attacked, a formula is being deleted from the game.

An Example Dialogue



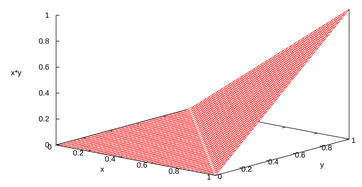
Łukasiewicz Logic

T-Norm Based Fuzzy Logics

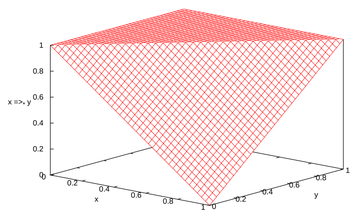
- many valued logics: 0 stands for absolute falsity, 1 for truth, but inifinitely many intermediate degrees of truth between 0 and 1
- truth function for (strong) conjunction $\&$ is a continuous t-norm
- a t-norm is a commutative, associative function $* : [0,1]^2 \to [0,1]$ with unit 1 which is order preserving
- truth function for implication $\to$ is the residuum of a t-norm
- the residuum $\Rightarrow_*$ of a t-norm $*$ is determined by $x \Rightarrow_* y := \sup\{z \mid x * z \leqslant y\}$
- other connectives $\wedge$, $\vee$, and $\neg$ are derived from $\&$,$\to$, and $\bot$

Łukasiewicz Logic

- one of three *fundamental* t-norm based fuzzy logics
- originally J. Łukasiewicz defined a three-valued logic for modelling future contingents, which has later been extended to infinitely many truth values
- Łukasiewicz t-norm: $x *_\mathrm{L} y = \max(0, x + y - 1)$
- associated residuum: $x \Rightarrow_\mathrm{L} y = \min(1, 1 - x + y)$
- the *unique* fuzzy logic where all truth functions are continuous
- all connectives can be derived from $\to$ and $\bot$



Łukasiewicz t-Norm $*_\mathrm{L}$

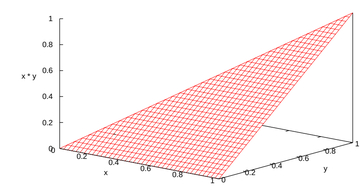


Residuum $\Rightarrow_\mathrm{L}$

Other Fuzzy Logics

Gödel Logic

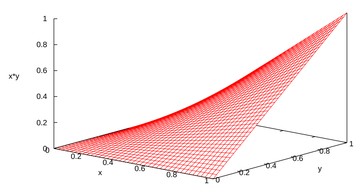
- also known as *Intuitionistic Fuzzy Logic*
- based on the Gödel t-norm $x *_\mathrm{G} y = \min(x, y)$
- associated residuum: $x \Rightarrow_\mathrm{G} y = y$ if $x > y$, and is 1 otherwise
- only the order of truth values is relevant for evaluating formulas



Gödel t-Norm $*_\mathrm{G}$

Product Logic

- introduced in 1996 by Hajek, Godo, and Esteva
- based on the Product t-norm $x *_\Pi y = x \cdot y$
- associated residuum: $x \Rightarrow_\Pi y = y/x$ if $x > y$, and is 1 otherwise



Product t-Norm $*_\Pi$

Adequateness of Giles's Game

For Łukasiewicz Logic

- Already proved by Giles in the 1970s:
- A formula $\mathtt{F}$ is valid in Łukasiewicz Logic iff **I** have a strategy to avoid risk (expected loss) in a game starting with me asserting $\mathtt{F}$ for any assignment of probability values to experiments.
- Moreover: given a fixed interpretation, my expected loss of money from asserting a formula in the game directly corresponds to a valuation in Łukasiewicz Logic.

For Gödel & Product Logic

- Variants of Giles's Game presented by Fermüller recently,
- alternative betting schemes: *selecting representatives* (Gödel Logic) and *joint bets* (Product Logic)
- dialogue rule for implication has to be extended as well,
- dialogue rules correspond to the logical rules of an analytic proof system based on relational hypersequents.

Alternative Dialogue Rules

- Presented in this thesis,
- another way to adapt the dialogue rule for implication for Gödel Logic and Product Logic,
- game gets simpler compared to the other approach,
- connection to the hypersequent calculus is lost.

Accompanying Implementation

Webgame

- Web-based application which allows playing Giles's Game interactively,
- simulates evaluation by dispersive experiments.
- see `http://logic.at/people/roschger/thesis/webgame`

Giles

- Small Haskell-program to display game trees of Giles's game,
- given a formula, computes a game tree of the corresponding game and outputs the tree as a dot-Graph specification.

Hypseq

- Utility to find derivations of hypersequents in the relational hypersequent calculus **rH**,
- computes all possible derivations and outputs the one with the smallest height.

TCGame

- Utility to find a winning strategy for the proponent **P** in a Truth Comparison Game,
- for Gödel Logic,
- winning strategy for **P** can be seen as a proof of the starting formula.

Contact: roschger@logic.at