

Demo: Reasoning with Arithmetic in the Attempto Reasoner RACE

Norbert E. Fuchs
*Institute of Computational Linguistics
University of Zurich*

Attempto Controlled English (ACE) offers numbers that are positive and negative integers and positive and negative reals. Furthermore, there are arithmetic expressions built with the help of the operators $+$, $-$, $*$, $/$, $^$ from numbers, variables, proper names and parenthesised subexpressions.

Boolean formulas are built from numbers, arithmetic expressions, proper names and variables with the help of the comparison operators $=$, \neq , $>$, \geq , $<$ and \leq . In RACE the copula *is* can be used as a synonym for the comparison operator $=$.

When reasoning with arithmetic expressions and formulas one encounters four phenomena that did not previously occur in the Attempto reasoner RACE.

- While RACE fundamentally relies on syntactic matching of logical atoms, numerical expressions – like those in the formula $100/50 + 8 = 4 + 6$ – cannot simply be matched, but must be numerically evaluated before being tested for equality.
- While previously the order of processing did not matter, the evaluation of expressions – as in $A \text{ is } B + C$. $C \text{ is } D - 1$. $B \text{ is } 2$. $D \text{ is } 3$. – must be delayed until all constituents have a value.
- Even after evaluation remain problems of relating formulas, as can be seen in the deduction attempt $X=1 \mid - X>0$.
- As in standard logic arithmetical contradictions can involve negation, as for instance in $A \text{ is } 1$. $A \text{ is not } 1$. But there are new forms of contradictions not involving negation, for example $A \text{ is } 1$. $A \text{ is } 2$. or simply $1=2$.

Using simple examples I will show in this demo how these problems can be solved so that RACE can effectively and efficiently reason with numbers, expressions and formulas.