LIONS EAT ONLY STRAWBERRIES
A variation on themes by Lewis Carroll and Mark Twain

Jack K. Horner

Last modified: 19 July 2020/0910 US Central Time

1.0 Introduction

In a scene in *Letters from the Earth* (Twain 1962, "Extract from Eve's Diary", 82), Adam and Eve stand in the Garden of Eden some distance from the First Lion -- "William McKinley" -- discussing whether McKinley is a vegetarian. Adam propounds a long-winded, first-principles argument for the claim, concluding that the First Lion eats no meat.

Eve, who had seen McKinley devour the First Lamb just a few days earlier, counters, "Adam, I think there is something better than logic."

"And what could that be?" he asks.

"Fact", she replies.

Adam's argument is in some ways akin to some of Lewis Carroll's "Logical Puzzles" (1896), which trade on contrapositive inference.

Here, I use the automated deduction functionality in *Mathematica (*2020) to produce an equational logic proof of a variant of Adam's argument. More particularly, let's suppose that Adam's argument is as shown in Figure 1:

> Premise 1. If x is not the First Lion, then x is not McKinley.
> Premise 2. If x is McKinley, then x eats only strawberries.
> Premise 3. x is McKinley.
>
> Conclusion: The First Lion eats only strawberries.

Figure 1. Adam's argument.

2.0 Method

The argument in Figure 1 was translated to a form suitable for *Mathematica*'s **FindEquationalProof** function. That function accepts a set of premises, and a conclusion, cast in either a first-order-logic or equational logic form and attempts to produce an equational logic proof of the conclusion from the premises.

The resulting script was executed on the following platform:

Mathematica (2020)
Windows 10
Dell Inspiron 545
-- Intel Core2 Quad CPU Q8200, clocked at 2.33 GHz, 4 Cores
-- 8 GB RAM
-- 1 TB disk

2.1 Some terminology

In this section, I assume the definitions of term, value of a term, variable, and constant contained in Baader and Nipkow 1999, Chapter 3.

A *rewriting system* is a system of R rules that transforms expressions that satisfy some well-defined set of formation rules to another expression that satisfy those formation rules. For the purposes of this paper, I restrict "rewriting system" to a rewriting system that concerns identities of terms.

Two terms are said to be *identical* if the values of the terms are equal for all values of variables occurring in them.

A *reduction of a term T to a term T'* is a (typically recursive) rewriting of T to T' using a set of rewriting rules R such that T' is "simpler than" T (given some definition of "simpler than"). A *reduction sequence of a term T to a term T'* is a sequence T0 = T, T1, T2, T3, ..., Tn = T', where each Ti is the result of applying R to Ti-1, i = 1, 2, ..., n.

If "simpler than" is a partial ordering (Suppes 1974, Df. 21, 72) on a reduction sequence that begins with T and ends with T' in a system with a set of rewriting rules R, "simpler than" induces a reduction order (Baader and Nipkow 1999, 102) on the reduction sequence that begins with T and ends with T'. A term *Tn is in normal form* if no application of R to Tn changes Tn.

A rewriting system is said to be *finitely terminating* if every reduction sequence of any term T produces, in a finite number of iterations, a normal form of T. A rewriting system is said to be *confluent* if the normal forms of all terms in the system are unique.

Some term rewriting systems are both finitely terminating and confluent (Baader and Nipkow 1999, esp. Chapter 9). Such rewriting systems have unique normal forms for all expressions. This permits us to use the the output of such a system to determine whether there is an identity between two terms T1 and T2 in the following manner. If T1 and T2 and have the same normal form, then there is an identity between T1 and T2. Otherwise, there is not an identity.

2.2 Mathematica's equational logic inference algorithm

The inference algorithm in Mathematica's ADF is the Knuth-Bendix completion algorithm (Knuth and Bendix 1970). KBC attempts to transform a given finite set of identities (an "input" to KBC) to a finitely terminating, confluent term rewriting system that preserves identity. At initialization, KBC attempts to "orient" the identities supplied in its input according to the KnuthBendix reduction order (Baader and Nipkow 1999, Section 5.4.4). This results in an initial set of reduction rules. KBC then attempts to complete this initial set of rules with additional rules, obtaining their normal forms, and adding a new rule for every pair of the normal forms in accordance with the reduction order.

KBC may

1. Terminate with success, yielding a finitely terminating, confluent set of rules, or
2. Terminate with failure, or
3. Loop without terminating.

3.0 Results (*Mathematica* script and outputs)

The proof can be produced by a single *Mathematica* statement.

In[1]:=
```
proofLionsEatOnlyStrawberries =
 FindEquationalProof[Not[Exists[x, And[FirstLion[x], Not[EatsOnlyStrawberries[x]]]]],
  {ForAll[x, Implies[Not[FirstLion[x]], Not[McKinley[x]]]],
   ForAll[x, Implies[McKinley[x], EatsOnlyStrawberries[x]]], ForAll[x, McKinley[x]]}]
```

Out[1]=
```
ProofObject[ Logic: Predicate/EquationalLogic   Steps: 76
             Theorem: ∀x !(FirstLion[x] && ! EatsOnlyStrawberries[x]) ]
```

A detailed version of the proof follows.

In[2]:=
```
proofLionsEatOnlyStrawberries["ProofNotebook"]
```

Axiom 1

We are given that:

```
∀x (!FirstLion[x]⇒!McKinley[x])
```

Axiom 2

We are given that:

```
∀x (McKinley[x]⇒EatsOnlyStrawberries[x])
```

Axiom 3

We are given that:

```
∀xMcKinley[x]
```

Hypothesis 1

We would like to show that:

```
∀x! (FirstLion[x]&&!EatsOnlyStrawberries[x])
```

Equationalized Axiom 1

We generate the "equationalized" axiom:

```
x1== (x1|| (x2&&!x2))
```

Equationalized Axiom 2

We generate the "equationalized" axiom:

```
x1== (x1&& (x2||!x2))
```

Equationalized Axiom 3

We generate the "equationalized" axiom:

```
(x1||x2) == (x2||x1)
```

Equationalized Axiom 4

We generate the "equationalized" axiom:

We generate the "equationalized" axiom:

(x1||(x2&&x3))==((x1||x2)&&(x1||x3))

Equationalized Axiom 5

We generate the "equationalized" axiom:

(FirstLion[x1]||!McKinley[x1])==(a_0||!a_0)

Equationalized Axiom 6

We generate the "equationalized" axiom:

(!McKinley[x1]||EatsOnlyStrawberries[x1])==(a_0||!a_0)

Equationalized Axiom 7

We generate the "equationalized" axiom:

((x1&&x2)||(x1&&x3))==(x1&&(x2||x3))

Equationalized Axiom 8

We generate the "equationalized" axiom:

(a_0||!a_0)==McKinley[x1]

Equationalized Axiom 9

We generate the "equationalized" axiom:

(x1&&x2)==(x2&&x1)

Equationalized Hypothesis 1

We generate the "equationalized" hypothesis:

(a_0||!a_0)==!(FirstLion[x_0]&&!EatsOnlyStrawberries[x_0])

Critical Pair Lemma 1

The following expressions are equivalent:

((x1&&!x1)||x2)==x2

Proof

Note that the input for the rule:

x1_||x2_↔x2_||x1_

contains a subpattern of the form:

x1_||x2_

which can be unified with the input for the rule:

x1_||(x2_&&!x2_)→x1

where these rules follow from Equationalized Axiom 3 and Equationalized Axiom 1 respectively.

Substitution Lemma 1

It can be shown that:

(!McKinley[x1]||FirstLion[x1])==(a_0||!a_0)

Proof

We start by taking Equationalized Axiom 5, and apply the substitution:

x1_||x2_→x2||x1

which follows from Equationalized Axiom 3.

Substitution Lemma 2

It can be shown that:

(EatsOnlyStrawberries[x1] || !McKinley[x1]) == (ạ$_0$ || !ạ$_0$)

Proof

We start by taking Equationalized Axiom 6, and apply the substitution:

x1_ || x2_ → x2 || x1

which follows from Equationalized Axiom 3.

Substitution Lemma 3

It can be shown that:

!McKinley[x1_] || FirstLion[x1_] → EatsOnlyStrawberries[x_0] || !McKinley[x_0]

Proof

We start by taking Substitution Lemma 1, and apply the substitution:

ạ$_0$ || !ạ$_0$ → EatsOnlyStrawberries[x_0] || !McKinley[x_0]

which follows from Substitution Lemma 2.

Critical Pair Lemma 2

The following expressions are equivalent:

(x1 && (x2 || !x1)) == (x1 && x2)

Proof

Note that the input for the rule:

(x1_ && x2_) || (x1_ && x3_) → x1 && (x2 || x3)

contains a subpattern of the form:

(x1_ && x2_) || (x1_ && x3_)

which can be unified with the input for the rule:

x1_ || (x2_ && !x2_) → x1

where these rules follow from Equationalized Axiom 7 and Equationalized Axiom 1 respectively.

Substitution Lemma 4

It can be shown that:

(EatsOnlyStrawberries[x_0] || !McKinley[x_0]) == McKinley[x1]

Proof

We start by taking Equationalized Axiom 8, and apply the substitution:

ạ$_0$ || !ạ$_0$ → EatsOnlyStrawberries[x_0] || !McKinley[x_0]

which follows from Substitution Lemma 2.

Substitution Lemma 5

It can be shown that:

!McKinley[x1_] || FirstLion[x1_] → McKinley[x_0]

Proof

We start by taking Substitution Lemma 3, and apply the substitution:

```
EatsOnlyStrawberries[x0]||!McKinley[x0]→McKinley[x0]
```

which follows from Substitution Lemma 4.

Critical Pair Lemma 3

The following expressions are equivalent:

```
McKinley[x1]==McKinley[x2]
```

Proof

Note that the input for the rule:

```
"0"
```

contains a subpattern of the form:

```
EatsOnlyStrawberries[x0]||!McKinley[x0]
```

which can be unified with the input for the rule:

```
"0"
```

where these rules follow from Substitution Lemma 4 and Substitution Lemma 4 respectively.

Critical Pair Lemma 4

The following expressions are equivalent:

```
((x1||!x1)&&x2)==x2
```

Proof

Note that the input for the rule:

```
x1_&&x2_↔x2_&&x1_
```

contains a subpattern of the form:

```
x1_&&x2_
```

which can be unified with the input for the rule:

```
x1_&&(x2_||!x2_)→x1
```

where these rules follow from Equationalized Axiom 9 and Equationalized Axiom 2 respectively.

Critical Pair Lemma 5

The following expressions are equivalent:

```
(x1&&x2)==(x1&&(!x1||x2))
```

Proof

Note that the input for the rule:

```
(x1_&&!x1_)||x2_→x2
```

contains a subpattern of the form:

```
(x1_&&!x1_)||x2_
```

which can be unified with the input for the rule:

```
(x1_&&x2_)||(x1_&&x3_)→x1&&(x2||x3)
```

where these rules follow from Critical Pair Lemma 1 and Equationalized Axiom 7 respectively.

Critical Pair Lemma 6

The following expressions are equivalent:

```
(x1||x2)==(x1||(!x1&&x2))
```

Proof

Note that the input for the rule:

```
(x1_||!x1_)&&x2_→x2
```

contains a subpattern of the form:

```
(x1_||!x1_)&&x2_
```

which can be unified with the input for the rule:

```
(x1_||x2_)&&(x1_||x3_)→x1||(x2&&x3)
```

where these rules follow from Critical Pair Lemma 4 and Equationalized Axiom 4 respectively.

Critical Pair Lemma 7

The following expressions are equivalent:

```
(McKinley[x1]&&EatsOnlyStrawberries[x1])==(McKinley[x1]&&(a0||!a0))
```

Proof

Note that the input for the rule:

```
x1_&&(x2_||!x1_)→x1&&x2
```

contains a subpattern of the form:

```
x2_||!x1_
```

which can be unified with the input for the rule:

```
"0"
```

where these rules follow from Critical Pair Lemma 2 and Substitution Lemma 2 respectively.

Substitution Lemma 6

It can be shown that:

```
(McKinley[x1]&&EatsOnlyStrawberries[x1])==McKinley[x1]
```

Proof

We start by taking Critical Pair Lemma 7, and apply the substitution:

```
x1_&&(x2_||!x2_)→x1
```

which follows from Equationalized Axiom 2.

Critical Pair Lemma 8

The following expressions are equivalent:

```
(McKinley[x0]&&EatsOnlyStrawberries[x0])==(McKinley[x0]&&McKinley[x1])
```

Proof

Note that the input for the rule:

```
x1_&&(x2_||!x1_)→x1&&x2
```

contains a subpattern of the form:

```
x2_||!x1_
```

which can be unified with the input for the rule:

"0"

where these rules follow from Critical Pair Lemma 2 and Substitution Lemma 4 respectively.

Substitution Lemma 7

It can be shown that:

(EatsOnlyStrawberries[x1]&&McKinley[x1]) == McKinley[x1]

PROOF

We start by taking Substitution Lemma 6, and apply the substitution:

x1_&&x2_→x2&&x1

which follows from Equationalized Axiom 9.

Critical Pair Lemma 9

The following expressions are equivalent:

McKinley[x1] == (EatsOnlyStrawberries[x1]&&McKinley[x2])

PROOF

Note that the input for the rule:

EatsOnlyStrawberries[x1_]&&McKinley[x1_]→McKinley[x1]

contains a subpattern of the form:

McKinley[x1_]

which can be unified with the input for the rule:

McKinley[x1_]↔McKinley[x2_]

where these rules follow from Substitution Lemma 7 and Critical Pair Lemma 3 respectively.

Critical Pair Lemma 10

The following expressions are equivalent:

McKinley[x1] == (McKinley[x2]&&EatsOnlyStrawberries[x1])

PROOF

Note that the input for the rule:

EatsOnlyStrawberries[x1_]&&McKinley[x2_]→McKinley[x1]

contains a subpattern of the form:

EatsOnlyStrawberries[x1_]&&McKinley[x2_]

which can be unified with the input for the rule:

x1_&&x2_↔x2_&&x1_

where these rules follow from Critical Pair Lemma 9 and Equationalized Axiom 9 respectively.

Substitution Lemma 8

It can be shown that:

McKinley[x_0] == (McKinley[x_0]&&McKinley[x1])

PROOF

We start by taking Critical Pair Lemma 8, and apply the substitution:

`McKinley[x1_]&&EatsOnlyStrawberries[x2_]→McKinley[x2]`

which follows from Critical Pair Lemma 10.

Critical Pair Lemma 11

The following expressions are equivalent:

`McKinley[x0]==(McKinley[x1]&&McKinley[x2])`

Proof

Note that the input for the rule:

`McKinley[x0]&&McKinley[x1_]→McKinley[x0]`

contains a subpattern of the form:

`McKinley[x0]`

which can be unified with the input for the rule:

`McKinley[x1_]↔McKinley[x2_]`

where these rules follow from Substitution Lemma 8 and Critical Pair Lemma 3 respectively.

Critical Pair Lemma 12

The following expressions are equivalent:

`(x1&&x1)==x1`

Proof

Note that the input for the rule:

`x1_&&(!x1_||x2_)→x1&&x2`

contains a subpattern of the form:

`x1_&&(!x1_||x2_)`

which can be unified with the input for the rule:

`x1_&&(x2_||!x2_)→x1`

where these rules follow from Critical Pair Lemma 5 and Equationalized Axiom 2 respectively.

Critical Pair Lemma 13

The following expressions are equivalent:

`True`

Proof

Note that the input for the rule:

`x1_||(!x1_&&x2_)→x1||x2`

contains a subpattern of the form:

`!x1_&&x2_`

which can be unified with the input for the rule:

`x1_&&x1_→x1`

where these rules follow from Critical Pair Lemma 6 and Critical Pair Lemma 12 respectively.

Critical Pair Lemma 14

The following expressions are equivalent:

```
(x1&&x1) == (x1&& (x1 || !x1))
```

PROOF

Note that the input for the rule:

```
x1_&& (x2_ || !x1_) →x1&&x2
```

contains a subpattern of the form:

```
x2_ || !x1_
```

which can be unified with the input for the rule:

```
x1_ || !x1_→x1 || !x1
```

where these rules follow from Critical Pair Lemma 2 and Critical Pair Lemma 13 respectively.

Substitution Lemma 9

It can be shown that:

```
(x1&&x1) ==x1
```

PROOF

We start by taking Critical Pair Lemma 14, and apply the substitution:

```
x1_&& (x2_ || !x2_) →x1
```

which follows from Equationalized Axiom 2.

Substitution Lemma 10

It can be shown that:

```
(x1&&x1) ==x1
```

PROOF

We start by taking Substitution Lemma 9, and apply the substitution:

```
x1_&&x2_→x2&&x1
```

which follows from Equationalized Axiom 9.

Substitution Lemma 11

It can be shown that:

```
True
```

PROOF

We start by taking Substitution Lemma 10, and apply the substitution:

```
x1_&&x1_→x1
```

which follows from Critical Pair Lemma 12.

Critical Pair Lemma 15

The following expressions are equivalent:

```
(!x1 || x2) == (!x1 || (x1&&x2))
```

PROOF

Note that the input for the rule:

```
x1_ || (!x1_&&x2_) →x1 || x2
```

contains a subpattern of the form:

`!x1_`

which can be unified with the input for the rule:

`x1_→x1`

where these rules follow from Critical Pair Lemma 6 and Substitution Lemma 11 respectively.

Substitution Lemma 12

It can be shown that:

`(!McKinley[x1]||FirstLion[x1])==McKinley[x0]`

Proof

We start by taking Substitution Lemma 5, and apply the substitution:

`x1_→x1`

which follows from Substitution Lemma 11.

Substitution Lemma 13

Out[2]=

It can be shown that:

`(FirstLion[x1]||!McKinley[x1])==McKinley[x0]`

Proof

We start by taking Substitution Lemma 12, and apply the substitution:

`x1_||x2_→x2||x1`

which follows from Equationalized Axiom 3.

Critical Pair Lemma 16

The following expressions are equivalent:

`McKinley[x0]==(FirstLion[x1]||!McKinley[x2])`

Proof

Note that the input for the rule:

`FirstLion[x1_]||!McKinley[x1_]→McKinley[x0]`

contains a subpattern of the form:

`McKinley[x1_]`

which can be unified with the input for the rule:

`McKinley[x1_]↔McKinley[x2_]`

where these rules follow from Substitution Lemma 13 and Critical Pair Lemma 3 respectively.

Critical Pair Lemma 17

The following expressions are equivalent:

`(McKinley[x1]&&FirstLion[x1])==(McKinley[x1]&&McKinley[x0])`

Proof

Note that the input for the rule:

`x1_&&(x2_||!x1_)→x1&&x2`

contains a subpattern of the form:

x2_||!x1_

which can be unified with the input for the rule:

FirstLion[x1_]||!McKinley[x1_]→McKinley[x_0]

where these rules follow from Critical Pair Lemma 2 and Substitution Lemma 13 respectively.

Substitution Lemma 14

It can be shown that:

(McKinley[x1]&&FirstLion[x1])==McKinley[x_0]

Proof

We start by taking Critical Pair Lemma 17, and apply the substitution:

McKinley[x1_]&&McKinley[x2_]→McKinley[x_0]

which follows from Critical Pair Lemma 11.

Substitution Lemma 15

It can be shown that:

(FirstLion[x1]&&McKinley[x1])==McKinley[x_0]

Proof

We start by taking Substitution Lemma 14, and apply the substitution:

x1_&&x2_→x2&&x1

which follows from Equationalized Axiom 9.

Critical Pair Lemma 18

The following expressions are equivalent:

McKinley[x_0]==(FirstLion[x1]&&McKinley[x2])

Proof

Note that the input for the rule:

FirstLion[x1_]&&McKinley[x1_]→McKinley[x_0]

contains a subpattern of the form:

McKinley[x1_]

which can be unified with the input for the rule:

McKinley[x1_]↔McKinley[x2_]

where these rules follow from Substitution Lemma 15 and Critical Pair Lemma 3 respectively.

Critical Pair Lemma 19

The following expressions are equivalent:

McKinley[x_0]==(McKinley[x1]&&FirstLion[x2])

Proof

Note that the input for the rule:

FirstLion[x1_]&&McKinley[x2_]→McKinley[x_0]

contains a subpattern of the form:

FirstLion[x1_]&&McKinley[x2_]

which can be unified with the input for the rule:

x1_&&x2_↔x2_&&x1_

where these rules follow from Critical Pair Lemma 18 and Equationalized Axiom 9 respectively.

Critical Pair Lemma 20

The following expressions are equivalent:

McKinley[x_0] == (!McKinley[x1]||FirstLion[x2])

Proof

Note that the input for the rule:

FirstLion[x1_]||!McKinley[x2_]→McKinley[x_0]

contains a subpattern of the form:

FirstLion[x1_]||!McKinley[x2_]

which can be unified with the input for the rule:

x1_||x2_↔x2_||x1_

where these rules follow from Critical Pair Lemma 16 and Equationalized Axiom 3 respectively.

Critical Pair Lemma 21

The following expressions are equivalent:

(!McKinley[x1]||FirstLion[x2]) == (!McKinley[x1]||McKinley[x_0])

Proof

Note that the input for the rule:

!x1_||(x1_&&x2_)→!x1||x2

contains a subpattern of the form:

x1_&&x2_

which can be unified with the input for the rule:

McKinley[x1_]&&FirstLion[x2_]→McKinley[x_0]

where these rules follow from Critical Pair Lemma 15 and Critical Pair Lemma 19 respectively.

Substitution Lemma 16

It can be shown that:

McKinley[x_0] == (!McKinley[x1]||McKinley[x_0])

Proof

We start by taking Critical Pair Lemma 21, and apply the substitution:

!McKinley[x1_]||FirstLion[x2_]→McKinley[x_0]

which follows from Critical Pair Lemma 20.

Critical Pair Lemma 22

The following expressions are equivalent:

(x1&&x1&&x2) == (x1&&(!x1||x2))

Proof

Note that the input for the rule:

```
x1_&&(!x1_||x2_)→x1&&x2
```

contains a subpattern of the form:

```
!x1_||x2_
```

which can be unified with the input for the rule:

```
!x1_||(x1_&&x2_)→!x1||x2
```

where these rules follow from Critical Pair Lemma 5 and Critical Pair Lemma 15 respectively.

Substitution Lemma 17

It can be shown that:

```
(x1&&x1&&x2)==(x1&&x2)
```

Proof

We start by taking Critical Pair Lemma 22, and apply the substitution:

```
x1_&&(!x1_||x2_)→x1&&x2
```

which follows from Critical Pair Lemma 5.

Critical Pair Lemma 23

The following expressions are equivalent:

```
(x1&&x2)==(x1&&x2&&x1)
```

Proof

Note that the input for the rule:

```
x1_&&x1_&&x2_→x1&&x2
```

contains a subpattern of the form:

```
x1_&&x2_
```

which can be unified with the input for the rule:

```
x1_&&x2_↔x2_&&x1_
```

where these rules follow from Substitution Lemma 17 and Equationalized Axiom 9 respectively.

Substitution Lemma 18

It can be shown that:

```
McKinley[x0]==(McKinley[x0]||!McKinley[x1])
```

Proof

We start by taking Substitution Lemma 16, and apply the substitution:

```
x1_||x2_→x2||x1
```

which follows from Equationalized Axiom 3.

Critical Pair Lemma 24

The following expressions are equivalent:

```
x1==(McKinley[x0]&&x1)
```

Proof

Note that the input for the rule:

```
(x1_||!x1_)&&x2_→x2
```

contains a subpattern of the form:

`x1_||!x1_`

which can be unified with the input for the rule:

`McKinley[x0]||!McKinley[x1_]→McKinley[x0]`

where these rules follow from Critical Pair Lemma 4 and Substitution Lemma 18 respectively.

Critical Pair Lemma 25

The following expressions are equivalent:

`EatsOnlyStrawberries[x1]⩵McKinley[x1]`

Proof

Note that the input for the rule:

`McKinley[x0]&&x1_→x1`

contains a subpattern of the form:

`McKinley[x0]&&x1_`

which can be unified with the input for the rule:

`McKinley[x1_]&&EatsOnlyStrawberries[x2_]→McKinley[x2]`

where these rules follow from Critical Pair Lemma 24 and Critical Pair Lemma 10 respectively.

Substitution Lemma 19

It can be shown that:

`EatsOnlyStrawberries[x1]⩵McKinley[x2]`

Proof

We start by taking Critical Pair Lemma 3, and apply the substitution:

`McKinley[x1_]→EatsOnlyStrawberries[x1]`

which follows from Critical Pair Lemma 25.

Substitution Lemma 20

It can be shown that:

`EatsOnlyStrawberries[x1]⩵EatsOnlyStrawberries[x2]`

Proof

We start by taking Substitution Lemma 19, and apply the substitution:

`McKinley[x1_]→EatsOnlyStrawberries[x1]`

which follows from Critical Pair Lemma 25.

Substitution Lemma 21

It can be shown that:

`(EatsOnlyStrawberries[x0]&&x1)⩵x1`

Proof

We start by taking Critical Pair Lemma 24, and apply the substitution:

`McKinley[x1_]→EatsOnlyStrawberries[x1]`

which follows from Critical Pair Lemma 25.

Substitution Lemma 22

It can be shown that:

(EatsOnlyStrawberries[x1]&&FirstLion[x2])==McKinley[x_0]

Proof

We start by taking Critical Pair Lemma 19, and apply the substitution:

McKinley[x1_]→EatsOnlyStrawberries[x1]

which follows from Critical Pair Lemma 25.

Substitution Lemma 23

It can be shown that:

(EatsOnlyStrawberries[x1]&&FirstLion[x2])==EatsOnlyStrawberries[x_0]

Proof

We start by taking Substitution Lemma 22, and apply the substitution:

McKinley[x1_]→EatsOnlyStrawberries[x1]

which follows from Critical Pair Lemma 25.

Critical Pair Lemma 26

The following expressions are equivalent:

(x1&&EatsOnlyStrawberries[x_0])==(EatsOnlyStrawberries[x_0]&&x1)

Proof

Note that the input for the rule:

EatsOnlyStrawberries[x_0]&&x1_→x1

contains a subpattern of the form:

EatsOnlyStrawberries[x_0]&&x1_

which can be unified with the input for the rule:

x1_&&x2_&&x1_→x1&&x2

where these rules follow from Substitution Lemma 21 and Critical Pair Lemma 23 respectively.

Substitution Lemma 24

It can be shown that:

(x1&&EatsOnlyStrawberries[x_0])==x1

Proof

We start by taking Critical Pair Lemma 26, and apply the substitution:

EatsOnlyStrawberries[x_0]&&x1_→x1

which follows from Substitution Lemma 21.

Critical Pair Lemma 27

The following expressions are equivalent:

x1==(EatsOnlyStrawberries[x2]&&x1)

Proof

Note that the input for the rule:

Note that the input for the rule:

`EatsOnlyStrawberries[x0]&&x1_→x1`

contains a subpattern of the form:

`EatsOnlyStrawberries[x0]`

which can be unified with the input for the rule:

`EatsOnlyStrawberries[x1_]↔EatsOnlyStrawberries[x2_]`

where these rules follow from Substitution Lemma 21 and Substitution Lemma 20 respectively.

Critical Pair Lemma 28

The following expressions are equivalent:

`x1==(x1&&EatsOnlyStrawberries[x2])`

Proof

Note that the input for the rule:

`x1_&&EatsOnlyStrawberries[x0]→x1`

contains a subpattern of the form:

`EatsOnlyStrawberries[x0]`

which can be unified with the input for the rule:

`EatsOnlyStrawberries[x1_]↔EatsOnlyStrawberries[x2_]`

where these rules follow from Substitution Lemma 24 and Substitution Lemma 20 respectively.

Substitution Lemma 25

It can be shown that:

`FirstLion[x1]==EatsOnlyStrawberries[x0]`

Proof

We start by taking Substitution Lemma 23, and apply the substitution:

`EatsOnlyStrawberries[x1_]&&x2_→x2`

which follows from Critical Pair Lemma 27.

Substitution Lemma 26

It can be shown that:

`(EatsOnlyStrawberries[x0]||!McKinley[x0])==!(FirstLion[x0]&&!EatsOnlyStrawberries[x0])`

Proof

We start by taking Equationalized Hypothesis 1, and apply the substitution:

`a0||!a0→EatsOnlyStrawberries[x0]||!McKinley[x0]`

which follows from Substitution Lemma 2.

Substitution Lemma 27

It can be shown that:

`McKinley[x0]==!(FirstLion[x0]&&!EatsOnlyStrawberries[x0])`

Proof

We start by taking Substitution Lemma 26, and apply the substitution:

`EatsOnlyStrawberries[x0]||!McKinley[x0]→McKinley[x0]` which follows from Substitution Lemma 4.

Substitution Lemma 28

It can be shown that:

`McKinley[x0]==!(!EatsOnlyStrawberries[x0]&&FirstLion[x0])`

Proof

We start by taking Substitution Lemma 27, and apply the substitution:

`x1_&&x2_→x2&&x1`

which follows from Equationalized Axiom 9.

Substitution Lemma 29

It can be shown that:

`EatsOnlyStrawberries[x0]==!(!EatsOnlyStrawberries[x0]&&FirstLion[x0])`

Proof

We start by taking Substitution Lemma 28, and apply the substitution:

`McKinley[x1_]→EatsOnlyStrawberries[x1]`

which follows from Critical Pair Lemma 25.

Substitution Lemma 30

It can be shown that:

`EatsOnlyStrawberries[x0]==!(!EatsOnlyStrawberries[x0]&&EatsOnlyStrawberries[x0])`

Proof

We start by taking Substitution Lemma 29, and apply the substitution:

`FirstLion[x1_]→EatsOnlyStrawberries[x0]`

which follows from Substitution Lemma 25.

Substitution Lemma 31

It can be shown that:

`FirstLion[x0]==!(!EatsOnlyStrawberries[x0]&&EatsOnlyStrawberries[x0])`

Proof

We start by taking Substitution Lemma 30, and apply the substitution:

`EatsOnlyStrawberries[x0]→FirstLion[x0]`

which follows from Substitution Lemma 25.

Substitution Lemma 32

It can be shown that:

`FirstLion[x0]==!(!FirstLion[x0]&&EatsOnlyStrawberries[x0])`

Proof

We start by taking Substitution Lemma 31, and apply the substitution:

`EatsOnlyStrawberries[x0]→FirstLion[x0]`

which follows from Substitution Lemma 25.

Substitution Lemma 33

It can be shown that:

`True`

Proof

We start by taking Substitution Lemma 32, and apply the substitution:

`x1_&&EatsOnlyStrawberries[x2_]→x1`

which follows from Critical Pair Lemma 28.

Conclusion 1

We obtain the conclusion:

`True`

Proof

Take Substitution Lemma 33, and apply the substitution:

`x1_→x1`

which follows from Substitution Lemma 11.

References

Baader F and Nipkow T. (1999). *Term Rewriting and All That*. Cambridge University Press

Carroll L. (1896). *Symbolic Logic*. Reprinted by Clarkson N.Potter, 1977.

Knuth DE and Bendix PB. (1970). Simple word problems in universal algebras. In J. Leech, ed. *Computational Problems in Abstract Algebra*. Pergamon Press. pp. 263-297.

McCune WW and Padmanabhan R. (1996). *Automated Deduction in Equational Logic and Cubic Curves.* Springer.

Suppes P. (1972). *Axiomatic Set Theory.* Dover.

Twain M. (1962). *Letters From the Earth*. B. DeVoto (ed.) Harper and Row.

Wolfram Research (2020). *Mathematica v12 .1 .0* Home Edition. https : // www.wolfram.com/mathematica - home - edition /. Accessed 19 March 2020.