

Improving Automated Feedback

Building a Rule Feedback Generator

Eric Bouwers

September 27, 2007

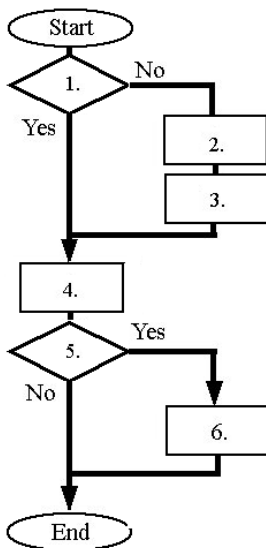


(Generic) Outline

- 1 The problem
- 2 Our solution
- 3 Evaluation
- 4 Conclusion



Procedural skills



1. Are the denominators the same?
If yes, skip to step 4.

2. Find the lowest common denominator.

3. Convert both fractions to the common denominator.

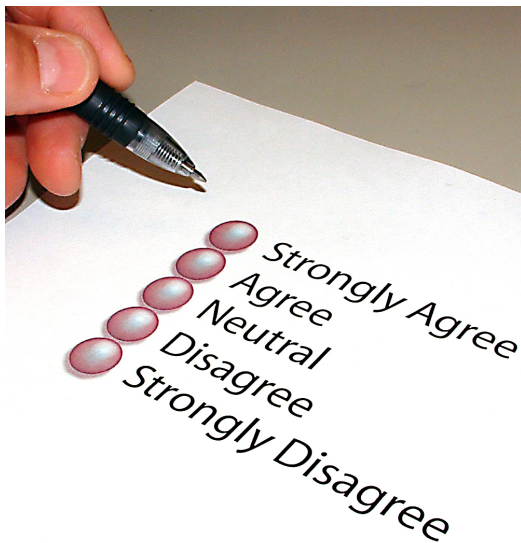
4. Add the numerators, and place the sum over the common denominator.

5. Can the answer be simplified? (If no, you are finished.)

6. Simplify the answer.



Feedback



Feedback in Education



Feedback in Education



Current solutions

MathXpert Calculus Assistant - [Solve a linear equation (practice) Problem 5]

File Edit Algebra Pre-Calculus Calculus Graph View Options Window Help

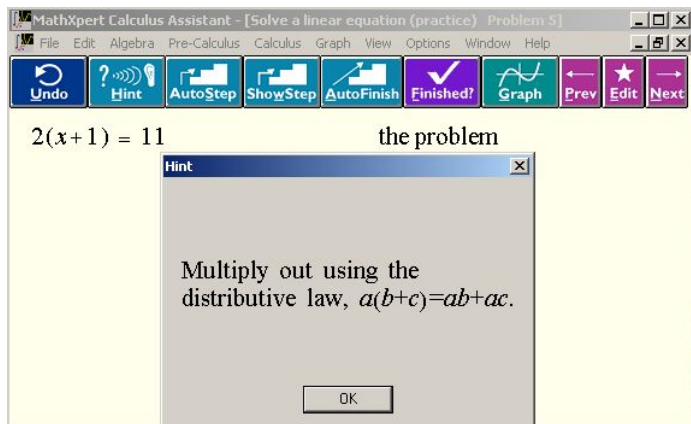
Undo Hint AutoStep ShowStep AutoFinish Finished? Graph Prev Edit Next

$2(x+1) = 11$

subtract ? from both sides
 divide both sides by ?
 make a substitution, $u = ?$
 express as polynomial
 $a(b+c) = ab+ac$
 $ab = ba$



Current solutions



Current solutions

Apluxix - Student : student

File Edit Step Calculation Parameters Past activities Help

Self-correction End of the exercise 1/1 Stop the self-correction

$2(x+1) = 11$

$2x+2 = 11$

~~$2x = 13$~~

$x = 6.5$

Virtual keyboard

Undo Redo Cut Copy Paste

or	=	≠	()	$\frac{\Box}{\Box}$	\Box^2	+	x	7	8	9	a	\leftarrow	\rightarrow
{	≤	<	()	\Box^0	-	y	4	5	6	b	\uparrow	\downarrow	
≥	>)	/	$\sqrt{\Box}$	x	0	1	2	3	.				

State : Well-formed expression



Current solutions

Aplusix - Student : student - exercise (CHABRO-1.0 A2)

File Edit Step Calculation Parameters Past activities Help

exercises (list) End of the exercise 1/10 Stop the list The Map

Calculate

$10 (1+2.6)$

\times

$10+0.26$

\parallel

10.26

APLUSIX

There are errors on the path going to 10.26 .

Ok Solution

State : Well-formed expression



Current solutions

A screenshot of a digital whiteboard interface. The top toolbar contains various mathematical symbols and buttons: $\sqrt{\quad}$, \square° , $\frac{\square}{\square}$, (\square) , $+$, $-$, \times , \div , $\cancel{\square}$, $\square\square$, \downarrow , **Terug**, and **Hulp**. The main area contains the following text:

$$4x + 6 = 2$$

$$4x = -4$$

X

$$x = 1$$

To the right of the equations, there are annotations:

- A right curly brace next to the first equation, with $- 6$ written next to it.
- A right curly brace next to the second equation, with the text "gelijkwaard met:" written next to it.



Current solutions

Equation Solver

Working area

$2^*x = 3 + 2^*3^*x - z - 5$
 $y = 3^*x - z - 5$
 $2^*z = 3^*x$

Submit

Undo

Hint

Progress:

Number of solved variables (goal:3):
0

Number of occurrences of variables (goal:3):
7

Size of left-hand side expressions (goal:3):
7

History

$2^*x = 3 + 2^*y$
 $y = 3^*x - z - 5$
 $2^*z = 3^*x$

 $2^*x = 3 + 2^*y$
 $y + 5 = 3^*x - z$
 $2^*z = 3^*x$

Feedback

Error: Since variable y has disappeared from the equation
 $2^*x = 3 + 2^*3^*x - z - 5$
 we assume you have tried to apply the substitution rule.
 Correctly applying the substitution rule for y gives
 $2^*x = 3 + 2^*(3^*x - z - 5)$
 Is this what you intended?

Welcome to Equation Solver





Current solutions

http://proxy.activemath.org - ActiveMath - Exercise - Mozilla Firefox

Print | Help

Le Math Active

Exercise

  **The derivative of a sum** ★★

The derivative of the function $f(x) = \frac{2}{3} \cdot x^2 \cdot y + 2 \cdot x^{-3}$ is

$f'(x) = \frac{4}{3} \cdot x$.

Sorry, no. The difficulties occur probably with the handling of the variable y and of the negative exponent.

Regard the variable y , which is to be considered as a constant, because we differentiate a function in x in variable x like usually.

The derivative of the function $f(x) = \frac{2}{3} \cdot x^2 \cdot y + 2 \cdot x^{-3}$ is

$f'(x) =$

☐ Activate Input Editor

Evaluate Input syntax help Hint



Current solutions

Mathematik heute - Bruchrechnung

Mathematik
heute

Addiere die Brüche:

Hilfe
Inhalt
Lexikon

$$\frac{3}{5} + \frac{1}{8}$$

$$+ \frac{3}{5} + \frac{1}{8}$$

$$= \frac{4}{13}$$

OK?

Fehler in der 2. Zeile:
Du verwendest eine falsche Regel bei der [Addition](#) der Brüche.
Beachte, dass du die Zähler nur addieren darfst, wenn die Nenner [gleichnamig](#) sind.
Korrigiere deine Rechnung.

Tipp

3.14(0)



Synopsis

- Feedback is important in learning
- Personal feedback is not feasible without (computerized) help
- Current solutions are either:
 - Only Correct/Incorrect solutions
 - Result of extensive research



Goal

Research question

How can we make the generation of high-quality, domain-specific feedback easier?



Goal

Research question

How can we make the generation of high-quality, domain-specific feedback easier?

Basic idea

Create a generic framework which separates the knowledge of rule-feedback generation from knowledge about the domain.



Requirements

- Should always be able to produce basic feedback
-
-
-



Requirements

- Should always be able to produce basic feedback
- More detailed/complete input leads to better feedback
-
-



Requirements

- Should always be able to produce basic feedback
- More detailed/complete input leads to better feedback
- Can be instantiated on different domains (with little effort)
-

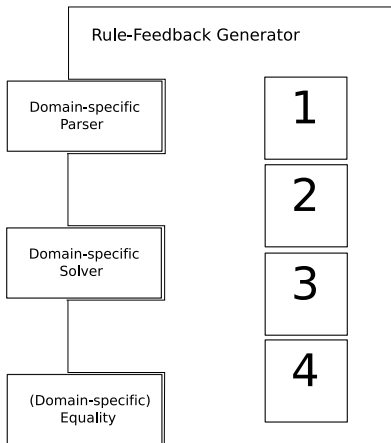


Requirements

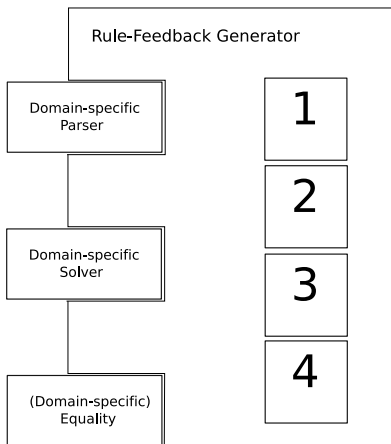
- Should always be able to produce basic feedback
- More detailed/complete input leads to better feedback
- Can be instantiated on different domains (with little effort)
- Adaptable to a single class-room (or student!)



Overall design



Overall design



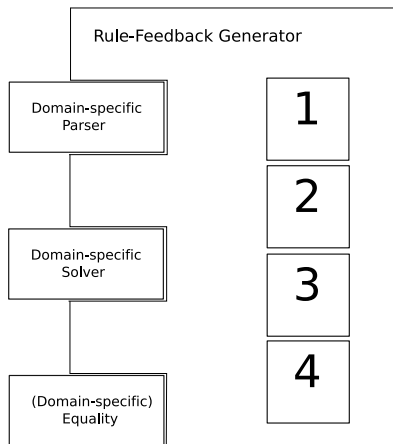
Phase 1:

Input:
CT + PT

Output:
Correct/Incorrect message



Overall design



Phase 2:

Input:

CT + PT

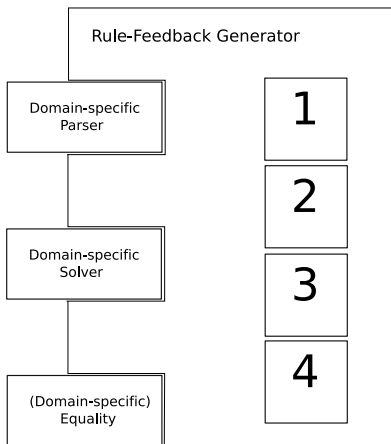
+ Student rule

Output:

Correct/Incorrect message-tuple



Overall design



Phase 3:

Input:

CT + PT

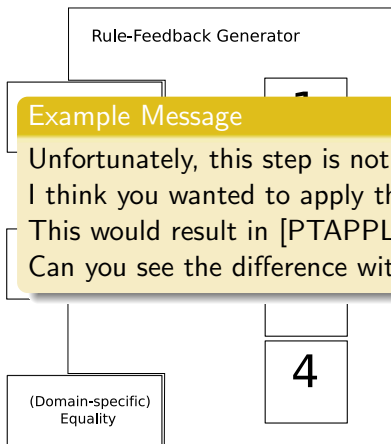
+ Allowed rules

Output:

Configurable message, phase level



Overall design



Phase 3:

Example Message

Unfortunately, this step is not correct.

I think you wanted to apply this rule: [RULE].

This would result in [PTAPPLIEDRULE].

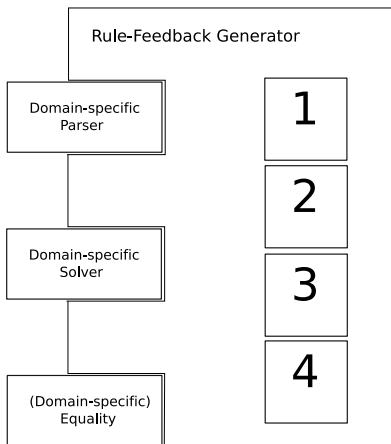
Can you see the difference with [CT]?

Output:

Configurable message, phase level



Overall design



Phase 4:

Input:

CT + PT

+ Allowed rules + Buggy rules

Output:

Configurable message, rule level



First phase

Input: PT, CT

Output: Correct or Incorrect

Algorithm: Solve both terms and check whether their results are semantically equal.



First phase

Input: PT, CT

Output: Correct or Incorrect

Algorithm: Solve both terms and check whether their results are semantically equal.

Implementation

```
firstPhase :: RFG a => RFGSolver a -> RFGEqual a
              -> a -> a -> String

firstPhase solve equal pt ct =
    let resultCt = solve ct
        resultPt = solve pt
    in if equal resultPt resultCt
        then getConfig Correct
        else getConfig Incorrect
```



First phase

Input: PT, CT

Output: Correct or Incorrect

Algorithm: Solve both terms and check whether their results are semantically equal.

Implementation

```
firstPhase :: RFG a => RFGSolver a -> RFGEqual a
            -> a -> a -> String

firstPhase solve equal pt ct =
    let resultCt = solve ct
        resultPt = solve pt
    in if equal resultPt resultCt
       then getConfig Correct
       else getConfig Incorrect
```



First phase

Input: PT, CT

Output: Correct or Incorrect

Algorithm: Solve both terms and check whether their results are semantically equal.

Implementation

```
firstPhase :: RFG a => RFGSolver a -> RFGEqual a
              -> a -> a -> String

firstPhase solve equal pt ct =
    let resultCt = solve ct
        resultPt = solve pt
    in if equal resultPt resultCt
       then getConfig Correct
       else getConfig Incorrect
```



Third phase

Input: PT, CT and allowed rules

Output: Configurable message



Third phase

Input: PT, CT and allowed rules

Output: Configurable message

Algorithm

Given two terms, calculate the rewrite rule δ between these terms.

Determine which allowed rule is closest to δ .

Apply rules from the set to the PT if possible and recurse.

The result is the rule which is a closest match.



Third phase

Input: PT, CT and allowed rules

Output: Configurable message

Algorithm

Given two terms, **calculate the rewrite rule δ** between these terms.

Determine which allowed rule is closest to δ .

Apply rules from the set to the PT if possible and recurse.

The result is the rule which is a closest match.



Third phase

Input: PT, CT and allowed rules

Output: Configurable message

Algorithm

Given two terms, calculate the rewrite rule δ between these terms.

Determine which allowed rule is **closest** to δ .

Apply rules from the set to the PT if possible and recurse.

The result is the rule which is a closest match.



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

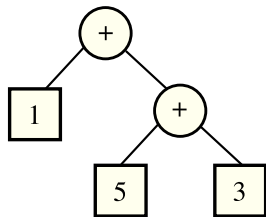
$$CT = 3 + 6$$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

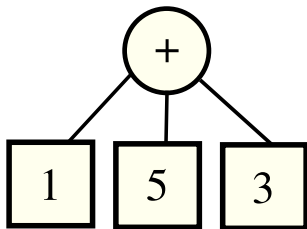
$$CT = 3 + 6$$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

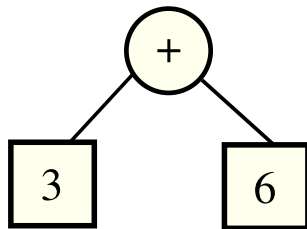
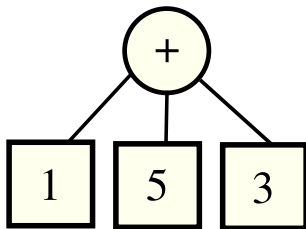
$$CT = 3 + 6$$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

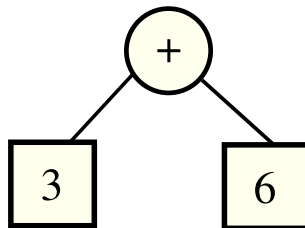
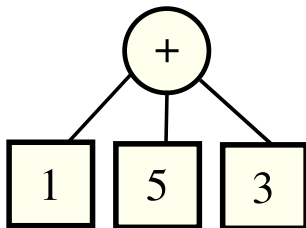
$$CT = 3 + 6$$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

$$CT = 3 + 6$$



Non-Equal

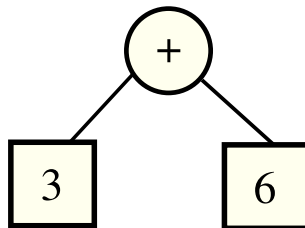
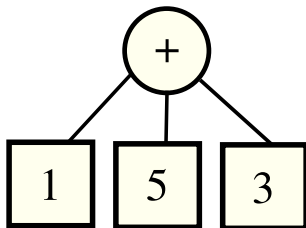
$([1,5,3],[3,6,-])$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

$$CT = 3 + 6$$



Equal, Associative

$([1,5,3],[3,6,-]) , ([1,5,3],[-,3,6])$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

$$CT = 3 + 6$$

Equal, Associative and Commutative

1

$([1,5,3],[3,6,-]), ([1,5,3],[-,3,6]), ([1,5,3],[6,3,-]),$
 $([1,5,3],[-,6,3]), ([1,3,5],[3,6,-]), ([1,3,5],[-,3,6]),$
 $([1,3,5],[6,3,-]), ([1,3,5],[-,6,3]), ([3,1,5],[3,6,-]),$
 $([3,1,5],[-,3,6]), ([3,1,5],[6,3,-]), ([3,1,5],[-,6,3]),$
 $([3,5,1],[3,6,-]), ([3,5,1],[-,3,6]), ([3,5,1],[6,3,-]),$
 $([3,5,1],[-,6,3]), ([5,3,1],[3,6,-]), ([5,3,1],[-,3,6]),$
 $([5,3,1],[6,3,-]), ([5,3,1],[-,6,3]), ([5,1,3],[3,6,-]),$
 $([5,1,3],[-,3,6]), ([5,1,3],[6,3,-]), ([5,1,3],[-,6,3])$

6



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

$$CT = 3 + 6$$

Equal, Associative and Commutative

([1,5,3],[3,6,-]), ([1,5,3],[-,3,6]), ([1,5,3],[6,3,-]),
 ([1,5,3],[-,6,3]), ([1,3,5],[3,6,-]), ([1,3,5],[-,3,6]),
 ([1,3,5],[6,3,-]), ([1,3,5],[-,6,3]), ([3,1,5],[3,6,-]),
 ([3,1,5],[-,3,6]), ([3,1,5],[6,3,-]), ([3,1,5],[-,6,3]),
 ([3,5,1],[3,6,-]), ([3,5,1],[-,3,6]), ([3,5,1],[6,3,-]),
 ([3,5,1],[-,6,3]), ([5,3,1],[3,6,-]), ([5,3,1],[-,3,6]),
 ([5,3,1],[6,3,-]), ([5,3,1],[-,6,3]), ([5,1,3],[3,6,-]),
 ([5,1,3],[-,3,6]), ([5,1,3],[6,3,-]), ([5,1,3],[-,6,3])

1

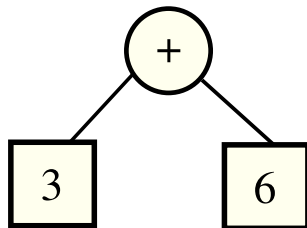
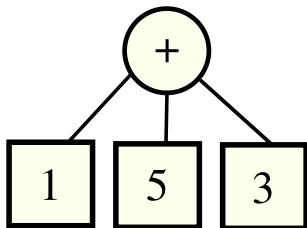
6



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

$$CT = 3 + 6$$



Result:

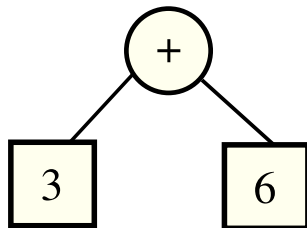
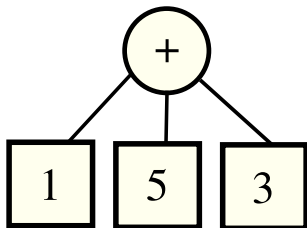
$([1,5],[6])$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

$$CT = 3 + 6$$



Result:

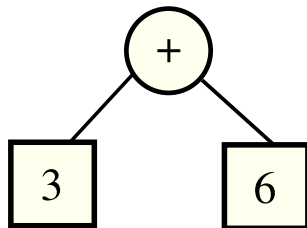
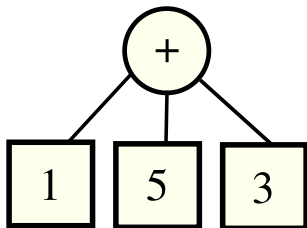
$$\delta = 1 + 5 \Rightarrow +6$$



Calculating rewrite rules

$$PT = 1 + 5 + 3$$

$$CT = 3 + 6$$



Result:

$$\delta = 1 + 5 \Rightarrow 6$$



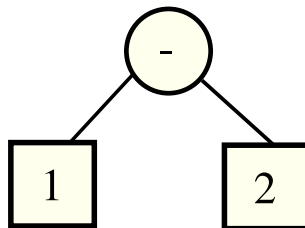
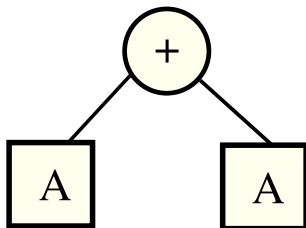
Defining the distance

Distance between rules:



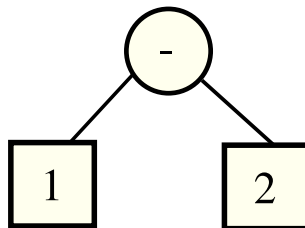
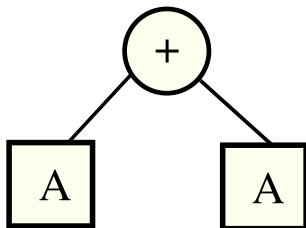
Defining the distance

Distance between terms:



Defining the distance

Distance between terms:

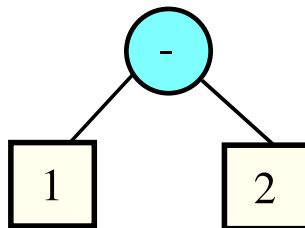
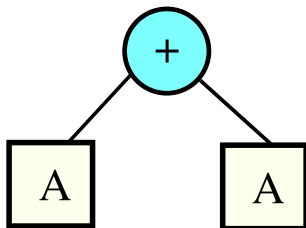


Environment = {}

Distance = 0

Defining the distance

Distance between terms:

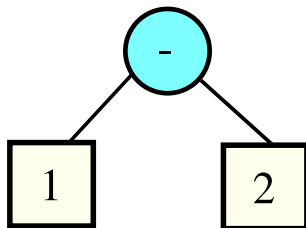
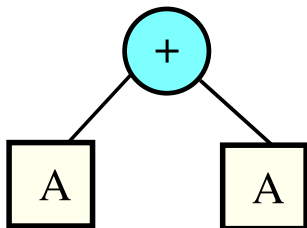


Environment = {}

Distance = 0

Defining the distance

Distance between terms:

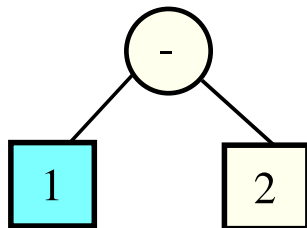
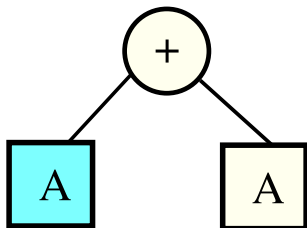


Environment = {}

Distance = 2

Defining the distance

Distance between terms:

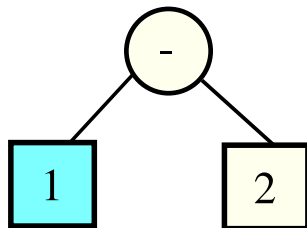
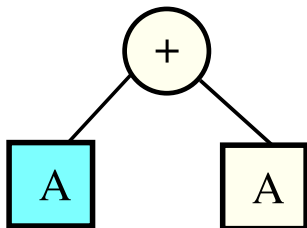


Environment = {}

Distance = 2

Defining the distance

Distance between terms:

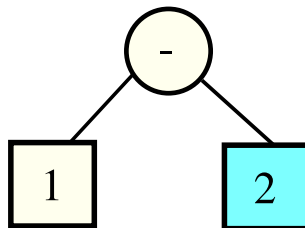
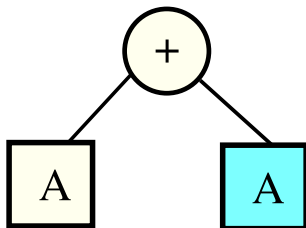


Environment = $\{(A, 1)\}$

Distance = 2

Defining the distance

Distance between terms:

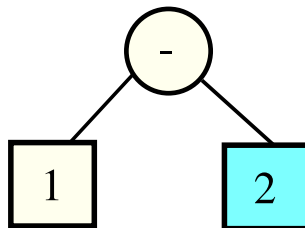
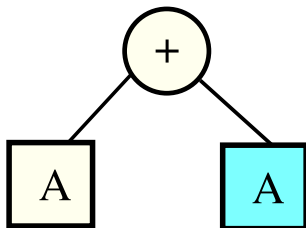


Environment = $\{(A, 1)\}$

Distance = 2

Defining the distance

Distance between terms:



Environment = $\{(A, 1)\}$

Distance = 4



Example

$$PT = (a \vee b) \rightarrow c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {

ImpElimination :

$$A \rightarrow B \Rightarrow \neg(A) \vee B$$

, MorganOr :

$$\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$$

}



Example

$$PT = (a \vee b) \rightarrow c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
 }

Calculated Rule

$$\delta = (a \vee b) \rightarrow c \Rightarrow (\neg a \wedge b) \vee c$$



Example

$$PT = (a \vee b) \rightarrow c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
 }

Calculated Rule

$$\delta = (a \vee b) \rightarrow c \Rightarrow (\neg a \wedge b) \vee c$$

Distances

$$\text{dist}(\delta, \text{ImpElimination}) = 9$$

$$\text{dist}(\delta, \text{MorganOr}) = 15$$



Example

$$PT = (a \vee b) \rightarrow c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {

ImpElimination :

$$A \rightarrow B \Rightarrow \neg(A) \vee B$$

, MorganOr :

$$\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$$

}

Calculated Rule

$$\delta = (a \vee b) \rightarrow c \Rightarrow (\neg a \wedge b) \vee c$$

Distances

$$\text{dist}(\delta, \text{ImpElimination}) = 9$$

$$\text{dist}(\delta, \text{MorganOr}) = 15$$



Example

$$PT' = \neg(a \vee b) \vee c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
}



Example

$$PT' = \neg(a \vee b) \vee c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {

ImpElimination :

$$A \rightarrow B \Rightarrow \neg(A) \vee B$$

, MorganOr :

$$\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$$

}

Calculated Rule

$$\delta = \neg(a \vee b) \Rightarrow (\neg a \wedge b)$$



Example

$$PT' = \neg(a \vee b) \vee c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
 }

Calculated Rule

$$\delta = \neg(a \vee b) \Rightarrow (\neg a \wedge b)$$

Distances

$$\text{dist}(\delta, \text{ImpElimination}) = 9$$

$$\text{dist}(\delta, \text{MorganOr}) = 4$$



Example

$$PT' = \neg(a \vee b) \vee c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
 }

Calculated Rule

$$\delta = \neg(a \vee b) \Rightarrow (\neg a \wedge b)$$

Distances

$$\text{dist}(\delta, \text{ImpElimination}) = 9$$

$$\text{dist}(\delta, \text{MorganOr}) = 4$$



Example

$$PT'' = (\neg a \wedge \neg b) \vee c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
 }



Example

$$PT'' = (\neg a \wedge \neg b) \vee c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
 }

Calculated Rule

$$\delta = \neg b \Rightarrow b$$



Example

$$PT'' = (\neg a \wedge \neg b) \vee c$$

$$CT = (\neg a \wedge b) \vee c$$

Rules = {
 ImpElimination :
 $A \rightarrow B \Rightarrow \neg(A) \vee B$
 , MorganOr :
 $\neg(A \vee B) \Rightarrow \neg(A) \wedge \neg(B)$
 }

Calculated Rule

$$\delta = \neg b \Rightarrow b$$

Distances

$$\text{dist}(\delta, \text{ImpElimination}) = 9$$

$$\text{dist}(\delta, \text{MorganOr}) = 10$$



Fourth phase

M. Hennecke

Online Diagnose in intelligenten mathematischen Lehr-Lern-Systemen.

PhD thesis, Hildesheim University, 1999.



Fourth phase

M. Hennecke

Online Diagnose in intelligenten mathematischen Lehr-Lern-Systemen.

PhD thesis, Hildesheim University, 1999.

Example

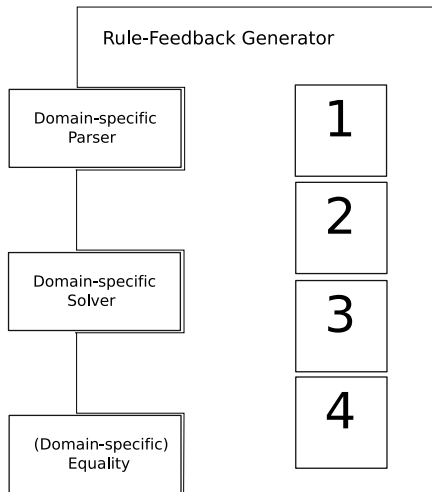
$$\frac{A}{B} + \frac{C}{D} \Rightarrow \frac{A+C}{B+D}$$

: AddError, You can only add fractions with equal denominators

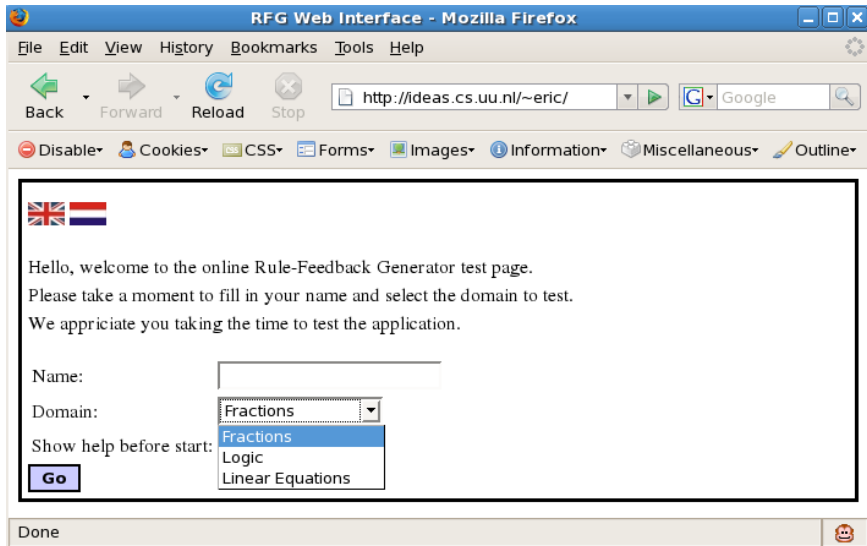
: 0.3



Overview



Receiving feedback



Feedback message

Graphical

Merged

Textual

$$p \rightarrow (q \rightarrow p)$$

Unfortunately, this step is not correct.

I think you wanted to apply this rule: $(A \rightarrow B) \Rightarrow (\neg(A) \vee B) : (\text{ImpElem})$.

It can be applied in multiple ways resulting in, for example, $(\neg(p) \vee (q \rightarrow p))$.

Can you see the difference with $(\neg(p) \wedge (q \rightarrow p))$?

$$\neg p \wedge (q \rightarrow p)$$

Answer here



Wrong Guesses

EqElem	: $A \leftrightarrow B$	\rightarrow	$(A \wedge B) \vee (\neg(A) \wedge \neg(B))$
ImpElem	: $A \rightarrow B$	\rightarrow	$\neg(A) \vee B$
MorganA	: $\neg(A \wedge B)$	\rightarrow	$\neg(A) \vee \neg(B)$
NotNot	: $\neg\neg A$	\rightarrow	A
DistAO	: $A \wedge (B \vee C)$	\rightarrow	$(A \wedge B) \vee (A \wedge C)$
IdemA	: $A \wedge A$	\rightarrow	A



Wrong Guesses

EqElem	: $A \leftrightarrow B$	\rightarrow	$(A \wedge B) \vee (\neg(A) \wedge \neg(B))$
ImpElem	: $A \rightarrow B$	\rightarrow	$\neg(A) \vee B$
MorganA	: $\neg(A \wedge B)$	\rightarrow	$\neg(A) \vee \neg(B)$
NotNot	: $\neg\neg A$	\rightarrow	A
DistAO	: $A \wedge (B \vee C)$	\rightarrow	$(A \wedge B) \vee (A \wedge C)$
IdemA	: $A \wedge A$	\rightarrow	A

PT	CT	Expected rule	Actual output
$a \rightarrow (b \wedge c)$	$a \vee (b \wedge c)$	ImpElem	NotNot
$\neg\neg a \leftrightarrow b$	$(\neg a \wedge b) \vee (\neg\neg\neg a \vee \neg b)$	EqElem	MorganA
$a \vee c \leftrightarrow b$	$(a \vee c \wedge b) \vee \neg(a \vee c) \wedge \neg b$	EqElem	IdemA
$a \leftrightarrow b \wedge (c \vee d)$	$(a \leftrightarrow b \wedge c) \vee (a \leftrightarrow b \wedge d)$	EqElem	DistAO



Definition of Rules

$$\text{Multiplication} : (A/B) * (C/D) \Rightarrow (A * C) / (B * D)$$

$$\text{Division} : (A/B) / (C/D) \Rightarrow (A/B) * (D/C)$$

$$\text{SameMixed} : A|B/B \Rightarrow A + 1$$

$$\text{AddFractions} : (A/B) + (C/B) \Rightarrow (A + C) / B$$

$$\text{SubFractions} : (A/B) - (C/B) \Rightarrow (A - C) / B$$



Definition of Rules

$$\text{Multiplication} : (A/B) * (C/D) \Rightarrow (A * C) / (B * D)$$

$$\text{Division} : (A/B) / (C/D) \Rightarrow (A/B) * (D/C)$$

$$\text{SameMixed} : A|B/B \Rightarrow A + 1$$

$$\text{AddFractions} : (A/B) + (C/B) \Rightarrow (A + C) / B$$

$$\text{SubFractions} : (A/B) - (C/B) \Rightarrow (A - C) / B$$

$$\text{SolveMin} : A - B \Rightarrow C \text{ where } C := \text{solve}(A - B)$$

$$\text{SolveAdd} : A + B \Rightarrow C \text{ where } C := \text{solve}(A + B)$$

$$\text{SolveMul} : A * B \Rightarrow C \text{ where } C := \text{solve}(A * B)$$



Conclusion

- Partial/Configurable approach is possible
- Adding an extra domain takes little effort
- Combining techniques leads to surprisingly good results
- Already useful, both for research as well as practical



Resources

Thesis page:

<http://www.cs.uu.nl/wiki/Students/EricBouwersThesisPage>

Prototype:

<http://ideas.cs.uu.nl/~eric/>

Contact me:

EricBouwers@gmail.com

