

# Um resolvidor de equações algébricas como ferramenta de apoio à sala de aula no ensino de equações algébricas

Henrique M. Seffrin, Geiseane Rubi, Talvany Carlotto,  
Gabriel Mello, Patrícia A. Jaques

Programa Interdisciplinar de Pós Graduação em Computação Aplicada (PIPICA)  
Universidade do Vale do Rio dos Sinos (UNISINOS)

`hseffrin@terra.com.br`, `pjaques@unisinossinos.br`

***Abstract.** In view of the difficulties faced by Brazilian students in elementary algebra, an equation solver is proposed, which can be used as an auxiliary class tool in algebra learning. This paper describes the implementation and the behavior of this solver, including its architecture, scenario and evaluation with local elementary and high school math teachers.*

***Resumo.** Tendo em vista as dificuldades enfrentadas por alunos brasileiros em álgebra elementar, foi implementado um resolvidor de equações algébricas, podendo este ser utilizado como uma ferramenta de auxílio no ensino de álgebra. Este artigo descreve o funcionamento e implementação deste resolvidor, incluindo a sua arquitetura, um cenário de execução e os testes, ao qual ele foi submetido, com professores de matemática do ensino fundamental e médio.*

## 1. Introdução

Estudos mostram que os alunos brasileiros têm grande dificuldade em assuntos relacionados à álgebra: “Nos resultados do Sistema Nacional de Avaliação da Educação Básica (SAEB), por exemplo, os itens referentes à álgebra raramente atingem um índice de 40% de acerto em muitas regiões do país” [Sef 1998]. Tendo em vista esta problemática, foi proposto um sistema que auxiliasse os alunos na resolução de equações algébricas, um importante conteúdo da álgebra elementar.

Este resolvidor foi implementado tendo em vista que um dos maiores problemas enfrentados pelos alunos é como chegar ao resultado da equação e não qual o resultado desta. Desta maneira, este resolvidor não só resolve determinada equação, mas também mostra ao aluno as operações e decisões tomadas, passo-a-passo, para resolver a equação. Por este motivo o resolvidor pode ser utilizado como uma ferramenta auxiliar no ensino de equações algébricas.

O resolvidor foi implementado na linguagem de programação Java. Todo o conhecimento procedural deste sistema de como resolver uma equação foi implementado como um sistema especialista baseada em regras de produção, o que tem se provado mais adequado para o conhecimento procedural matemático [Anderson *et al.* 1992]. Para implementação do sistema especialista foi utilizada a *shell* Drools, onde cada passo da resolução de uma equação foi codificado para uma ou mais regras nesta *shell*.

Este artigo está organizado como segue. A Seção 2 descreve o ensino de álgebra descrevendo os problemas encontrados e exemplificando alguns erros mais comuns dos alunos. A Seção 3 aborda a *shell* de sistema especialista utilizada na implementação do resolvedor, descrevendo sua arquitetura. A Seção 4 compara este resolvedor com outros disponíveis pela *web*. A Seção 5 apresenta o resolvedor, descrevendo sua implementação, estruturas de dados utilizadas, regras implementadas e um cenário de execução. A Seção 6 relata a avaliação qualitativa a qual o resolvedor foi submetido, por parte de professores de matemática de escolas de ensino médio e fundamental. Fechando este artigo, a Seção 7 descreve os trabalhos futuros que serão realizados com o resolvedor.

## 2. Ensino de Álgebra

A disciplina de Matemática, em geral, é considerada desafiadora para muitos alunos. A álgebra, área da matemática julgada muito abstrata, tanto na Educação Básica quanto na Superior, provoca a ruptura de conceitos já internalizados pelos alunos [Cury e Konzen 2006], tendo em vista o forte desenvolvimento aritmético dos mesmos. Com isto, pesquisadores e professores buscam alternativas para aprimorar o ensino do conteúdo de Equações Algébricas, amenizando as dificuldades encontradas pelos alunos.

Como estão habituados somente com a Aritmética, ao se depararem com uma equação algébrica, os alunos enfrentam grandes dificuldades de interpretação, pois têm como único objetivo o cálculo e não compreendem o que significa a incógnita, ou ainda, o que precisa ser descoberto. A compulsão por calcular impede os discentes de pensarem algebricamente, ou seja, analisar e interpretar os dados da equação, para então, estabelecerem estratégias para a resolução.

Muitos alunos utilizam técnicas mecanizadas que vão desde a exaustão até mesmo às falsas regras, como, “passar para o outro lado com o sinal invertido”. Realizam a operação inversa, mas invertem o sinal, demonstrando apenas que aplicaram a regra. Conforme Lima (2007, p. 97), essas técnicas são procedimentos ligados ao mundo simbólico, que representam uma compreensão de propriedades matemáticas, que pode ter ocorrido por meio da repetição, mas nem sempre estão de acordo. É plausível que essas regras tenham significado matemático e algébrico, tornando-se apenas “macetes”.

Portanto é de extrema importância que o professor, enquanto orientador da aprendizagem, não dê ênfase em um único método de resolução, mas que desenvolva o pensamento flexível, mostrando as diversas possibilidades relacionadas à aplicação das equações algébricas, dando significado às regras utilizadas pelos alunos. Faz-se necessário fortalecer conteúdos base para a aprendizagem das equações. E o docente deve utilizar diferentes metodologias e instrumentos didáticos, que os estimulem a aprendizagem, proporcionando-os experiências matemáticas para que se tornem cidadãos independentes e críticos.

## 3. *Shell* de Sistema Especialista

*Shells* de sistemas especialistas são ferramentas desenvolvidas para serem utilizadas na construção de sistemas especialistas. Antes da existência destes, os sistemas especialistas eram sempre criados do zero, geralmente em LISP ou Prolog. Porém, foi se percebendo que a maioria dos sistemas especialistas utilizava fatos e regras para se representar o conhecimento e um interpretador para executá-los, e, então, começaram a

separar o interpretador dos fatos e regras. A esse interpretador é dado o nome de *shell*.

Basicamente toda *shell* de sistema especialista possui os seguintes módulos [Nikolopoulos 1997]:

- a base de conhecimento estruturada;
- o mecanismo de inferência;
- ferramentas de aquisição de conhecimento;
- *interface* do usuário;
- *interface* com outros *softwares* (por exemplo: planilhas, bancos de dados, linguagens de programação, etc.);

Tendo-se em vista que essas *shells* possuem toda a arquitetura de sistemas especialistas, o uso delas facilita o desenvolvimento de tais sistemas. É tarefa do desenvolvedor implementar o seu núcleo, ou seja, a formalização do conhecimento do sistema através de fatos e regras.

### 3.1. Drools

O JBoss Drools (<http://www.jboss.org/drools/downloads.html>) é uma *shell* de sistema especialista baseada em regras de produção desenvolvido em Java, sob a licença *Open Source*. O Drools utiliza fatos e regras para representar o conhecimento do sistema. Sua arquitetura é dividida em **Memória de Trabalho** (*Working Memory*), local onde os fatos são armazenados com a possibilidade de serem removidos ou alterados; **Base de Conhecimento** (*Production Memory*), onde são armazenadas as regras de produção; e **Agenda**, responsável pela ordem de ativação das regras e também pela resolução de conflitos [Proctor *et al.* 2008].

```
rule "Divide por 1"
  dialect "java"
  when
    e: Expression( );
    eval (e.getNewExpression().contains("="));
    eval (canDivideBy1(e.getRoot()));
  then
    System.out.println("# Dividir por 1");
    e.removeDaArvore(e.find("/").getDir());
    e.setNewRoot(e.getRoot());
    update(e);
end;
```

Figura 1: Exemplo de Regra

As Regras são o meio pelo qual o conhecimento do sistema é expresso e, geralmente, possuem o formato *if* <condição> *else* <ação>. Sua estrutura basicamente divide-se em duas partes: a *Left Hand Side (LHS)* e a *Right Hand Side (RHS)*. A LHS é a área onde são escritas as condições para que a regra execute, ou seja, os fatos que devem ser verdadeiros ou que devem existir na Memória de Trabalho para que a regra seja chamada. Ela é representada pela cláusula *when* no Drools, correspondendo ao *if* utilizado por outras *shells*. Já a RHS é onde são escritas as ações a serem realizadas quando as condições forem verdadeiras, como, por exemplo, valores a serem calculados, fatos a serem inseridos/removidos/alterados da Memória de Trabalho. Esta é representada pela cláusula *then*. Na Figura 1 pode ser visualizado um exemplo de regra representada segundo a sintaxe do Drools.

Os Fatos são os dados sobre os quais as regras trabalham, ou seja, as conclusões chegadas a partir da execução de uma determinada regra ou uma hipótese a ser testada pelas regras. Eles são representados como instâncias de objetos na Memória de Trabalho.

A Agenda é responsável pela ativação das regras, uma vez que um fato inserido na Memória de Trabalho é testado com as condições das regras. Este é realizado através do algoritmo ReteOO, uma versão otimizada do algoritmo Rete [Nikolopoulos 1997]. Quando uma condição se torna verdadeira sobre este fato, é criado um registro de ativação na Agenda contendo uma referência para a regra e os fatos que a ativaram (em caso de haver mais de um). Podem ocorrer casos, e geralmente ocorrem, de um fato ativar mais de uma regra. Nesses casos, a Agenda utiliza uma estratégia de resolução de conflitos. O Drools por padrão utiliza o DephtConflictResolver, onde as regras com maior prioridade (*Salience*) são selecionadas primeiramente. Para regras com mesmas prioridades, é selecionada aquela que foi adicionada na Agenda por último, isto é, o algoritmo utiliza uma política LIFO (*Last In First Out*) [Lafore 2005].

#### 4. Trabalhos Relacionados

Nesta seção é descrita uma comparação do resolvidor proposto com outros resolvidores de equações *web*.

O resolvidor proposto foi implementado em Java e possui assim a vantagem de ser multiplataforma, ou seja, funciona em qualquer sistema operacional que seja compatível com a máquina virtual Java. Como todo o seu conhecimento, em relação à resolução de uma equação, foi implementado em uma *shell* de sistema especialista baseado em regras de produção, o resolvidor pode ser expandido simplesmente escrevendo-se novas regras, seja elas para a resolução de outros tipos de equações ou para detecção de concepções errôneas (*misconceptions*), e o fato do seu código ser aberto (*Open-Source*) contribui para isso.

Dentre os resolvidores de equações disponíveis na internet, temos, como exemplo, o Álgebra Help (<http://www.algebrahelp.com/calculators/equation/>) e o webMath (<http://www.webmath.com/>). O primeiro resolve passo-a-passo uma equação, porém não é capaz de trabalhar com a fórmula de Bhaskara e com frações. Já o segundo resolve equações apenas por assuntos separados, por exemplo, há uma seção para trabalhar apenas com distributiva. Se o usuário entrar com uma equação que necessite apenas uma operação de soma neste último, ele considera a equação desconhecida. Ambos são aplicações fechadas, não havendo descrição detalhada sobre suas implementações.

#### 5. Trabalho proposto

O trabalho proposto consiste em um resolvidor de equações algébricas do 2º grau, voltado especificamente para o conteúdo de 7ª série do ensino fundamental.

O Resolvidor pode ser utilizado como uma ferramenta para auxiliar o aluno na resolução de equações, pois ele é capaz de resolver equações e mostrar ao aluno passo-a-passo todas as operações realizadas na equação até chegar ao resultado final. São exemplos de operações: MMC, simplificação, fatoração entre outros.

No desenvolvimento do resolvidor foi utilizada a linguagem de programação Java (<http://java.sun.com/>), pois esta possui a característica de ser multiplataforma, ou

seja, é capaz de executar em qualquer sistema operacional que tenha suporte a máquina virtual Java. O sistema especialista JBoss Drools foi escolhido por possuir grande integração com a linguagem Java, e por ter sido desenvolvido na mesma, garantindo assim que o resolvedor seja multiplataforma.

O resolvedor é distribuído em um arquivo jar – um formato de arquivo compactado e executável da linguagem Java. A sua execução é simples, sendo apenas necessários a instalação da máquina virtual Java no computador do usuário e o arquivo “.jar” do resolvedor.

### 5.1. Funcionamento

O funcionamento do resolvedor pode ser dividido em duas etapas: (1) a transformação da equação passada pelo usuário para um formato aceito pelo resolvedor, e (2) a resolução propriamente dita da equação.

Na primeira etapa a equação passada pelo usuário é transformada em uma árvore de expressões. Este formato foi escolhido, pois possibilita representar a precedência de operadores, onde os operadores com menor precedência se localizam mais próximo da raiz da árvore. Na Figura 2 pode ser visualizado um exemplo de árvore de expressões.

A montagem da árvore consiste em transformar a expressão do usuário, que está na notação infixa (ex:  $2x+3=8$ ), para a notação pós-fixa (ex:  $2x3+8=$ ), conforme o algoritmo de [Goodrich 2007]. Foi necessária a transformação para pós-fixa pois nesta forma a precedência é mantida sem o auxílio dos parênteses, facilitando a montagem da árvore. Após a expressão ser representada na notação pós-fixa, é gerada a árvore binária de expressão segundo o algoritmo descrito em [Lafore 2005].

Na segunda etapa, com a árvore já formada, a equação é inserida na Memória de Trabalho (*Working Memory*) do Drools, onde, então, o sistema especialista vai buscar as regras cujas condições fecham com a equação em memória. Ao final de cada regra, a equação é reinserida na memória (*retract* e *insert* ou *update*), visto que após aplicar-se uma regra sobre a equação obtém-se uma nova equação e, conseqüentemente, novas regras matemáticas aplicam-se sobre ela. A cada nova equação inserida na memória a mesma é exibida na tela bem como uma descrição do passo executado. A execução encerra quando a equação for reduzida ao máximo, e não houver mais regras que fechem com a mesma em memória.

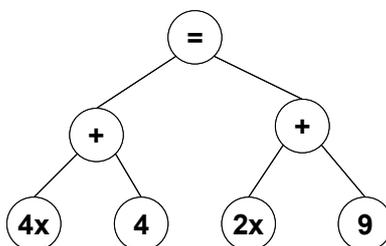


Figura 2: Exemplo de árvore de expressão para a equação "4x+4=2x+9"

### 5.2. Regras

Foram implementadas, ao todo, dezoito regras, sendo cada uma é responsável pela resolução de uma parte da equação. Tendo em vista que na matemática há mais de uma forma de se resolver um problema, no resolvedor há algumas regras que resolvem o mesmo problema, mas utilizando-se de métodos diferentes como, por exemplo, as

regras “Simplifica” e “Calculo”. A regra “Simplifica” pode remover de uma fração os operandos de uma multiplicação que são iguais no numerador e no denominador, evitando assim duas chamadas da regra “Calculo” para realizar as multiplicações. Mas, de qualquer forma, ambas estariam corretas, sendo que uma levaria mais ou menos passos que a outra para chegar ao resultado final. Na Figura 3 pode-se visualizar a resolução de uma equação através das regras “Simplifica” e “Calculo”. O resolvidor, por si só, utiliza as regras que possuem mais prioridade. Porém, ele será incorporado futuramente em Sistema Tutor Inteligente (STI) de equações algébricas que se encontra em desenvolvimento (ver seção 7), no qual o resolvidor será estendido para ser o modelo cognitivo do tutor. Neste caso, ele deve ser capaz de reconhecer as formas de resolver as equações a fim de identificar a operação realizada pelo aluno.

$$\begin{aligned}
 \text{(a) } x &= \frac{2(x-1)}{3(x-1)} \Rightarrow x = \frac{2}{3} \\
 \text{(b) } x &= \frac{2(x-1)}{3(x-1)} \Rightarrow x = \frac{2x-2}{3x-3} \Rightarrow (3x-3)x = 2x-2 \\
 &\Rightarrow 3x^2 - 3x = 2x - 2 \Rightarrow 3x^2 - 5x + 2 = 0 \\
 &\Rightarrow x = \frac{5 \pm \sqrt{5^2 - 4 \times 6}}{6} \Rightarrow x = \frac{5 \pm \sqrt{1}}{6} \\
 &\Rightarrow x = \frac{5 \pm 1}{6} \Rightarrow x' = \frac{6}{6} = 1 \Rightarrow x'' = \frac{4}{6} = \frac{2}{3}
 \end{aligned}$$

**Figura 3: Exemplo de resolução de uma equação onde (a) uso da regra Simplifica (b) uso da regra Calculo**

A fim de evitar conflito na execução das regras, foi colocado em cada uma um valor de prioridade. Desse modo as regras que facilitam a resolução da equação, como a regra “Simplifica”, possuem prioridades maiores e logo serão executadas primeiramente.

Segue abaixo a lista de algumas regras implementadas, em ordem alfabética, bem como sua descrição:

- **Bhaskara:** regra responsável pela aplicação da fórmula de Bhaskara em uma equação. Sua execução consiste em percorrer a árvore da expressão e identificar os valores necessários ( $ax^2+bx+c$ ) para a aplicação da fórmula. Uma vez que os valores tenham sido coletados, a árvore original é substituída por duas árvores, sendo ambas da fórmula de Bhaskara, uma com o sinal negativo antes da raiz e outra com o sinal positivo, como mostra a equação “b” da Figura 3.
- **Calculo:** regra responsável por qualquer cálculo a ser realizado em uma árvore, com exceção da divisão, já que esta é tratada por várias regras (“Divide Iguais”, “Divide por 1”, “Elimina Denominador”, “MMC” e “Resultado”).
- **Isola X:** em equações algébricas, o objetivo principal é obter o valor da incógnita. Para isso, deve-se isolá-la em um dos lados da equação. Esta regra tem exatamente este objetivo: isolar a incógnita em um dos lados da equação para que o seu valor seja calculado no outro lado.
- **MMC:** responsável pela primeira parte do processo de eliminação de frações da equação. Seu funcionamento consiste basicamente em coletar os denominadores das frações e realizar o cálculo do MMC. O processo termina com a regra “Elimina Denominador”. O cálculo do MMC é feito a partir do método eficiente descrito em [Paterlini 1988].
- **Move INT:** assim como a regra Isola X, esta regra contribui para a descoberta

do valor da incógnita. Esta regra é responsável por agrupar valores inteiros em um mesmo lado da equação.

- **Multiplicação de frações:** responsável por realizar a operação de multiplicação entre duas frações [Sullivan 2005].
- **Potência de polinômio:** resolve a potência de polinômio baseado na regra:  $(x+y)^2 = x^2+2*x*y+y^2$ . Resolve apenas esses polinômios simples, ou seja, com apenas dois termos.
- **Raiz Denominador:** no caso de uma fração apresentar uma raiz no denominador, esta regra é responsável por remover essa raiz simplesmente multiplicando o resultado da incógnita por uma fração cujo numerador e denominador sejam iguais. Essa operação resulta em um número inteiro no denominador, pois raiz de um número multiplicado pela raiz desse número ira resultar no próprio número [Sullivan 2005].

$$x = \frac{2}{\sqrt{5}} \Rightarrow x = \frac{2}{\sqrt{5}} \times \frac{\sqrt{5}}{\sqrt{5}} \Rightarrow x = \frac{2\sqrt{5}}{\sqrt{25}} \Rightarrow x = \frac{2\sqrt{5}}{5}$$

- **Resultado:** regra responsável pela descoberta do valor da incógnita, quando uma equação apresenta em um lado a incógnita multiplicada por um inteiro, e no outro lado somente um inteiro. A regra passa o inteiro da incógnita para o outro lado da equação invertendo a operação, ou seja, esse valor torna-se divisor. Ex:

$$2x = 4 \Rightarrow x = \frac{4}{2} \Rightarrow x = 2$$

- **Resultado  $x^2$ :** regra também responsável pela descoberta do valor da incógnita. A diferença em relação à regra anterior é o fato da incógnita estar elevada ao quadrado. A regra remove a potência e aplica a raiz quadrada no outro lado da equação.
- **Simplifica:** responsável pela simplificação de frações. Também é responsável pela divisão da fração quando o resultado da incógnita é encontrado, mas, se e somente se, a fração não produzir valores com casas decimais. Caso isso ocorra, a fração é dividida por valores primos até se tornar irredutível.

Algumas outras regras do resolvidor foram suprimidas devido ao fato de serem menos importantes e devido ao limite de espaço para este artigo.

### 5.3. Cenário

A seguir é apresentado um cenário de execução do resolvidor para uma dada entrada:

1. Considere a entrada  $4x + 4 = 2x + 9$  ;
2. Uma vez que a expressão é recebida pelo resolvidor, ela é então convertida para a notação pós-fixa e, em seguida, é montada a árvore de expressão;
3. A árvore é então inserida na Memória de Trabalho do Drools;
4. Uma vez inserida, ela é testada pelas condições das regras;
5. É realizada uma varredura na árvore, e não são encontrados indícios de operações a serem feitas (soma, subtração e multiplicação), tampouco de frações. Porém são encontrados dois valores inteiros, diferentes de zero, um em cada lado da equação. Dois valores inteiros podem ser operados se estiverem no mesmo lado da equação, o que não é o caso. Então, o valor que se localiza à esquerda da equação é movido para a direita, para posteriormente ser operado (regra "Move INT").
6. Uma nova varredura é feita na árvore, desta vez foram encontrados dois valores compatíveis para se realizar uma operação, porém estes estão em

- sub-árvores diferentes (-4 e 9), sendo necessário agrupá-los sob uma mesma sub-árvore: o nó da sub-árvore onde se encontra o inteiro troca de lugar com o outro inteiro compatível (regra “Soma Subtração subarvores diferentes”);
7. Havendo agora possibilidade de execução de uma operação, esta é executada sobre esses dois nós compatíveis. É realizada a operação descrita no nó pai, nesse caso a soma (regra “Calculo”);
  8. Novamente, não é encontrada uma operação para ser realizada, mas são encontradas duas incógnitas, uma em cada lado da equação ( $4x$  e  $2x$ ). A incógnita que se localiza no lado direito da equação é deslocada para o lado esquerdo, realizando-se as devidas trocas de operação (regra “Isola X”);
  9. Como há duas incógnitas ( $-2x+4x$ ) do lado esquerdo da equação e estas são compatíveis para que se aplique o cálculo, é realizada a operação de soma sob estes termos, visto que esta é a operação do nó pai dos mesmos (regra “Calculo”);
  10. Por fim, como há somente dois termos ( $2x=5$ ), um em cada lado da equação, e um destes possui uma incógnita, é necessário apenas descobrir o valor da incógnita: move-se o inteiro que acompanha a incógnita para o outro lado da equação resultando numa fração ( $x=5/2$ ) (regra “Resultado”);
  11. Como a fração resultante é irredutível, então não há mais regras para serem aplicadas e a execução termina;

O resultado de cada passo de resolução bem como a descrição do passo feito é exibido na tela para o usuário, como mostra a Figura 4. Na tela, encontra-se na parte superior a equação a ser resolvida, que é construída utilizando-se os botões à direita. No centro, são exibidos os passos da resolução da equação, que é iniciada ao se clicar no botão “Resolver Equação”, localizado na parte inferior da janela. Os botões permitem a construção de equações que possuam incógnitas, seja com potências ou não, soma, subtração, multiplicação e divisão.

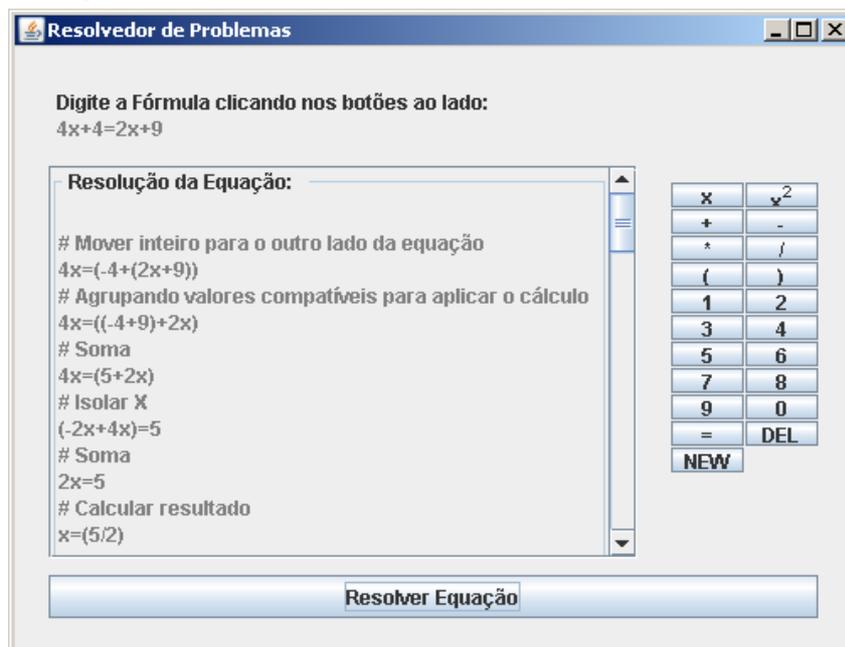


Figura 4: Tela do resolvidor

## 6. Resultados da Avaliação Qualitativa

Uma avaliação qualitativa foi realizada com cinco professores que lecionam o conteúdo de álgebra elementar, em escolas de ensino fundamental e médio de Porto Alegre, com experiência em ensino do conteúdo de 10 a 20 anos. Esta avaliação qualitativa teve como objetivo verificar com os professores a utilidade do resolvedor em sala de aula, melhorias a serem realizadas, assim como estratégias pedagógicas que os professores aplicariam utilizando o resolvedor como ferramenta de aprendizagem de apoio.

Essa avaliação contou com o apoio de uma professora de matemática e seguiu a seguinte metodologia. Inicialmente, os professores entraram em contato com a ferramenta, quando foi mostrada a resolução de algumas equações pré-definidas pelo entrevistador. Após essa primeira etapa, os professores foram convidados a manipular o resolvedor, inserindo equações de sua preferência para desenvolvimento pelo *software* proposto.

Todos os professores acharam o resolvedor uma ferramenta auxiliar rica e interessante na resolução de equações algébricas. Isso porque, segundo os professores, ele permite que o aluno visualize passo-a-passo as decisões tomadas e as operações realizadas, como pode ser observado na Figura 4, pois o resolvedor mostra tanto o resultado como uma descrição da operação realizada. Além disso, os professores observaram algumas melhorias a serem realizadas e também forneceram estratégias de aplicação do resolvedor como ferramenta de apoio à sala de aula. Essas observações e comentários resultantes desta análise são descritos a seguir. Os nomes dos professores são omitidos a fim de respeitar a confidencialidade da informação.

Como principais melhorias, os professores consideraram que se deve modificar a nomenclatura utilizada pelo sistema na descrição das operações realizadas em cada passo, buscando a forma usual da escrita matemática. Não se deve utilizar termos como “Mover inteiro para o outro da Equação”, “Agrupar valores Compatíveis”, ou ainda, “Isolar x”, pois são regras decorrentes do treinamento, mas que, muitas vezes, não fazem sentido para os alunos. A forma de apresentação também pode ser melhorada: os professores indicam a estrutura vertical de frações ( $\frac{3}{9}$ ) no lugar da horizontal (3/9). Também seria importante dar liberdade ao aluno para escolher equações a serem resolvidas, maneiras de visualização da resposta, método de resolução, e método de interação com a ferramenta computacional (com o teclado ou utilizando o mouse). Além disso, ao utilizar o resolvedor o professor deve descrever equações que estejam ligadas ao cotidiano do aluno. Também consideraram importante a utilização de animações, cores e outros recursos computacionais que motivem e chamem a atenção dos alunos. Um último ponto é que o sistema deve ser testado exaustivamente, pois os alunos serão curiosos em inserir operações não válidas com o único objetivo de verificar se o software funciona corretamente.

## 7. Conclusão e trabalhos futuros

Futuramente o resolvedor será incorporado ao modelo cognitivo do STI Pat2Math. O Pat2Math é um STI afetivo voltado ao ensino de álgebra elementar, que além de auxiliar o aluno na aprendizagem de álgebra elementar, irá inferir as emoções deste a fim de adaptar o sistema também aos aspectos emocionais e de motivação do aluno.

O Resolvedor fará parte da base de domínio do tutor juntamente com o componente declarativo, parte responsável pelo armazenamento do conteúdo. Ele atuará

com o Modelo do Aluno para aplicar a técnica de *model tracing*. A técnica consiste em comparar a resposta do aluno com a resposta do resolvidor sobre o possível próximo passo de resolução da equação. Se as respostas forem equivalentes, a resposta do aluno é considerada válida. Caso contrário, então o resolvidor aplica mais uma regra sobre a equação que ele encontrou e compara novamente. Caso a resposta seja ainda negativa, o resolvidor irá repetir as etapas acima até encontrar o resultado parcial fornecido pelo aluno. Muitas vezes, ele terá que recorrer ao *backtracking*, ou seja, retornar alguns passos e aplicar uma das outras regras que foram descartadas pela estratégia de resolução de conflitos do Drools, indo assim por outro caminho, isto é, por outras regras, até achar a solução. Se após uma verificação exaustiva com todas as regras presentes na base do conhecimento, o sistema não conseguir encontrar a resposta fornecida pelo aluno, a resposta do aluno é considerada inválida. Para esses casos o tutor irá acionar as regras de *misconceptions*, que são regras que contêm os erros mais comuns dos alunos em relação à álgebra, como, por exemplo, somar incógnitas com potências diferentes, para que seja possível identificar a linha de pensamento do aluno. A partir destas informações, o tutor pode dar mais ênfase a estes assuntos em que o aluno possui maior dificuldade.

### **Agradecimentos**

O grupo agradece a Unisinos, CNPq e FAPERGS pelo apoio através das bolsas de Iniciação Científica e a Capes pelo apoio dado através do projeto Capes/Cofecub.

### **Referências Bibliográficas**

- Anderson, J. R., A. Corbett, et al. (1992) General principles for an intelligent tutoring architecture. In: V. Shute e W. Regian (Ed.). *Cognitive Approaches to Automated Instruction*. Hillsdale, NJ: Erlbaum, p.81-106.
- Cury, H. N.; Konzen, B. (2006) Análise de resoluções de questões em matemática: as etapas do processo. *Educação Matemática em Revista-RS*, v. 7, n. 7, p. 33-41.
- Goodrich, M. T. and Tamassia, R. (2007) *Estruturas de dados e algoritmos em Java*. Porto Alegre: Bookman, 4. ed . 600 p.
- Lafore, R. (2005) *Estruturas de dados e algoritmos em Java*. Ciência Moderna.
- Lima, R. (2007). *Equações Algébricas no Ensino Médio: Uma jornada por diferentes Mundos da Matemática*. Tese de Doutorado. PUC-SP.
- Nikolopoulos, C. (1997) *Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems*, Marcel Dekker Inc. Press.
- Paterlini, R. (1988) Um Método para o Cálculo do MDC e do MMC. In *Revista do Professor de Matemática n<sup>o</sup>. 13*, págs. 34-37.
- Proctor, M. , Neale, M. , Frandsen, M. , Griffith, S., Tirelli, E. , Meyer, F. and Verlaenen K. (2008) *Drools User Guide*, [http://downloads.jboss.com/drools/docs/4.0.7.19894.GA/html\\_single/index.html](http://downloads.jboss.com/drools/docs/4.0.7.19894.GA/html_single/index.html)
- Sef. (1998) *Parâmetros Curriculares Nacionais: matemática. Volume 3. MEC/SEF*. Brasília, p.148.
- Sullivan, M. (2005). *Algebra and trigonometry*. Upper Saddle River: Pearson, 7. ed. 1160 p.