IComb: Um sistema para o ensino e aprendizagem de combinatória em ambiente Web

ALEXANDRE LUÍS KUNDRÁT EISENMANN

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

PROGRAMA: CIÊNCIA DA COMPUTAÇÃO

ORIENTADOR: PROF. DR. LEÔNIDAS DE OLIVEIRA BRANDÃO

São Paulo, março de 2009

IComb: Um sistema para o ensino e aprendizagem de combinatória em ambiente Web

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Alexandre Luís Kundrát Eisenmann
e aprovada pela Comissão Julgadora:

Banca Examinadora

- Prof.Dr. Leônidas de Oliveira Brandão (orientador) - IME-USP
- Prof.Dr. Marco Aurélio Gerosa - IME-USP
- Profa.Dra. Maria Alice Grigas Varella Ferreira - EP-USP

Agradecimentos

Ao professor *Lêonidas* que mesmo nas situações mais adversas me apoiou, incentivou e tornou possível o presente trabalho.

À colega *Janine* com suas valiosas sugestões e participação fundamental na fase de experimentação.

À minha mãe *Regina* na qual me inspirei e herdei a paixão pela matemática e pela ciência pura.

Ao meu pai *José Henrique* que, com sua ótica de economista, me mostrou a beleza das ciências aplicadas e a recompensa de um trabalho realizado com seriedade.

A meu avô *Victor* que os anos de docência em matemática ilustraram a nobreza de seu ofício.

À minha esposa *Fabiana* que me incentivou e acompanhou ao longo de todo o tempo. Nada teria sido possível sem sua compreensão, paciência e carinho. Como se não bastasse, Fabiana participou ativamente do trabalho, produzindo as imagens das camisetinhas dos jogadores presentes no universo "Futebol" a ser discutido posteriormente.

E finalmente a minha filha *Sandra*, no momento com 3 meses de idade, que mesmo sem ter consciência disso me inspirou profundamente.

Resumo

Neste trabalho apresentamos o desenho e desenvolvimento de um novo sistema de apoio ao ensino/aprendizagem via Web, o ***iComb* - Combinatória Interativa na Internet**. Também apresentamos alguns experimentos didáticos com a atual proposta do ***iComb***. Este sistema é baseado na experiência do ***Combien?***, sistema desenvolvido pela Université Pierre et Marie Curie - LIP6, que tem bons recursos para auxílio à resolução de exercícios ligados a análise combinatória, mas apresenta limitações quanto a incorporação de novos exercícios e principalmente apresenta restrições para seu uso via Web. Uma inovação advinda deste projeto é que o ***iComb*** pode ser integrado a sistemas gerenciadores de cursos, como o SAW - Sistema de Aprendizagem pela Web. Além disso, o ***iComb*** pode ser incorporado a páginas pessoais, redes sociais e blogs, na forma de um "widget".

Abstract

In this work, we present the software ***iComb* - Interactive Combinatorics on Internet**, a new software that can be used to teach and learning combinatorics trough the Word Wide Web. The ***iComb*** is based on the ***Combien?*** experience, a software developed by Université Pierre et Marie Curie - LIP6. We can easily integrate the ***iComb*** software in a web-based learning management system like the **SAW - Web-based learning system**. Moreover, the ***iComb*** can be easily added to a personal web sites, social networks systems and blogs using the copy-and-past pattern resulting in a powerful collaborative tool to the conceptual movement named as Web 2.0.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

"Não se pode ensinar coisa alguma a alguém, pode-se apenas auxiliá-lo a descobrir por si mesmo"

Galileu Galilei

O crescimento do uso da Internet no ensino no Brasil e no mundo e o subseqüente aumento na demanda por pesquisas nesta área (Litto et al., 2004) têm evidenciado algumas características desta abordagem em relação ao método tradicional de ensino.

Utilizando a infra-estrutura da Internet é possível aproximar professores e alunos. Os modernos sistemas computacionais[1] de ensino a distância permitem a interação entre professores e alunos não só distantes fisicamente, mas também distantes temporalmente, não havendo a necessidade de que esta relação ocorra de forma sincronizada.

Potencialmente, essa mudança de contexto desloca a atividade de ensino que conhecemos para um plano mais pessoal, possibilitando que o aprendiz siga seu próprio ritmo de aprendizado no local e horário que lhe for mais conveniente.

Por outro lado, os sistemas de educação a distância competem pela atenção do aprendiz contra uma miríade de outros recursos facilmente disponíveis na Web. Desta maneira, um sistema de boa qualidade, com desenho adequado e que incorpore interatividade, pode, não só viabilizar um ensino a distância, como ser atrativo para o aluno e, mais importante, incorporar as vantagens do uso do computador (como velocidade e precisão).

O projeto de um novo sistema educacional deve levar em conta a concepção do ensino. O desenho do sistema deve considerar questões-chave, tais como; *como o aluno aprende?*

[1] A partir deste ponto utilizaremos a forma reduzida *sistema*, como equivalente ao termo em língua inglesa software.

ou *como maximizar o aprendizado do aluno?*. Apesar destas perguntas não possuirem respostas definitivas, existem propostas com bons resultados relatados.

Como observado por Polya, a matemática é uma ciência na qual a resolução de problemas desempenha papel fundamental (Pólya, 1981). Qualquer uma de suas subdisciplinas possui abundância de exercícios e situações problemas que o professor pode encontrar na bibliografia disponível, ou mesmo criar baseado em sua própria experiência. Deste modo, dentro da área de educação matemática, o ensino baseado na resolução de problemas é um campo fértil para pesquisa e exploração. A educação "problematizadora" preconiza que o aluno aprende através das situações problemas expostas pelo professor (Clements, 2000). O esforço do aluno na resolução dos problemas propostos é extremamente importante no processo de aprendizagem.

Outro aspecto importante para o processo de aprendizagem são os erros cometidos pelos alunos. Estes erros são importantes, pois constituem momentos particularmente favoráveis para a aprendizagem (Giroire et al., 2006). Por outro lado, a falta de agilidade da resposta pode atrapalhar o aprendizado uma vez que, segundo alguns trabalhos, a falta de uma avaliação/validação imediata dificulta a aprendizagem e pode causar a desmotivação do aluno (Hara and Kling, 1999, Kirby, 1999, Hentea et al., 2003).

1.1 Caracterização do Problema

O ensino da matemática pode ser bastante beneficiado pela utilização de computadores e da infra-estrutura da Internet. Alguns tópicos de matemática permitem bons mecanismos de validação/verificação automática, e possibilita a implementação de sistemas que validam automaticamente respostas de exercícios. De fato, o desenvolvimento de aplicativos voltados para o ensino da matemática vem aumentado no Brasil e no mundo (Litto et al., 2004).

Por outro lado, definir uma boa metodologia, escolher os componentes adequados e construir um bom projeto de sistema voltado ao ensino da matemática requer boa compreensão do domínio da matemática pesquisado.

A geometria é um bom exemplo de sucesso da aplicação dos computadores. A geometria implementada em computador, chamada Geometria Dinâmica (GD) ou Geometria Interativa, começou a ganhar destaque na década de 90, principalmente com a populariza-

ção dos programas comerciais Cabri Geometry e Geometer's Sketchpad (Isotani, 2005).

Neste trabalho o foco principal é o ensino/aprendizagem de combinatória via Web, tendo como modelo um projeto de sistema de um grupo francês da *Université Pierre et Marie Curie- LIP6*, o ***Combien?***. Este sistema é gratuito, mas apresenta algumas restrições para seu uso adequado sob a Web: não pode ser facilmente integrado a sistemas Web e não permite que novos exercícios/problemas sejam incorporados.

Estas duas limitações motivaram o presente estudo. Inicialmente foi tentado desenhar um sistema que tivesse o ***Combien?*** como motor, mas devido às dificuldades de alteração do sistema, optou-se pelo desenho de um novo sistema todo desenvolvido em Java. No próximo capítulo é detalhada esta abordagem inicial e suas dificuldades que resultaram no ***iComb***.

1.2 Justificativa

Apesar do sucesso da implementação do projeto ***Combien?*** o sistema foi concebido como uma ferramenta de instalação local e portanto não aderente ao paradigma Web, característica que limita sua utilização em maior escala, como por exemplo no contexto da educação a distância.

O sistema não possui versão para o português, dificultando sua aplicação em território brasileiro, além disso, não é óbvio que a natureza dos exercícios propostos para os alunos do sistema educacional francês tenham o mesmo impacto para o aprendiz brasileiro.

1.3 Objetivos

O objetivo do presente trabalho é propor e implementar o sistema ***iComb*** - *Combinatória Interativa na Internet* baseado na experiência do projeto ***Combien?***. O ***iComb*** será personalizado para a realidade brasileira e concebido para ser utilizado através da internet. Além disso, o ***iComb*** será integrado ao sistema *Sistema de Aprendizado pela Web (SAW)* de maneira a propiciar as características de um EaD.

Desta forma o ***iComb*** será mais um módulo de aprendizagem integrável ao sistema SAW (Janine, 2007, Brandão et al., 2006), enriquecendo o sistema que já conta com o

iGeom - *Geometria Interativa na Internet* e o ***iGraf*** - *Gráfico Interativo na Internet*. O sistema ***iComb*** - www.matematica.br/icomb - será distribuido gratuitamente e a disponibilização de seu código-fonte se encontra em estudo.

Coordenado pelo professor Leônidas de Oliveira Brandão, um dos objetivos do desenvolvimento desta infra-estrutura de sistemas, bem como dos módulos de aprendizagem, é "democratizar" o uso destes módulos de aprendizagem, permitindo que qualquer estudante ou professor, com acesso a um micro-computador, possa usufruir de seus benefícios.

1.4 Conteúdo da dissertação

No capítulo 2 é apresentada a fundamentação da proposta do sistema ***iComb***. Inicialmente apresenta-se o sistema ***Combien?*** e um resumo de experimentos didáticos que fundamentaram o ***Combien?***. A seguir é apresentado o sistema SAW, que foi utilizado como modelo para receber o ***iComb*** como um de seus *módulos de aprendizagem* (Brandão et al., 2006, Moura et al., 2007). Ainda no capítulo 2 é apresentado um protótipo para incorporar o ***Combien?*** em sistemas Web e apontada suas principais dificuldades, que levaram a descontinuidade deste modelo.

No capítulo 3 é apresentado o ***iComb*** - *Combinatória Interativa na Internet* propriamente dito, partindo de sua motivação e princípios fundamentais até a descrição de alguns componentes importantes, em especial é apresentada a chamada arquitetura de *plugins* definida para o ***iComb*** que permite a inclusão de novos universos de exercícios não estando restrito apenas a exercícios relacionados a cartas e baralhos. No final do capítulo é discutido a proposta de disponibilização do ***iComb*** com um *widget*, possibilitando a fácil integração e incorporação do ***iComb*** em *blogs* ou páginas pessoais.

No capítulo 4 é realizada uma discussão um pouco mais formal da possibilidade de um mecanismo de detecção automática de erros. A partir de algumas definições foi possível propor e demonstrar alguns teoremas que culminaram na descrição formal do que é enfim uma solução para um problema do ***iComb***. Conhecida a estrutura formal de uma solução, foi estabelecido o algoritmo de correção automática utilizado no sistema ***iComb***. O capítulo termina apresentando algumas limitações do método, bem como a sugestão de futuros desenvolvimento do sistema.

No capítulo 5 são apresentados alguns dos experimentos didáticos já realizados com

a atual versão do ***iComb***. No capítulo 6 são apresentadas as conclusões deste trabalho e apontados alguns trabalhos futuros.

Capítulo 2

Fundamentação

> *"Computers are incredibly fast, accurate, and stupid; humans are incredibly slow, inaccurate and brilliant; together they are powerful beyond imagination."*
>
> Albert Einstein

2.1 Análise Combinatória

A Análise Combinatória é, de maneira mais geral, a parte da matemática que analisa estruturas e relações discretas. Um dos tipos de problemas que ocorre com frequência neste domínio é contar ou classificar os subconjuntos de um conjunto finito que satisfazem certas condições dadas sem que seja necessário enumerar seus elementos (Morgado et al., 1991).

Embora a Análise Combinatória possua técnicas gerais que permitem atacar certos tipos de problemas, a solução de um problema combinatório exige quase sempre engenhosidade e a compreensão plena da situação descrita pelo problema. Por outro lado, se a aprendizagem destes conceitos se faz de maneira mecânica, padronizada, cria-se a impressão de que a Análise Combinatória é somente um jogo de fórmulas complicadas (Morgado et al., 1991).

Problemas de contagem e combinatória em geral não são resolvidos através de um método dedutivo, onde a partir de um conjunto de axiomas aplica-se regras de inferência para a obtenção de novos fatos até a resolução do problema. Segundo (Giroire et al., 2006), a reação dos estudantes frente à resolução de problemas de combinatória reflete uma dificuldade que os autores interpretam como dificuldade de representação, como ilustrado

nas frases a seguir: “Não consigo representar o problema”, “Não sei por onde começar”, “Eu entendo a solução dada pelo professor, mas não entendo porque a minha está errada”, “Eu propus uma solução mas não sei dizer se está certa ou errada”.

Encontrar um equilíbrio entre expor a Combinatória como um assunto somente acessível a alunos muito criativos ou como um receituário de fórmulas que só resolvem alguns casos, é tarefa árdua para o professor. Por outro lado, esforços para que os alunos “enxerguem” o problema e “visualizem” as respostas, podem contribuir para uma melhor representação mental dos mesmos, e ajudar a alcançarmos este tênue equilíbrio.

2.2 *Combien?*

O grupo ***Combien?*** da *Université Pierre et Marie Curie, LIP6* construiu um sistema pedagógico para ajudar estudantes do sistema educacional francês a aprender combinatória utilizando linguagem matemática. Neste projeto é oferecida aos estudantes uma interface cuja linguagem é próxima àquela que eles espontaneamente tendem a utilizar na formulação dos problemas e que ao mesmo tempo, permite uma argumentação matemática rigorosa.

O sistema ***Combien?*** propõe ao aluno uma meta imediatamente compreensível: Quantos? Para ressaltar a meta de determinar quantos possíveis elementos existem com determinada propriedade eles adotaram o termo francês correspondente, *Combien?*. Inicialmente, o sistema apresenta ao aluno um problema em forma textual e para que ele consiga atender o pedido, fornecendo a quantidade requerida pelo problema, deverá seguir alguns passos envolvendo ações, modelagem e raciocínio (Le Calvez et al., 2003, Giroire et al., 2006, Tisseau et al., 2000).

A fundamentação matemática subjacente ao método utilizado pelo sistema foi chamada de “método construtivo”. A idéia é permitir que o aluno, raciocinando a partir dos opções disponíveis sistema, descreva um algoritmo de enumeração que seja solução do problema. Essa é uma forma eficiente de resolver o problema uma vez que não será necessário “rodar” o algoritmo, apenas descrevê-lo criteriosamente (Le Calvez et al., 2003, Giroire et al., 2006, Tisseau et al., 2000).

Os especialistas no domínio da Combinatória foram capazes de estabelecer classes de problemas, de maneira que para cada uma delas o ***Combien?*** possui uma *interface* específica, também chamada de “máquinas” (Le Calvez et al., 2003). Seguindo o comando

dos alunos as máquinas irão direcionar a solução ao mesmo tempo em que interagem com os alunos solicitando respostas parciais.

As "máquinas" do ***Combien?*** foram projetadas para auxiliar o aluno em qualquer sequência de resolução escolhida, independentemente da ordem em que os passos são realizados. Assim, qualquer solução correta será considerada como tal, ao mesmo tempo que qualquer solução incorreta será detectada pelo sistema.

O ***Combien?*** foi desenvolvido para duas plataformas, MacOS e Windows. A linguagem de programação utilizada foi Smalltalk, no ambiente Visualworks (VW, 2009).

2.2.1 Detecção de Erros

Geralmente no processo de aprendizagem, os erros cometidos pelos estudantes são importantes pois constituem situações particularmente favoráveis para o aprendizado. Quando os estudantes resolvem exercícios na sala de aula, eles têm que esperar por períodos de tempos longos para terem acesso a modelos de resposta. Quando a solução é finalmente dada eles já esqueceram o contexto do problema e o melhor momento para apresentar correções passara (Giroire et al., 2006). Vários outros trabalhos apontam para a importância de retorno rápido para as respostas de alunos, principalmente quando se trata de atividades via Web. Por exemplo, (Hara and Kling, 1999) e (Hentea et al., 2003), afirmam que uma das principais frustrações de alunos envolvidos em cursos Web (incluindo aqueles com parte do curso sendo presencial) é a falta de retorno rápido sobre as tarefas enviadas pelos alunos. Já (Moura et al., 2007) relata maior envolvimento dos alunos quando é providenciado pelo sistema uma resposta imediata às soluções enviadas pelos alunos.

O sistema ***Combien?*** pode, a cada passo da solução, verificar se as ações até então realizadas podem ou não levar a um resultado correto. Desta maneira, o sistema poderá apontar rapidamente ao aluno um erro no desenvolvimento da solução. Alternativamente, o sistema armazena o erro para futura notificação.

Segundo o grupo ***Combien?***, a correção imediata do erro aumenta em muito a compreensão do mesmo uma vez que está ligado ao último raciocínio utilizado (Giroire et al., 2006).

O "método construtivo" utilizado no sistema ***Combien?*** permite que só os erros interessantes sob o ponto de vista pedagógico sejam analisados, afinal, a ação das máquinas

"direciona" os estudantes a seguir determinado processo de raciocínio, evitando possíveis erros irrelevantes.

De modo geral, em cada "máquina" está codificada uma lista dos possíveis erros, cada qual associado a uma estrutura de representação de erros (*error schema*). A cada ação do aluno, a máquina consulta a estrutura de representação de erro e verifica a correção da ação realizada.

2.2.2 Um exemplo concreto

Quatro máquinas foram definidas, a saber, *SetBuilding*, *ListBuilding*, *CaseSetBuilding* e *CaseListBuilding*. Cada uma delas foi projetada para resolver uma certa classe de problemas de combinatória. A fim de ilustrar o funcionamento do ***Combien?*** a seguir será apresentado o problema número 5 pertencente a classe *SetBuilding* do ***Combien?***. Neste problema é proposta a utilização do baralho de pôquer, um baralho com 32 cartas: a partir de um baralho comum (52 cartas) retira-se todas as cartas inferiores a 7, de forma que cada naipe contenhas as cartas 8, 9, 10, Valete, Dama, Rei e Ás.

Problema nº5: Com um baralho de 32 cartas, quantas mãos de 20 cartas é possível formar com 3 ases, 4 espadas e 16 cartas vermelhas.

Uma vez escolhido o exercício o ***Combien?*** solicitará ao aluno a definição do Universo conforme o seguinte formato: "Eu quero subconjuntos com `<número>` elementos escolhidos em `<universo>`". Neste particular exercício o aluno poderá completar a frase de forma a obter "Eu quero subconjuntos com 20 elementos escolhidos em um baralho de 32 cartas".

Definido o Universo[1], o aluno dará início a tarefa de construção, que consiste na definição de estágios. Seguindo um formato pré-definido, o aluno deverá raciocinar de maneira tal que ao final, o resultado de cada estágio contribuirá para a obtenção da resposta desejada.

Em cada estágio o aluno solicitará o número de elementos (no caso, cartas) e as restrições desejadas. Logo em seguida o mesmo deverá informar quantas ocorrências espera ter

[1] A definição de Universo utilizada no ***Combien?*** é: "Em um exercício onde a meta é contar as configurações que satisfazem certas restrições, o Universo é o conjunto de todas as possíveis configurações, do qual iremos selecionar aquelas que verificam as restrições". A definição utilizada não é diferente do conceito de Universo na teoria das probabilidades.

selecionado, utilizando as estruturas combinatórias clássicas como combinação, arranjo, fatorial e exponenciação.

A figura 2.1 representa a composição do 3º estágio de uma das possíveis respostas:

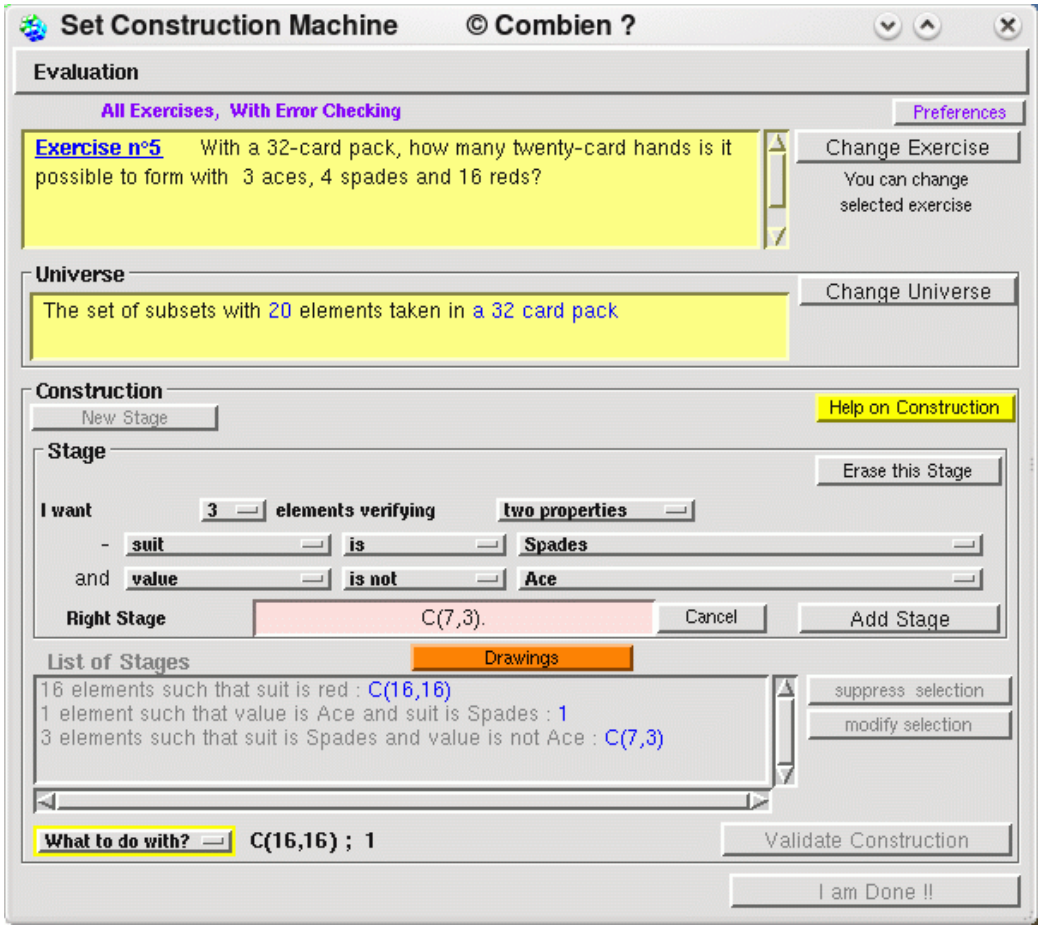

Figura 2.1: Interface *SetBuilding* do ***Combien?***.

- **(1º estágio)** 16 elementos tais que o naipe é Vermelho: $C(16, 16)$.
- **(2º estágio)** 1 elemento tal que naipe é Espadas e o valor é Ás: 1
- **(3º estágio)** 3 elementos tais que naipe é Espadas e o valor não é Ás: $C(7, 3)$

Como é possível verificar no exemplo da figura 2.1, cada estágio define sub problemas cuja quantidade pode ser calculada com a utilização de apenas uma estrutura combinatória

conhecida, seja ela arranjo, combinação, etc... O ***Combien?*** possui um poderoso mecanismo de detecção automática de erros e a cada ação incorreta avisa o aluno com possíveis dicas.

Para finalizar, o aluno deverá informar o que fazer com os estágios selecionados na caixa de combinação apropriada, escolhendo uma entre as duas possíveis opções: multiplicar ou somar. Definida esta ação, o aluno solicita a validação final e, em caso de uma construção bem sucedida, o sistema irá apresentar a tela como mostra a figura 2.2.

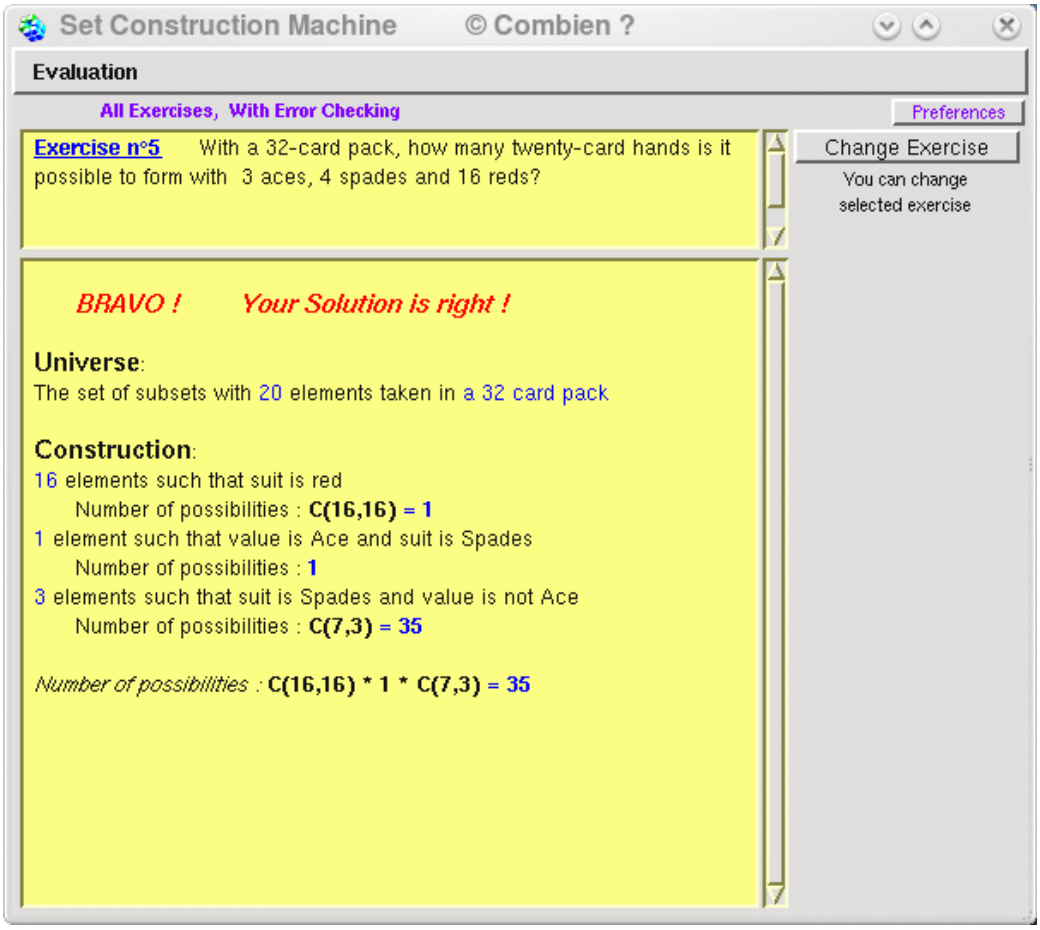

Figura 2.2: Final da construção no ***Combien?***.

2.2.3 Experimentos

A seguir são apresentados resumos de vários dos experimentos realizados pelo grupo ***Combien?*** todos eles aplicados no sistema educacional francês (Le Calvez et al., 2003).

Em 2002 foram realizados três tipos de experimentos. O público alvo escolhido foi diferente para cada um deles: estudantes, professores de matemática e ergonomistas. Os experimentos foram realizados utilizando-se a máquina SetBuilding que contém originalmente dez exercícios em ordem crescente de dificuldade.

Em 2003 o grupo iniciou uma série de experimentos com as quatro máquinas: *SetBuilding*, *ListBuilding*, *CaseSetBuilding* e *CaseListBuilding*.

Para cada experimento, um livro texto foi disponibilizado tanto na forma on-line, quanto em papel. Em cada um dos experimentos, as máquinas foram apresentadas como ferramentas de resolução de problemas. O público era livre para escolher a ordem que os problemas seriam resolvidos. Foi solicitado para que alguns estudantes expressassem o raciocínio utilizado em voz alta de forma que pudessem ser gravados enquanto usavam as máquinas. No final de cada sessão, cada estudante completou um questionário para avaliar o sistema.

Público alvo: estudantes do último ano da escola secundária

Em 2002, dois experimentos foram realizados com estudantes voluntários. No primeiro grupo, nenhum dos 3 estudantes tinha qualquer tipo de conhecimento em combinatória. Eles consideraram a atividade como um desafio e trocaram muitas informações entre eles. Quase não utilizaram o help. O grupo estava bastante entusiasmado e conseguiu resolver todos os problemas propostos, ao contrário do que acreditavam os pesquisadores.

Então eles solicitaram exercícios de outros tipos. Os pesquisadores apresentaram as outras máquinas, adequadas para solucionar diferentes tipos ou classes de problemas. Na primeira aula de combinatória subsequente ao experimento eles sabiam como resolver os exercícios da classe *SetBuilding* , porém não tinham idéia de como resolver os exercícios das outras três classes.

O segundo grupo, novamente com 3 alunos, trabalhou com as máquinas depois de ter tido 5 horas de trabalho resolvendo exercícios de combinatória no papel, porém sem uma apresentação apropriada sobre o assunto. Embora, menos motivado do que o primeiro grupo, eles foram mais rápidos e gostaram de resolver os exercícios.

Nos dois grupos, os estudantes começaram com o foco no valor numérico a ser obtido, então observavam a demonstração que a máquina os forçava a realizar. Depois de certo tempo, eles não mostravam mais interesse no valor numérico, mas focavam na definição

dos diferentes passos da demonstração.

Público alvo: Estudantes de matemática discreta do segundo ano da universidade

Novamente aqui, os estudantes mostraram grande interesse no experimento. Eles estavam motivados e resolveram os problemas com facilidade. Eles apreciaram a possibilidade de sistematizar uma solução. Alguns alunos relataram que "descobriram" que é possível utilizar um "método": "Agora eu sei que existe um forma sistemática de resolver este tipo de exercício" (depoimento de um aluno).

Público alvo: Estudantes em seu treinamento para ensino em matemática

Mais uma vez o experimento foi apreciado. Alguns estudantes acharam difícil seguir o método e se sentiram incomodados com as restrições do sistema; eles gostariam de poder resolver os exercícios de forma diferente. Entretanto, expressaram o desejo de utilizar o sistema eles mesmos e mostraram interesse em utilizar o sistema em sala de aula.

Público alvo: Professores de matemática

Experimentaram mais dificuldade em seguir o método induzido pelas máquinas. Geralmente quando ensinam combinatória, eles não encorajam seus alunos a seguir sistematicamente um método. Portanto, eles teriam que mudar a forma de pensar. Para os primeiros problemas propostos, eles prefeririam dar a solução sem qualquer justificativa. Porém, para os problemas mais difíceis da mesma classe, eles apreciaram o fato de justificar cada passo.

Resultados

Todos os usuários apreciaram a aparência da máquina *SetBuilding*. Também gostaram do fato de que as outras máquinas possuem o mesmo padrão visual. Isso permite que possam se concentrar no conceito subjacente. Acharam que o método é interessante porque permite resolver problemas cuja solução não é óbvia. Eles resolveram todos os exercícios, seguindo a ordem que foi dada, o que não foi pedido que fizessem. Apreciaram o progresso pedagógico e a maneira que o experimento foi conduzido, o que permitiu no primeiro

momento, familiarização com a máquina, e depois que se concentrassem na dificuldade dos seus problemas.

Um dos objetivos do projeto é convencer os estudantes de que eles precisam "provar" os resultados e que utilizando uma abordagem sistemática é possível resolver exercícios cada vez mais difíceis. O sistema mostrou para os pesquisadores que o ***Combien?*** é uma ferramenta eficiente para fazer os estudantes justificarem os resultados numéricos obtidos em problemas de combinatória.

2.3 Sistema de Aprendizagem pela Web (SAW)

Atualmente existem vários sistemas complexos para gerenciar cursos pela Web, inclusive alguns gratuitos, como o Teleduc (Rocha et al., 2002) e o Moodle (Moodle, 2004). Entretanto estes ambientes são desprovidos de recursos especializados para o aprendizado de conteúdos específicos, como a Geometria e Combinatória. Além disso, muitos deles restringem a incorporação de novos recursos impossibilitando o uso de novas ferramentas educacionais. Um sistema gerenciador de cursos na Web que pode incorporar, de forma simples, módulos educacionais na forma de *applets*[2] é o SAW - Sistema de Aprendizagem pela Web (Brandão et al., 2006).

A grande vantagem do uso do *SAW* consiste em sua capacidade de receber **módulos de aprendizagem (MA)**, na forma de *applets*, específicos para áreas do aprendizado. O primeiro estudo de caso deste conceito no SAW foi feito com o ***iGeom***, um sistema gratuito para ensino/aprendizagem de geometria. O autores relatam várias vantagens do uso do ***iGeom*** integrado ao *SAW*, destacando as seguintes:

- Um professor pode produzir os exercícios em sua máquina e enviá-los ao servidor ou criá-los diretamente no MA acoplado ao sistema gerenciador;
- Um professor pode utilizar exercícios em MA elaborados por outros professores (formando uma comunidade de produtores de compoentes didáticos digitais);
- O aluno pode ter acesso imediato ao resultado de validação de sua resposta para o exercício;

[2]Um applet é uma aplicação *Java* que pode ser utilizada dentro de um navegador web.

- O servidor pode anotar a resposta ao exercício enviada pelo aluno, permitindo que o próprio aluno ou o professor possam examinar posteriormente a solução enviada.

Na figura 2.3 são apresentadas duas telas do *SAW*, sendo aquela em primeiro plano a *interface* do exercício utilizando o *iGeom* como MA. Nela fica claro que nesta abordagem o MA e gerenciador de cursos parece um só sistema. Um requisito necessário para que um *applet* seja utilizado como MA no *SAW* é que o mesmo disponha de um método para enviar dados de formulário via Web (utilizando o protocolo HTTP). Deste modo, ao finalizar o exercício, o aluno pode enviar sua resposta e o *SAW* pode armazená-la para futuras consultas, seja pelo aluno ou pelo professor.

2.4 *CombienDelivery*

A primeira idéia que surgiu para a criação de um sistema voltado ao ensino de combinatória em ambiente Web foi a transformação do sistema ***Combien?*** num módulo de aprendizagem (MA) integrável ao SAW.

A utilização do ***Combien?*** tal como distribuído pela Universidade de Paris é bastante atraente, uma vez que estaríamos utilizando um sistema muito bem testado. Além disso, poderíamos contar com apoio do grupo ***Combien?***. Futuras alterações no sistema francês poderiam ser facilmente incorporadas ao projeto. Enfim, toda a evolução e esforço do ***Combien?*** seriam aproveitados.

A idéia resultou na criação do *CombienDelivery*, módulo especialmente desenvolvido para integrar o ***Combien?*** ao sistema SAW. O *CombienDelivery* é um *applet Java* desenvolvido como um módulo de aprendizagem integrado ao sistema SAW. No pano de fundo da figura 2.4 podemos ver o sistema SAW com o *CombienDelivery*, este último distinguível pela seta verde e os dizeres ***Combien?***. No primeiro plano está o sistema ***Combien?*** aberto em resposta a um clique sobre o *CombienDelivery*.

2.4.1 Problemas apresentados

Apesar do desenvolvimento do *CombienDelivery* ter sido realizado como previsto, alguns inconvenientes da abordagem foram evidenciados:

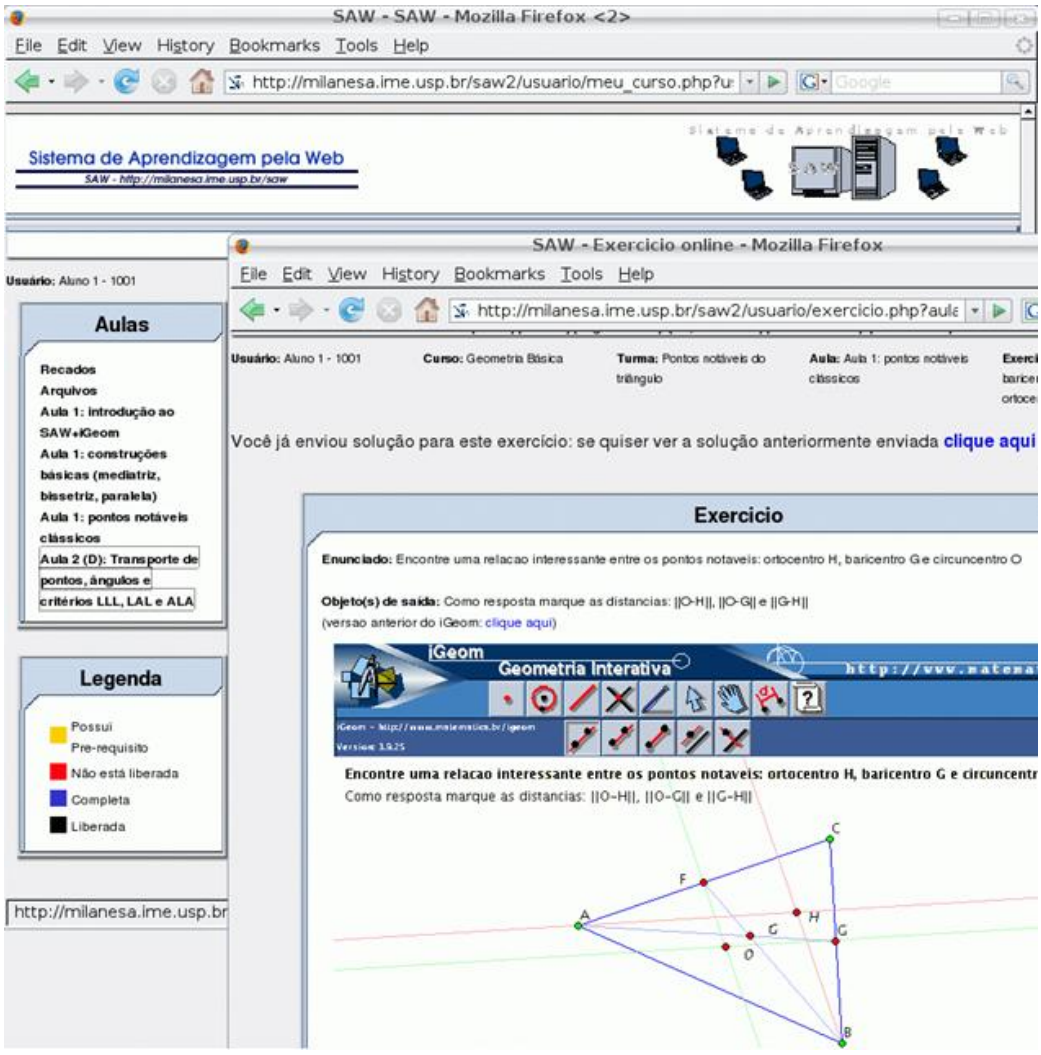

Figura 2.3: Interfaces do SAW

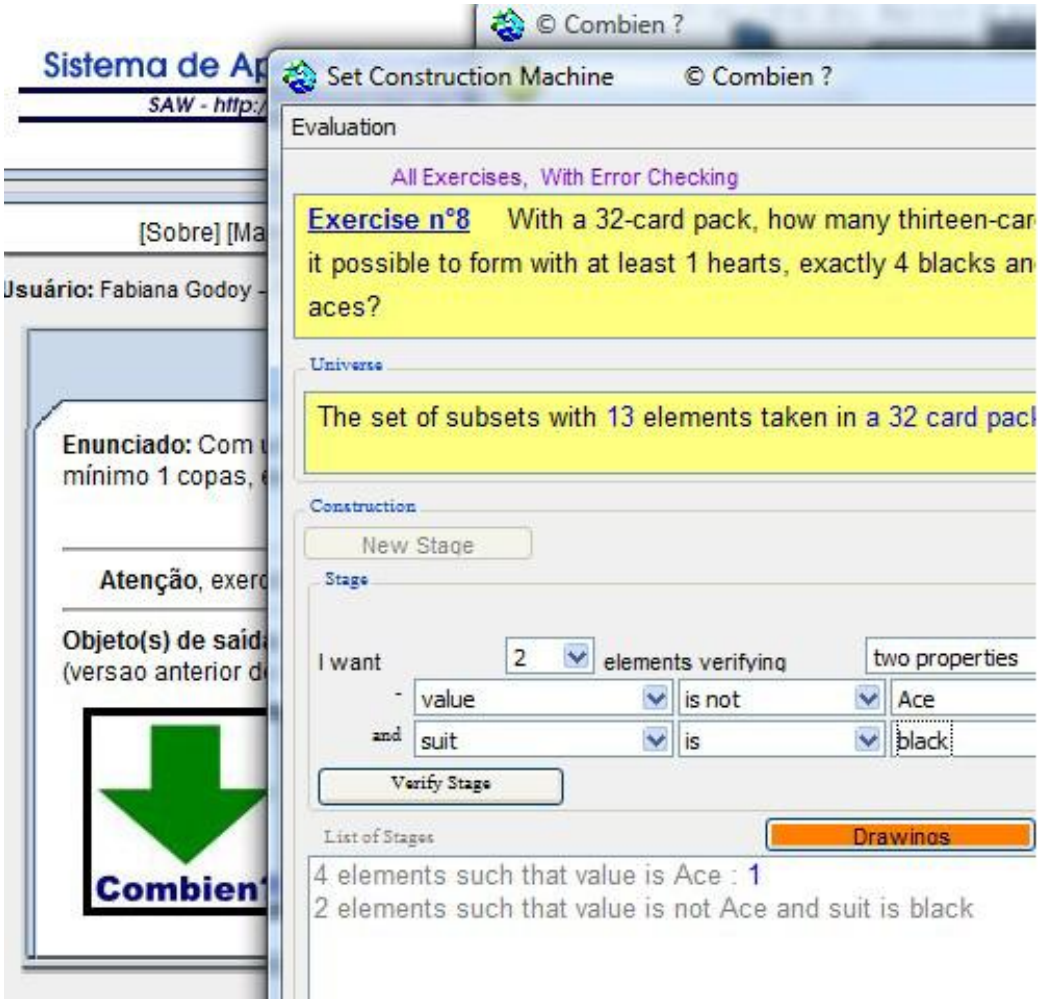

Figura 2.4: Resolução do Exercício pelo ***Combien?***

- O *CombienDelivery* exige uma primeira instalação do ***Combien?*** nas máquinas. Esse passo pode onerar em muito a disponibilidade do ***Combien?*** para alunos e professores. É provável que frente a este tipo de dificuldade os professores simplesmente desistam de utilizar a alternativa sem ao menos tentar.

- Apesar de mínima, o *CombienDelivery* supõe pequenas alterações no sistema ***Combien?***, em especial a chamada via linha de comando. Desta forma, o desenvolvimento do *CombienDelivery* está condicionado a alteração do sistema ***Combien?***, fato externo e não controlável.

- O *CombienDelivery* necessita monitorar o sistema de arquivos da máquina a partir da aplicação Web. Desta forma, foi necessário a utilização de *applets* assinados exigindo do usuário a permissão explícita da ação do componente.

- A atual distribuição do ***Combien?*** possui um conjunto pré-definido de exercícios, não sendo possível a inclusão de novos. Naturalmente esta característica é essencial para a verificação e validação do sistema. Uma vez que o universo dos exercícios é fixo é possível comparar a experiência entre várias amostras de alunos. Porém, no contexto da educação a distância, é desejável certo grau de flexibilidade.

- A atual distribuição do ***Combien?*** não foi traduzido para o português e não possui mecanismo de internacionalização, fato essencial para a aplicação do sistema no Brasil.

Capítulo 3

iComb - Combinatória Interativa na Internet

> *"The difference between school and life? In school, you're taught a lesson and then given a test. In life, you're given a test that teaches you a lesson."*
>
> Tom Bodett

A experiência do grupo ***Combien?*** mostrou que é possível conceber um sistema eficaz no aprendizado de combinatória. A concepção do "método construtivo" possibilitou a criação de interfaces que direcionam o aluno à correta solução do problema. Além disso, conforme reportado pelo grupo, o sistema possibilita a compreensão do modelo subjacente aos problemas e, portanto, uma maior compreensão da disciplina.

A detecção automática de erros possibilita resposta imediata ao aluno de forma a reduzir o atraso entre surgimento da dúvida e apresentação da explicação. Desta forma, evita-se o esquecimento por parte do aluno do contexto onde a dúvida surgiu.

O registro das ações do aluno é uma ferramenta eficaz de análise, uma vez que é possível comparar resoluções entre vários alunos e perceber padrões de erros específicos, informação valiosa para que o professor possa direcionar seus esforços.

3.1 Motivação

Se por um lado o ***Combien?*** mostrou-se proveitoso ao ensino de combinatória, por outro ele apresenta várias dificuldades para uma boa incorporação a um sistema Web, como ficou

claro na seção 2.4. Deste modo, a motivação deste trabalho foi conceber, implementar e testar um modelo que contornasse a incorporação das idéias didáticas do ***Combien?*** ao ambiente Web. Assim, buscou-se resolver:

- **Baixa Integração**: O ***Combien?*** não foi concebido para ser extensível e integrável. Não há modo natural de integrá-lo a grandes sistemas LMS, como o Moodle ou o próprio SAW. O ***Combien?***, portanto, não é sensível ao contexto das aulas e da relação aluno-professor. O professor não pode, por exemplo, criar um novo exercício e enviar a seus alunos remotamente.

- **Baixa Disponibilidade**: Tipicamente, qualquer esforço de inclusão de novas tecnologias em escolas e universidades brasileiras parte da iniciativa de um grupo ou até mesmo de um único professor. Uma aplicação *desktop* exige a realização do *download* e a subsequente instalação. O problema ganha escala quando várias máquinas devem possuir a instalação, seja na residência dos alunos ou num laboratório da instituição de ensino. Comparado com uma aplicação Web, a disponibilidade de uma aplicação *desktop* é bastante baixa e pode diminuir a probabilidade de sucesso do projeto.

- **Diferença Cultural**: Muitos dos exercícios do ***Combien?*** são baseados no universo do baralho. Um modelo conceitual simples como este permite que os alunos tenham uma compreensão imediata do problema exposto não sendo necessárias maiores explicações.

  Não é óbvio que este modelo conceitual é tão natural para os alunos brasileiros como são para os alunos franceses, principalmente quando levamos em consideração as grandes diferenças sócio econômicas que o Brasil possui. Um sistema preparado para trabalhar com universos alternativos seria de grande utilidade para ambientes heterogêneos.

- **Idioma**: O ***Combien?*** não possui versão para a língua portuguesa. A instalação atual conta com os idiomas Francês, Inglês e Espanhol.

3.2 Princípios Fundamentais

O objetivo deste trabalho foi definir, implementar e testar um sistema Web baseado no ***Combien?***, de tal modo que o mesmo possa ser incorporado a sistema gerenciadores de cursos como um módulo interativo de aprendizagem. O sistema resultante deste esforço é o *iComb: Combinatória Interativa na Internet*. A partir das dificuldades encontradas com o sistema ***Combien?***, estabeleceu-se que o ***iComb*** deveria atender aos seguintes requisitos:

- **Integrável**: O ***iComb*** deve ser integrável a outros sistemas Web.
- **Disponível**: A instalação e utilização do ***iComb*** deve ser tão fácil quanto possível, devendo funcionar em computadores com alguns anos de defasagem (para não inviabilizar seu uso em escolas, ou por alunos, com equipamentos defasados).
- **Aderência cultural**: Os alunos devem entender facilmente os problemas propostos, permitindo a utilização de variados universos (uma vez que o Brasil é um país grande, com diferenças culturais).
- **Idioma português**

A partir da especificação destes requisitos ficou clara a necessidade de um "novo sistema", ***iComb***, a ser implementado em linguagem Java. A partir daí foram estabelecidos mais claramente os requisitos que o ***iComb*** deveria atender, sendo:

- **Módulo de aprendizagem (MA)**: O ***iComb*** deve ser facilmente integrado a sistemas gerenciadores de curso, como o SAW, além de poder utilizá-lo como aplicativo ou como um *applet* em qualquer página Web.
- **Internacionalização**: O sistema deverá ser preparado a funcionar em vários idiomas. A versão inicial contemplará o idioma português.
- **Autoria de exercícios**: O professor deverá ser capaz de publicar novos exercícios.
- **Modelos de Universos**: o ***iComb*** deve permitir sua utilização em vários universos distintos, não apenas cartas de baralhos, visando um maior escopo de problemas adequados às realidades locais.

3.3 Componentes

3.3.1 Arquitetura de *Plugins*

Um dos princípios do ***iComb*** é permitir a criação de novos exercícios de forma a enriquecer a experiência de aprendizes e professores. Além disso, o ***iComb*** não está restrito apenas a um único domínio ou universo, como por exemplo o universo das cartas de baralho. No ***iComb*** poderemos não só definir novos exercícios, como também novos universos de forma não programática.

Chamaremos de *plugin* um pacote de informações sobre determinado universo que acompanha a distribuição do ***iComb***. Naturalmente não deverá existir qualquer acoplamento entre o ***iComb*** e um determinado *plugin*, em outras palavras o ***iComb*** não deverá ter qualquer conhecimento sobre qualquer *plugin* de universo de forma a propiciarmos a extensibilidade.

Como módulo acadêmico integrável ao sistema SAW o ***iComb*** é uma *applet* Java. O núcleo básico do ***iComb*** está empacotado num arquivo `icomb.jar`. A definição de um universo específico é distribuída num arquivo distinto. Por exemplo, o *plugin* do universo *baralho* é distribuído no arquivo `baralho.jar`.

Um *plugin* de universo consiste de um arquivo `.xml` que define o universo e de um diretório `imagens` que contém as imagens de todos os elementos do universo. Compactando estes arquivos no formato `.jar` temos um *plugin* definido e *plugável* ao ***iComb***.

A figura 3.1 apresenta um trecho do arquivo `baralho.xml`. A *tag* `<conjunto>` agrega um número arbritário de elementos, definidos pela *tag* de mesmo nome. Cada elemento possui um nome e uma imagem, informações definidas como propriedades da *tag* `<elemento>`. Na propriedade *imagem* espera-se que o nome do arquivo de imagem que representa este elemento seja informado, enquanto o arquivo de imagem propriamente dito deverá estar localizado no diretório `imagens`.

Cada elemento possui atributos definidos em *tag* específica. Um atributo possui nome e valor. O nome de um atributo é definido como propriedade da *tag* `atributo` enquanto o valor é definido numa *tag* distinta.

Cada atributo pode ter uma coleção de predicados. Predicados são informações simples que qualificam o atributo. Em nosso exemplo estamos qualificando o naipe “Ouros”

```
<universo nome="Baralho de 32 cartas" >
  <elemento nome="Ás de ouros" imagem="ad.gif">
     <atributo nome="suit">
       <valor>diamonds</valor>
       <predicado>red</predicado>
     </atributo>
     <atributo nome="value">
       <valor>ace</valor>
     </atributo>
  </elemento>
  <elemento nome="7 de ouros" imagem="7d.gif">
     <atributo nome="suit">
       <valor>diamonds</valor>
       <predicado>red</predicado>
     </atributo>
     <atributo nome="value">
       <valor>7</valor>
     </atributo>
  </elemento>
  <elemento nome="8 de ouros" imagem="8d.gif">
     <atributo nome="suit">
       <valor>diamonds</valor>
       <predicado>red</predicado>
     </atributo>
     <atributo nome="value">
       <valor>8</valor>
     </atributo>
  </elemento>
```

Figura 3.1: Arquivo baralho.xml

como sendo “vermelho”. Um atributo poderá ter mais do que um predicado, por exemplo poderemos qualificar a carta “Rei de Copas” como “figura”.

Cada exercício refere-se a determinado universo, assim o ***iComb*** poderá reformatar sua *interface* conforme o universo em questão. A figura 3.2 ilustra o momento em que o aprendiz informa a condição para um estágio. Observe que a caixa de combinação mais a esquerda apresenta todos os atributos possíveis enquanto que a caixa de combinação mais a direita, que está aberta, apresenta os predicados relacionados aquele atributo.

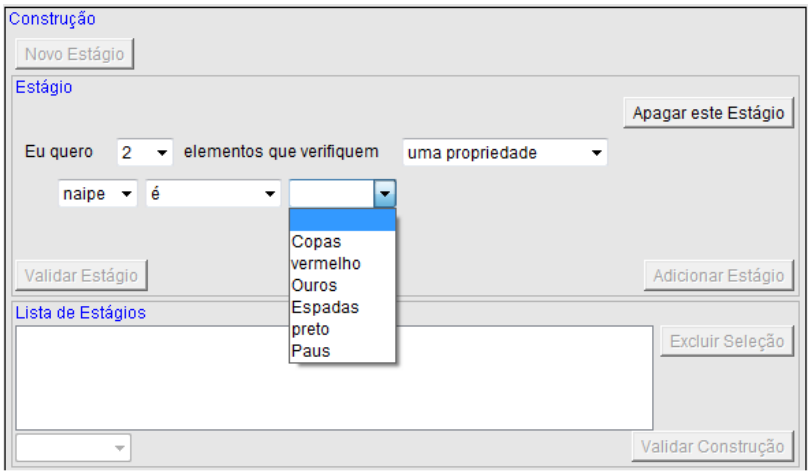

Figura 3.2: Secção construção do ***iComb***

Conforme o aprendiz vai construindo os estágios, o ***iComb*** apresenta uma instância possível de elementos selecionados conforme ilustra a figura 3.3.

Figura 3.3: Elementos apresentados em secção do ***iComb***

A figura 3.4 apresenta os elementos do universo *futebol*. Neste domínio, cada imagem representa um jogador das quatro possíveis equipes: azul, verde, vermelha e amarela. O goleiro possui uma imagem levemente diferenciada comparado aos demais jogadores da equipe.

Figura 3.4: Elementos do universo *futebol* apresentados em secção do ***iComb***

3.3.2 Internacionalização

Internacionalização de um sistema é um processo ou princípio de desenvolvimento que procura adaptar um sistema à língua e cultura de um país. Um importante aspecto da internacionalização é que o sistema não é reescrito para ser adaptado a outro idioma ou cultura. Todas as alterações são feitas em tempo de configuração e de forma não programática. O acrônico `I18n` é muitas vezes utilizado para designar o processo de internacionalização uma vez que entre a primeira letra "I" e a última letra "n" da palavra inglesa *internacionalization* há 18 letras.

Dependendo dos requisitos do sistema em questão é desejado prover uma estrutura de internacionalização capaz de tornar o sistema adaptado em relação a vários aspectos, entre eles idiomas, formato de data e tempo e formatação de unidades monetárias. Enquanto formatação de datas e unidades monetárias pode ser confiada a bibliotecas previamente escritas, a internacionalização referente a mudança de idiomas exige a criação de um arquivo específico para cada idioma desejado, uma vez que este conhecimento é dependente do domínio, em outras palavras, é dependente do sistema que está sendo desenvolvido.

O sistema de internacionalização do ***iComb*** é bastante simples, refere-se apenas a mudança de idiomas. Não há necessidade de mudanças específicas relacionadas a padrões de datas ou valores monetários uma vez que o ***iComb*** não utiliza estes conceitos. É importante observar também que a arquitetura de *plugins* de universos não prevê internacionalização. Isso significa que os universos são mono idioma não havendo mecanismos para *internacionalizá-los*.

Na atual distribuição o ***iComb*** conta com três idiomas, português, inglês e francês, respectivamente definidos em `i18n_pt_BR.properties`, `i18n_en_US.properties` e `i18n_fr_FR.properties`.

3.3.3 Mecanismo de Ajuda

O sistema de ajuda do ***iComb*** possui três estruturas de ajuda aos usuários, cada qual voltada a uma dimensão específica. O sistema de ajuda tradicional, acionado pelo botão `Ajuda`, apresenta os conceitos do ***iComb*** de forma geral (figura 3.5). O sistema de ajuda contextual apresentado pelo quadro de nome `Ajuda` como vemos na figura 3.6. Este quadro é atualizado em função da operação que o usuário vai realizar.

Por fim o ***iComb*** pode apresentar ao usuário os elementos do universo relacionado ao exercício em questão, não só a imagem de cada elemento como também a descrição de seus atributos. A figura 3.7 apresenta a tela que é acionada pelo botão `Mostrar Universo`.

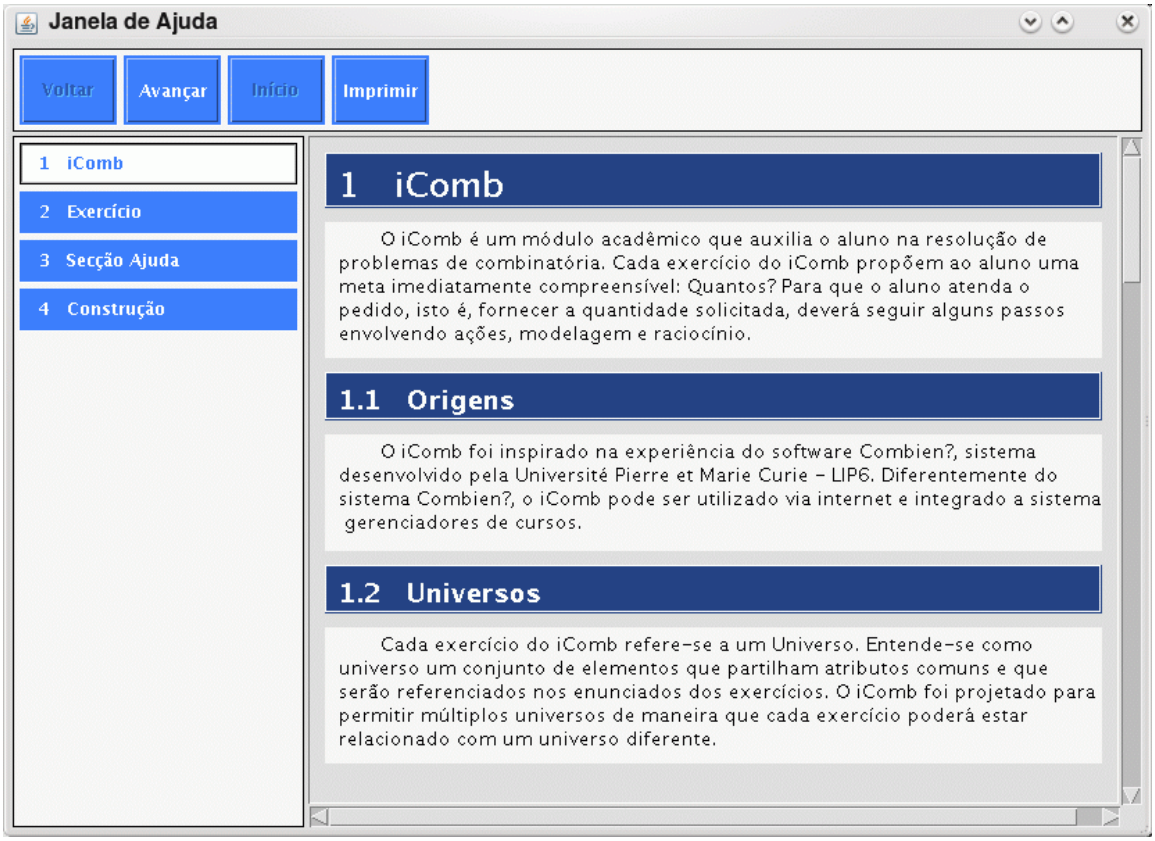

Figura 3.5: Sistema de ajuda do ***iComb***

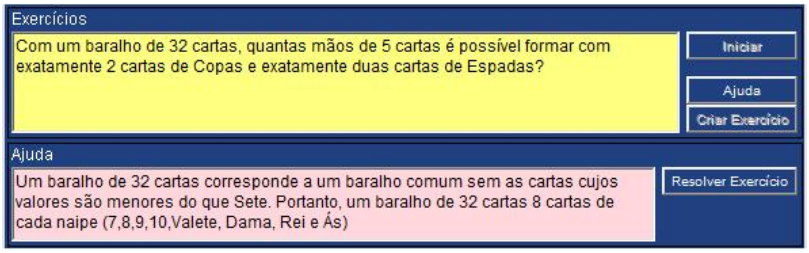

Figura 3.6: Sistema de ajuda contextual do ***iComb***

Figura 3.7: Tela que apresenta o universo em questão

3.3.4 Integração com outros sistemas

O ***iComb*** foi projetado para ser facilmente integrável com outros sistemas LMS, em especial o sistema SAW. O ***iComb*** é um *applet* Java, portanto executado pela máquina virtual Java presente no navegador. Dependendo dos parâmetros passados, o ***iComb*** terá seu comportamento ajustado para aquele conjunto de informações, como por exemplo o exercício desejado, a solução esperada e o universo em questão.

A comunicação entre servidor-***iComb*** se dá através de parâmetros do *applet*, em outras palavras, o cliente solicita uma requisição para o servidor e o mesmo responde com uma página `HTML` e a sua respectiva *tag* `APPLET` devidamente montada, incluindo todos os parâmetros necessários. Neste ponto o aluno estará interagindo diretamente com o ***iComb*** em seu navegador. Uma vez concluído o exercício as informações do ***iComb*** devem ser reenviadas ao servidor. Esta comunicação, na direção ***iComb***-servidor é realizada por um método `javascript` específico que irá colher as informações do *applet* e enviá-las finalmente ao servidor.

A figura 3.8 mostra os parâmetros necessários para uma chamada ao ***iComb***. O parâmetro `EnumGabarito` contém o enunciado do exercício em questão. O parâmetro `solucao` contém a codificação de uma solução para o exercício em questão, como vimos anteriormente, esta informação é suficiente para que o ***iComb*** corrija qualquer solução válida. O parâmetro `resposta` contém o valor esperado como resposta, independente da solução fornecida. Em `Dica` espera-se um texto que será apresentado como dica ao aprendiz no momento em que dá início à resolução do exercício.

A definição de qual é o universo em questão deve ser informada no parâmetro `ArquivoUniverso`, especificamente o endereço do arquivo `.xml` que contém a definição do universo. Observe que o ***iComb*** irá procurar tal arquivo dentro dos arquivos `.jar` listados na `tag ARCHIVE`.

A figura 3.9 ilustra a comunicação no sentido ***iComb***-servidor. O método `pegaResposta()` é capaz de acessar diretamente métodos da *applet* para enviá-los ao servidor.

3.3.5 Segurança

A segurança de um sistema pode ser ameaçada em todos os momentos de seu ciclo de vida, seja de forma não intencional ou na forma de ataques deliberados realizado ora por pessoas ligadas à própria organização, os chamados *insiders*, ou por pessoas externas à

```
<APPLET name="iComb" CODE="icomb.IComb.class" CODEBASE="."
        ARCHIVE="iComb.jar, baralho.jar" WIDTH=600 HEIGHT=685>
        <PARAM NAME=EnumGabarito
                VALUE="Com um baralho de 32 cartas, quantas mãos de 13 cartas
                       é possível formar com 12 cartas vermelhas,
                       exatamente 3 Ases e no máximo 4 cartas de Ouros?">
        <PARAM NAME=solucao
                VALUE="8:suit={hearts};1:suit={diamonds} E value={ace};1:suit={black}
                       E value={ace};3:suit={diamonds} E value#{ace}">
        <PARAM NAME=resposta VALUE="70">
        <PARAM NAME=ArquivoUniverso VALUE="baralho/baralho.xml">
        <PARAM NAME=Dica
                VALUE="Um baralho de 32 cartas corresponde a um baralho
                comum sem as cartas cujos valores são menores do que Sete.
                Portanto, um baralho de 32 cartas 8 cartas de cada
                naipe (7,8,9,10,Valete, Dama, Rei e Ás)">
</APPLET>
```

Figura 3.8: Parâmetros para a *applet* do ***iComb***

```
<script type="text/javascript">
 function pegaResposta() {
   var resposta = document.iComb.getAnswer();
   var avaliacao = document.iComb.getAvaliacao();
   var trace = document.iComb.getTrace();
   ...
 }
```

Figura 3.9: Método javascript que acessa métodos da *applet* e os envia ao servidor

organização ou *outsiders* (Goertzel, 2009). Ataques contra a segurança podem explorar vulnerabilidades do sistema na fase de desenvolvimento, distribuição e instalação, operação e manutenção (Goertzel, 2009) de maneira que um sistema robusto com relação a segurança exige alto grau de esforço e investimento.

Dependendo da importância e confidenciabilidade das informações processadas pelo sistema, a exigência quanto a segurança pode ser tornar maior ou menor. A tecnologia utilizada e abrangência do sistema também influencia nos modelos de segurança utilizados, de maneira que, tipicamente, uma aplicação Web, por estar disponível para qualquer pessoa exige preocupação elevada com segurança. Como todos sabemos, por exemplo, a expectativa quanto a segurança em sistemas bancários na Web é bastante elevada. A questão da segurança num módulo acadêmico como o ***iComb*** está relacionada com a garantia de que um agente externo esteja impossibilitado de conseguir o gabarito de um determinado exercício por vias não previstas pelo sistema. Porém, como iremos verificar a seguir, este é um problema difícil e ainda em aberto.

A figura 3.10 apresenta três etapas do ciclo de vida da utilização do ***iComb***. A etapa

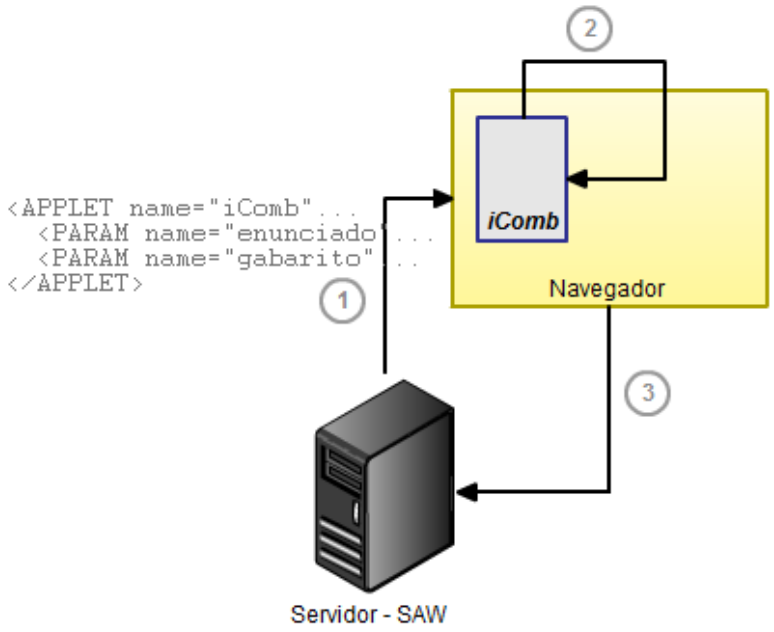

Figura 3.10: Segurança

(1) é a resposta da solicitação do cliente para a utilização do ***iComb***, onde a *tag* `APPLET` é enviada ao navegador com pelo menos os parâmetros `enunciado` e `gabarito`.

A etapa (2) representa a execução do ***iComb***. É importante observar que durante a operação do ***iComb*** o servidor não é mais solicitado e todo o processamento ocorre no navegador cliente. Esta característica da arquitetura é fundamental para discutirmos segurança uma vez que o ***iComb*** necessita de toda informação da resposta antes mesmo do aluno começar a utilizá-lo.

A etapa (3) representa o envio da resposta fornecida pelo aluno ao servidor.

iComb é auto contido

Uma importante decisão quanto a arquitetura do ***iComb*** é a exigência de que o mesmo seja auto contido, isto é, seja independente de outros componentes ou outras camadas durante seu ciclo de vida. A figura 3.10 mostra que depois da etapa (1) *instanciar* o ***iComb*** e antes da etapa (3) enviar os dados novamente ao servidor, o ***iComb*** não realiza qualquer comunicação com outros componentes que não seja ele próprio.

Algumas razões para a manutenção do ***iComb*** independente e auto contido são:

- Simplifica a integração do ***iComb*** com diversos sistemas LMS e permite a inclusão do ***iComb*** até mesmo em *blogs* através do conceito de *widgets*.
- Torna o desenvolvimento da aplicação bem mais simples, uma vez que só existe uma camada a ser considerada.
- O ***iComb*** registra todas as decisões tomadas pelos alunos assim como o tempo necessário para a execução. Numa arquitetura que envolve mais componentes e camadas, o registro temporal sofreria algum tipo de viés decorrente do tempo de comunicação entre as camadas. Se, por exemplo, cada ação do usuário fosse validada no servidor, não seria possível diferenciar tempo de conexão de demora do aluno.

Por outro lado, a independência do ***iComb*** expõe o módulo a um problema de segurança. Como vimos em capítulo anterior, para que o ***iComb*** possa corrigir os alunos a cada decisão, uma das possíveis soluções deve ser de conhecimento do ***iComb***, assim o mesmo

deverá ser informado pelo servidor não só sobre qual é o exercício em questão, mas também qual é a solução para o problema. Enfim, a solução deve ser enviada para o navegador e para o ***iComb*** mas não pode ser lida pelos alunos.

Segurança nos navegadores

Muita energia e esforço foi gasto para garantir que as informações enviadas aos navegadores não sejam interceptadas e lidas por agentes externos durante este processo. O sistema criptográfico de chave pública e privada garante que a informação que chega ao nosso navegador (a) não foi lida de forma indevida por pessoas maliciosas e (b) foi enviada pelo remetente que esperávamos. Assim, quando movimentamos nosso dinheiro pelo *internet banking* ou quando realizamos a compra de algum artigo através da *internet* temos grande confiança que as operações serão realizadas de forma segura.

Por outro lado, a garantia oferecida pelos sistemas de segurança usuais na *internet* não é suficiente para as necessidades do ***iComb***. Gostaríamos, para os propósitos do ***iComb***, de que as informações enviadas aos navegadores não pudessem ser lidas pelo próprio usuário, mas apenas pelo *applet* do ***iComb***. Fazendo uma analogia com os sistemas bancários, é como se desejássemos criar um *internet banking* onde o próprio usuário não pudesse ler o código `HTML` gerado para apresentar seu extrato bancário.

Como o ***iComb*** é executado no navegador, toda informação passada para o mesmo pode ser potencialmente lida por um aluno suficientemente motivado e talentoso. Enfim, em termos absolutos não é possível passar informações ao módulo ***iComb*** de forma totalmente segura. Observe que mesmo que as informações fossem criptografadas o problema não seria resolvido, afinal o ***iComb*** deverá descriptografá-la e portanto receber a informação da chave criptográfica. Mesmo utilizando um algoritmo sem chave criptográfica, o aluno poderá descompilar a classe java do ***iComb*** e ler o algoritmo perfeitamente. Enfim, não há como resolver de maneira absoluta o problema.

3.3.6 Web 2.0 e *iComb* como *Widget*

A Web 2.0 é hoje um conceito largamente utilizado porém bastante impreciso e fugidio para uma definição formal. Como exposto em (Oreilly) o conceito Web 2.0 não possui limites rígidos mas sim um "núcleo gravitacional" onde um conjunto de princípios e práti-

cas é amarrado a um verdadeiro "sistema solar" de páginas que demonstram alguns destes princípios, numa distância variável deste núcleo.

Este conjunto de princípios e práticas reforça a noção de que os sistemas possuem mais valor à medida que mais pessoas os utilizam, de maneira que a colaboração e a troca de informação é essencial. À medida que o indivíduo ganha expressividade com as novas ferramentas da Web 2.0 como páginas *wiki*, *blogs* ou redes sociais, os mesmos tornam-se centrais para o desenvolvimento do mundo virtual.

Widgets são componentes computacionais integráveis a páginas Web através da simples operação de copiar e colar pedaços de código `HTML`. A simplicidade e independência dos *widgets* permite que qualquer pessoa possa integrá-los a *blogs* ou páginas pessoais tornando o conceito totalmente aderente a plataforma Web 2.0. A figura 3.11 mostra a disponibilidade da *widget* do ***iComb*** na página do projeto.

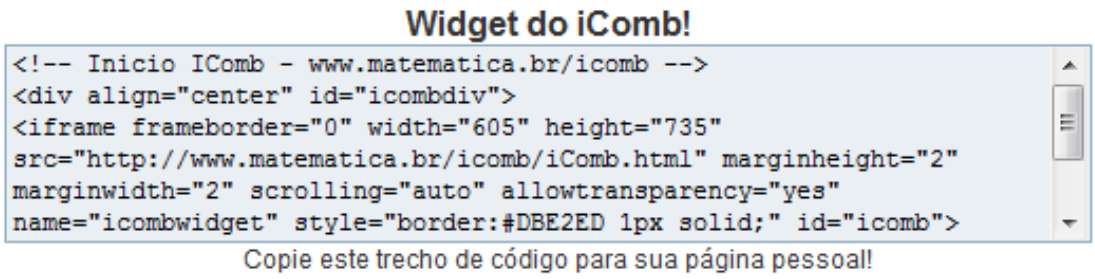

Figura 3.11: *Widget* do ***iComb***

Disponibilizar o ***iComb*** como um *widget* foi motivado pela idéia de trazer a Matemática para a Web 2.0 e encorajar o desenvolvimento de uma família de "*widgets* matemáticos" para enriquecer o cenário da educação matemática na *internet*. A utilização e desenvolvimento de *widgets* no contexto da educação matemática é raramente feito, por outro lado indivíduos e instituições de ensino demandam ferramentas colaborativas que possam ser facilmente compartilhadas em redes sociais.

3.4 Funcionamento do *iComb*

Nesta secção apresentaremos o funcionamento do ***iComb*** e sua integração com o sistema SAW. A sequência que iremos seguir apresenta a utilização do ***iComb*** do ponto de vista do

aprendiz utilizando o sistema SAW. O exercício escolhido faz parte do universo *"Cartas do baralho"*.

Supondo que o professor cadastrou um exercício no sistema SAW atrelado ao módulo de aprendizagem ***iComb***, o aprendiz, ao utilizar o SAW, observará em seu contexto a presença de mais um novo exercício. Diferentemente do ***Combien?***, no ***iComb*** é permitido que o professor defina o exercício a ser resolvido.

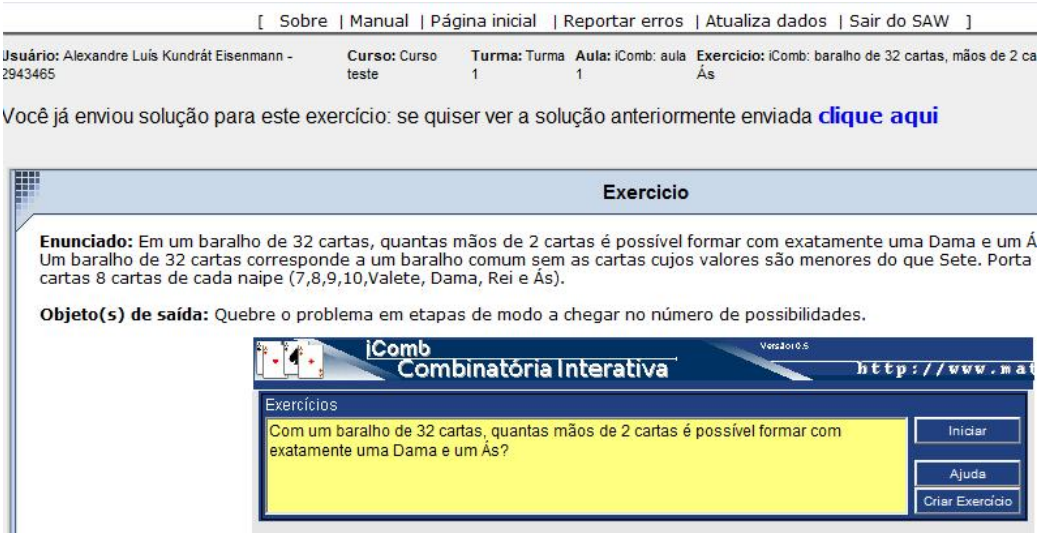

Figura 3.12: Tela inicial do ***iComb***

A figura 3.12 apresenta a tela inicial do sistema ***iComb***. O aluno poderá visualizar o enunciado do exercício previamente cadastrado pelo professor com a tela do ***iComb*** abaixo.

Ao acionar o botão `Iniciar`, o aluno dará início à resolução do exercício diretamente. Observe a sensível diferença entre o ***iComb*** e o módulo *CombienDelivery* tratado anteriormente. Como o *CombienDelivery* abre o próprio ***Combien?*** não é possível definir previamente o exercício sem que o código do próprio ***Combien?*** seja alterado. Já o ***iComb*** foi preparado para este tipo de utilização, então o exercício cadastrado pelo professor será justamente aquele que o aluno irá observar como mostra a figura 3.12.

Na sequência o aluno deverá incluir os estágios necessários a resolução do exercício, etapa representada na figura 3.13.

A figura 3.14 apresenta a interface do ***iComb*** após a inclusão de dois estágios. No momento em que a imagem foi capturada o aluno preparava o terceiro estágio. Observe a

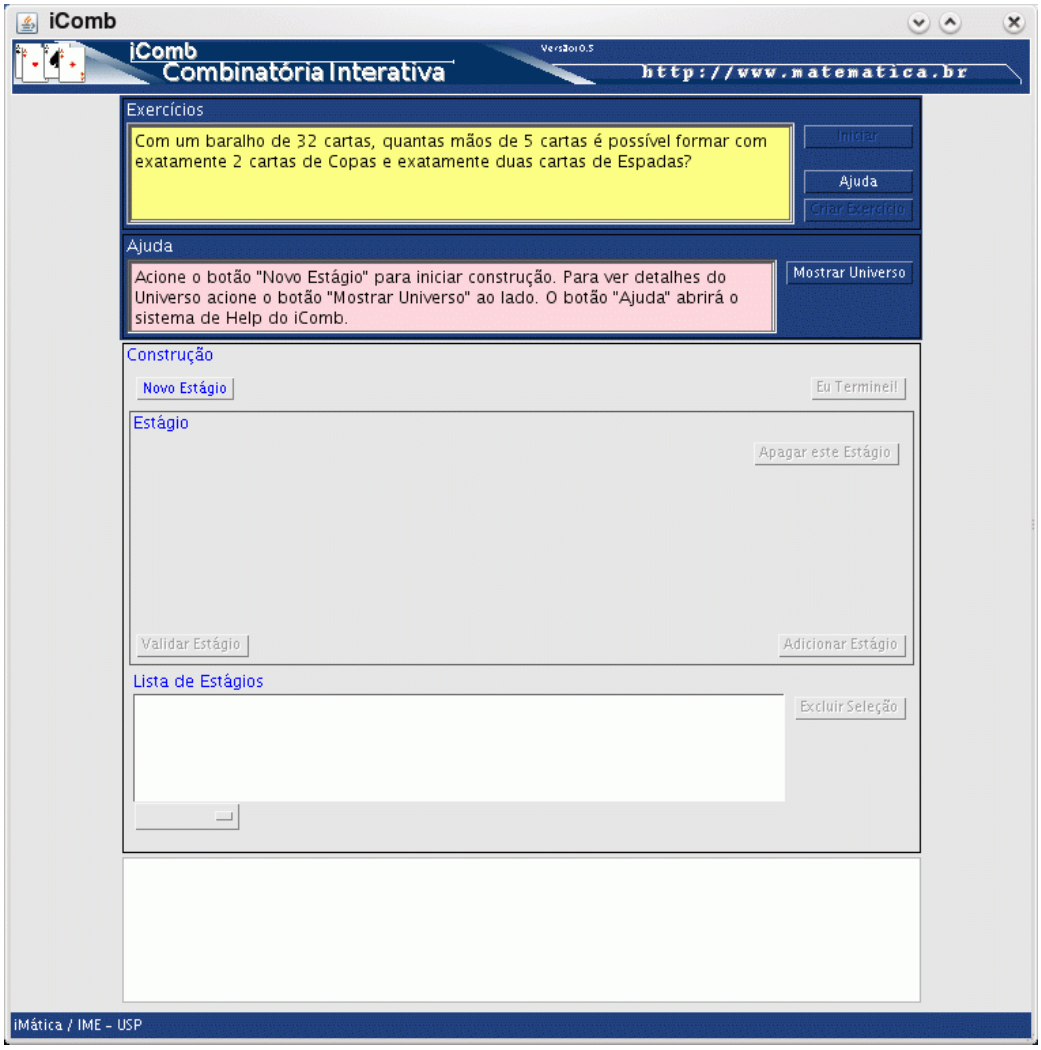

Figura 3.13: Tela do ***iComb***

lista de estágios previamente criados. Uma diferença essencial em relação ao ***Combien?*** é apresentação de imagens reais das cartas ao invés da indicação textual. Neste caso específico, duas cartas de copas e duas cartas de espadas são apresentadas em resposta a inclusão dos dois primeiros estágios.

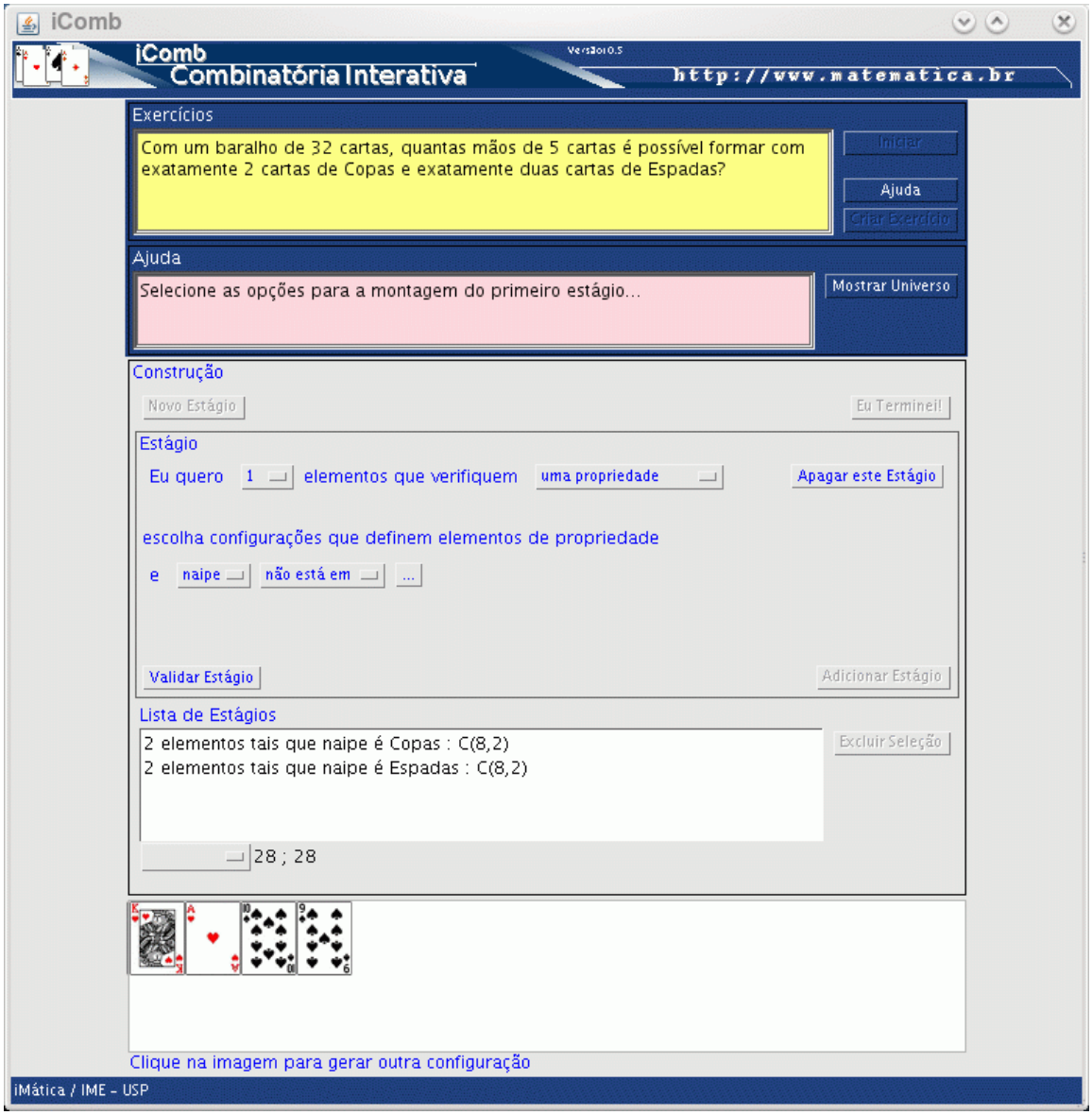

Figura 3.14: Resolvendo exercícios no ***iComb***

Depois do último estágio criado, o aprendiz caminha para os momentos finais. Neste ponto, como apresentado na figura 3.15, o conjunto de cartas abaixo apresenta as cinco

cartas solicitadas no exercício. A qualquer tempo o aprendiz poderá observar uma configuração de cartas coerente com os estágios definidos até o momento.

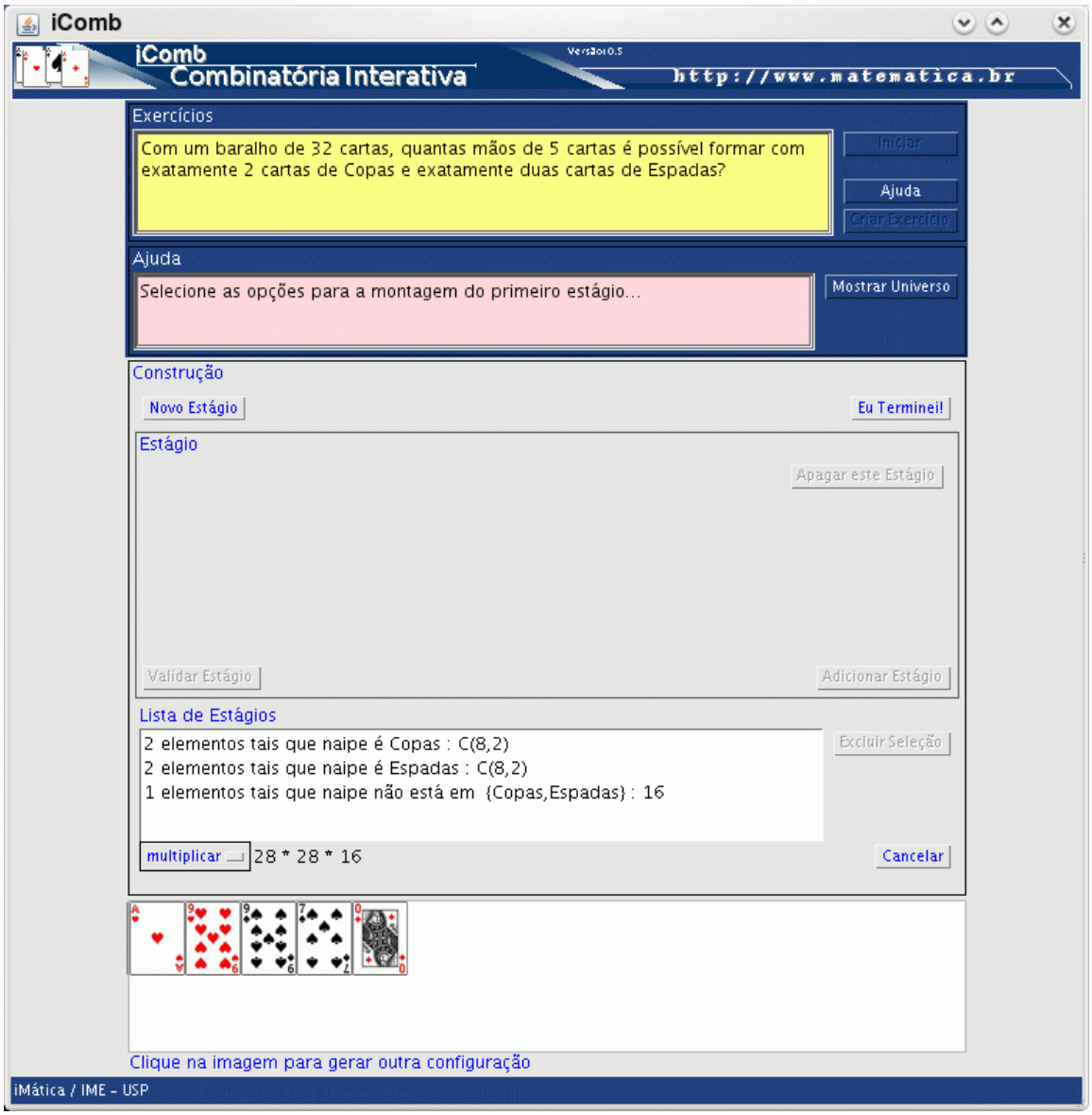

Figura 3.15: Etapa Estágio finalizada

Depois de validada a construção, uma última tela é apresentada ao aluno. Neste ponto o aluno irá enviar a solução para a infra-estrutura do SAW acionando o botão "Enviar" como mostra a figura 3.16.

Ao contrário do *CombienDelivery*, podemos mandar a solução diretamente para o

SAW. No *CombienDelivery* era necessário monitorar o sistema de arquivos do computador a espera de um novo arquivo, além disso, o aprendiz era obrigado a escolher o diretório adequado. No ***iComb***, com o controle do código fonte, é possível eliminar totalmente esses passos de maneira que o exercício é enviado ao sistema SAW imediatamente após o comando "Enviar".

Figura 3.16: Enviar exercício concluído ao SAW

Finalmente a solução é enviada para o SAW e o professor poderá verificar a solução consultando o menu específico do sistema SAW.

Capítulo 4

Detecção Automática de Erros

"We learn wisdom from failure much more than from success. We often discover what will do by finding out what will not do; and probably he who never made a mistake never made a discovery."

Samuel Smiles

"A verdade surge mais facilmente do erro do que da confusão."

Francis Bacon

4.1 Introdução

Um mecanismo de detecção automática de erros é um componente capaz de analisar cada interação do usuário e detectar ações que podem resultar em erro. Um bom mecanismo irá advertir o usuário em todos os momentos onde o mesmo cometer um erro, porém nem sempre é possível projetar um mecanismo com esta característica. Em alguns casos somente uma detecção parcial será possível, de maneira que pode haver situações em que o usuário cometeu uma decisão errada, mas o mecanismo não é capaz de detectá-la. Assim, a criação de algoritmos de detecção de erros depende da classe de problemas, ainda podendo apresentar variado grau de precisão.

Nessa seção será analisada a natureza e estrutura dos problemas propostos pelo ***iComb***, discutindo-se até que ponto é possível a criação de um algoritmo de detecção de erro. O objetivo será definir uma classe de problemas e um algoritmo que seja correto para esta classe. Neste contexto, será dito que um *algoritmo é correto para uma classe de*

problemas quando, para qualquer problema da classe e qualquer que seja a proposta de solução apresentada, o algoritmo consegue detectar se esta é correta ou não.

4.2 Estrutura do Problema

Todos os problemas propostos pelo ***iComb*** tem um enunciado que respeita sempre a mesma estrutura: "Quantos conjuntos de N objetos é possível formar respeitando as restrições ...". Indiretamente os problemas do ***iComb*** também referem-se a um determinado universo, facilmente reconhecível pelo aprendiz, de maneira que é possível transformar o modelo do enunciado acima para "Considerando o universo U, quantos conjuntos de N objetos é possível formar respeitando as restrições...".

Por exemplo, considere o problema "Com um baralho de 32 cartas, quantas mãos de 5 cartas é possível formar com exatamente duas cartas de Copas e exatamente duas cartas de Espadas?". O universo considerado é "Um baralho de 32 cartas", o tamanho dos conjuntos de interesse é $N = 5$ e as restrições são "duas cartas de Copas"e "duas cartas de Espadas".

Uma vez apresentado o problema, o ***iComb*** desafia o aprendiz a encontrar a cardinalidade do conjunto resposta para o problema utilizando a *interface* proposta. O aprendiz deverá então definir uma série de *Estágios* que resultará na *Solução* para o problema.

Um Estágio possui uma estrutura bem definida, é composto de uma condição e de um número desejado, por exemplo, um dos estágios para resolver o problema acima poderia ser: “Eu quero `[2]` elementos que verifiquem `[uma propriedade]: [naipe] [é] [Copas]`”.

Formalmente será utilizada a letra U para representar o conjunto universo. Quando há ambiguidade será utilizado um identificador como subscrito. Por exemplo U_{32} representa o conjunto das cartas de um baralho de 32 cartas, U_{52} representa o conjunto das cartas de um baralho de 52 cartas, enquanto U_F representa o conjunto dos jogadores do domínio Futebol. Observe abaixo o universo das figuras de um baralho U_{16}.

$$U_{16} = \{ {}^{J}_{\diamondsuit}, {}^{Q}_{\diamondsuit}, {}^{K}_{\diamondsuit}, {}^{A}_{\diamondsuit}, {}^{J}_{\spadesuit}, {}^{Q}_{\spadesuit}, {}^{K}_{\spadesuit}, {}^{A}_{\spadesuit}, {}^{J}_{\heartsuit}, {}^{Q}_{\heartsuit}, {}^{K}_{\heartsuit}, {}^{A}_{\heartsuit}, {}^{J}_{\clubsuit}, {}^{Q}_{\clubsuit}, {}^{K}_{\clubsuit}, {}^{A}_{\clubsuit} \}$$

A seguir serão apresentadas uma série de definições importantes para formalizar este tipo de abordagem presente no ***iComb***.

Definição 4.1. Um *Universo* é um conjunto de elementos que partilham características

comuns, em outras palavras, que partilham os mesmos atributos. Cada problema proposto deixará bem claro qual é o universo de interesse. Um universo pode ser composto por cartas de um baralho, jogadores de times de futebol ou qualquer outro conjunto a ser definido futuramente. Como já foi citado, um universo será usualmente tratado por U e, quando necessário, um indentificador em subscrito acompanhará a notação. Assim, por exemplo U_{52} representará o universo das 52 cartas de um baralho comum.

Definição 4.2. Um *Problema* apresenta uma questão sobre um determinado universo. Precisamente, um problema será uma pergunta sobre qual a quantidade de subconjuntos de um universo é possível formar tal que determinadas condições sobre os elementos sejam respeitadas.

Definição 4.3. Será denominado como *Conjunto Resposta* de um problema o conjunto formado conforme solicita o enunciado do problema. Naturalmente a *Resposta* do problema é igual a cardinalidade do conjunto resposta.

Definição 4.4. Uma *Condição* é uma expressão que avalia um elemento do universo resultando em verdadeiro ou em falso. Assim, considerando o universo U_{16} e a condição "c: naipe é Copas", o conjunto dos elementos onde a condição dada resulta em verdadeiro é $\{{}^{J}_{\heartsuit}, {}^{Q}_{\heartsuit}, {}^{K}_{\heartsuit}, {}^{A}_{\heartsuit}\}$.

Definição 4.5. Um estágio E_c^p é o conjunto de todos os subconjuntos de p elementos cuja condição c é avaliada como verdadeira. Por exemplo, considerando a condição "c: naipe é Copas", o estágio $E_c^3 = \{\{{}^{Q}_{\heartsuit}, {}^{K}_{\heartsuit}, {}^{A}_{\heartsuit}\}, \{{}^{J}_{\heartsuit}, {}^{K}_{\heartsuit}, {}^{A}_{\heartsuit}\}, \{{}^{J}_{\heartsuit}, {}^{Q}_{\heartsuit}, {}^{A}_{\heartsuit}\}, \{{}^{J}_{\heartsuit}, {}^{Q}_{\heartsuit}, {}^{K}_{\heartsuit}\}\}$.

No presente trabalho só interessa os estágios que de fato selecionam algum elemento, isto é, só será considerado estágio quando há no mínimo um elemento cuja condição é avaliada como verdadeira. Quando $p = 0$ considera-se que $E_c^p = \emptyset$.

Definição 4.6. Dado um estágio E_c^p de um universo U, o conjunto de elementos deste universo onde a condição c é avaliada como verdadeira será chamado de *Domínio de um estágio* e representado pela notação $\tilde{E}$. Já o número de elementos desejado será representado por $p(E)$. Assim, num estágio E_c^p, $p(E) = p$.

Observação 4.1. Deve-se notar que os estágios utilizados devem ser gerados a partir de combinações de elementos e deste modo a cardinalidade de um estágio E_c^p é dado por $\binom{|\tilde{E}|}{p(E)}$. Note que a cardinalidade do estágio acima, definido no universo U_{16} é igual $|E| = \binom{|\tilde{E}|}{p(E)} = \binom{4}{3} = 4$.

Definição 4.7. Dois estágios E_1 e E_2 são iguais, se e somente se, $\tilde{E_1} = \tilde{E_2}$. Um corolário desta definição é que, se E_1 e E_2 são iguais, então $p(E_1) = p(E_2)$ e do mesmo modo se $\tilde{E_1}$ e $\tilde{E_2}$ são iguais, então $p(E_1) = p(E_2)$. Observe portanto, que os estágios $E^3_{\text{naipe é ouros}}$ e $E^3_{\text{naipe não está em \{espadas,copas,paus\}}}$ são iguais.

Definição 4.8. Um estágio E^p_c é chamado de *estágio vazio* quando $p(E) = 0$, isto é quando o estágio não seleciona nenhum elemento. Assim um estágio vazio tem cardinalidade nula, $|E^0_c| = 0$.

Definição 4.9. Um estágio E^p_c é chamado de *estágio completo* quando $p(E) = |\tilde{E}|$, isto é quando o estágio seleciona todos os elementos do domínio $\tilde{E}$. A cardinalidade de um *estágio completo* é $|E| = 1$. Observe que se $0 < k \leq n$ então $\frac{n!}{(n-k)!k!} = k \Leftrightarrow k = n$.

Definição 4.10. Um estágio E^p_c é dito *trivial* quando $p(E) = 0$ ou quando $p(E) = |\tilde{E}|$, isto é, quando o estágio é *vazio* ou o estágio é *completo*. Observe que nestes casos os estágios selecionariam nenhum ou todos os elementos de seus domínios. Assim, um estágio G é *não trivial* quando $0 < p(G) < |\tilde{G}|$. Uma constatação interessante sobre um estágio não trivial é que seu domínio possui no mínimo 2 elementos, portanto, todos os estágios cujos domínios possuem 1 elemento são estágios triviais.

Como os problemas serão decompostos em estágios, sendo que cada estágio define um conjunto parcial de elementos para comporem a solução final, será necessário utilizar produto de estágios.

Definição 4.11. O produto de n estágios será dado por $X_1 \times \ldots \times X_n = \{\{x_1 \cup \ldots \cup x_n\} | x_1 \in \tilde{X_1} \wedge \ldots \wedge x_n \in \tilde{X_n}\}$.

Definição 4.12. Uma *Solução* para um problema do ***iComb*** é um conjunto de estágios $E_1, E_2, \ldots, E_n$ onde:

- $E_1 \times E_2 \times \ldots \times E_n = R$, isto é, o conjunto produto dos estágios é igual ao conjunto resposta R.
- Os conjuntos $\{\tilde{E_1}, \tilde{E_2}, \ldots, \tilde{E_n}\}$ são dois a dois disjuntos, isto é, cada elemento do conjunto universo só pode estar presente em, no máximo, um dos estágios.

Exemplo 4.1. Por exemplo, suponha o seguinte problema: "Num baralho de 16 cartas, contendo somente as figuras[1], quantas mãos de 3 cartas é possível formar com 2 cartas de copas e 1 cartas de paus?"

Observe que o conjunto resposta R é igual a

$$\{\{\tfrac{J}{\heartsuit},\tfrac{Q}{\heartsuit},\tfrac{J}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{J}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{J}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{J}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{J}{\clubsuit}\},\{\tfrac{K}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{J}{\clubsuit}\},$$
$$\{\tfrac{J}{\heartsuit},\tfrac{Q}{\heartsuit},\tfrac{Q}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{Q}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{Q}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{Q}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{Q}{\clubsuit}\},\{\tfrac{K}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{Q}{\clubsuit}\},$$
$$\{\tfrac{J}{\heartsuit},\tfrac{Q}{\heartsuit},\tfrac{K}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{K}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{K}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{K}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{K}{\clubsuit}\},\{\tfrac{K}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{K}{\clubsuit}\},$$
$$\{\tfrac{J}{\heartsuit},\tfrac{Q}{\heartsuit},\tfrac{A}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{A}{\clubsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{A}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{K}{\heartsuit},\tfrac{A}{\clubsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{A}{\clubsuit}\},\{\tfrac{K}{\heartsuit},\tfrac{A}{\heartsuit},\tfrac{A}{\clubsuit}\}\}$$

Agora este problema será analisado decompondo-o em estágios. Para ilustrar a decomposição será utilizada uma em dois estágios: $A = E^2_{\text{naipe é copas}}$ e $B = E^1_{\text{naipe é paus}}$. Assim,

$A = E^2_{\text{naipe é copas}} = \{\{\tfrac{J}{\heartsuit},\tfrac{Q}{\heartsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{K}{\heartsuit}\},\{\tfrac{J}{\heartsuit},\tfrac{A}{\heartsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{K}{\heartsuit}\},\{\tfrac{Q}{\heartsuit},\tfrac{A}{\heartsuit}\},\{\tfrac{K}{\heartsuit},\tfrac{A}{\heartsuit}\}\}$ e

$B = E^1_{\text{naipe é paus}} = \{\{\tfrac{J}{\clubsuit}\},\{\tfrac{Q}{\clubsuit}\},\{\tfrac{K}{\clubsuit}\},\{\tfrac{A}{\clubsuit}\}\}$

É fácil verificar que $A \times B = E^2_{\text{naipe é copas}} \times E^1_{\text{naipe é paus}} = R$ portanto o conjunto $\{A, B\}$ é uma solução para o problema. Para finalmente responder a pergunta, isto é, a quantidade de "mãos", deve-se descobrir a cardinalidade do conjunto R que pode ser expressa por $|R| = |A| \times |B| = 6 \times 4 = 24$.

Também pode-se escrever $|A| = \binom{|\tilde{A}|}{p(A)}$ e $|B| = \binom{|\tilde{B}|}{p(B)}$. Portanto $|R| = \binom{4}{2} \times \binom{4}{1} = 6 \times 4 = 24$.

Definição 4.13. Um problema será chamado de *simples* ou do tipo *simples* quando possui uma *solução* nos termos discutidos na definição 4.12. Em outras palavras, se um problema é *simples* e R é o conjunto resposta, existe um conjunto de estágios $E_1, E_2, \ldots, E_n$ onde:

- $E_1 \times E_2 \times \ldots \times E_n = R$
- Os conjuntos $\{\tilde{E_1}, \tilde{E_2}, \ldots, \tilde{E_n}\}$ são dois a dois disjuntos.

A atual versão do ***iComb*** está restrita a resolução de problemas do tipo *simples*. Como será mostrado posteriormente, o ***iComb*** pode reconher qualquer solução para um problema *simples* além de poder apontar uma proposta de solução incorreta.

[1]Figura de baralho são as cartas Valete, Dama, Rei e Ás.

4.2.1 Filtro e Frequência

Visando formalizar o reconhecimento de diferentes soluções para um mesmo problema, serão utilizados mais dois conceitos complementares, os conceitos de *filtro* e de *extração*.

Definição 4.14. Suponha dois conjuntos A e B, $ftr_B(A)$ é o *filtro* de B por A definido por $\{x \in B | A \subset x\}$. Informalmente, o *filtro* de um conjunto B por A é o subconjunto de B cujos elementos contém os elementos de A.

Suponha

$$R = \{\{\tfrac{J}{\heartsuit}, \tfrac{Q}{\heartsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{K}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{J}{\clubsuit}\},$$
$$\{\tfrac{J}{\heartsuit}, \tfrac{Q}{\heartsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{K}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{Q}{\clubsuit}\},$$
$$\{\tfrac{J}{\heartsuit}, \tfrac{Q}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{K}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\},$$
$$\{\tfrac{J}{\heartsuit}, \tfrac{Q}{\heartsuit}, \tfrac{A}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{A}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{A}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{A}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{A}{\clubsuit}\}, \{\tfrac{K}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{A}{\clubsuit}\}\}$$

Então:

- $ftr_R(\{\tfrac{K}{\clubsuit}\}) = \{\{\tfrac{J}{\heartsuit}, \tfrac{Q}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{J}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{K}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}\}$
- $ftr_R(\{\tfrac{K}{\clubsuit}, \tfrac{A}{\heartsuit}\}) = \{\{\tfrac{J}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{Q}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{K}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}\}$
- $ftr_R(\{\tfrac{K}{\clubsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\heartsuit}\}) = \{\{\tfrac{K}{\heartsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\clubsuit}\}\}$
- $ftr_R(\{\tfrac{K}{\spadesuit}\}) = \emptyset$
- $ftr_R(\{\tfrac{K}{\clubsuit}, \tfrac{A}{\heartsuit}, \tfrac{K}{\heartsuit}, \tfrac{J}{\heartsuit}\}) = \emptyset$
- $ftr_R(\{\}) = R$

Portanto, é fácil ver que

$$|ftr_E(A)| = \binom{|\tilde{E}| - |A|}{p(E) - |A|} = \frac{(|\tilde{E}| - |A|)!}{((|\tilde{E}| - |A|) - (p(E) - |A|))!(p(E) - |A|)!} \tag{4.1}$$

Definição 4.15. Suponha dois conjuntos A e B, $ext_B(A)$ é a *extração* de B por A definido por $\{x \in B | A \not\subset x\}$. A *extração* de um conjunto B por A é o subconjunto de B cujos elementos não contém os elementos de A. Uma *extração* é o complemento de um *filtro* no sentido em que $ftr_B(A) \cup ext_B(A) = B$.

- $ext_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}) = \{\{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}\}$
- $ext_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}\}) = \{\{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}\}$
- $ext_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}\}) = \{\{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\},$
  $\{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}\}$
- $ext_R(\{\begin{smallmatrix}K\\\spadesuit\end{smallmatrix}\}) = R$
- $ext_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\heartsuit\end{smallmatrix}\}) = R$
- $ext_R(\{\}) = \emptyset$

Definição 4.16. Define-se a *frequência* com que um conjunto de elementos A ocorre em B como $F_B(A) = \frac{|ftr_B(A)|}{|B|}$

Então:

- $F_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}) = \frac{1}{4}$
- $F_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}\}) = \frac{1}{8}$
- $F_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}\}) = \frac{1}{24}$
- $F_R(\{\begin{smallmatrix}K\\\spadesuit\end{smallmatrix}\}) = 0$
- $F_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}J\\\heartsuit\end{smallmatrix}\}) = 0$
- $F_R(\{\}) = 1$

Observação 4.2. Seja um universo U e um elemento $e \in U$. Se $E_1 \times E_2 \times \ldots \times E_n = R$ e $e \in \tilde{E_1}$ então a *frequência* $F_{E_1}(\{e\})$ é igual a $F_R(\{e\})$. Em outras palavras, a *frequência* que um elemento ocorre no estágio que o contém, deve ser idêntica a *frequência* que o elemento ocorre no conjunto resposta R.

Exemplo 4.2. Para ilustrar este resultado considere o conjunto R descrito logo acima e os estágios:

- $E^2_{\text{naipe é copas}} = \{\{\tiny\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}\}, \{\begin{smallmatrix}J\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}K\\\heartsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}\}, \{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}, \begin{smallmatrix}A\\\heartsuit\end{smallmatrix}\}\}$

- $E^1_{\text{naipe é paus}} = \{\{\begin{smallmatrix}J\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}Q\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}, \{\begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}\}$

Note que $F_{E^2_{\text{naipe é copas}}}(\{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}\}) = F_R(\{\begin{smallmatrix}K\\\heartsuit\end{smallmatrix}\}) = \frac{1}{2}$ e $F_{E^1_{\text{naipe é paus}}}(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}) = F_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}) = \frac{1}{4}$.

Assim, se os estágios forem definidos corretamente, a *frequência* de ocorrência de um elemento em seu estágio é exatamente igual a *frequência* que o mesmo ocorre no conjunto resposta. Ainda mais, supondo um estágio E, tal que $1 \leq p(E) \leq |\tilde{E}|$, é possível calcular a *frequência* do elemento $e \in \tilde{E}$.

Teorema 4.1. A *frequência* com que um elemento $e \in \tilde{E}$ ocorre no conjunto resposta R é igual a $F_R(\{e\}) = \frac{p(E)}{|\tilde{E}|}$.

Demonstração. Como foi visto na observação 4.2, a *frequência* que um elemento ocorre no estágio que o contém, deve ser idêntica a *frequência* que o elemento ocorre no conjunto resposta R, portanto $F_R(\{e\}) = F_E(\{e\})$. Utilizando a definição 4.16 e a identidade 4.1, segue que

$$
\begin{aligned}
F_R(\{e\}) = F_E(\{e\}) = \frac{|ftr_E(\{e\})|}{|E|} &= \frac{\binom{|\tilde{E}|-1}{p(E)-1}}{\binom{|\tilde{E}|}{p(E)}} \\
&= \frac{\frac{(|\tilde{E}|-1)!}{\cancel{(|\tilde{E}|-1-p(E)+1)!}(p(E)-1)!}}{\frac{(|\tilde{E}|)!}{\cancel{(|\tilde{E}|-p(E))!}p(E)!}} \\
&= \frac{\frac{\cancel{(|\tilde{E}|-1)!}}{\cancel{(p(E)-1)!}}}{\frac{(|\tilde{E}|)\cancel{(|\tilde{E}|-1)!}}{p(E)\cancel{(p(E)-1)!}}} \\
&= \frac{p(E)}{|\tilde{E}|}
\end{aligned}
\tag{4.2}
$$

□

Teorema 4.2. A *frequência* com que dois elementos pertencentes ao mesmo estágio $e \in \tilde{E}$ e $f \in \tilde{E}$ ocorram simultaneamente é dado por $F_R(\{e, f\}) = \frac{p(E)(p(E)-1)}{|\tilde{E}|(|\tilde{E}|-1)}$.

Demonstração.

$$
\begin{aligned}
F_R(\{e, f\}) = F_E(\{e, f\}) = \frac{|ftr_E(\{e, f\})|}{|E|} &= \frac{\binom{|\tilde{E}|-2}{p(E)-2}}{\binom{|\tilde{E}|}{p(E)}} \\
&= \frac{\frac{(|\tilde{E}|-2)!}{\cancel{(|\tilde{E}|-2-p(E)+2)!}(p(E)-2)!}}{\frac{(|\tilde{E}|)!}{\cancel{(|\tilde{E}|-p(E))!}p(E)!}} \\
&= \frac{p(E)(p(E)-1)}{|\tilde{E}|(|\tilde{E}|-1)}
\end{aligned}
\tag{4.3}
$$

□

Teorema 4.3. A *frequência* com que dois elementos pertencentes a estágios distintos $e_1 \in \tilde{E_1}$ e $e_2 \in \tilde{E_2}$ ocorrem simultaneamente é dado por $F_R(\{e_1, e_2\}) = \frac{p(E_1)}{|\tilde{E_1}|} \times \frac{p(E_2)}{|\tilde{E_2}|}$

Demonstração. $F_R(\{e_1, e_2\}) = F_{E_1}(\{e_1\}) \times F_{E_2}(\{e_2\}) = \frac{p(E_1)}{|\tilde{E_1}|} \times \frac{p(E_2)}{|\tilde{E_2}|}$ □

O mesmo raciocínio pode ser facilmente estendido para o cálculo da *frequência* com mais de dois elementos. Entretanto, para os propósitos do desenvolvimento a seguir esse exercício não será necessário.

Exemplo 4.3. *Em um baralho de 32 cartas, quantas mãos de 5 cartas podemos formar com exatamente 3 Ases e 3 cartas de Paus?*

Uma solução para o problema é o conjunto de estágios $\{A, B, C\}$ definidos abaixo:

$$A = E^1_{\text{valor é ás e naipe é paus}} = \{\{\tfrac{A}{\clubsuit}\}\}$$

$$B = E^2_{\text{valor é ás e naipe não é paus}} = \{\{\tfrac{A}{\diamondsuit}, \tfrac{A}{\spadesuit}\}, \{\tfrac{A}{\diamondsuit}, \tfrac{A}{\heartsuit}\}, \{\tfrac{A}{\spadesuit}, \tfrac{A}{\heartsuit}\}\}$$

$$C = E^2_{\text{valor não é ás e naipe é paus}} = \{\{\tfrac{7}{\clubsuit}, \tfrac{8}{\clubsuit}\}, \{\tfrac{7}{\clubsuit}, \tfrac{9}{\clubsuit}\}, \{\tfrac{7}{\clubsuit}, \tfrac{10}{\clubsuit}\}, \{\tfrac{7}{\clubsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{7}{\clubsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{7}{\clubsuit}, \tfrac{K}{\clubsuit}\},$$
$$\{\tfrac{8}{\clubsuit}, \tfrac{9}{\clubsuit}\}, \{\tfrac{8}{\clubsuit}, \tfrac{10}{\clubsuit}\}, \{\tfrac{8}{\clubsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{8}{\clubsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{8}{\clubsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{9}{\clubsuit}, \tfrac{10}{\clubsuit}\},$$
$$\{\tfrac{9}{\clubsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{9}{\clubsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{9}{\clubsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{10}{\clubsuit}, \tfrac{J}{\clubsuit}\}, \{\tfrac{10}{\clubsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{10}{\clubsuit}, \tfrac{K}{\clubsuit}\},$$
$$\{\tfrac{J}{\clubsuit}, \tfrac{Q}{\clubsuit}\}, \{\tfrac{J}{\clubsuit}, \tfrac{K}{\clubsuit}\}, \{\tfrac{Q}{\clubsuit}, \tfrac{K}{\clubsuit}\}\}$$

Como foi visto anteriormente, a resposta para o problema é $|A| \times |B| \times |C| = 1 \times 3 \times 21 = 63$. A tabela 4.1 apresenta todos os 63 elementos do conjunto resposta para o problema enquanto a tabela 4.2 mostra de forma concisa a *frequência* com que cada elemento e cada dupla de elementos aparece no conjunto resposta.

A tabela de *frequência* contém apenas os elementos do universo que aparecem em alguma configuração da resposta, isto é que possuem *frequências* maior do que zero. A primeira linha contém as *frequências* individuais de cada elemento enquanto que a região central apresenta as *frequências* com que dois elementos ocorrem simultaneamente. Por exemplo, consultando a tabela de *frequência* é fácil observar que os elementos $\frac{A}{\diamondsuit}$ e $\frac{7}{\clubsuit}$ estão ambos presentes em $\frac{4}{21}$ das configurações válidas para o exercício. Enfim, estas duas cartas presentes em exatamente 12 configurações das 63 apresentadas na tabela 4.1.

4.3 Estrutura da Solução

Na perspectiva de um corretor automático não basta conhecer uma solução de um problema. Na realidade será necessário conhecer todas as soluções possíveis. Só desta maneira será possível munir o sistema de um bom algoritmo de correção automática. Naturalmente, se um problema possui mais do que uma solução e o sistema só conhece uma, como poderia verificar que uma solução alternativa está errada?

Observando o exemplo 4.2 da página 48, não há nada que indique que o conjunto de estágios $\{E^2_{\text{naipe é copas}}, E^1_{\text{naipe é paus}}\}$ é a única solução para o problema. De fato, vários dos problemas do ***iComb*** e do ***Combien?*** possuem mais do que uma solução. Desta

A ♣	A ♢	A ♠	7 ♣	8 ♣	A ♣	A ♢	A ♡	7 ♣	8 ♣	A ♣	A ♠	A ♡	7 ♣	8 ♣
A ♣	A ♢	A ♠	7 ♣	9 ♣	A ♣	A ♢	A ♡	7 ♣	9 ♣	A ♣	A ♠	A ♡	7 ♣	9 ♣
A ♣	A ♢	A ♠	7 ♣	10 ♣	A ♣	A ♢	A ♡	7 ♣	10 ♣	A ♣	A ♠	A ♡	7 ♣	10 ♣
A ♣	A ♢	A ♠	7 ♣	J ♣	A ♣	A ♢	A ♡	7 ♣	J ♣	A ♣	A ♠	A ♡	7 ♣	J ♣
A ♣	A ♢	A ♠	7 ♣	Q ♣	A ♣	A ♢	A ♡	7 ♣	Q ♣	A ♣	A ♠	A ♡	7 ♣	Q ♣
A ♣	A ♢	A ♠	7 ♣	K ♣	A ♣	A ♢	A ♡	7 ♣	K ♣	A ♣	A ♠	A ♡	7 ♣	K ♣
A ♣	A ♢	A ♠	8 ♣	9 ♣	A ♣	A ♢	A ♡	8 ♣	9 ♣	A ♣	A ♠	A ♡	8 ♣	9 ♣
A ♣	A ♢	A ♠	8 ♣	10 ♣	A ♣	A ♢	A ♡	8 ♣	10 ♣	A ♣	A ♠	A ♡	8 ♣	10 ♣
A ♣	A ♢	A ♠	8 ♣	J ♣	A ♣	A ♢	A ♡	8 ♣	J ♣	A ♣	A ♠	A ♡	8 ♣	J ♣
A ♣	A ♢	A ♠	8 ♣	Q ♣	A ♣	A ♢	A ♡	8 ♣	Q ♣	A ♣	A ♠	A ♡	8 ♣	Q ♣
A ♣	A ♢	A ♠	8 ♣	K ♣	A ♣	A ♢	A ♡	8 ♣	K ♣	A ♣	A ♠	A ♡	8 ♣	K ♣
A ♣	A ♢	A ♠	9 ♣	10 ♣	A ♣	A ♢	A ♡	9 ♣	10 ♣	A ♣	A ♠	A ♡	9 ♣	10 ♣
A ♣	A ♢	A ♠	9 ♣	J ♣	A ♣	A ♢	A ♡	9 ♣	J ♣	A ♣	A ♠	A ♡	9 ♣	J ♣
A ♣	A ♢	A ♠	9 ♣	Q ♣	A ♣	A ♢	A ♡	9 ♣	Q ♣	A ♣	A ♠	A ♡	9 ♣	Q ♣
A ♣	A ♢	A ♠	9 ♣	K ♣	A ♣	A ♢	A ♡	9 ♣	K ♣	A ♣	A ♠	A ♡	9 ♣	K ♣
A ♣	A ♢	A ♠	10 ♣	J ♣	A ♣	A ♢	A ♡	10 ♣	J ♣	A ♣	A ♠	A ♡	10 ♣	J ♣
A ♣	A ♢	A ♠	10 ♣	Q ♣	A ♣	A ♢	A ♡	10 ♣	Q ♣	A ♣	A ♠	A ♡	10 ♣	Q ♣
A ♣	A ♢	A ♠	10 ♣	K ♣	A ♣	A ♢	A ♡	10 ♣	K ♣	A ♣	A ♠	A ♡	10 ♣	K ♣
A ♣	A ♢	A ♠	J ♣	Q ♣	A ♣	A ♢	A ♡	J ♣	Q ♣	A ♣	A ♠	A ♡	J ♣	Q ♣
A ♣	A ♢	A ♠	J ♣	K ♣	A ♣	A ♢	A ♡	J ♣	K ♣	A ♣	A ♠	A ♡	J ♣	K ♣
A ♣	A ♢	A ♠	Q ♣	K ♣	A ♣	A ♢	A ♡	Q ♣	K ♣	A ♣	A ♠	A ♡	Q ♣	K ♣

Tabela 4.1: Conjunto Resposta para o problema do exemplo 4.3.

$\frac{2}{7}$	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{1}{1}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	
7 ♣	8 ♣	9 ♣	10 ♣	J ♣	Q ♣	K ♣	A ♣	A ♢	A ♠	A ♡	
	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{4}{21}$	7 ♣
		$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{4}{21}$	8 ♣
			$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{4}{21}$	9 ♣
				$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{4}{21}$	10 ♣
					$\frac{1}{21}$	$\frac{1}{21}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{4}{21}$	J ♣
						$\frac{1}{21}$	$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{4}{21}$	Q ♣
							$\frac{2}{7}$	$\frac{4}{21}$	$\frac{4}{21}$	$\frac{4}{21}$	K ♣
								$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	A ♣
									$\frac{1}{3}$	$\frac{1}{3}$	A ♢
										$\frac{1}{3}$	A ♠
											A ♡

Tabela 4.2: Tabela de Frequências para o conjunto Resposta do problema do exemplo 4.3.

forma, será necessário compreender se há uma estrutura geral que descreve a solução para um problema do ***iComb*** de modo a poder usá-la na forma de um algoritmo de detecção automática de erro.

Por exemplo, considere o problema: "Num baralho de 16 cartas, contendo somente as figuras, quantas mãos de 5 cartas é possível formar com 4 ases e duas cartas de espadas?"

O conjunto resposta para este problema é

$R = \{\{{}^{A}_{\diamond}, {}^{A}_{\spadesuit}, {}^{A}_{\heartsuit}, {}^{A}_{\clubsuit}, {}^{J}_{\spadesuit}\}, \{{}^{A}_{\diamond}, {}^{A}_{\spadesuit}, {}^{A}_{\heartsuit}, {}^{A}_{\clubsuit}, {}^{Q}_{\spadesuit}\}, \{{}^{A}_{\diamond}, {}^{A}_{\spadesuit}, {}^{A}_{\heartsuit}, {}^{A}_{\clubsuit}, {}^{K}_{\spadesuit}\}\}$

Existe mais de uma decomposição em estágios para este problema, duas destas decomposições são:

- $\{E^4_{\text{valor é ás}}, E^1_{\text{naipe é espadas e valor não é ás}}\}$
- $\{E^2_{\text{valor é ás e naipe é preto}}, E^2_{\text{valor é ás e naipe é vermelho}}, E^1_{\text{naipe é espadas e valor não é ás}}\}$.

O conceito de *frequência* será útil para discutir em que condições um problema terá mais do que uma solução. Dado um conjunto resposta, quanto maior for o número de soluções, maior será o custo computacional do mecanismo de detecção automática de erros. Por outro lado, se este número é reduzido, a criação do mecanismo será na mesma medida mais simples.

Teorema 4.4. Seja $S = \{S_1, S_2, \ldots, S_n\}$ e $Z = \{Z_1, Z_2, \ldots, Z_m\}$ duas soluções para um problema P de conjunto resposta R num universo U. Suponha um estágio $S_i, 0 \leq i \leq n$ tal que $0 < p(S_i) < |\tilde{S}_i|$. Se dois elementos $a, b \in \tilde{S}_i$ então $a, b \in \tilde{Z}_j, 0 \leq j \leq m$.

Em outras palavras, deseja-se provar que se dois elementos a, b pertencem a um estágio não trivial da solução S, os dois elementos deverão pertencer a um mesmo estágio na solução Z. Lembrando que estágio trivial é um estágio que seleciona todos ou nenhum elemento de seu domínio.

Demonstração. Sabe-se que a, b pertencem ao mesmo estágio S_i portanto:

$$F_R(\{a\}) = F_R(\{b\}) = \frac{p(S_i)}{|\tilde{S}_i|} \tag{4.4}$$

$$F_R(\{a,b\}) = \frac{p(S_i)(p(S_i)-1)}{|\tilde{S}_i|(|\tilde{S}_i|-1)} \tag{4.5}$$

Observe que $0 < p(S_i) < |\tilde{S}_i|$ implica $|\tilde{S}_i| \geq 2$, portanto a expressão 4.5 é válida pois não há possibilidade de divisão por zero.

Suponha, por contradição, que $a, b \in S_i$ pertencem a estágios distintos em Z, por exemplo $a \in Z_f$ e $b \in Z_g$. Quando dois elementos pertencem a estágios distintos, sua *frequência* em conjunto é dada pela expressão:

$$F_R(\{a,b\}) = F_{Z_g}(\{a\}) \times F_{Z_f}(\{b\}) = F_R(\{a\}) \times F_R(\{b\}) = \frac{p(S_i)^2}{|\tilde{S}_i|^2} \tag{4.6}$$

Igualando as equações (4.5) e (4.6) tem-se:

$$\frac{p(S_i)(p(S_i)-1)}{|\tilde{S}_i|(|\tilde{S}_i|-1)} = \frac{p(S_i)^2}{|\tilde{S}_i|^2} \tag{4.7}$$

Se $p(S_i) = 0$, a igualdade acima será verdadeira. Se $p(S_i) \neq 0$, temos:

$$\frac{p(S_i)-1}{|\tilde{S}_i|-1} = \frac{p(S_i)}{|\tilde{S}_i|}$$

$$|\tilde{S}_i|(p(S_i)-1) = p(S_i)(|\tilde{S}_i|-1)$$

$$|\tilde{S}_i|p(S_i) - |\tilde{S}_i| = p(S_i)|\tilde{S}_i| - p(S_i) \Rightarrow |\tilde{S}_i| = p(S_i) \tag{4.8}$$

O que é absurdo já que por hipótese $0 < p(S_i) < |\tilde{S}_i|$.

□

Teorema 4.5. Seja $S = \{S_1, S_2, \ldots, S_n\}$ e $Z = \{Z_1, Z_2, \ldots, Z_m\}$ duas soluções para um problema P de conjunto resposta R num universo U. Suponha dois estágios $S_i, 0 \leq i \leq n$ e $S_j, 0 \leq j \leq n$ tal que $0 < p(S_i) < |\tilde{S}_i|$ e $0 < p(S_j) < |\tilde{S}_j|$. Se dois elementos a, b pertencem a estágios distintos em S, também pertencerão a estágios distintos em Z.

Demonstração. Novamente será utilizada uma prova por contradição. Seja $a \in S_i$ e $b \in S_j$. Como a e b pertencem a estágios distintos então

$$F_R(\{a,b\}) = F_{S_i}(\{a\}) \times F_{S_j}(\{b\}) = F_R(\{a\}) \times F_R(\{b\}) \tag{4.9}$$

Suponha, por contradição, que a e b pertençam ao mesmo estágio Z_f de Z. Assim sabe-se que:

$$F_R(\{a\}) = F_R(\{b\}) = \frac{p(Z_f)}{|\tilde{Z}_f|} \tag{4.10}$$

$$F_R(\{a,b\}) = \frac{p(Z_f)(p(Z_f)-1)}{|\tilde{Z}_f|(|\tilde{Z}_f|-1)} \tag{4.11}$$

Resultado análogo ao teorema anterior. Finalmente, substituindo 4.10 em 4.9 e igualando 4.11 e 4.9, tem-se

$$\frac{p(Z_f)(p(Z_f)-1)}{|\tilde{Z}_f|(|\tilde{Z}_f|-1)} = \frac{p(Z_f)^2}{|\tilde{Z}_f|^2} \tag{4.12}$$

Como anteriormente, analisa-se os dois casos: $p(Z_f) = 0$ ou quando $p(Z_f) = |\tilde{Z}_f|$.

Se $p(Z_f) = 0$

Como o elemento a pertence tanto a S_i quanto a Z_f, sua *frequência* pode ser expressa de duas maneiras diferentes resultando na igualdade $\frac{p(S_i)}{|\tilde{S}_i|} = \frac{p(Z_p)}{|\tilde{Z}_f|}$. Se $p(Z_f) = 0$ então $p(S_i) = 0$, o que contraria a hipótese.

Se $p(Z_f) = |\tilde{Z}_f|$

Da mesma maneira, se $p(Z_f) = |\tilde{Z}_f|$, então $\frac{p(Z_f)}{|\tilde{Z}_f|} = 1$ e $p(S_i) = |\tilde{S}_i|$ o que contraria a hipótese novamente, já que $0 < p(S_i) < |\tilde{S}_i|$. Enfim, nos dois casos a hipótese foi negada, resultando num absurdo.

□

Os dois resultados acima mostram que se dois elementos estão presentes em um estágio não trivial de uma solução, então devem pertencer a um mesmo estágio numa solução alternativa. Analogamente, se dois elementos estão presentes em estágios não triviais distintos de uma solução, deverão estar presentes em estágios distintos de uma solução alternativa.

Em qualquer um dos casos, uma alteração neste padrão ocasionaria alteração das *frequências* dos elementos, seja a *frequência* individual do elemento, seja a *frequência* conjunta considerando os elementos dois a dois.

Teorema 4.6. Sejam S e Z duas soluções para um problema P de conjunto resposta R num universo U. Se um elemento a pertence a um estágio não trivial de S, então a pertence a um estágio não trivial de Z.

Demonstração. Considere a pertencente ao domínio de um estágio S_i não trivial em S. Como S_i é não trivial sabe-se que $0 < p(S_i) < |\tilde{S}_i|$. A frequência de a em R é dado por $\frac{p(S_i)}{|\tilde{S}_i|}$. Dividindo a expressão $0 < p(S_i) < |\tilde{S}_i|$ por $|\tilde{S}_i|$ temos $0 < F_R(\{a\}) < 1$.

Suponha, por contradição que a pertença a um estágio trivial de Z. Ora se o estágio é trivial então a frequência de a em R deverá ser igual a zero ou igual a 1. Absurdo!

□

Teorema 4.7. Seja $S = \{S_1, S_2, \ldots, S_n\}$ e $Z = \{Z_1, Z_2, \ldots, Z_m\}$ duas soluções para um problema P de conjunto resposta R num universo U. Suponha um estágio $S_i, 0 \leq i \leq n$ tal que $0 = p(S_i) \leq |\tilde{S}_i|$. Se um elemento $a \in \tilde{S}_i$ e $a \in \tilde{Z}_j, 0 \leq j \leq m$ então $p(Z_j) = 0$

Demonstração. Se a pertence tanto a S_i quanto a Z_j então $F_R(\{a\}) = F_{S_i}(\{a\}) = F_{Z_j}(\{a\}) = \frac{p(S_i)}{|\tilde{S}_i|} = \frac{p(Z_j)}{|\tilde{Z}_j|} = 0$, pois $p(S_i) = 0$. Então $p(Z_j) = 0$.

□

Teorema 4.8. Seja $S = \{S_1, S_2, \ldots, S_n\}$ e $Z = \{Z_1, Z_2, \ldots, Z_m\}$ duas soluções para um problema P de conjunto resposta R num universo U. Suponha um estágio $S_i, 0 \leq i \leq n$ tal que $0 \leq p(S_i) = |\tilde{S}_i|$. Se um elemento $a \in \tilde{S}_i$ e $a \in \tilde{Z}_j, 0 \leq j \leq m$ então $p(Z_j) = |\tilde{Z}_j|$

Demonstração. Se a pertence tanto a S_i quanto a Z_j então $F_R(\{a\}) = F_{S_i}(\{a\}) = F_{Z_j}(\{a\}) = \frac{p(S_i)}{|\tilde{S}_i|} = \frac{p(Z_j)}{|\tilde{Z}_j|} = 1$, pois $p(S_i) = |\tilde{S}_i|$. Então $p(Z_j) = |\tilde{Z}_j|$.

□

Os três teoremas acima mostram que se um elemento pertence a um estágio trivial de uma solução, deverá pertencer a um estágio trivial de uma solução alternativa. Além disso mostram que se um elemento pertence a um estágio não trivial, ele deverá pertencer a um estágio não trivial de uma solução alternativa.

Teorema 4.9. Seja $S = \{S_1, S_2, \ldots, S_n\}$ e $Z = \{Z_1, Z_2, \ldots, Z_m\}$ duas soluções para um problema P de conjunto resposta R num universo U. Se S_i e Z_j são estágios não triviais e existe um elemento a tal que $a \in S_i$ e $a \in Z_j$ então $S_i = Z_j$.

Demonstração. Primeiramente mostra-se que $\tilde{S}_i = \tilde{Z}_j$.

Suponha, por contradição, que existe um elemento $e \neq a$ em $\tilde{S}_i$ tal que $e \notin \tilde{Z}_j$. Ora, mas isso é absurdo pois pelo teorema 4.4 dois elementos que pertençam a um mesmo estágio não trivial de uma solução, devem pertencer a um mesmo estágio na solução alternativa. Observe que tanto a quanto e pertencem a $\tilde{S}_i$ mas somente a pertence a $\tilde{Z}_j$, então conclui-se que $\tilde{S}_i \subseteq \tilde{Z}_j$

Pelo mesmo raciocínio conclui-se que $\tilde{Z}_j \subseteq \tilde{S}_i$. Ora, se $\tilde{S}_i \subseteq \tilde{Z}_j$ e $\tilde{Z}_j \subseteq \tilde{S}_i$ então $\tilde{Z}_j = \tilde{S}_i$

Observe que $F_R(\{a\}) = F_{S_i}(\{a\}) = F_{Z_j}(\{a\}) = \frac{p(S_i)}{|\tilde{S}_i|} = \frac{p(Z_j)}{|\tilde{Z}_j|}$

Como $|\tilde{Z}_j| = |\tilde{S}_i|$ então $p(S_i) = p(Z_j)$

Enfim, como $\tilde{Z}_j = \tilde{S}_i$ e $p(S_i) = p(Z_j)$ então $S_i = Z_j$

□

Teorema 4.10. Seja $S = \{S_1, S_2, \ldots, S_n\}$ e $Z = \{Z_1, Z_2, \ldots, Z_m\}$ duas soluções para um problema P de conjunto resposta R num universo U. Se S_i é um estágio não trivial de S então existe $Z_j \in Z$ não trivial tal que $S_i = Z_j$

Demonstração. Seja $e \in \tilde{S}_i$. Pelo teorema 4.6 existe um Z_j não trivial tal que $e \in Z_j$. Então, pelo teorema 4.9, $S_i = Z_j$ □

Teorema 4.11. Seja S e Z duas soluções para um problema P de conjunto resposta R num universo U. Seja $S_{NT} = \{S_1, S_2, \ldots, S_n\}$ o conjunto dos estágios não triviais de S e seja $Z_{NT} = \{Z_1, Z_2, \ldots, Z_m\}$ o conjunto dos estágios não triviais de Z. Então $S_{NT} = Z_{NT}$.

Demonstração. Pelo teorema 4.10 sabe-se que para cada elemento em S_{NT} existe um estágio não trivial em Z. Portanto $S_{NT} \subseteq Z_{NT}$. Por raciocínio análogo conclui-se que $Z_{NT} \subseteq S_{NT}$. Portanto $S_{NT} = Z_{NT}$. □

Reescrevendo o resultado acima de maneira mais geral, se S e Z são conjuntos de estágios solução para o mesmo problema P, então.

$$\{x \in S : 0 < p(x) < |\tilde{x}|\} = \{x \in Z : 0 < p(x) < |\tilde{x}|\} \tag{4.13}$$

Para completar a análise necessita-se discutir os casos restantes, isto é, quando $p(E) = 0$, quando $p(E) = |\tilde{E}|$ e quando $0 < |\tilde{E}| < 2$ pois no desenvolvimento acima supõe-se que $\tilde{E}$ tem no mínimo 2 elementos. Começando do final, é simples observar que se $0 < |\tilde{E}| < 2$, então $p(E) = 0$ ou $p(E) = |\tilde{E}|$ de maneira que não vai ser necessário comentar este último caso em separado.

Observe que se $p(E) = 0$, a *frequência* dos elementos de E será necessariamente igual a 0, isto é, nunca serão selecionados para o conjunto resposta. Por outro lado, se $p(E) = |\tilde{E}|$ a *frequência* dos elementos de E será sempre igual a 1, ou seja, estes elementos estarão presente em todos as respostas.

Nosso objetivo é preservar as *frequências* dos elementos. Note que se $p(E) = 0$ todo elemento de $\tilde{E}$ poderá pertencer a qualquer outro estágio G sem alterar sua *frequência* se $p(G)$ for igualmente igual a 0. Da mesma forma, se $p(E) = |\tilde{E}|$, todo elemento de $\tilde{E}$ poderá pertencer a qualquer outro estágio G sem alterar sua *frequência* se $p(G) = |\tilde{G}|$.

Em outras palavras se S_1 e S_2 são dois conjuntos de estágios solução para o mesmo problema P, então:

$$\bigcup_{\substack{E_i \in S_1 \\ p(E_i)=0}} \tilde{E_i} = \bigcup_{\substack{G_i \in S_2 \\ p(G_i)=0}} \tilde{G_i} \tag{4.14}$$

$$\bigcup_{\substack{E_i \in S_1 \\ p(E_i)=|\tilde{E_i}|}} \tilde{E_i} = \bigcup_{\substack{G_i \in S_2 \\ p(G_i)=|\tilde{G_i}|}} \tilde{G_i} \tag{4.15}$$

Finalmente, considerando os resultados (4.13), (4.14) e (4.15) pode-se enunciar uma forma geral para verificar se um conjunto de estágios é uma solução para um problema P.

Teorema 4.12. Seja S um conjunto de estágios solução para um problema P. Um conjunto de estágios Z será solução para o problema P se, e somente se, as três igualdades abaixo forem verdadeiras:

$$\{x \in S : 0 < p(x) < |\tilde{x}|\} = \{x \in Z : 0 < p(x) < |\tilde{x}|\} \tag{4.16}$$

$$\bigcup_{\substack{x\in S\\ p(x)=0}} \tilde{x} = \bigcup_{\substack{x\in Z\\ p(x)=0}} \tilde{x} \tag{4.17}$$

$$\bigcup_{\substack{x\in S\\ p(x)=|\tilde{x}|}} \tilde{x} = \bigcup_{\substack{x\in Z\\ p(x)=|\tilde{x}|}} \tilde{x} \tag{4.18}$$

O exemplo abaixo ilustra a aplicação do teorema a um problema específico. A partir de uma solução dada será discutido se outras duas soluções candidatas são de fato soluções, ou não, para o problema. Como será visto, a primeira solução candidata é uma solução alternativa ao problema enquanto a outra não é.

Exemplo 4.4. *Com um baralho de 32 cartas, quantas mãos de 20 cartas é possível formar com 3 Ases, 4 cartas de Espadas e 16 cartas vermelhas?*

Como o universo é U_{32} e que uma solução para o problema é:

$S = \{E^{16}_{\text{naipe é vermelho}}, E^{1}_{\text{naipe é espadas e valor é ás}}, E^{3}_{\text{naipe é espadas e valor não é ás}}\}$

Assim, o conjunto resposta R para o problema é igual a:

$R = E^{16}_{\text{naipe é vermelho}} \times E^{1}_{\text{naipe é espadas e valor é ás}} \times E^{3}_{\text{naipe é espadas e valor não é ás}}$

Para simplificar a notação, será considerado $E_A = E^{16}_{\text{naipe é vermelho}}$, $E_B = E^{1}_{\text{naipe é espadas e valor é ás}}$ e $E_C = E^{3}_{\text{naipe é espadas e valor não é ás}}$. Observe também que:

$p(E_A) = 16$ e $|\tilde{E_A}| = 16$.

$p(E_B) = 1$ e $|\tilde{E_B}| = 1$.

$p(E_C) = 3$ e $|\tilde{E_A}| = 7$.

Suponha duas soluções candidatas C_1 e C_2 tal que:

$C_1 = \{E^{8}_{\text{naipe é copas}}, E^{8}_{\text{naipe é ouros}}, E^{1}_{\text{naipe é espadas e valor é ás}}, E^{3}_{\text{naipe é espadas e valor não é ás}}\}$

$C_2 = \{E^{16}_{\text{naipe é vermelho}}, E^{4}_{\text{naipe é espadas}}\}$

A pergunta que deve ser respondida é: C_1 e C_2 são soluções para o problema P dado?

Para responder a pergunta, pode-se utilizar o resultado obtido no teorema 4.12. Sabe-se que o conjunto $S = \{E_A, E_B, E_C\}$ é uma solução para o problema. Agora será verificado se os conjuntos C_1 e C_2 são também soluções, para tanto será necessário verificar cada uma das equações 4.16, 4.17 e 4.18.

Observe que:

- $\{x \in S : 0 < p(x) < |\tilde{x}|\} = \{E_C\} = \{E^3_{\text{naipe é espadas e valor não é ás}}\}$
- $\bigcup_{\substack{x \in S \\ p(x)=0}} \tilde{x} = \emptyset$
- $\bigcup_{\substack{x \in S \\ p(x)=|\tilde{x}|}} \tilde{x} = E_A \cup E_B = \{\overset{7}{\heartsuit}, \overset{8}{\heartsuit}, \overset{9}{\heartsuit}, \overset{10}{\heartsuit}, \overset{J}{\heartsuit}, \overset{Q}{\heartsuit}, \overset{K}{\heartsuit}, \overset{A}{\heartsuit}, \overset{7}{\diamondsuit}, \overset{8}{\diamondsuit}, \overset{9}{\diamondsuit}, \overset{10}{\diamondsuit}, \overset{J}{\diamondsuit}, \overset{Q}{\diamondsuit}, \overset{K}{\diamondsuit}, \overset{A}{\diamondsuit}, \overset{A}{\spadesuit}\}$

Realizando as mesmas operações para C_1 temos:

- $\{x \in C_1 : 0 < p(x) < |\tilde{x}|\} = \{E^3_{\text{naipe é espadas e valor não é ás}}\}$
- $\bigcup_{\substack{x \in C_1 \\ p(x)=0}} \tilde{x} = \emptyset$
- $\bigcup_{\substack{x \in C_1 \\ p(x)=|\tilde{x}|}} \tilde{x} = \{\overset{7}{\heartsuit}, \overset{8}{\heartsuit}, \overset{9}{\heartsuit}, \overset{10}{\heartsuit}, \overset{J}{\heartsuit}, \overset{Q}{\heartsuit}, \overset{K}{\heartsuit}, \overset{A}{\heartsuit}, \overset{7}{\diamondsuit}, \overset{8}{\diamondsuit}, \overset{9}{\diamondsuit}, \overset{10}{\diamondsuit}, \overset{J}{\diamondsuit}, \overset{Q}{\diamondsuit}, \overset{K}{\diamondsuit}, \overset{A}{\diamondsuit}, \overset{A}{\spadesuit}\}$

Como as três igualdades 4.16, 4.17 e 4.18 são verificadas, conclui-se que C_1 é solução para o problema. Analisando agora C_2:

- $\{x \in C_2 : 0 < p(x) < |\tilde{x}|\} = \{E^4_{\text{naipe é espadas}}\}$
- $\bigcup_{\substack{x \in C_2 \\ p(x)=0}} \tilde{x} = \emptyset$
- $\bigcup_{\substack{x \in C_2 \\ p(x)=|\tilde{x}|}} \tilde{x} = \{\overset{7}{\heartsuit}, \overset{8}{\heartsuit}, \overset{9}{\heartsuit}, \overset{10}{\heartsuit}, \overset{J}{\heartsuit}, \overset{Q}{\heartsuit}, \overset{K}{\heartsuit}, \overset{A}{\heartsuit}, \overset{7}{\diamondsuit}, \overset{8}{\diamondsuit}, \overset{9}{\diamondsuit}, \overset{10}{\diamondsuit}, \overset{J}{\diamondsuit}, \overset{Q}{\diamondsuit}, \overset{K}{\diamondsuit}, \overset{A}{\diamondsuit}\}$

Observe que a igualdade falha para a primeira e terceira equações de maneira que pode-se concluir que C_2 não é solução para o problema.

Portanto, como foi visto, C_1 é uma solução igualmente válida para o problema. Note que os dois primeiros estágios de C_1 selecionam todas as cartas vermelhas, do mesmo modo que o estágio E_A da solução S. Estes dois estágios de C_1 tem a propriedade $p(E) = |\tilde{E}|$. Como E_A também possui a mesma propriedade, a equação 4.17 é válida. Os outros estágios são idênticos, deste modo pode-se encerrar a análise neste ponto.

Já C_2 não é solução para o problema P uma vez que a equação 4.16 não é verificada. A equação 4.16 exige que todos os estágios onde $0 < p(E) < |\tilde{E}|$ de uma solução sejam

idênticos aos estágios que possuem a mesma popriedade na outra solução, o que não é verdade uma vez que $E^3_{\text{naipe é espadas e valor não é ás}} \neq E^4_{\text{naipe é espadas}}$. De fato, observe que a solução candidata C_2 permite que conjuntos que não contém Ás de Espadas pertençam à resposta do problema.

Outra maneira de verificar a validade de uma solução é montar uma tabela de frequência para a solução candidata. A tabela 4.3 apresenta a tabela de frequência do exercício para S e portanto, para C_1. Entretanto, se o leitor montar a correspondente tabela relativa a C_2, verá que resulta uma tabela completamente diferente.

4.4 Algoritmo

O teorema 4.12 define as condições em que uma solução candidata será considerada uma solução de fato para um problema do ***iComb***. Pode-se expressar o teorema da seguinte maneira. Seja S uma solução para um problema, a solução candidata Z também será solução se, e somente se:

- Todos os estágios não triviais de Z forem idênticos aos estágios não triviais de S
- Os conjunto dos elementos que pertencem aos domínios de *estágios vazios* são idênticos em S e em Z
- Os conjunto dos elementos que pertencem aos domínios de *estágios completos* são idênticos em S e em Z

Esta equivalência possibilita que a partir de uma solução sabidamente correta, é possível avaliar a corretude de quaisquer outras soluções candidatas de maneira relativamente simples. O algoritmo de detecção automática de erros do ***iComb*** é precisamente a implementação do teorema 4.12 e pode ser visto no anexo da página 78.

Cada decisão do aluno realizada é avaliada imediatamente e caso qualquer uma das três condições seja falsa, o aluno será notificado no mesmo momento.

7 ♡	8 ♡	9 ♡	10 ♡	J ♡	Q ♡	K ♡	A ♡	7 ◇	8 ◇	9 ◇	10 ◇	J ◇	Q ◇	K ◇	A ◇	7 ♠	8 ♠	9 ♠	10 ♠	J ♠	Q ♠	K ♠	A ♠	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	7 ♡
		1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	8 ♡
			1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	9 ♡
				1	1	1	1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	10 ♡
					1	1	1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	J ♡
						1	1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	Q ♡
							1	1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	K ♡
								1	1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	A ♡
									1	1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	7 ◇
										1	1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	8 ◇
											1	1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	9 ◇
												1	1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	10 ◇
													1	1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	J ◇
														1	1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	Q ◇
															1	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	K ◇
																$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$	1	A ◇
																	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{3}{7}$	7 ♠
																		$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{3}{7}$	8 ♠
																			$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{3}{7}$	9 ♠
																				$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{3}{7}$	10 ♠
																					$\frac{1}{7}$	$\frac{1}{7}$	$\frac{3}{7}$	J ♠
																						$\frac{1}{7}$	$\frac{3}{7}$	Q ♠
																							$\frac{3}{7}$	K ♠
																								A ♠

Tabela 4.3: Tabela de Frequências do exemplo 4.4.

4.5 Limites do *iComb*

A partir dos resultados anteriores fica claro o grande potencial do ***iComb*** para aplicações de exercícios. Entretanto, nota-se que ele foi definido apenas para a classe de *problemas simples* e portanto deve-se investigar o quão grande é esta classe. Infelizmente, chega-se a conclusão que existem problemas combinatórios, semelhante aos considerados, que não são do tipo *simples*. Como no exemplo seguinte.

Exemplo 4.5. Em um baralho de 32 cartas, quantas mãos de 6 cartas podemos formar com exatamente 3 Ases e 3 cartas de Paus?

Antes de ser mostrado que de fato este problema não é do tipo *simples*, e portanto não solúvel pela atual versão do ***iComb***, vale notar que o exemplo a seguir é *simples* e é muito parecido com o exemplo 4.5.

Exemplo 4.6. Em um baralho de 32 cartas, quantas mãos de 5 cartas podemos formar com exatamente 3 Ases e 3 cartas de Paus?

Este exemplo é o mesmo do exemplo 4.3 que foi apresentado e resolvido na página 50. Porém, quando se altera a quantidade solicitada de 5 para 6 cartas, o exercício deixa de ser do tipo *simples*, não podendo mais ser resolvido com ajuda do ***iComb***. Mesmo que a quantidade solicitada fosse 7, o problema também não seria do tipo *simples*. Entretanto, é possível alterar sutilmente o problema de maneira a torná-lo um problema do tipo *simples*. Isso é feito no exemplo seguinte, acrescentando-se a restrição "nada a mais".

Exemplo 4.7. Em um baralho de 32 cartas, quantas mãos de 6 cartas podemos formar com exatamente 3 Ases e 3 cartas de Paus e **nada mais**?

A razão para que exercícios como o exemplo 4.5 não podem ser resolvidos com o auxílio do ***iComb*** é decorrente da seguinte propriedade.

Teorema 4.13. *(Teste de simplicidade)* Dado um problema P, do tipo *simples*, sobre um universo U e um conjunto resposta R de P. Sejam dois elementos $a, b \in U$ tal que $F_R(\{a\}) \neq 0$, $F_R(\{b\}) \neq 0$ e $ext_R(\{b\}) \neq \emptyset$, então:

$$F_R(\{a\}) \neq F_{ext_R(\{b\})}(\{a\}) \Rightarrow F_R(\{a\}) = F_R(\{b\}) \tag{4.19}$$

Demonstração. Como P é do tipo *simples* e R é o conjunto resposta, existe um conjunto de estágios $\{E_1, E_2, \ldots, E_n\}$ tal que $E_1 \times E_2 \times \ldots \times E_n = R$

Quando dois elementos a, b pertencem a estágios distintos então $F_R(\{a\}) = F_{ext_R(\{b\})}(\{a\})$ e $F_R(\{b\}) = F_{ext_R(\{a\})}(\{b\})$, isto é, se forem eliminadas todas as configurações onde um dos elementos existe, a frequência com que o outro elemento ocorre não é alterada. Este resultado é baseado no fato de que a frequência de um elemento no conjunto resposta é idêntica a frequência deste elemento no estágio em que pertence, não importando quais são os outros estágios.

Se $F_R(\{a\}) \neq F_{ext_R(\{b\})}(\{a\})$ então a e b pertencem ao mesmo estágio, digamos E. Ora, dois elementos que pertencem ao mesmo estágio possuem exatamente a mesma frequência, isto é, $F_R(\{a\}) = F_R(\{b\})$. □

Supondo o exemplo 4.5 com 7 cartas ao invés de 6, *Em um baralho de 32 cartas, quantas mãos de 7 cartas podemos formar com exatamente 3 Ases e 3 cartas de Paus?*, vamos usar o resultado acima para provar que não se pode resolvê-lo pelo ***iComb*** em seu estágio atual.

Para facilitar a análise, o conjunto Resposta R será escrito como a união de dois conjuntos que formam uma partição de R: o conjunto dos elementos que possuem o $\clubsuit^{A}$ e o conjunto dos elementos que não o possuem. Sendo R seja o conjunto resposta do exercício, então:

$$R = ftr_R(\left\{\clubsuit^{A}\right\}) \cup ext_R(\left\{\clubsuit^{A}\right\})$$

onde $ftr_R(\{\clubsuit^{A}\})$ representa o conjunto onde todas as configurações contém $\clubsuit^{A}$ enquanto $ext_R(\{\clubsuit^{A}\})$ representa o conjunto das configurações que não contém $\clubsuit^{A}$.

Neste ponto é importante observar que pode-se resolver cada uma das partes do problema pensando novamente em estágios. Assim, $ftr_R(\{\clubsuit^{A}\})$ e $ext_R(\{\clubsuit^{A}\})$ podem ser reescritos como:

- $ftr_R(\{\clubsuit^{A}\}) = E^1_{\text{valor é ás e naipe é paus}} \times E^2_{\text{valor é ás e naipe não é paus}} \times E^2_{\text{valor não é ás e naipe é paus}} \times E^2_{\text{valor não é ás e naipe é vermelho}}$
- $ext_R(\{\clubsuit^{A}\}) = E^3_{\text{valor é ás e naipe não é paus}} \times E^3_{\text{valor não é ás e naipe é paus}} \times E^1_{\text{valor não é ás e naipe é vermelho}}$

Como

- $\left|E^1_{\text{valor é ás e naipe é paus}}\right| = 1$
- $\left|E^2_{\text{valor é ás e naipe não é paus}}\right| = C^3_2 = 3$
- $\left|E^2_{\text{valor não é ás e naipe é paus}}\right| = C^7_2 = 21$
- $\left|E^2_{\text{valor não é ás e naipe é vermelho}}\right| = C^{14}_2 = 91$
- $\left|E^3_{\text{valor é ás e naipe não é paus}}\right| = 1$
- $\left|E^3_{\text{valor não é ás e naipe é paus}}\right| = C^7_3 = 35$
- $\left|E^1_{\text{valor não é ás e naipe é vermelho}}\right| = 14$

Temos

- $ftr_R(\{\clubsuit^A\}) = 1 \times 3 \times 21 \times 91 = 5733$
- $ext_R(\{\clubsuit^A\}) = 1 \times 35 \times 14 = 490$

Como $ftr_R(\{\clubsuit^A\}) \cap ext_R(\{\clubsuit^A\}) = \emptyset$ então

$$|R| = \left|ftr_R(\left\{\clubsuit^A\right\})\right| + \left|ext_R(\left\{\clubsuit^A\right\})\right| = 5733 + 490 = 6223$$

A partir do resultado acima é fácil verificar que $F_R(\{\clubsuit^A\}) = \frac{5733}{6233} = \frac{117}{127}$. Para provar que o exercício não pode ser resolvido pelo ***iComb*** é importante calcular a frequência de outro elemento com frequência maior do que zero. Escolhido o elemento $\heartsuit^{10}$, pode-se calcular sua frequência em duas etapas, primeiro no conjunto $R_1 = ftr_R(\{\clubsuit^A\})$ e depois no conjunto $R_2 = ext_R(\{\clubsuit^A\})$.

Seja $E_1 = E^2_{\text{valor não é ás e naipe é vermelho}}$ e $E_2 = E^1_{\text{valor não é ás e naipe é vermelho}}$. Então:

- $F_{R_1}(\{\heartsuit^{10}\}) = \frac{p(E_1)}{|\tilde{E}_1|} = \frac{2}{14} = \frac{1}{7}$
- $F_{R_2}(\{\heartsuit^{10}\}) = \frac{p(E_2)}{|\tilde{E}_2|} = \frac{1}{14}$

Com os resultados acima pode-se calcular através da fórmula

$F_R(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\}) = \frac{|R_1|.F_{R_1}(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\})+|R_2|.F_{R_2}(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\})}{|R|} = \frac{5733\times\frac{1}{7}+490\times\frac{1}{14}}{6223} = \frac{819+35}{6223} = \frac{854}{6223} = \frac{122}{889}$

Do teorema 4.19 temos que se o exercício P puder ser resolvido pelo ***iComb*** então $F_R(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\}) \neq F_{ext_R(\{\begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\})}(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\}) \Rightarrow F_R(\{\begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}) = F_R(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\})$. Como $R_2 = ext_R(\{\begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\})$ temos

$$F_R(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\}) \neq F_{ext_R(\{\begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\})}(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\}) \Rightarrow F_R(\{\begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}) = F_R(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\})$$

$$F_R(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\}) \neq F_{R_2}(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\}) \Rightarrow F_R(\{\begin{smallmatrix}A\\\clubsuit\end{smallmatrix}\}) = F_R(\{\begin{smallmatrix}10\\\heartsuit\end{smallmatrix}\})$$

$$\frac{122}{889} \neq \frac{1}{14} \Rightarrow \frac{117}{127} = \frac{122}{889}$$

O que é claramente um resultado falso. Portanto, o exercício proposto não pode ser resolvido pelo ***iComb***.

No caso do exemplo 4.6 da página 63, qualquer dupla de elementos que respeitem as hipóteses do teorema 4.13 resultam em *verdadeiro* quando submetidos a implicação 4.19. Considere, por exemplo, os elementos $\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}$ e $\begin{smallmatrix}A\\\diamondsuit\end{smallmatrix}$. Como $F_R(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}) = F_{ext_R(\{\begin{smallmatrix}A\\\diamondsuit\end{smallmatrix}\})}(\{\begin{smallmatrix}K\\\clubsuit\end{smallmatrix}\}) = \frac{2}{7}$ então 4.19 resulta em *verdadeiro*.

A expressão $F_R(\{\begin{smallmatrix}A\\\diamondsuit\end{smallmatrix}\}) \neq F_{ext_R(\{\begin{smallmatrix}A\\\spadesuit\end{smallmatrix}\})}(\{\begin{smallmatrix}A\\\diamondsuit\end{smallmatrix}\}) \Rightarrow F_R(\{\begin{smallmatrix}A\\\diamondsuit\end{smallmatrix}\}) = F_R(\{\begin{smallmatrix}A\\\spadesuit\end{smallmatrix}\})$ também resulta em *verdadeiro* quando são considerados os elementos $\begin{smallmatrix}A\\\diamondsuit\end{smallmatrix}$ e $\begin{smallmatrix}A\\\spadesuit\end{smallmatrix}$ já que $\frac{2}{3} \neq \frac{1}{2} \Rightarrow \frac{2}{3} = \frac{2}{3}$ é uma expressão *verdadeira*. Naturalmente para provar que o exemplo 4.6 é do tipo *simples* será necessário avaliar a expressão para todas as duplas de elementos, operação desnecessária pois os estágios já foram apresentados anteriormente.

4.6 Nova classe de exercícios

A limitação do ***iComb*** apresentada na seção 4.5 mostra que há uma classe de exercícios que não são do tipo *simples*, portanto não solucionáveis pelo ***iComb*** em seu estágio atual. Porém, como foi visto, é possível resolver "pedaços" do exercício pelo mesmo modelo utilizado no ***iComb***. Na realidade, no exemplo dado, o exercício pode ser agrupado em dois casos *simples* e ao final soma-se os resultados parciais dos dois grupos para obter a resposta desejada.

Esta técnica sugere que é possível desenvolver para o sistema ***iComb*** uma *interface* mais geral, para tratar de exercícios como o proposto no exemplo 4.5. De fato, uma *in-*

terface similar foi criada para o sistema ***Combien?*** e chamada de *CaseSetBuilding* para ilustrar o fato de que cada sub problema do exercício é um caso do chamado, na nomenclatura do ***Combien?***, de *SetBuilding*.

A fim de melhorar a compreensão desta restrição do ***iComb***, será definida uma nova classe de problemas, os problemas *compostos*.

Definição 4.17. Um problema será chamado de *composto* ou do tipo *composto* quando possui uma *solução* que pode ser expressa como uma soma de 2 ou mais parcelas diferentes de zero, cada qual uma multiplicação de estágios. Assim, um exercício do tipo *composto* terá uma solução, por exemplo, da forma $E_1 \times E_2 + E_3 \times E_4 \times E_5$.

Observe que não se sabe se, dado um universo, o conjunto de todos os problemas *simples* é disjunto do conjunto de todos os problemas *compostos*. Foi mostrado através do *teste de simplicidade* apresentado no teorema 4.13 da página 63 que existem problemas que não são do tipo *simples* e são do tipo *composto* mas não é possível ter certeza de que não existem problemas que sejam ao mesmo tempo *simples* e *composto*. Se for possível mostrar que estes conjuntos não possuem intersecção, bastará verficar se um problema tem uma solução do tipo *simples* ou do tipo *composto* para reconhecer em qual grupo pertence sem a necessidade da aplicação de *teste de simplicidade*.

A figura 4.1 representa a relação entre os conjuntos de problemas num diagrama de Venn. Note que não se sabe ainda se existem problemas que sejam *simples* e *compostos* ao mesmo tempo, apesar da certeza de que existem problemas *compostos* que não são do tipo *simples*.

De qualquer maneira, independente da discussão acima, é possível estender o ***iComb***. Na realidade, ao menos duas escolhas são possíveis no esforço de dotar o ***iComb*** da possibilidade de resolução de exercícios como o do exemplo 4.5. A primeira opção é o caminho escolhido pelo ***Combien?***: criação de uma nova *interface*, ou máquina (novamente na nomenclatura do ***Combien?***) especializada neste tipo de exercício. A segunda opção é sutilmente diferente: alteração e generalização da *interface* existente de maneira a incorporar a possibilidade da separação de um exercício em outros sub problemas num todo coerente. Nesta alternativa, o aluno não deve escolher *a priori* com qual máquina irá trabalhar, porém, de alguma maneira, deverá ser capaz de dividir o problema em instâncias menores.

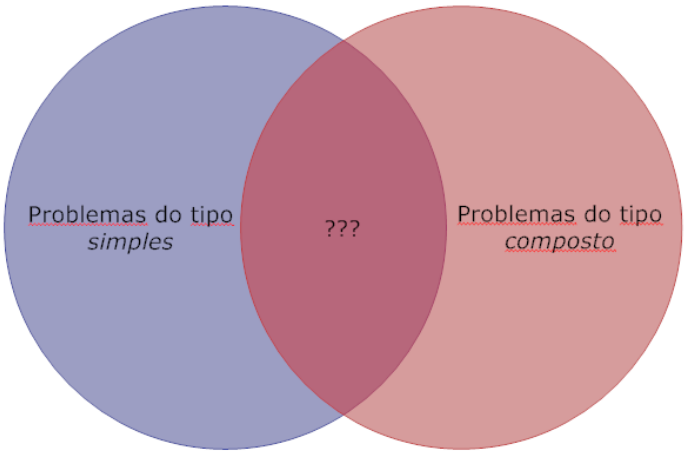

Figura 4.1: Classes de Problemas

4.7 Conclusão

No presente capítulo apresentou-se um estudo sobre condições necessárias e suficientes para o desenvolvimento de um algoritmo de detecção automática de erros para a classe de problemas combinatórios considerados. Foi definida uma classe de problemas, denominado *simples*, para a qual provou-se que é sempre possível validar uma proposta de solução, ou seja, se esta proposta apresenta erro é possível apontá-lo e se a solução proposta está correta, pode-se verificar a ausência de erro. Estes resultados possibilitaram a geração de um algoritmo de detecção automática de erros para os problemas do tipo *simples* - apresentado no anexo 1, a partir da página 78.

Além disso, foi mostrada a existência de problemas que não são do tipo *simples* e que portanto não podem ser utilizados no ***iComb*** por não ser solucionáveis pelo seu algoritmo verificador. Foi apresentado um caso particular de problema que não é do tipo simples (denominado *composto*) e demonstrado não ser possível resolvê-lo através dos mecanismos disponíveis no ***iComb***. Também foi apresentado um teste para verificar se um problema não pode ser do tipo *simples* a partir de seu conjunto resposta. Entretanto, foi mostrado que é possível decompor este problema em outros menores, cada um deles sendo do tipo simples, e portanto utilizáveis no ***iComb***. Ao final foram discutidas possibilidades de mudança na *interface* atual do ***iComb*** para permitir a utilização de problemas compostos.

Capítulo 5

Resultados

"Tell me and I forget, teach me and I remember, involve me and I learn."

Benjamin Franklin

"O aluno ouve e esquece, vê e se lembra, mas só compreende quando faz."

Confúcio

5.1 Experimento

O ***iComb*** foi avaliado por um grupo de seis professores de matemática do ensino médio, de maneira que não houve explicação de conceitos de combinatória. O experimento consistiu na resolução de um exercício de forma assistida pelo instrutor utilizando o ***iComb***. Na sequência foi solicitado que os professores resolvessem o restante dos exercícios utilizando o sistema.

Os exercícios apresentados foram:

1) Com um baralho de 32 cartas, quantas mãos de 2 cartas é possível formar com exatamente uma Dama e um Ás?

2) Com um baralho de 32 cartas, quantas mãos de 5 cartas é possível formar com exatamente 2 copas e 2 espadas?

3) Com um baralho de 52 cartas, quantas mãos de 8 cartas é possível formar com exatamente 1 ás, 2 reis e 2 valetes?

4) Com um baralho de 32 cartas, quantas mãos de 8 cartas é possível formar com exatamente 4 ases e 4 reis?

5) Com um baralho de 32 cartas, quantas mãos de 20 cartas é possível formar com 3 Ases, 4 cartas de espadas e 16 cartas vermelhas?

6) Com um baralho de 32 cartas, quantas mãos de 12 cartas é possível formar com o mínimo de sete cartas de copas, exatamente 4 ases, 2 cartas de espadas e nada mais?

7) Com um baralho de 32 cartas, quantas mãos de 18 cartas é possível formar com 6 cartas de ouros, no máximo 4 pretas, exatamente 4 valetes e exatamente 4 ases?

8) Com um baralho de 32 cartas, quantas mãos de 13 cartas é possível formar com no mínimo 1 carta de copas, exatamente 5 cartas pretas e exatamente 4 ases?

9) Com um baralho de 32 cartas, quantas mãos de 13 cartas é possível formar com 12 cartas vermelhas, exatamente 3 ases e no máximo 4 cartas de ouros?

10) Quantos times de 13 jogadores podemos formar com exatamente 5 jogadores do time azul e exatamente 5 jogadores do time verde? [1]

Ao final, o seguinte questionário foi entregue ao grupo de 6 professores:

1) Você conseguiu resolver os exercícios propostos?

2) Quando o ***iComb*** apresentou mensagens de erro, facilitou na resolução no exercício?

3) Você acha que o ***iComb*** pode ajudar no ensino de combinatória?

4) Você tem alguma sugestão para melhorar o ***iComb***?

5) Sobre a utilização do ***iComb***, você acha que: a) fácil b) razoável c) difícil

6) De que modo você utilizaria o ***iComb*** em sala de aula?

7) Comentários Gerais

[1] exercício do universo *futebol*

Vamos analisar questão a questão os resultados obtidos.

1) *Você conseguiu resolver os exercícios propostos?*
   Três professores responderam que sim (com facilidade comentou um dos professores) enquanto os outros 3 responderam que resolveram alguns dos exercícios propostos. É importante observar que o exercício 3 do experimento estava configurado de forma incorreta, de maneira que qualquer tentativa de resolução resulta em um erro.

2) *Quando o **iComb** apresentou mensagens de erro, facilitou na resolução no exercício?*
   Cinco dos seis professores respoderam que sim. Alguns professores sugeriram melhoria na qualidade das mensagens, de maneira a torná-las mais claras aos alunos.

3) *Você acha que o **iComb** pode ajudar no ensino de combinatória?*
   Novamente, cinco entre os seis professores disseram que o sistema pode ajudar no ensino de combinatória. Neste ponto várias observações foram feitas: "... quem estiver apresentando em sala de aula (laboratório) precisa ter domínio do assunto...", "... recurso fácil de ser aplicado; poderia ser desenvolvido atividades com níveis avançados."; "certeza que sim, se as mensagens forem mais didáticas."; "Sim, sem dúvida"; "Sim, é uma boa ferramenta de auxílio ao ensino-aprendizagem.". O professor que respondeu que o ***iComb*** não pode ajudar no ensino de combinatório disse: "Não, porque o programa não está muito claro do que se pede no exercício".

4) Você tem alguma sugestão para melhorar o ***iComb***?
   Neste ponto uma série de sugestões foram propostas: "Dicas para resolução de exercícios"; "Aceitar respostas similares" [2]; "Elaboração de questões para grupos diferentes" [3]; "A tela inicial, até o momento, apresentável. A quantidade de exercícios é pequena, provavelmente, haverá a necessidade de se acrescentar recursos para facilitar a inserção de questão e visualização. Trabalhar com Ergonomia do Software e Design de Interação"; "Sim, que as mensagens sejam mais didáticas, e que se coloquem links explicativos."; "Acredito [que] se houve[r] uma mensagem de erro por

[2]Na realidade, como vimos no decorrer deste trabalho, o ***iComb*** aceita todas as respostas possíveis e rejeita todas as respostas incorretas.

[3]O ***iComb*** permite a inclusão de exercício de maneira que é possível trabalhar com um repositório de exercícios. É possível a elaboração de exercícios para grupos diferentes quando utilizamos o ***iComb*** junto a algum sistema LMS como o SAW.

escrito já ajuda ou um manual para facilitar" [4]; "Melhorar as mensagens de erros por que não conseguimos ver exatamente qual era o erro"; "Sim, um número maior de exercícios."

5) *Sobre a utilização do **iComb**, você acha que: a) fácil b) razoável c) difícil*
Obtivemos 5 alterativas (b) e uma alternativa (a)

6) *De que modo você utilizaria o **iComb** em sala de aula?*
As respostas a esta questão foram bastante variadas: "Montaria questões para resolução em grupos"; "Utilizaria nas aulas de pós-graduação em Informática na Educação que leciono como forma de apoio aos profissionais da área de informática e matemática"; "Após várias aulas de combinatória, permuta e arranjo, como apoio nas resoluções de exercícios e maturação de conceitos"; "Laboratório de matemática ou sala de informática no estudo de probabilidade."; "Se ele fosse mais claro, utilizaria com certeza, mas como está os alunos teriam dificuldade em resolver as questões onde desmotivaria os alunos."

7) *Comentários Gerais*
Os comentários gerais foram: "O projeto é muito bom, porém ainda falta um pouquinho de aplicabilidade em sala de aula"; "Faça um ambiente de ensino que facilite o modo de utilização para o usuário e não a facilidade do raciocínio. Em Design de Iteração de um produto de software deve-se pensar na experiência para o usuário nas metas de usabilidade do produto e na qualidade do ensino e aprendizagem para apoio ao aluno e profissional da área."; "o iComb trava muito (pode ser o sistema?) e deve abranger exercícios bem iniciais de combinatória geral (com menos baralhos) e, claro, bem mais exercícios."; "A idéia é ótima, valeu. Espero que em breve esteja com mais opções e facilidades visando também o aluno em sala de aula.".

5.2 Análise dos Resultados

No geral, a avaliação do ***iComb*** foi positiva com a percepção geral de que o sistema era uma boa idéia e útil para o ensino de combinatória. Houve um grande número de comentários com relação a quantidade reduzida de exercícios do ***iComb*** ilustrando o não entendimento

[4] Este professor havia comentado sobre mensagens de erro em branco anteriormente. Provavelmente se refere a algum erro ocorrido durante a utilização do ***iComb***.

do conceito de repositório de exercícios. De fato, o sistema ***iComb*** pode trabalhar com qualquer número de exercícios, não havendo qualquer referência aos exercícios no código fonte do programa. Este comentário dos professores reforça a importância da idéia de repositório de exercícios.

Críticas a qualidade das mensagens de erro foram feitas por boa parte dos professores. Além da deficiência do sistema, este conjunto de comentários mostra que a utilização do sistema ***iComb*** não é trivial e imediata, pois se fosse, os professores dariam menos importância as mensagens de erro obtidas. O fato de que 5 entre 6 professores avaliaram a utilização do ***iComb*** como de *razoável* dificuldade corrobora esta última observação.

Capítulo 6

Conclusões

O grupo ***Combien?*** definiu as bases matemáticas do chamado método construtivo (Le Calvez et al., 2003, Giroire et al., 2006, Tisseau et al., 2000) para ajudar os alunos a resolverem exercícios de combinatória estabelecendo quatro interfaces pedagógicas especiais. O sistema ***Combien?***, que implementou estes conceitos, é um aplicativo concebido para ser localmente instalado na máquina de professores e alunos. O sistema foi testado com sucesso no sistema educacional francês.

Adequar o ***Combien?*** ao paradigma Web e a educação a distância motivou o desenvolvimento deste trabalho. Em especial, pode ser bastante produtivo integrá-lo a sistemas gerenciadores de curso, como ocorre no sistema SAW com os módulos de aprendizagem *iGeom - Geometria Interativa na Internet* e *iGraf - Gráficos Interativos na Internet*.

A primeira abordagem tentada foi uma integração direta do sistema ***Combien?***, esta abordagem levou a definição e implementação do *CombienDelivery*. Entretando, esta não se mostrou uma boa alternativa, por exigir alterações no sistema ***Combien?*** que deveriam ser feitas pelo grupo francês, mas isso mostrou-se muito lento. Além disso, tal alternativa reduziria o espaço para futuras alterações e adequações à realidade do aluno brasileiro, restringindo o potencial de adequação e evolução do sistema.

A definição do *iComb - Combinatória Interativa na Internet* baseado na experiência do ***Combien?*** e respeitando a arquitetura do ***iGeom*** e ***iGraf*** foi a alternativa encontrada para o desenvolvimento de um sistema voltado ao aprendizado de combinatória aderente ao paradigma Web. O ***iComb*** foi definido para ser facilmente integrado a sistemas como o SAW e procurando estender a idéia do ***Combien?***, incorporando algumas características

inéditas como uma arquitetura de *plugins*. Além disso foi possível demonstrar que há uma classe de exercícios que não podem ser resolvidos pelo ***iComb*** em seu atual estágio de desenvolvimento.

A proposta da arquitetura de *plugins* foi flexibilizar o sistema de maneira a permitir que novos universos e seus respectivos exercícios sejam criados. A motivação por trás desta idéia é a suspeita de que exercícios de determinados universos podem ser mais efetivos para a aprendizagem do que outros, dependendo do ambiente sócio cultural dos alunos em questão. De fato, alguns professores presentes no experimento realizado propuseram a criação de exercícios que não utilizassem baralhos ou cartas. A atual versão do ***iComb*** possui dois *plugins*, o universo "Baralho" e o universo "Futebol", este último possivelmente mais ao gosto dos alunos brasileiros.

A fundamentação matemática apresentada no capítulo 4 é nova, não tendo sido encontrada nos trabalhos relativos ao ***Combien?***. Ela foi resultado do esforço para dotar o ***iComb*** de um bom algoritmo para a detecção automática de erros. É importante destacar que uma das contribuições deste trabalho foi encontrar condições necessárias e suficientes que caracterizam uma solução correta. Esta caracterização viabilizou um algoritmo rápido para avaliação automática de respostas dos alunos. Entretanto, esta fundamentação também mostrou a existência de uma classe de exercícios que não é do tipo *simples* e que portanto não pode ser resolvido pelo ***iComb*** em seu atual estágio de desenvolvimento.

A disponibilização do ***iComb*** como *widget* , de maneira a permitir sua incorporação em *blogs* ou páginas pessoais tem o potencial de aumentar e estimular conteúdos relativos a matemática pelo mundo virtual. A atual implementação permite a inclusão da *widget* do ***iComb*** através da simples operação de copiar e colar, bastando a cópia do trecho específico na página oficial do sistema. Uma vez incorporado na página hospedeira o ***iComb*** irá sortear um dos exercícios presentes na sua base de dados a cada vez que a página é recarregada.

O sistema ***iComb*** - www.matematica.br/icomb - será distribuido gratuitamente e a disponibilização de seu código-fonte se encontra em estudo.

6.1 Trabalhos Futuros

Um trabalho que ainda deverá ser feito com o ***iComb*** são estudos mais acurados sobre sua utilização em ambientes educacionais reais. Também seria interessante criar uma comunidade de professores que desenvolvessem novos problemas e depois os testassem com seus alunos.

A atual versão do sistema ***iComb*** registra todas as decisões realizadas pelo aluno durante o processo de resolução de um exercício. Além disso, o momento em que a decisão ocorre também é armazenado no sistema e está disponível para análise. Assim, um trabalho interessante a ser realizado, seria o desenvolvimento de uma interface que apresentasse a resposta do aluno ao professor de forma dinâmica, como se fosse um filme, possibilitando ao professor "assistir" a resolução do aluno. Este formato mostrará claramente ao professor todas as decisões tomadas, as mensagens de erro apresentadas e o tempo gasto em cada etapa da resolução.

No capítulo 4 apresentamos o desenvolvimento matemático que resultou no algoritmo de detecção automática de erros. Além disso mostramos que existe uma classe de exercícios que não pode ser resolvido pelo ***iComb***. Por outro lado, vimos também que esta classe de exercício possui sub problemas, cada qual *resolvível* pelo ***iComb*** de forma isolada. Esta constatação nos dá confiança de que é possível estender o ***iComb*** de maneira a torná-lo adequado a trabalhar com este novo grupo de exercícios, além de preservar boa parte do desenvolvimento uma vez que o formato atual resolve os sub-problemas. Este futuro trabalho irá tornar o ***iComb*** uma ferramenta pedagógica mais abrangente.

Anexo 01

Algoritmo de Deteção Automática de Erros

Segue o código fonte do algoritmo de detecção automática de erros fruto dos resultados descritos neste trabalho.

```
package icomb.erro;

import icomb.formula.Binomio;
import icomb.objects.Condicao;
import icomb.objects.Element;
import icomb.objects.Estagio;
import icomb.objects.Universo;

import java.util.HashSet;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.Vector;


/**
 * Esta classe verifica a solução de um problema do IComb.
 * No construtor informamos a lista de elementos do universo
 * e um conjunto solução de condições.
 *
 * @author eisenmann
 *
 */
public class Avaliador
{

  private Element[] elementos;
  private Condicao[] condicoes;
  private Map elemParticao;
  private Map elemResposta;
  private Map particoes;

  /**
   * O array de condicoes representa um conjunto solução para o problema
   *
   * @param universo
   * @param condicoes que são uma solução para o problema
   */
  public Avaliador(Universo universo, Condicao[] condicoes)
  {

Vector vecElementos = universo.getElementos();
elementos = new Element[vecElementos.size()];
for(int i=0 ; i<vecElementos.size(); i++)
{
elementos[i] = (Element) vecElementos.get(i);
}

    this.condicoes = condicoes;
    this.elemParticao = new Hashtable();
    this.elemResposta = new Hashtable();
    this.particoes = new Hashtable();
    for(int i=0; i<elementos.length ; i++)
    {
      for(int j=0 ; j<condicoes.length ; j++)
      {
        if (condicoes[j].evaluate(elementos[i]))
        {
          if (elemParticao.get(elementos[i])!=null)
            throw new RuntimeException("Elemento não pode pertencer a mais do que uma condição!");
          else
          {
```

```
          elemParticao.put(elementos[i], condicoes[j]);

          Set set = (Set) particoes.get(condicoes[j]);
          if (set == null)
            set = new HashSet();

          set.add(elementos[i]);
          particoes.put(condicoes[j],set);
        }
      }
    }
  }
}


/**
 * Este método permite adicionarmos uma condição candidata para o problema.
 * A Condição não precisa ser idêntica a alguma das condições informadas no
 * contrutor, porém precisa ser compatível.
 *
 * A todo o momento o conjunto de soluções candidatas será confrontado com
 * o conjunto solução informado no construtor e uma mensagem de erro será
 * lançada quanto os conjuntos não forem compatíveis
 *
 * @param condicao candidata
 */
public void adicionaCondicao(Condicao condicao)
{

  Set setCondicoes = new HashSet();
  Set setElementos = new HashSet();
  for(int i=0; i<elementos.length ; i++)
  {
      if (condicao.evaluate(elementos[i]))
      {
        setElementos.add(elementos[i]);
        setCondicoes.add(elemParticao.get(elementos[i]));
      }
  }
  if (setElementos.size()==0)
    throw new RuntimeException("Nenhum elemento do conjunto universo atende as restrições informadas. Cada estágio criado deverá selecionar ao me

  if (setCondicoes.size()==0 || setCondicoes.size()==1 && (setCondicoes.iterator().next() == null))
    throw new RuntimeException("As restrições informadas selecionam elementos do conjunto univero que não deveriam ser escolhidos para este exerc

  // Frequencia dos elementos não é igual a 1 isto é
  // nem todos os elementos estão sendo selecionados
  if (condicao.getQuantidade() != setElementos.size())
  {
    if (setCondicoes.size()>1)
    {
     Iterator it = setCondicoes.iterator();
     Condicao c1 = (Condicao) it.next();
     Condicao c2 = (Condicao) it.next();

     HashSet conjElementos1 = (HashSet) particoes.get(c1);
     HashSet conjElementos2 = (HashSet) particoes.get(c2);

     Iterator it2 = setElementos.iterator();
     Element e1=null;
     while (it2.hasNext())
     {
     Element elemento = (Element) it2.next();
     if (conjElementos1.contains(elemento))
     {
     e1 = elemento;
     break;
     }
     }

     Iterator it3 = setElementos.iterator();
     Element e2=null;
     while (it3.hasNext())
     {
     Element elemento = (Element) it3.next();
     if (conjElementos2.contains(elemento))
     {
     e2 = elemento;
     break;
     }
     }


        throw new RuntimeException("Este estágio certamente irá alterar a distribuição desejada dos elementos. " +
         "Ela envolve elementos que deveriam ser selecionados em estágios distintos." +
         " Por exemplo, os elementos " + e1 + " e " + e2 +
         " devem estar presentes em estágios distintos.");
    }
```

```
    Condicao baseCondicao = (Condicao) setCondicoes.iterator().next();
    Set baseElementos = (Set) particoes.get(baseCondicao);

    if (setElementos.equals(baseElementos))
    {
      if (condicao.getQuantidade() != baseCondicao.getQuantidade())
        throw new RuntimeException("A condição selecionada está provavelmente correta, porém o número de elementos está errado. ");
    }
    else
      throw new RuntimeException("Esta seleção irá alterar a distribuição esperada de alguns elementos.");

  }
  else
  {
    Iterator it = setCondicoes.iterator();
    while (it.hasNext())
    {
      Condicao baseCondicao = (Condicao) it.next();
      Set baseElementos = (Set) particoes.get(baseCondicao);

      if (baseCondicao.getQuantidade() != baseElementos.size())
          throw new RuntimeException("Alguns elementos terão sua frequência alterada para 1, isto é, sempre irão aparecer.");
    }
  }
  // Se chegou aqui, condicao individual é ok
  consolida(condicao,setElementos);

}

/**
 * Depois das verificacoes, consolida condição ao conjunto de elementos.
 *
 * @param cond
 * @param setElementos
 */
private void consolida(Condicao cond, Set setElementos)
{
  Iterator it = setElementos.iterator();
  while (it.hasNext())
  {
    Element e = (Element) it.next();
    Condicao c = (Condicao) elemResposta.get(e);
    if (c != null)
      throw new RuntimeException("Mais do que 1 estágio seleciona um elemento ");
  }

  it = setElementos.iterator();
  while (it.hasNext())
  {
    Element e = (Element) it.next();
    elemResposta.put(e,cond);
  }

}


/**
 * Valida solução completa. Mesmo que todas as condições
 * candidatas sejam individualmente compatíveis é possível
 * que ao final do processo alguns elementos não tenham sido
 * selecionados.
 *
 */
public void valida()
{


  Iterator it = elemParticao.keySet().iterator();
  while (it.hasNext())
  {
    Element e = (Element) it.next();
    Condicao cSolucao = (Condicao) elemParticao.get(e);
    Condicao cResposta = (Condicao) elemResposta.get(e);

    if (cSolucao == null && cResposta !=null)
      throw new RuntimeException("Elementos indevidos estão sendo selecionados.");

    if (cResposta == null && cSolucao !=null)
      throw new RuntimeException("Há elementos que não estão sendo selecionados. Ex:" + e);

  }

}

/**
 * Reinicia a verificação
 */
```

```
  public void reset()
  {
     this.elemResposta = new Hashtable();

  }

  /**
   * Solução informada está correta? 1 - Sim, 0 - Não
   *
   * @return 1 - Sim, 0 - Não
   */

  public int getResposta()
  {
    try
    {
      valida();
    }
    catch(Exception e)
    {
      return 0;
    }
    return 1;

  }

  /**
   * Verifica se a fórmula do estágio está correta
   * @param universo
   * @param stage
   */
  public static void deteccaoDeErroFormula(Universo universo,Estagio stage)
  {

  Condicao condicao = stage.criaCondicao();
  Vector elementos = universo.getElementos();
  int n=0;
  for(int i=0 ; i<elementos.size() ; i++)
  {
  Element element = (Element) elementos.get(i);
  if (condicao.evaluate(element))
  n++;
  }


  int p = condicao.getQuantidade();
  Binomio binomio = new Binomio();
  long resultado1 = binomio.calcula(n,p);
  long resultado2 = 0;
  resultado2 = stage.getFormula().calcula(stage.getN(), stage.getP());

  if (resultado1!=resultado2)
  throw new RuntimeException("Cálculo Inválido");

  }


}
```

Referências Bibliográficas

L.O. Brandão, S. Isotani, and J.G. Moura. Imergindo a geometria dinâmica em sistemas de educação a distância: igeom e saw. *Revista Brasileira de Informática na Educação*, 2006. Volume 14 - Número 1 - Janeiro a Abril de 2006.

D.H. Clements. From exercises and tasks to problems and projects Unique contributions of computers to innovative mathematics education. *Journal of Mathematical Behavior*, 19(1):9–47, 2000.

H. Giroire, F. Le Calvez, and G. Tisseau. Benefits of Knowledge-Based Interactive Learning Environments: A Case in Combinatorics. *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, pages 285–289, 2006.

K.M. Goertzel. Introduction to Software Security. 2009.

N. Hara and R. Kling. Student's frustrations with a web-based distance education course. *First Monday: Journal on the Internet 4(12)*, 1999.

M. Hentea, M.J. Shea, and L. Pennington. A perspective on fulfilling the expectations of distance education. *Conference On Information Technology Education*, pages 160–167, 2003.

S. Isotani. Desenvolvimento de ferramentas no igeom: utilizando a geometria dinâmica no ensino presencial e a distância. Master's thesis, Universidade de São Paulo, 2005.

M. Janine. Saw - sistema de aprendizado pela web: Motivações e desenvolvimento. Master's thesis, Universidade de São Paulo, 2007.

E. Kirby. Student's frustration with a web-based distance. *In Technology and Teacher Education Annual*, pages 199–205, 1999.

F. Le Calvez, H. Giroire, J. Duma, G. Tisseau, and M. Urtasun. Combien? a Software to Teach Students How to Solve Combinatorics Exercises. *Proceedings of Artificial Intelligence in Education, Sydney, Australia, IOS Press, Amsterdam*, pages 447–453, 2003.

F. M. Litto, A. Filatro, and C. André. Brazilian research on distance learning, 1999-2003: A state-of-the art study. *In Proceedings of International Congress of Distance Education*, 2004. http://www.abed.org.br/congresso2004/por/pdf/180-TC-D4.pdf.

DG Moodle. Moodle-A Free, Open Source Course Management System for Online Learning. *Available http://moodle.org/[27 Jan 2004]*, 2004.

A. C. O. Morgado, J. B. P. Carvalho, P. C. P. Carvalho, and P. Fernandez. *Análise Combinatória e Probabilidade*. Coleção do Professor de Matemática. Sociedade Brasileira de Matemática, 9 edition, 1991.

J.G. Moura, L.O. Brandão, and A.A.F. Brandão. A web-based learning management system with automatic assessment resources. *Frontiers In Education Conference, 2007, Milwaukee, EUA.*, 2007. Proceedings of the Frontiers in Education Conference FIE'2007, 2007.

T. Oreilly. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software.

G. Pólya. Mathematical discovery : on understanding, learning and teaching problems solving. *Imprenta New York : John Wiley, 1981*, 1981.

H.V. Rocha et al. Projeto TelEduc: Pesquisa e Desenvolvimento de Tecnologia para Educação a Distância. *IX Congresso Internacional de Educação a Distância da ABED (Associação Brasileira de Educação a Distância)*, 2002.

G. Tisseau, H. Giroire, F. Le Calvez, M. Urtasun, and J. Duma. Design principles for a system to teach problem solving by modelling. *Lecture Notes in Computer Science*, 1839:393–402, 2000.

VW. Visual works. http://www.cincomsmalltalk.com/userblogs/cincom/blogView, 2009.