

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

BRUNO PINTO SILVA

**Modelagem do Conhecimento de  
Estudantes através de Redes Bayesianas**

Monografia apresentada como requisito parcial  
para a obtenção do grau de Bacharel em Ciência  
da Computação

Orientador: Prof. Dr. Rosa Maria Vicari

Porto Alegre  
2016

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do Curso de Ciência de Computação: Prof. Carlos Arthur Lang Lisbôa

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“A alegria está na luta, na tentativa, no sofrimento envolvido. Não na vitória propriamente dita.”*

— MOHANDAS GANDHI

## **AGRADECIMENTOS**

Agradeço à minha família pela amizade, carinho, risadas compartilhadas, união e dedicação. Em especial à minha mãe e meu pai, pela sabedoria transmitida e por me ensinarem os valores que hoje são os mais preciosos para mim, como trabalho duro, dedicação, perseverança e honestidade, e à Cristiane, que segurava minha mão com amor enquanto descobríamos um mundo novo.

Agradeço também ao pessoal da Cognitiva Brasil por ter acreditado em mim, pela parceria e por terem me ajudado nesse e em outros trabalhos. Agradeço à minha orientadora, Rosa Vicari, por dividir seu conhecimento comigo para o desenvolvimento desse trabalho. Agraço àqueles professores da graduação que foram exemplos de dedicação e competência que almejo seguir durante minha carreira profissional. Agradeço também aos meus amigos, Fernando, Pedro e William, que me acompanharam durante o curso e também foram parte dessa conquista.

## RESUMO

A educação cumpre um papel importante na formação de cada indivíduo. Com o avanço da tecnologia nas últimas décadas, surgiram novas possibilidades para a democratização do conhecimento. Assim, devemos nos preocupar com a criação de ferramentas que aproveitem esse potencial. Nesse contexto, o objetivo deste trabalho é criar um sistema de avaliação de conhecimento sobre algoritmos e programação em linguagem C que maximize a quantidade de informação obtida, minimizando o número de medições, otimizando, dessa forma, este processo. Com o objetivo de buscar soluções para o problema do número de medições foi desenvolvido um sistema interativo que estima a probabilidade de o aluno saber cada componente de conhecimento baseando-se em suas respostas. Optou-se por usar Redes Bayesianas para fazer essa modelagem. Ao final, o trabalho apresenta execuções simuladas que indicam coerência nas ideias apresentadas, porém, devido à falta de experimentos com alunos, não se pode afirmar a correspondência dessa modelagem com o estado real de conhecimento de um estudante.

**Palavras-chave:** Redes Bayesianas. Probabilidade Condicional. *Knowledge Tracing*.

## **Students' Knowledge Modeling through Bayesian Networks.**

### **ABSTRACT**

Education plays an important role in the formation of each individual. With the advancement of technology in recent decades, there were new possibilities for the democratization of knowledge. So we should be concerned with the creation of tools to harness this potential. In this context, the aim of this work is to create a knowledge assessment system of algorithms and programming in C language that maximizes the amount of information obtained by minimizing the number of measurements, optimizing, thus, this process. In order to find solutions to the problem of the number of measurements an interactive system that estimates the probability of the student to know every component of knowledge based on his responses was developed. We chose to apply Bayesian networks to build a model of the student's knowledge. At the end, we present simulated executions that indicate consistency in the ideas presented, however, due to lack of experiments with students, we can not state that the generated models correspond to the actual state of knowledge of a student.

**Keywords:** Bayesian Networks, Conditional Probability, Knowledge Tracing.

## LISTA DE FIGURAS

Figura 1.1 Grafo direcionado representando conexões em série .....	15
Figura 1.2 Grafo direcionado representando conexões divergente .....	16
Figura 1.3 Grafo direcionado representando conexões convergente.....	17
Figura 2.1 Exemplo de topologia de espinha dorsal. ....	26
Figura 2.2 Exemplo de relação entre componente de conhecimento e probabilidade de acerto ou erro.....	27
Figura 2.3 Espinha dorsal do modelo de conhecimento para linguagem C .....	30
Figura 2.4 Função de entropia.....	33
Figura 2.5 Tela de login do sistema .....	34
Figura 2.6 Exibição de pergunta de múltipla escolha .....	35
Figura 2.7 Diagrama de classes resumido da aplicação backend.....	36
Figura 2.8 Sequência de chamadas na obtenção da próxima questão a ser trabalhada...36	
Figura 2.9 Diagrama de classes das entidades do sistema .....	37
Figura 2.10 Diagrama entidade relacionamento da base de dados .....	37
Figura 3.1 Ordem da obtenção das evidências na primeira simulação .....	40
Figura 3.2 Ordem da obtenção das evidências na segunda simulação.....	41

## LISTA DE TABELAS

Tabela 3.1 Primeira Simulação de Execução .....	39
Tabela 3.2 Segunda Simulação de Execução .....	41

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>10</b>
<b>1.1 Motivação</b> .....	<b>10</b>
<b>1.2 Objetivo</b> .....	<b>10</b>
<b>1.3 Referencial Teórico</b> .....	<b>11</b>
1.3.1 Fundamentos da Teoria Probabilística .....	11
1.3.2 Probabilidade Condicional.....	12
1.3.3 Cálculo de Probabilidades.....	12
1.3.4 Cálculo de Probabilidades para Variáveis.....	13
1.3.5 Redes Causais e d-separação .....	15
1.3.6 Redes Bayesianas.....	18
1.3.7 Entropia.....	20
<b>1.4 Estado da Arte</b> .....	<b>20</b>
1.4.1 CMU General Programming Languages Tutor.....	20
1.4.2 Uso de Redes Bayesianas.....	21
1.4.3 Outros Trabalhos Relacionados .....	22
<b>2 TRABALHO DESENVOLVIDO</b> .....	<b>25</b>
<b>2.1 Conceitos Usados na Modelagem</b> .....	<b>25</b>
2.1.1 Relacionamento Entre as Componentes de Conhecimento .....	25
2.1.2 Componente de Conhecimento e Probabilidade de Acerto .....	26
<b>2.2 Domínio de Conhecimento e Estrutura Topológica</b> .....	<b>27</b>
<b>2.3 Descrição da Implementação</b> .....	<b>30</b>
2.3.1 Valores Usados nas Tabelas de Probabilidades.....	31
2.3.2 Seleção de Componente de Conhecimento.....	32
2.3.3 Soluções de Software Utilizadas.....	33
2.3.4 Descrição da Interface de Usuário .....	34
2.3.5 Arquitetura do Sistema Backend .....	34
2.3.6 Base de Dados do Sistema .....	35
<b>3 DESCRIÇÃO DE EXPERIMENTOS E RESULTADOS</b> .....	<b>38</b>
<b>4 CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>42</b>
<b>REFERÊNCIAS</b> .....	<b>44</b>

## 1 INTRODUÇÃO

### 1.1 Motivação

A educação cumpre um papel importante na formação de cada indivíduo, contribuindo para seu bem-estar econômico, saúde e felicidade. No âmbito social, em decorrência dos crescentes avanços tecnológicos, a sociedade precisa cada vez mais de trabalhadores bem preparados e que facilmente se adaptem a novas ferramentas, como consequência disso, torna-se mais competitiva e, de certa forma, mais desigual. Em decorrência desses fatores, aumenta a demanda por novas tecnologias que reduzam os custos da educação e ao mesmo tempo a tornem mais efetiva (CARNOY, 2005). De fato, a disponibilização de cursos abertos massivos online tem aumentado nos últimos anos. Entre os exemplos mais populares estão Khan Academy<sup>1</sup>, EdX<sup>2</sup> e Coursera<sup>3</sup>, grandes empresas que oferecem materiais de alta qualidade gratuitamente, possibilitando acesso à informação a muitas pessoas que de outra forma não o teriam. Entretanto, a esses cursos ainda falta o nível de personalização adequado para que tais alunos os aproveitem ao máximo (REDDY; LABUTOV; JOACHIMS, 2016). Como mostrado em Bloom (1984), a tutoria um a um (na qual um tutor dedica-se exclusivamente a um aluno) é, na média, muito mais efetiva do que ensino para grandes grupos de alunos, chegando a dois desvios padrão de diferença entre a média dos alunos de cada grupo. Embora a abordagem individualizada usando tutores humanos frequentemente se mostre economicamente inviável, o crescente desenvolvimento da capacidade dos sistemas computacionais nas últimas décadas criou possibilidades para a tutoria automática individualizada (SPOEL et al., 2015), pois, técnicas de aprendizado de máquina podem diminuir seus custos sem comprometer sua qualidade (PIECH et al., 2015).

### 1.2 Objetivo

Objetivo Geral: criar uma sistema de avaliação de conhecimento sobre algoritmos e programação em linguagem C que maximize a quantidade de informação obtida minimizando o número de medições.

---

<sup>1</sup><https://pt.khanacademy.org/>

<sup>2</sup><https://www.edx.org/>

<sup>3</sup><https://www.coursera.org/>

Objetivos específicos: obter um método de avaliação que considere incerteza diante às evidências oferecidas pelo aluno.

### 1.3 Referencial Teórico

As definições e teoremas apresentados neste capítulo foram baseados no livro Jansen and Nielsen (2007).

#### 1.3.1 Fundamentos da Teoria Probabilística

Fazer **experimentos** é uma possibilidade quando se quer investigar os possíveis resultados envolvendo um problema. Para isso, diversas variáveis são observadas com o objetivo avaliar a relação causal entre elas. Os experimentos resultam em algumas de muitas possibilidades, cada uma com dada probabilidade de acontecer. Um **espaço amostral** é o conjunto de todos os possíveis resultados para um experimento. Ao jogar um dado, por exemplo, o espaço amostral é formado pelo conjunto  $S = \{1, 2, 3, 4, 5, 6\}$ . Um **evento** é um subconjunto do espaço amostral, ou seja, um conjunto de possíveis resultados de um experimento. Usando novamente o exemplo do dado, o evento de jogar um dado e esse cair com um número maior que 4 é dado por  $\{5, 6\}$ .

#### *Axiomas de Kolmogorov*

Para medir o grau de incerteza sobre um experimento, considerando o espaço amostral  $S$ , nós designamos a probabilidade  $P(A)$  a cada evento  $A \subseteq S$ . Essas probabilidades devem obedecer os seguintes axiomas que dão a base a toda a Teoria Probabilística:

**Axioma 1:**  $P(S) = 1$

A probabilidade de algum evento no espaço amostral aconteça é 1, ou seja, não existem eventos fora do espaço amostral.

**Axioma 2:**  $\forall A \subseteq S, P(A) \geq 0$

Nenhum evento pode ter probabilidade negativa.

**Axioma 3:** Se  $A \subseteq S, B \subseteq S$  e  $A \cap B = \emptyset$  então  $P(A \cup B) = P(A) + P(B)$

Se dois eventos são disjuntos, então a probabilidade de um evento combinado é a soma das probabilidades individuais desses eventos.

Uma regra complementar ao terceiro axioma pode ser facilmente comprovada:

Se  $A \subseteq S, B \subseteq S$  e  $A \cap B \neq \emptyset$  então  $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

### 1.3.2 Probabilidade Condicional

**Notação:**  $P(A|B) = p$  significa "Dado o evento  $B$ , a probabilidade de  $A$  é  $p$ ", ou seja, se  $B$  é verdade, tudo mais é irrelevante para  $A$ , então a probabilidade de  $A$  é  $p$ .

Afirmações sobre probabilidades estão sempre condicionadas ao que mais é sabido. Esse condicionamento pode ocorrer entre eventos, por exemplo, dizer que a probabilidade  $P(B|A) = p$

Se  $A$  e  $B$  são eventos pertencentes a  $S$  e soubermos que  $B$  irá acontecer, as possíveis saídas em que  $A$  pode ser verdade também estão contidas em  $B$ , ou seja, estão em  $A \cap B$ . Porém, ainda existe a probabilidade de que  $A$  não aconteça, dessa forma chegamos à **equação da probabilidade condicional:**

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Essa fórmula pode ser generalizada para qualquer quantidade de eventos, por exemplo, se soubermos também que o evento  $C$  irá acontecer, então a fórmula passa a ser:

$$P(A|B \cap C) = \frac{P(A \cap B \cap C)}{P(B \cap C)}$$

### 1.3.3 Cálculo de Probabilidades

Da equação da probabilidade condicional apresentada em 1.3.2 pode-se derivar a **regra fundamental para cálculo de probabilidade:**

$$P(A|B) * P(B) = P(A \cap B)$$

Como a operação de intersecção é comutativa, ou seja,  $P(A \cap B) = P(B \cap A)$  e já que  $P(B|A) * P(A) = P(B \cap A)$ , então, por transitividade obtemos a seguinte igualdade:

$$P(B|A) * P(A) = P(A|B) * P(B)$$

E assim chegamos à **Regra de Bayes:**

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

A Regra de Bayes provê uma forma de recalculer a crença sobre o evento  $A$  tendo obtido informações sobre o evento  $B$ . Por isso  $P(A)$  é chamado de probabilidade a priori,  $P(A|B)$  é chamado de probabilidade a posteriori e  $P(B|A)$  é chamado de predisposição<sup>4</sup> de  $A$  dada a ocorrência de  $B$

### 1.3.4 Cálculo de Probabilidades para Variáveis

O cálculo de probabilidades, além de considerar apenas simples eventos, também pode ser estendido para variáveis, que são coleções de espaços amostrais. Uma variável pode ser considerada um experimento e cada possível saída para um experimento é chamada de estado.

Para as explicações acerca de cálculo de probabilidade para variáveis, será considerada a variável  $A$  com os possíveis estados  $(a_1, a_2, \dots, a_i, \dots, a_m)$ , a variável  $B$  com os possíveis estados  $(b_1, b_2, \dots, b_j, \dots, b_n)$  e a variável  $C$  com os possíveis estados  $(c_1, c_2, \dots, c_k, \dots, c_p)$ .

$sp(A)$  denota o conjunto de variáveis de estado de  $A$ . Esses estados devem ser mutuamente exclusivos, ou seja, cada saída corresponde a apenas um estado, e exaustivos, ou seja, cada saída deve corresponder a algum estado existente.

Sendo  $P(A)$  a distribuição de probabilidade dos estados de  $A$  e  $x_i$  a probabilidade de cada estado  $i$  de  $A$ :

$$P(A) = (x_1, x_2, \dots, x_n), x_i \geq 0; \sum_{i=1}^n x_i = 1$$

**Notação:** A probabilidade de  $A$  estar no estado  $a_i$  é denotada por  $P(A = a_i)$ . Caso o contexto deixe óbvio qual é a variável em questão, usa-se  $P(a_i)$ .

Podemos estender a probabilidade condicional para variáveis, dessa forma,  $P(A_i|B_j)$ , com  $A$  tendo  $m$  possíveis estados e  $B$  tendo  $n$  possíveis estados, nos leva a  $n * m$  combinações de probabilidades condicionais.

Essas diversas probabilidades geralmente são exibidas usando tabelas, da seguinte forma:

---

<sup>4</sup>Em tradução livre do termo em inglês *likelihood*.

	$b_1$	$b_2$	...	$b_n$
$a_1$				
$a_2$				
...				
$a_m$				

Como mostrado no primeiro axioma de *Kolmogorov* em 1.3.1, a soma das probabilidades para todos os possíveis estados deve ser igual a 1, portanto:

$$\sum_{i=1}^m P(A = a_i | B = b_j) = 1 \text{ para cada } b_j.$$

A **probabilidade conjunta**  $P(A = a_i, B = b_j)$  é a probabilidade de ocorrência do estado  $a_i$  juntamente com o estado  $b_j$ . Essas probabilidades geralmente também são representadas em uma tabela de dimensões  $n * m$ . Note que, dessa vez, a probabilidade em questão refere-se às duas variáveis, dessa forma:

$$P(A, B) = \sum_{i=1}^m \sum_{j=1}^n P(A = a_i, B = b_j) = 1.$$

A regra fundamental também pode ser aplicada a variáveis:

**Teorema: A regra fundamental para variáveis.**

$$P(A, B) = P(A|B) * P(B)$$

e condicionada a outra variável  $C$ :

$$P(A, B|C) = P(A|B, C) * P(B|C)$$

A fim de se obter a probabilidade isolada de cada possível estado de uma variável tendo-se as probabilidades conjuntas entre variáveis, é necessário apenas somar as probabilidades dos estados das demais variáveis relativos a cada estado da variável que se deseja isolar. Essa operação é chamada de **marginalização**:

$$P(A = a_i) = \sum_{j=1}^n P(A = a_i, B = b_j)$$

Pode-se dizer que a probabilidade de  $P(A)$  foi marginalizada de  $P(A, B)$ . A notação

adequada é:

$$P(A) = \sum_B P(A, B)$$

Com isso, já que  $P(A, B) = P(A|B) * P(B)$  e que  $P(A) = \sum_B P(A, B)$ , pode-se chegar à Regra de Bayes para Variáveis apenas com as probabilidades conjuntas  $P(A, B)$ .

Sumarizando:

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)} = \frac{P(A, B)}{\sum_B P(A, B)}$$

### 1.3.5 Redes Causais e d-separação

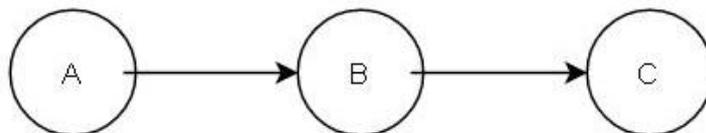
**Redes Causais** consistem de um conjunto de variáveis com uma relação de causalidade entre si. Elas podem ser vistas como grafos direcionados nos quais a direção das arestas denota a direção da causalidade entre os nodos a que estão ligadas. Se uma aresta sai do nodo  $N_0$  apontando na direção do nodo  $N_1$ ,  $N_0$  é um nodo pai<sup>5</sup> de  $N_1$  e  $N_1$  um nodo filho de  $N_0$ .

Em uma **Rede Causal** a crença em torno das variáveis pode influenciar na crença sobre as outras de acordo com suas conexões. Essas influências devem respeitar algumas limitações de acordo com seu tipo de relação. A seguir serão apresentadas algumas explicações considerando cada tipo de conexão.

#### Conexão em Série

Considerando três variáveis  $A$ ,  $B$  e  $C$ ,  $A$  tendo uma relação causal com  $B$  e  $B$  tendo uma relação causal com  $C$ , como mostrado na figura 1.1, essas são chamadas de conexões em série. Nessa configuração, uma evidência sobre o estado de  $A$  influencia

Figura 1.1: Grafo direcionado representando conexões em série



no estado de  $B$  e esse, por consequência, influencia no estado de  $C$ . Da mesma forma, uma evidência sobre  $C$  influencia  $A$  através de  $B$ . Porém, se  $B$  for **instanciada** (variável

<sup>5</sup>parent foi traduzido para pai, embora o termo original em inglês sirva tanto para pai quanto para mãe.

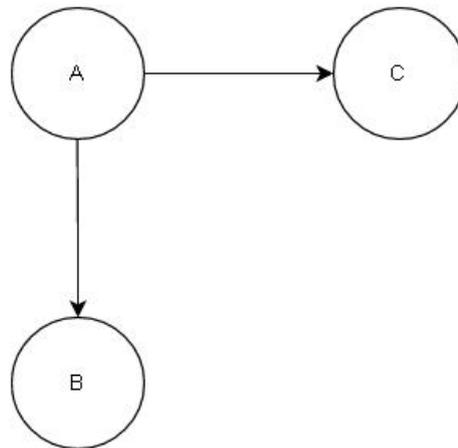
que tem seu estado conhecido, também chamada de variável com evidência sólida<sup>6</sup>) a influência entre  $A$  e  $C$  é interrompida. Assim,  $A$  e  $C$  passam a ser chamados de d-separados ou de independentes.

Portanto, podemos chegar à conclusão de que, em uma conexão em série, uma evidência pode ser transmitida através de uma variável, a não ser que essa esteja instanciada.

### *Conexão Divergente*

Considerando três variáveis  $A$ ,  $B$  e  $C$ ,  $A$  tendo uma relação causal com  $B$  e  $C$ , essas são chamadas de conexões divergentes:

Figura 1.2: Grafo direcionado representando conexões divergente



Nessa configuração, uma evidência sobre o estado de  $A$  influencia no estado de  $B$  e de  $C$ . Da mesma forma, evidências sobre  $B$  e  $C$  influenciam seus estados mutuamente através de  $A$ . Porém, se  $A$  for **instanciada**, a influência entre  $B$  e  $C$  é interrompida. Assim,  $B$  e  $C$  passam a ser d-separados ou independentes.

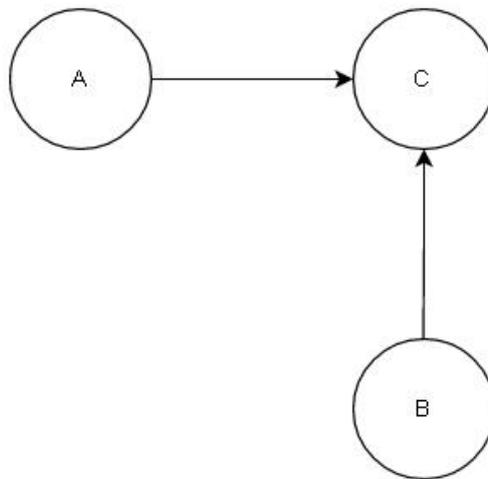
Portanto, podemos chegar à conclusão de que, em uma conexão divergente, uma evidência pode ser transmitida através de um nodo pai até outros nodos filhos, a não ser que este esteja instanciado.

### *Conexão Convergente*

Considerando três variáveis  $A$ ,  $B$  e  $C$ ,  $A$  e  $B$  tendo uma relação causal com  $C$ , essas são chamadas de conexões convergentes:

<sup>6</sup>Em tradução livre do termo *hard evidence*

Figura 1.3: Grafo direcionado representando conexões convergente



Nessa configuração, evidências sobre os estados de  $A$  e de  $B$  influenciam no estado de  $C$  e evidências sobre  $C$  influenciam o estado destes. Porém, evidências sobre  $A$  e  $B$  não influenciam seus estados mutuamente através de  $C$  a não ser que  $C$  ou um de seus descendentes seja instanciado.

Portanto, podemos chegar à conclusão de que, em uma conexão convergente, uma evidência pode ser transmitida a nodos com um filho em comum, apenas se esse ou um de seus descendentes estiver instanciado.

#### *Conexões em Série, Convergente, Divergente: Resumo*

*Duas variáveis  $A$  e  $B$  são d-separadas se para todos os caminhos entre  $A$  e  $B$  existe alguma variável intermediária  $V$ , além de  $A$  e  $B$ , tal que:*

- a conexão é serial ou divergente e  $V$  é instanciada.*
- a conexão é convergente e nem  $V$  nem nenhum de seus descendentes recebeu evidência.*

#### *Moralização*

Em um grafo, a tarefa de definir se um par de variáveis é d-separado pode ser muito complexa se for necessário considerar todos os tipos de conexões entre essas. Considerando as variáveis  $A$  e  $B$ , e o conjunto de variáveis  $C$ , para as quais se tem evidência sólida, uma maneira mais simples de definir a d-separação entre  $A$  e  $B$  seria seguir os seguintes passos:

- Montar um grafo contendo  $A$ ,  $B$ , todas as variáveis de  $C$  e todas as variáveis ligadas diretamente a  $A$ ,  $B$  ou alguma variável de  $C$ . Esse grafo é chamado de **grafo ancestral**.
- Conectar todos os nodos do grafo ancestral que têm filhos em comum.
- Remover a direção de todas as arestas do grafo, tornando-o um grafo não direcionado.

Se todos os caminhos entre  $A$  e  $B$  passarem por algum nodo de  $C$ , então  $A$  e  $B$  são d-separados. Esse processo é chamado de **moralização**.

A moralização pode ser generalizada para trabalhar com conjuntos de variáveis, ao invés de apenas com variáveis individuais. Para os conjuntos de variáveis  $A$  e  $B$ , executa-se o mesmo processo descrito anteriormente, caso não haja caminhos entre qualquer variável contida em  $A$  e qualquer variável contida em  $B$ , então esses são conjuntos d-separados.

### 1.3.6 Redes Bayesianas

#### *Definição de Redes Bayesianas*

Um grafo ou rede causal pode ter definida entre suas ligações a noção de força da ligação. Essa força pode ser representada por um número e é bastante intuitivo tratá-la como uma probabilidade condicional. Para ilustrar essa ideia será usada como exemplo a figura 1.3. Nesse caso teria-se duas probabilidades condicionais:  $P(C|A)$  e  $P(C|B)$ . Como ambas as probabilidades condicionais relacionam-se a  $C$ , é adequado representar tais probabilidades como uma única:  $P(C|A, B)$ . Outro ponto a se observar é que já que  $A$  e  $B$  não têm pais, suas tabelas de probabilidade são representadas apenas por suas probabilidades individuais,  $P(A)$  e  $P(B)$ .

#### *Regra da Cadeia para Redes Bayesianas*

Como mostrado em 1.3.4, tendo a tabela de probabilidades conjuntas de um conjunto de variáveis, pode-se chegar às probabilidades condicionais, dada uma evidência e as probabilidades individuais. Entretanto, trabalhar com todo o conjunto de probabilidade conjuntas entre variáveis pode se tornar impraticável, uma vez que o número de probabilidades cresce exponencialmente com o número de variáveis. Portanto, é conveniente

ter uma forma mais compacta de representar a tabela de probabilidades conjuntas de um conjunto de variáveis.

Para a distribuição de probabilidades sobre um conjunto de variáveis temos uma equação chamada *regra da cadeia*.

**Regra da Cadeia Geral:** Consideremos o conjunto de variáveis  $U = \{A_1, A_2, \dots, A_n\}$ . Para a distribuição probabilística  $P(U)$  temos:

$$P(U) = P(A_n|A_1, A_2, \dots, A_{n-1})P(A_{n-1}|A_1, A_2, \dots, A_{n-2})\dots P(A_2|A_1)P(A_1)$$

Para Redes Bayesianas essa regra tem uma forma especial:

**Regra da Cadeia para Redes Bayesianas:** Consideremos uma Rede Bayesiana BN sobre as variáveis  $U = \{A_1, A_2, \dots, A_n\}$ . Assim, BN especifica uma distribuição probabilística conjunta única  $P(U)$  dada pelo produto de todas as tabelas de probabilidades condicionais de BN:

$$P(U) = \prod_{i=1}^n P(A_i|pa(A_i))$$

onde  $pa(A_i)$  denota os pais de  $A_i$  em BN e  $P(U)$  reflete as probabilidades de BN.

A regra da cadeia demonstra que uma Rede Bayesiana é uma representação compacta da distribuição de probabilidades conjuntas.

### *Inserindo Evidências*

Assumindo que  $A$  é uma variável com  $n$  possíveis estados  $(x_1, x_2, \dots, x_n)$  e que obtemos a informação  $e$  de que  $A$  pode estar apenas no estado  $i$  ou no estado  $j$ , temos a distribuição probabilística  $P(A, e) = (0, 0, \dots, x_i, \dots, x_j, \dots, 0, 0)$ . Note que  $P(e)$ , a probabilidade a priori de  $e$ , é obtida marginalizando  $A$  de  $P(A, e)$ . Note também que  $P(A, e)$  é o resultado da multiplicação de  $P(A)$  por  $(0, \dots, 1, \dots, 1, \dots, 0)$  onde as posições com 1's são a  $i$ -ésima e a  $j$ -ésima.

Agora, assumindo que temos a tabela de probabilidades conjuntas  $P(U)$ , a tabela de probabilidade conjunta  $P(U, e)$  é a tabela  $P(U)$  substituindo todas as entradas de  $A$  em qualquer estado diferente de  $i$  e  $j$  por zeros. Isso é o mesmo que multiplicar  $P(U)$  por  $e$ :

$$P(U, e) = P(U).e$$

Note que  $P(e) = \sum_U P(U, e) = \sum_U (P(U).e)$ . Usando a regra da cadeia para Redes Bayesianas nos chegamos ao seguinte teorema.

**Teorema 1.3.1** Sendo  $BN$  uma Rede Bayesiana sobre o conjunto universo  $U$  e  $e_1, e_2, \dots, e_m$  evidências, então

$$P(U, e) = \prod_{A \in U} P(A|pa(A)) \cdot \prod_{i=1}^m e_i$$

e para  $A \in U$ , sendo que  $U \setminus \{A\}$  denota o conjunto  $U$  sem o subconjunto  $\{A\}$ , temos:

$$P(A|e) = \frac{\sum_{U \setminus \{A\}} P(U, e)}{P(e)}$$

### 1.3.7 Entropia

Com um conjunto de  $n$  possíveis eventos com probabilidades  $p_1, p_2, \dots, p_n$ , estamos interessados em saber qual é o tamanho da incerteza de ocorrência de uma determinada saída. A medida de incerteza criada por Shannon (1948) é chamada de entropia e é denotada por  $H$ . A função usada para cálculo de  $H$  é dada por:

$$H = \sum_{i=1}^n p_i * \log p_i$$

## 1.4 Estado da Arte

*Knowledge Tracing* é a tarefa de prever aspectos sobre a próxima interação de um estudante com um sistema, baseado-se principalmente em seu histórico progresso de interações com esse, permitindo a avaliação mais precisa sobre o domínio dele em relação a assuntos específicos (PIECH et al., 2015). As informações resultantes desse processo podem ser usadas para sugerir atividades baseando-se em necessidades individuais do aluno e, dessa forma, otimizar o seu processo de aprendizagem (ANDERSON et al., 1990).

### 1.4.1 CMU General Programming Languages Tutor

Durante as décadas de oitenta e noventa, foram desenvolvidos diversos trabalhos sobre modelagem cognitiva durante a aquisição de conhecimento (ANDERSON, 1982) (CORBETT; KOEDINGER, 1997) (ANDERSON; BOYLE; YOST, 1985) (ANDERSON et al., 1990). Corbett e Anderson (CORBETT; ANDERSON, 1992) desenvolveram um

estudo que obteve muito destaque propondo uma abordagem de modelagem de conhecimento de estudantes aplicada a um tutor para prática e aprendizado de programação (CMU General Programming Languages Tutor) que, através de um conjunto de centenas de regras de escrita de programas (chamado de modelo de estudante ideal), acompanha o estudante durante o desenvolvimento de pequenos programas e estima passo a passo a probabilidade de aprendizado do aluno de cada regra de formação da linguagem (CORBETT; ANDERSON, 1992). O modelo usado por esse trabalho supõe que a cada interação o conhecimento do estudante está evoluindo e assume apenas dois estados para cada regra, aprendido ou não aprendido, não prevendo esquecimento das mesmas. Essa abordagem se propõe a estimar duas probabilidades:

1. A probabilidade de que o aluno soubesse a habilidade antes da última interação, considerando a evidência oferecida por essa (acerto ou erro):

$$P(L_{n-1}|Acerto_n) = \frac{P(L_{n-1}) * (1 - P(S))}{P(L_{n-1}) * (1 - P(S)) + (1 - P(L_{n-1})) * P(G)}$$

$$P(L_{n-1}|Erro_n) = \frac{P(L_{n-1}) * P(S)}{P(L_{n-1}) * P(S) + (1 - P(L_{n-1})) * (1 - P(G))}$$

2. Probabilidade de ter aprendido a habilidade depois da última interação, ou seja, a probabilidade de o aluno saber a habilidade:

$$P(L_n|Evidencia_n) = P(L_{n-1}|Evidencia_n) + ((1 - P(L_{n-1}|Evidencia_n)) * P(T))$$

Os parâmetros usados nessa equação foram estimados empiricamente. São eles: (p(L0)) probabilidade à priori de uma regra estar no estado aprendido antes de qualquer interação do aluno com o sistema, (p(T)) probabilidade de o aluno aprender a regra durante a interação atual, (p(G) - G vem da palavra em inglês Guess) probabilidade de o aluno aplicar uma regra corretamente mesmo sem ter o conhecimento necessário para isso e (p(S) - S vem da palavra em inglês Slip) probabilidade de o aluno cometer um erro mesmo tendo o conhecimento necessário (CORBETT; ANDERSON, 1992).

#### 1.4.2 Uso de Redes Bayesianas

Foram selecionados alguns trabalhos relacionados que se aproximam da nossa proposta, os quais serão apresentados a seguir:

Reye (1996) afirmou que, ao estimar o conhecimento, diferente do que fora proposto por Corbett and Anderson (1992), com o modelo de probabilidade condicional de quatro variáveis, a probabilidade de aprendizado de cada tópico sendo avaliado não deveria ser atualizada independentemente das outras. Definindo que um dado tópico A é um pré-requisito para o tópico B, haveria uma inconsistência ao dizer que o aluno sabe o tópico B sem saber o tópico A. Dessa forma, manter uma ordenação parcial entre os tópicos do domínio sendo tratado é importante, não só por sugerir melhores formas de atualizar as probabilidades, como por oferecer uma garantia maior de consistência no modelo. Sendo assim, fez uso de Redes Bayesianas, uma forma gráfica de projetar modelos probabilísticos baseada no conceito de probabilidade condicional, para fazer a modelagem de conhecimento. Dessa forma foi projetado um modelo no qual, a cada interação do aluno, são atualizadas as probabilidades desse ter habilidades relacionadas à que está sendo trabalhada. Outros tipos de relação mais complexas também podem ser modeladas usando probabilidade condicional, como por exemplo, determinar que duas ou mais habilidades são, ao mesmo tempo, pré-requisito para outra ou determinar que “O estudante que sabe B tem 90% de chance de saber A.” ( $p(\text{sabe}(A) | \text{sabe}(B)) = 0.9$ ). Esse trabalho também questiona a consideração de apenas dois estados (sabe ou não sabe), afirmando que um grau de detalhamento maior sobre a probabilidade de conhecimento de determinado assunto pode ser útil para calibrar uma abordagem de seleção de atividades, em outras palavras, afirma que um aluno tendo probabilidades 0,4 e 0,7 de saber duas habilidades simplesmente estaria no estado de “não sabe”. A diferença de probabilidade entre as duas é alta o suficiente para que, dependendo da abordagem usada, se fizesse uma diferenciação entre a escolha de abordar novamente uma ou a outra (REYE, 1996).

### 1.4.3 Outros Trabalhos Relacionados

Desde a proposta do conceito de *Knowledge Tracing*, diversos trabalhos foram feitos a fim de aprimorá-lo. A seguir, são descritos alguns trabalhos relacionados ao conceito, acompanhados de uma breve descrição de suas abordagens e resultados:

- Chang et al. (2006) propõem o uso de Rede Bayesianas para modelar o conhecimento do aluno e um modelo de ajuda, e assim, testa a efetividade da disponibilização de ajuda ao estudante durante a interação com um sistema tutor. O trabalho sugere que os estudantes têm maior chance de absorver o conhecimento se houver

ajuda disponibilizada pelo sistema.

- Cen, Koedinger and Junker (2007) trabalham com a hipótese de algumas habilidades serem trabalhadas em excesso devido ao fato de o modelo originalmente proposto usar os mesmos valores para os quatro parâmetros para todas as habilidades. O trabalho propõe uma abordagem na qual as diversas habilidades avaliadas pelo sistema recebem conjuntos de parâmetros ajustados através de aprendizagem de máquina usando históricos de interações de estudantes. Como resultado, obtêm uma diminuição no tempo de interação do aluno com o sistema sem diferença significativa na taxa de aprendizado em relação à proposta original.
- Baker, Corbett and Alevan (2008) propõem um novo método para estimar os quatro parâmetros através de análise contextual, oferecendo uma alternativa que supera os problemas de identificabilidade apontados por Beck (2007), no qual diferentes conjuntos de dados se podem ser compatíveis com diferentes combinações dos quatro parâmetros, que na prática representam realidades diferentes .
- Pardos and Heffernan (2010) propõem, com o objetivo de aumentar o nível de individualização da abordagem, três soluções para estimar as probabilidades a priori dos quatro parâmetros. O estudo chegou à conclusão de que a melhor abordagem entre as propostas é uma que estima a probabilidade a priori de conhecimento das habilidades baseado na média de acertos do estudante até o momento. Caso o aluno ainda não tenha interagido com o sistema, é usada a média dos outros alunos como probabilidade inicial.
- Piech et al. (2015) faz uma adaptação radical na ideia de *Knowledge Tracing* usando Redes Neurais Recorrentes. O trabalho sugere que o uso de redes neurais propicia uma representação do domínio sendo tratado, sem que esse tenha que ser codificado por algum especialista. Usando validação cruzada, o estudo demonstra atingir melhores resultados preditivos do que as abordagens anteriores e, com isso, ter dado início a uma linha promissora de pesquisa sobre *Knowledge Tracing*.
- Reddy, Labutov and Joachims (2016) trabalha com recomendação personalizada de atividades usando modelos probabilísticos do conhecimento do estudante. O algoritmo para modelagem proposto não necessita de uma descrição do modelo mental do aluno, pois aprende através das interações desse. Através de uma avaliação empírica, o trabalho conclui que o modelo proposto tem capacidade de predição equiparável a modelos *benchmark*.
- Lindsey and Mozer (2016) faz comparação entre a eficiência de *Bayesian Kno-*

*wledge Tracing* (BKT) (REYE, 1996) e de *Deep Knowledge Tracing* (DKT)(PIECH et al., 2015). Esse trabalho sugere que DKT é mais eficiente que a abordagem original de BKT, porém, propõe melhorias no modelo, como passar a considerar que o aluno pode fazer a transição do estado "sabe" para "não sabe", e afirma que assim os dois tipos modelagem passam a ser equiparáveis. Com isso, o uso de BKT passaria a ser mais vantajoso, visto que essa é uma abordagem mais simples para modelagem do ponto de vista de raciocínio.

Em nosso trabalho, utilizamos a proposta de estruturação do domínio de conhecimento em forma de Rede Bayesiana proposta por Reye (1996), utilizada com extensões também por Reye (2004), Chang et al. (2006), Cen, Koedinger and Junker (2007), Baker, Corbett and Aleven (2008), Pardos and Heffernan (2010).

## 2 TRABALHO DESENVOLVIDO

O propósito deste trabalho é desenvolver uma modelagem para avaliação do conhecimento de estudantes através de sistemas interativos. Essa modelagem está baseada na intercalação dos dois passos descritos a seguir:

- Avaliar o estado estimado de conhecimento do aluno e, baseado nesse, apresentar a ele uma atividade referente à componente de conhecimento<sup>1</sup> da qual se tiver mais incerteza sobre sua competência. As atividades consideradas aqui são apenas perguntas de múltipla escolha, porém, pode-se extrapolar as ideias apresentadas para outros tipos de questão ou até para recomendação de atividades de estudo.
- Atualizar o modelo baseando-se na interação do aluno com a atividade. Os possíveis resultados para o formato de atividade usado no sistema aqui apresentado são apenas acerto ou erro dos exercícios oferecidos. Em caso de extrapolação, já comentada no item anterior, a estimativa de conhecimento de uma componente pode ser aumentada simplesmente pelo fato de o aluno ter lido um material recomendado, por exemplo.

### 2.1 Conceitos Usados na Modelagem

O modelo implementado por este trabalho visa maximizar a quantidade de informação obtida através das interações com o sistema. Para isso, as componentes de conhecimento do domínio a ser avaliadas foram relacionadas de forma a se supor o conhecimento de certas componentes para as quais ainda não se obteve informações baseado em suas relações de pré-requisito com outras componentes de conhecimento. Esse modelo baseia-se fortemente no uso de Redes Bayesianas da maneira proposta em (REYE, 1996) e nas melhoras e extensões desse modelo propostas em (REYE, 2004).

#### 2.1.1 Relacionamento Entre as Componentes de Conhecimento

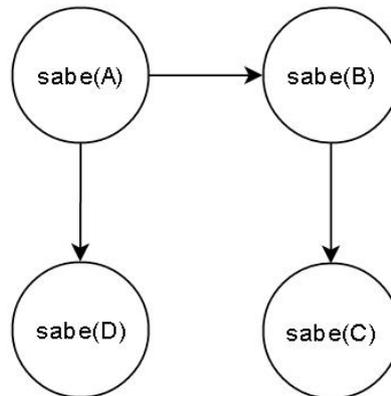
Como já apresentado anteriormente em 1.4.2, as Redes Bayesianas proporcionam um formalismo que tem relação intuitiva com a estruturação do domínio de um tópico,

---

<sup>1</sup>O termo componente de conhecimento representa cada uma das sub-áreas ou assuntos do domínio em questão. Esse termo foi escolhido a fim de manter consistência de nomenclatura com a literatura científica usada de base.

já que, muito frequentemente, alguns conhecimentos são naturalmente pré-requisitos para outros (REYE, 1996). Dessa forma, como proposto em Reye (1996), o sistema define uma ordenação parcial entre as componentes de conhecimento usando probabilidades condicionais a fim de estruturá-los em forma de uma Rede Bayesiana. A estrutura resultante é chamada de espinha dorsal da Rede Bayesiana.

Figura 2.1: Exemplo de topologia de espinha dorsal.



A cada atualização na probabilidade de o aluno conhecer uma dada componente de conhecimento, esta exerce uma influência em outras componentes de conhecimento, direta ou indiretamente relacionadas à componente atualizada. No exemplo contido na figura 2.1, **A** é um pré-requisito para **B** e **D**.

### 2.1.2 Componente de Conhecimento e Probabilidade de Acerto

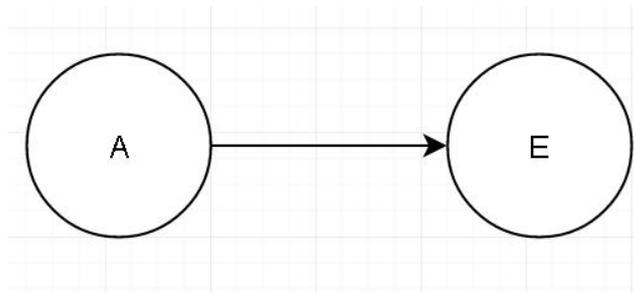
Além da relação causal entre as componentes de conhecimento sendo avaliados pelo sistema, ainda existe uma relação condicional com a probabilidade de acerto ou erro em uma interação, dada a probabilidade de conhecimento de cada componente. Quanto maior a probabilidade de o aluno conhecer a componente de conhecimento, maior a chance de acerto.

A atualização do modelo se dá através da revisão da probabilidade de o aluno saber a componente de conhecimento baseado no evidência obtida:  $P(A = sabe|E)$

O exemplo a seguir será usado para ilustrar a atualização. Nele serão usadas as variáveis  $A = \{sabe, nao\_sabe\}$  e  $E = \{acerto, erro\}$  que são relacionadas de acordo com a figura 2.2.

1. A probabilidade condicional de um acerto, dado que o aluno sabe a componente de

Figura 2.2: Exemplo de relação entre componente de conhecimento e probabilidade de acerto ou erro.



conhecimento é de 0.95 (levando em conta que o aluno possa cometer um engano), isso é,  $P(E = \textit{acerto} | A = \textit{sabe}) = 0.95$ ; e

2. A probabilidade condicional de um acerto, dado que o aluno não sabe a componente de conhecimento é de 0.20 (levando em conta que o aluno possa fazer um "chute" correto), isso é,  $P(E = \textit{acerto} | A = \textit{nao\_sabe}) = 0.20$

Atribuiremos o valor 0.5 à probabilidade a priori de o aluno saber, isso é, não se tem ideia de seu conhecimento prévio. A partir disso, podemos calcular que a probabilidade inicial de o aluno acertar, que é igual à probabilidade de ele acertar sabendo somada à probabilidade de ele acertar não sabendo, é  $0.95 * 0.5 + 0.2 * 0.5 = 0.575$ .

A atualização da probabilidade de o aluno saber é feita aplicando a Regra de Bayes:

**Caso 1:** Se a evidência fornecida for um acerto, a probabilidade revisada de saber A é  $0.95 * 0.5 / 0.575 = 0.82$ .

**Caso 2:** Se a evidência fornecida for um erro, a probabilidade revisada de que ele soubesse A é  $(1 - 0.95) * 0.5 / (1 - 0.575) = 0.058$ .

Os valores obtidos tanto no primeiro quanto no segundo caso influenciaram as probabilidades, porém não até os extremos, pois as probabilidades condicionais levam em conta as possibilidades de "chute" e engano, já citadas anteriormente. Isso vai ao encontro do objetivo de avaliar o aluno considerando incerteza diante às evidências oferecidas por esse.

## 2.2 Domínio de Conhecimento e Estrutura Topológica

O domínio de conhecimento modelado pelo sistema foi inspirado no currículo da disciplina de Algoritmos e Programação (INF01202) ministrada no Instituto de Infor-

mática da Universidade Federal do Rio Grande do Sul (UFRGS). Essa é uma disciplina introdutória para os cursos de Engenharia da Computação e Ciência da Computação e apresenta diversos conceitos básicos de programação fazendo uso da Linguagem C. A seguir, são listados os tópicos considerados como componentes de conhecimento:

- Tipos de Dados - O Aluno conhece os tipos de dados primitivos da linguagem e a finalidade de cada um.
- Declaração de Variáveis - O aluno entende a finalidade das variáveis e sabe como declarará-las corretamente usando o tipo apropriado para a finalidade desejada.
- Ponteiros - O aluno sabe declarar ponteiros, como usá-los e a diferença desses para as variáveis normais.
- Comandos de Seleção - O aluno entende como os comandos if, if-else e switch afetam o fluxo do programa e sabe como usá-los.
- Comandos Iterativos - O aluno entende como os comandos while, do-while e for afetam o fluxo do programa, sabe escolher o comando correto para a situação desejada e sabe como aplicá-los.
- Tipos Estruturados Básicos - O aluno entende o conceito de arranjos e sabe defini-los corretamente, inserir elementos e acessar elementos.
- Subprogramação - O aluno entende o conceito de modularidade e sabe fazer definição e chamadas de funções com passagem de parâmetros por valor. Também entende o conceito de variável local ou global.
- Recursividade - O aluno entende a lógica das chamadas recursivas, sabe como são contruídos algoritmos simples usando recursão e entende como são definidas as condições de parada.
- Estruturas - O aluno sabe definir e entende a utilidade de tipos compostos complexos.

Esses componentes de conhecimento foram relacionados em forma de pré-requisitos resultando na estrutura em árvore apresentada na figura 2.3. Essas relações são listadas e justificadas a seguir:

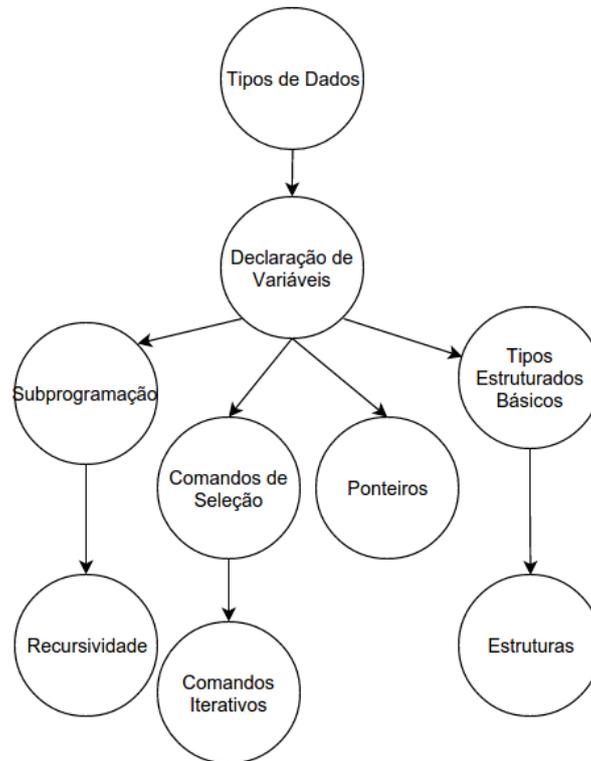
- 'Tipos de Dados' pré-requisito para 'Declaração de Variáveis': toda variável usada na linguagem C deve ser declarada com um tipo definido, dessa forma, a declaração de variáveis é dependente do conhecimento dos tipos de dados primitivos e suas finalidades.

- 'Declaração de Variáveis' é pré-requisito para 'Tipos Estruturados Básicos': os arranjos são listas de posições de memória subsequente referenciadas por identificadores, muito semelhantes às variáveis comuns. Dessa forma, é conveniente conhecer 'Declaração de Variáveis' antes de 'Tipos Estruturados Básicos'.
- 'Declaração de Variáveis' é pré-requisito para 'Ponteiros': Ponteiros são um tipo especial de variáveis que referenciam endereços de memória. Dessa forma, é necessário entender a relação que variáveis têm com endereços de memória para que se possa aprender ponteiros.
- 'Declaração de Variáveis' é pré-requisito para 'Comandos de Seleção': os comandos de seleção executam comandos dependendo de testes. Esses testes geralmente são feitos usando variáveis, 'Declaração de Variáveis' é um conhecimento que naturalmente deve ser obtido antes de 'Comandos de Seleção'.
- 'Declaração de Variáveis' é pré-requisito para 'Subprogramação': a definição de uma função em linguagem C segue um padrão semelhante ao da declaração de uma variável. Além disso, as funções podem receber variáveis como parâmetros ou retornar variáveis. Sendo assim, para a compreensão de 'Subprogramação', é necessário o conhecimento sobre 'Definição de Variáveis'.
- 'Tipos Estruturados Básicos' é pré-requisito para 'Estruturas': antes de criar tipos estruturados complexos, é importante conhecer tipos estruturados básicos, pois através do entendimento desses o aluno passa a entender que é possível fazer a definição de novos tipos através de composição.
- 'Comandos de Seleção' é pré-requisito para 'Comandos Iterativos': através da aprendizagem de comandos de seleção, o aluno passa a ter a noção de que pode manipular o fluxo de um programa. Como esse é um tipo de comando mais simples, passa a ser natural compreendê-lo antes de comandos iterativos.
- 'Subprogramação' é pré-requisito para 'Recursão': a recursão acontece quando uma função faz chamadas a si própria, por isso, é necessário saber criar e manipular funções para o uso da recursão.

É importante ressaltar que com essa topologia nunca haverá d-separação entre dois nodos responsáveis por probabilidades de componentes de conhecimento. Isso se dá devido a dois fatos:

- As relações entre todos os pares de nodos se dão apenas através da composição de conexões divergentes ou em série.

Figura 2.3: Espinha dorsal do modelo de conhecimento para linguagem C



- As evidências não são adicionadas diretamente no nodo da componente de conhecimento, mas sim nos nodos responsáveis por estimar a probabilidade de acerto para cada uma dessas.

O fato de não haver d-separação entre os nodos faz com que qualquer nodo que tenha sua probabilidade revisada cause alterações em todos os outros nodos da rede.

### 2.3 Descrição da Implementação

A implementação deste trabalho obteve como resultado um sistema simples que permite demonstração e avaliação do comportamento da modelagem feita. Devido a tal simplicidade, seria conveniente que os artefatos de software resultantes desse trabalho sejam integrados a outros sistemas mais maduros.

### 2.3.1 Valores Usados nas Tabelas de Probabilidades

As probabilidades a priori de o aluno saber as componentes de conhecimento dos nodos raiz foram inicializadas com o valor 0.5. Já que nada se sabe sobre estes, há igualmente chance de saber ou não os assuntos tratados:

$$P(sabe_0) = 0.5$$

Dessa forma, as probabilidades a priori dos nodos raiz são representadas pela seguinte tabela:

<i>sabe</i>	<i>nao_sabe</i>
0.5	0.5

As probabilidades de acerto dado que o aluno sabe cada assunto  $N$  tratado receberam o valor 0.9, dando margem de 10% de chance de erro para casos de engano:

$$P(E_n = acerto | N = sabe) = 0.9$$

Levando em conta que o sistema tem apenas questões que múltipla escolha que oferecem 4 opções, a chance de um "chute" certo é de 25%, portanto, as probabilidades de acerto, dado que o aluno não sabe o assunto tratado, receberam o valor 0.25:

$$P(E_n = acerto | N = nao_sabe) = 0.25$$

Dessa forma, as tabelas de probabilidades condicionais dos nodos que recebem as evidências de interação do usuário são:

	<i>sabe</i>	<i>nao_sabe</i>
<i>acerto</i>	0.9	0.25
<i>erro</i>	0.1	0.75

Para todo par de componentes de conhecimento  $A$  e  $B$ , nos quais  $A$  é pré-requisito para  $B$ , deseja-se modelar duas características:

- Se o aluno sabe  $A$ , tem grande chance de saber  $B$ . O valor escolhido para representar essa probabilidade foi 0,7, logo:

$$P(B = sabe | A = sabe) = 0.7$$

- Por outro lado, o fato de o aluno não saber A não deve ser um total impedimento para que saiba B, já que nem sempre o conhecimento é obtido de forma linear de acordo com os pré-requisitos. Portanto, mesmo que o aluno não saiba A, ainda existe uma pequena probabilidade de que saiba B. O valor escolhido para representar essa probabilidade foi 0,2, logo:

$$P(B = sabe|A = nao_sabe) = 0.2$$

Com isso chegamos à seguinte tabela de probabilidades:

	<i>sabe<sub>A</sub></i>	<i>nao_sabe<sub>A</sub></i>
<i>sabe<sub>B</sub></i>	0.7	0.2
<i>nao_sabe<sub>B</sub></i>	0.3	0.8

### 2.3.2 Seleção de Componente de Conhecimento

A fim minimizar a quantidade de incerteza sobre o conhecimento de um aluno, este trabalho usa o conceito de entropia para fazer a seleção da componente de conhecimento a ser disponibilizada ao aluno cada interação.

Para mostrar o funcionamento, serão considerados os possíveis estados "*sabe*" e "*não sabe*", onde  $P(sabe)$  é a probabilidade de o aluno saber um certo assunto, e  $P(naosabe)$  a probabilidade de não saber. De acordo com a teoria probabilística  $P(sabe) + P(naosabe) = 1$ , logo  $P(naosabe)$  pode ser representado como  $1 - P(sabe)$ . Assim, podemos expandir a fórmula da entropia<sup>2</sup> para esse caso apenas em termos de  $P(sabe)$ :

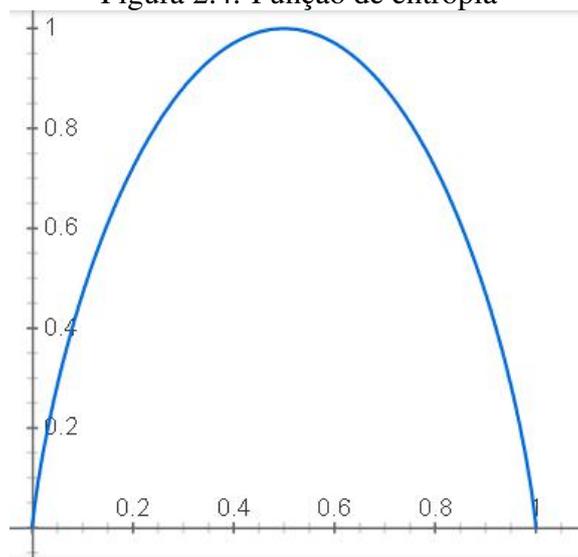
$$-(P(sabe) * \log P(sabe) + (1 - P(sabe)) * \log(1 - P(sabe)))$$

Dessa forma, podemos gerar um gráfico que mostra o comportamento dessa função. Esse gráfico é exibido na figura 2.4.

Nesse gráfico,  $P(sabe)$  é exibido no eixo X e a entropia no eixo Y. Nele podemos perceber que, quanto mais próximo de 0.5, maior é o valor de  $H$ , ou seja, quanto mais uniforme for a distribuição probabilística entre os possíveis estados, mais incerteza se tem sobre o conhecimento do estudante. Sendo assim, é interessante investigar mais sobre os nodos nos quais a entropia é maior. Portanto, a escolha da componente de conhecimento

<sup>2</sup> $H = \sum_{i=1}^n p_i * \log p_i$ , com  $p_i$  sendo a probabilidade do estado  $i$

Figura 2.4: Função de entropia



que será testada, ou seja, o tipo de questão que será dada ao aluno, é feita calculando a entropia de todos os nodos e selecionando a que apresentar maior valor.

### 2.3.3 Soluções de Software Utilizadas

O sistema foi desenvolvido usando o modelo cliente-servidor através do protocolo HTTP. A aplicação servidor foi implementada usando: Java<sup>3</sup>, que é uma linguagem de programação de alto nível que usa orientação a objetos e roda sobre uma máquina virtual, o que permite que seus códigos compilados sejam portáveis entre diversas plataformas, Spring Framework<sup>4</sup>, um framework para desenvolvimento em java que disponibiliza, entre outras coisas, suporte para injeção de dependências, gerenciamento de transações, aplicações web e teste, Postgres<sup>5</sup>, um sistema de banco de dados objeto-relacional *open-source* e Smile<sup>6</sup>, uma biblioteca que facilita a modelagem e inferência usando Redes Bayesianas. A aplicação cliente foi implementada usando: Javascript<sup>7</sup>, que é uma linguagem para programação web, HTML<sup>8</sup> (Hypertext Markup Language), que é uma linguagem de marcação que provê a estruturação da página web, CSS<sup>9</sup> (Cascading Style Sheets),

<sup>3</sup>[https://www.ibm.com/support/knowledgecenter/ssw\\_aix\\_71/com.ibm.aix.performance/java\\_perf\\_mon.htm](https://www.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.performance/java_perf_mon.htm)

<sup>4</sup><https://projects.spring.io/spring-framework/>

<sup>5</sup><http://www.postgresql.org/about/>

<sup>6</sup><http://www.bayesfusion.com/>

<sup>7</sup><https://www.javascript.com/>

<sup>8</sup><https://www.w3.org/standards/webdesign/htmlcss.html>

<sup>9</sup><https://www.w3.org/standards/webdesign/htmlcss.html>

serve para definir a aparência da página web e AngularJS<sup>10</sup>, que é um framework para desenvolvimento em Javascript que facilita o uso de visualizações dinâmicas em aplicações web.

### 2.3.4 Descrição da Interface de Usuário

O sistema conta com uma interface simples. Ao acessá-lo, são requisitados usuário e senha para autenticação. A tela de autenticação é exibida na figura 2.5.

Figura 2.5: Tela de login do sistema



Após a autenticação, o usuário passa a receber as questões a serem respondidas. Na figura 2.6 é mostrado o exemplo da exibição de uma questão. Após selecionar uma opção e clicar do botão "next", o resultado dessa interação é enviado à aplicação *back-end*. As questões são recebidas, uma após a outra, até que o usuário saia do sistema.

### 2.3.5 Arquitetura do Sistema Backend

No framework Spring Web MVC são usadas classes denominadas classes de controle<sup>11</sup>. Essas classes, em linhas gerais, são responsáveis por tratar as requisições HTTP<sup>12</sup>. No diagrama de classes mostrado na figura 2.7, as classes de controle usadas são exibidas à esquerda na cor azul. Outro tipo de classes usado no framework Spring são as classes

<sup>10</sup><https://angularjs.org/>

<sup>11</sup> Chamadas de *controllers* no original em inglês

<sup>12</sup><http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>

Figura 2.6: Exibição de pergunta de múltipla escolha

Qual desses não é um tipo de dado primitivo da linguagem C?

int

char

string

float

NEXT

© eLearning | Design: HTML5 UP

de serviço<sup>13</sup>, exibidas no centro da figura, em amarelo. Essas classes são responsáveis por operações que não pertencem especificamente a nenhum objeto, ou seja, tratam da lógica da aplicação manipulando outros elementos<sup>14</sup>. À direita, em verde, são mostradas as classes representantes de entidades. Essas classes são responsáveis por representar os elementos básicos envolvidos na aplicação, dessa forma, elas refletem, ao nível da linguagem de propósito geral, as principais tabelas do banco de dados.

As sequências de execução mais importantes do sistema são: o registro de uma resposta, que irá atualizar o estado da rede bayesiana do aluno, e a busca da próxima questão a ser trabalhada. Na figura 2.8 é mostrado o diagrama de sequência dessa última.

### 2.3.6 Base de Dados do Sistema

Para a modelagem e manipulação da base de dados foi usado o framework Spring Data JPA. Esse framework permite que a base de dados relacional seja criada apenas usando anotações em classes java. Cada classe que se deseja transformar em uma tabela no banco de dados é anotada com `@Entity`. Isso gera uma tabela de banco de dados com o mesmo nome da classe java. Além disso, cada tabela deve ter um identificador.

<sup>13</sup> Chamadas de *services* no original em inglês

<sup>14</sup> <http://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/stereotype/Service.html>

Figura 2.7: Diagrama de classes resumido da aplicação backend

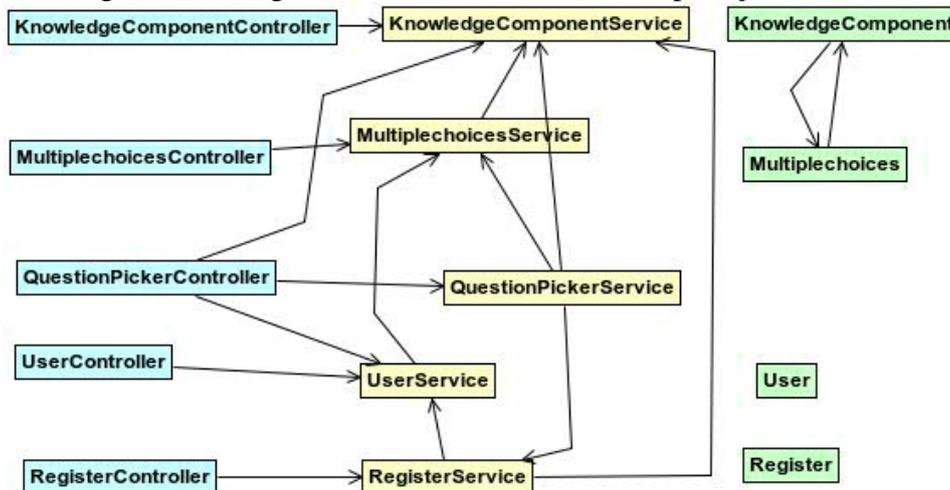
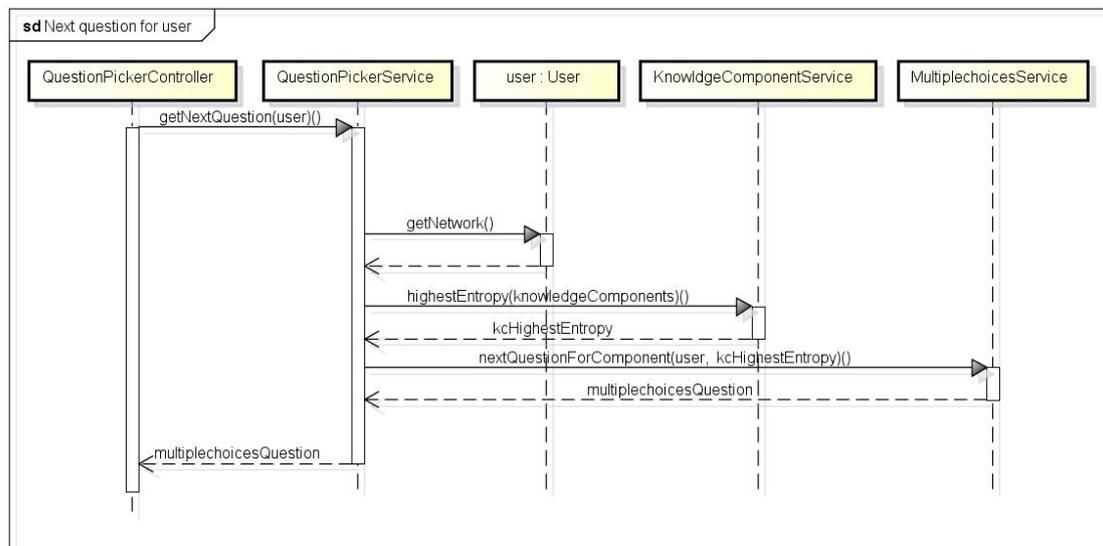


Figura 2.8: Sequência de chamadas na obtenção da próxima questão a ser trabalhada



powered by Astah

Para isso, cada classe a ser mapeada para uma tabela deve ter um atributo de um tipo primitivo com a anotação `@Id`. Esse atributo será a chave primária da tabela. Os relacionamentos entre as tabelas também são definidos através de atributos anotados. No sistema apresentado neste trabalho, por exemplo, existe a classe *KnowledgeComponent*, que representa as componentes de conhecimento, e a classe *Multiplechoices*, que representa as questões do sistema. Cada componente de conhecimento pode ter diversas questões associadas a si, dessa forma, a classe *KnowledgeComponent* tem um atributo do tipo `List<Multiplechoices>` anotado com `@OneToMany`, indicando que cada componente de conhecimento estará relacionada a zero ou mais questões de múltipla escolha. Sendo assim, o sistema é composto pelas entidades apresentadas na figura 2.9 que foram mapeadas

para o banco de dados relacional representado na figura 2.10. .

Figura 2.9: Diagrama de classes das entidades do sistema

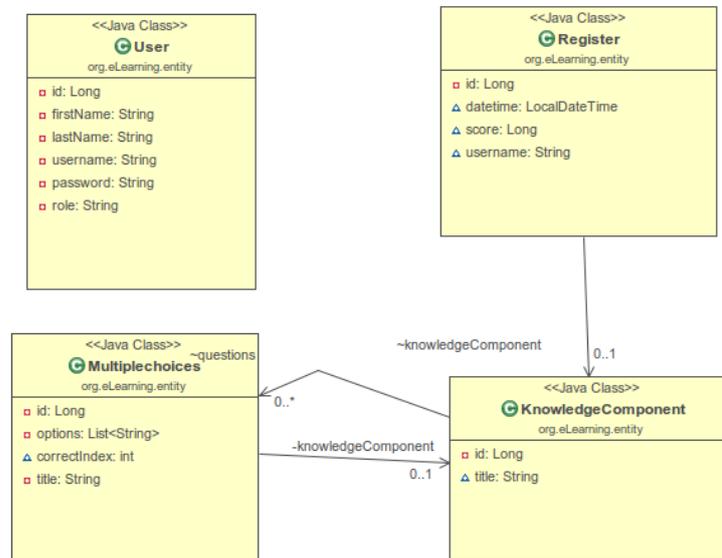
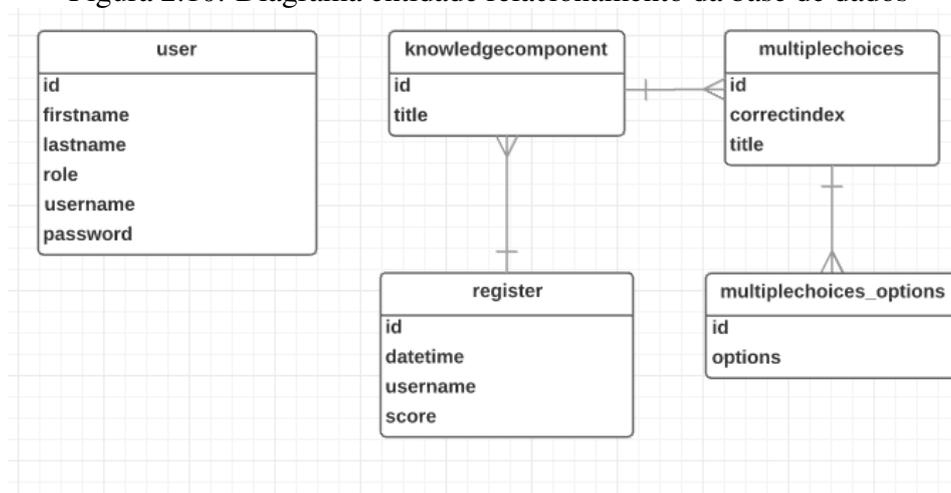


Figura 2.10: Diagrama entidade relacionamento da base de dados



Cabe ressaltar que as redes bayesianas de cada aluno não aparecem na base de dados. Isso se dá devido ao fato de essas, por questões de simplicidade, terem sido registradas em arquivos de texto, funcionalidades já oferecidas pela biblioteca usada para manipulação de redes bayesianas. Outro ponto que cabe ser notado é o fato de não ter sido desenvolvido nenhum *parser* ou interface para alimentar a base de dados. Essa foi abastecida de forma *hardcoded* através de uma classe de inicialização em java.

### 3 DESCRIÇÃO DE EXPERIMENTOS E RESULTADOS

Foram registradas duas simulações de sequências de interação com a Rede Bayesiana modelada. As simulações foram feitas executando a aplicação *back-end* e fazendo as requisições HTTP diretamente via *browser*, não envolvendo assim a aplicação *front-end*. Por isso, as simulações não fizeram uso de exercícios reais, foram feitas apenas inserções de acerto ou erro envolvendo diretamente as componentes de conhecimento. Essas simulações são exibidas nas tabelas 3.1 e 3.2. Nessas tabelas, as colunas **CC** representam cada componente de conhecimento. Nas colunas **P0** são mostradas as probabilidades iniciais de cada componente de conhecimento da rede. Depois dessa, são mostradas colunas com evidências adicionadas ('A' para acerto ou 'E' para erro) seguidas do resultado da adição de tal evidência para cada componente. Dessa forma, cada tabela pode ser vista como uma linha temporal indo da esquerda para a direita.

#### Primeira Simulação

Nessa simulação foram adicionadas apenas evidências de acerto para uma análise simples das influências na rede. Os valores obtidos em cada iteração dessa simulação são exibidos na tabela 3.1. Na figura 3.1 é mostrada uma representação gráfica da rede bayesiana na qual os nodos estão marcados com a ordenação na qual as evidências foram obtidas para essa simulação.

Inicialmente, 'Tipos de Dados', que é pré-requisito para todas as outras componentes de conhecimento, tem nível de incerteza máximo. Todos os outros nodos, à medida que se distanciam desse nodo raiz têm menores valores para  $P(sabe)$  probabilidade, pois tratam de assuntos cada vez mais avançados. Assim que é adicionada uma evidência de acerto para 'Tipos de Dados', todas as probabilidades aumentam, porém, os nodos mais distantes da raiz continuam tendo probabilidades menores.

Em seguida, é adicionada uma evidência de acerto em 'ponteiros'. Esse é um nodo lateral que não é pré-requisito para nenhum outro. Dessa forma, o nodo mais afetado com adição dessa evidência, além dele próprio, é 'Declaração de Variáveis', que é pré-requisito para todos os outros, com exceção de 'Tipos de Dados'. Através dessa alteração em 'Declaração de Variáveis', a evidência adicionada resulta em uma alteração fraca em todo o resto da rede, justamente pelo fato de 'Ponteiros' ter sido considerado como uma área de conhecimento mais isolada, diferente dos outros nodos folha, que tem pelo menos

um nodo a mais como pré-requisito indireto.

Após isso, é adicionada uma evidência para 'Estruturas'. Esse também é um nodo folha, porém, tem como pré-requisito 'Tipos Estruturados Básicos'. Já que a evidência adicionada foi um acerto, as chances de que se saiba esse pré-requisito são aumentadas e esse aumento é propagado para trás nos pré-requisitos, e esses por sua vez influenciam os outros ramos. Esse processo se repetiu até o final da simulação, já que os demais nodos para o qual se recolheu evidências também eram folhas.

Ao final dessa simulação não haviam sido recolhidas evidências para os nodos 'Declaração de Variáveis', 'Tipos Estruturados Básicos', 'Comandos de Seleção' e 'Subprogramação'. Desses nodos, o que apresenta maior probabilidade de o aluno saber é 'Declaração de Variáveis', pois esse é o que tem relação mais próxima com mais nodos que receberam evidências. Dessa forma, mesmo não tendo sido feita coleta de evidência para esse nodo, ter evidências de acerto para 'Ponteiros', 'Estruturas', 'Comandos Iterativos' e 'Recursividade', para os quais esse é direta ou indiretamente pré-requisito, são bons indícios de que o aluno também o sabe.

Tabela 3.1: Primeira Simulação de Execução

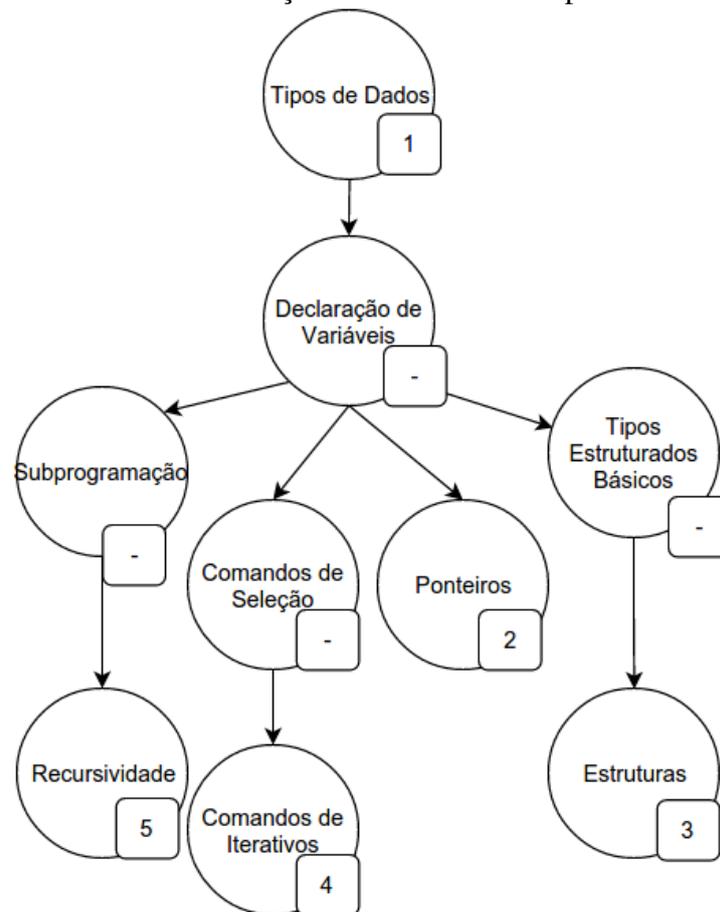
CC	P0	E1	P1	E2	P2	E3	P3	E4	P4	E5	P5
Tipos de Dados	0,5	A	0,783		0,831		0,851		0,868		0,882
Declaração de variáveis	0,45		0,591		0,729		0,786		0,833		0,872
Ponteiros	0,425		0,496	A	0,78		0,804		0,824		0,84
Tipos Estruturados Básicos	0,425		0,496		0,564		0,706		0,73		0,74
Comandos de Seleção	0,425		0,496		0,564		0,593		0,73		0,749
Subprogramação	0,425		0,496		0,564		0,593		0,617		0,749
Estruturas	0,413		0,448		0,482	A	0,77		0,78		0,788
Comandos Iterativos	0,413		0,448		0,482		0,496	A	0,78		0,788
Recursividade	0,413		0,448		0,482		0,496		0,508	A	0,788

## Segunda Simulação

Na figura 3.2 é mostrada uma representação gráfica da rede bayesiana na qual os nodos estão marcados com a ordenação na qual as evidências foram obtidas na segunda simulação.

Nessa simulação foram introduzidos erros nas componentes 'Ponteiros' e 'Recur-

Figura 3.1: Ordem da obtenção das evidências na primeira simulação

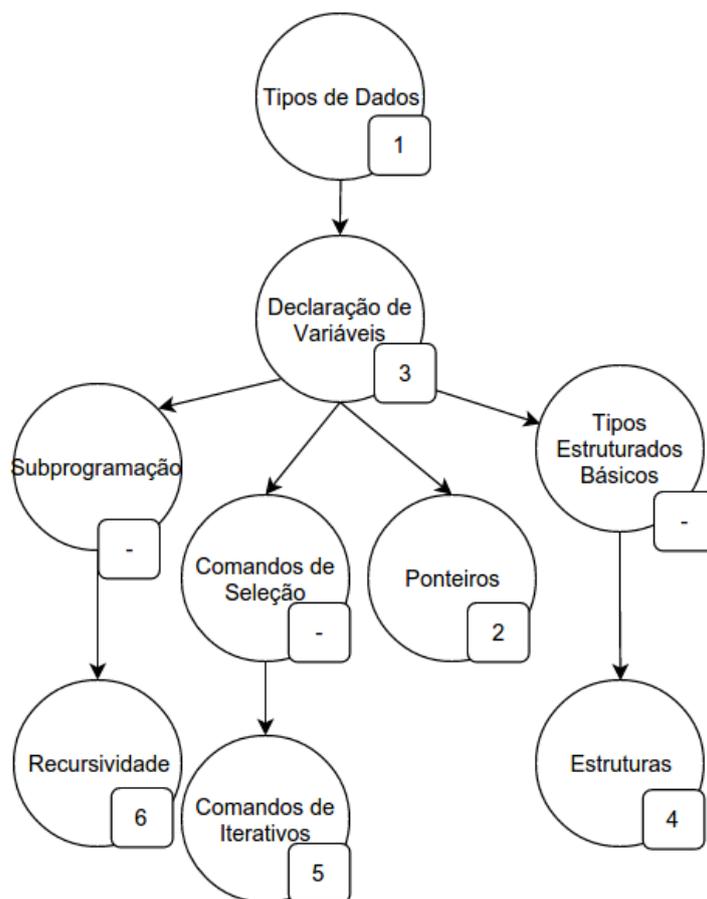


sividade'. O erro da componente 'Ponteiro' ocasionou em uma grande queda na probabilidade de saber 'Declaração de Variáveis', devido à proximidade na relação entre esses nodos. Essa queda teve como consequência uma diminuição nas probabilidades dos nodos dependentes de 'Declaração de Variáveis'. Após isso, o nodo com maior entropia passou a ser o próprio 'Declaração de Variáveis'. Após a coleta de uma evidência de acerto, esse nodo teve uma drástica subida em sua probabilidade, indicando bem a estimativa do modelo de que a limitação desse aluno hipotético está em 'Ponteiros' apenas e não em seu pré-requisito. A adição da evidência de erro para a componente 'Recursividade' ocasionou em uma probabilidade muito menor para a componente 'Subprogramação' quando comparado à simulação anterior. Já a componente 'Comandos de seleção' que tem o mesmo pré-requisito que 'Subprogramação' tem uma probabilidade muito maior do que essa, já que a componente para a qual esta é pré-requisito, 'Comandos Iterativos', tem uma evidência de acerto. Com isso, essa simulação dá indicações de que a modelagem feita é coerente.

Tabela 3.2: Segunda Simulação de Execução

CC	P0	E1	P1	E2	P2	E3	P3	E4	P4	E5	P5	E6	P6
Tipos de Dados	0,5	A	0,78		0,71		0,82		0,84		0,86		0,84
Declaração de variáveis	0,45		0,59		0,408	A	0,71		0,77		0,82		0,76
Ponteiros	0,42		0,49	E	0,11		0,17		0,19		0,20		0,18
Tipos Estruturados Básicos	0,42		0,49		0,404		0,55		0,69		0,72		0,69
Comandos de Seleção	0,42		0,49		0,404		0,55		0,58		0,72		0,69
Subprogramação	0,42		0,49		0,404		0,55		0,58		0,61		0,42
Estruturas	0,41		0,44		0,402		0,47	A	0,76		0,77		0,76
Comandos Iterativos	0,41		0,44		0,402		0,47		0,49	A	0,77		0,76
Recursividade	0,41		0,44		0,402		0,47		0,49		0,50	E	0,12

Figura 3.2: Ordem da obtenção das evidências na segunda simulação



#### 4 CONCLUSÃO E TRABALHOS FUTUROS

Nesse trabalho foi apresentada uma revisão sobre teoria probabilística, redes causais, Redes Bayesianas e conceitos relacionados, que servem de base para o entendimento dos capítulos subsequentes. Também foram revisados trabalhos contidos na literatura acadêmica relacionados ao uso de sistemas automatizados para a tutoria de estudantes. Em seguida, baseado-se na revisão feita anteriormente, foi apresentado o desenvolvimento de uma ferramenta que faz modelagem de conhecimento durante a interação de alunos. Por fim, foram apresentadas simulações com o intuito de demonstrar a coerência das ideias apresentadas em execuções reais. Embora as simulações exibidas na seção 3 mostrem a coerência das ideias apresentadas, em razão da falta de prova formal ou experimentação com alunos, não se pode afirmar de fato que esse sistema consegue fazer uma aproximação satisfatória do estado real de conhecimento. Entretanto, o fato de trabalhos anteriores terem obtido bons resultados (CORBETT; ANDERSON, 1992) (REYE, 2004) (CHANG et al., 2006) (CEN; KOEDINGER; JUNKER, 2007) (BAKER; CORBETT; ALEVEN, 2008) (PARDOS; HEFFERNAN, 2010) com pequenas melhorias no modelo básico proposto em Anderson et al. (1990) indica que o uso de Redes Bayesianas tem ainda potencial a ser explorado.

Na modelagem aqui apresentada, cada nodo representante de uma componente de conhecimento está associado a um nodo representante de sua probabilidade de acerto. Dessa forma, não é possível a adição de mais de uma evidência para a mesma componente. No caso de sistemas de avaliação que acompanham a evolução do aluno durante um período de tempo maior, seria desejável reiterar a avaliação do conhecimento do aluno sobre um mesmo assunto. Sendo assim, como trabalhos futuros, seria pertinente a adaptação dessa modelagem para a aplicação em uma Rede Bayesiana Dinâmica. Isso permitiria a modelagem temporal da evolução do aluno.

Além disso, as componentes de conhecimento aqui apresentadas ainda são bastante abrangente, dessa forma, seria interessante dividir cada uma delas em componentes mais específicas, criando também uma relação de pré-requisitos entre elas. Assim seria possível diagnosticar com mais precisão quais são as deficiências do aluno.

Outro trabalho futuro seria a integração dos artefatos desenvolvidos a sistemas de ensino mais maduros. Isso possibilitaria a aplicação prática, permitindo assim refinar a modelagem feita, tanto do ponto de vista dos parâmetros usado nas relações modeladas, quando das próprias relações. Essa integração também permitiria coleta de dados para

teste e evolução da abordagem proposta usando aprendizagem de máquina, por exemplo, como feito por Cen, Koedinger and Junker (2007).

## REFERÊNCIAS

ANDERSON, J. R. Acquisition of cognitive skill. **Psychological Review**, v. 89, n. 4, p. 369–406, 1982. ISSN 0033-295X.

ANDERSON, J. R. et al. **Cognitive modeling and intelligent tutoring**. 1990. 7–49 p.

ANDERSON, J. R.; BOYLE, C. F.; YOST, G. The geometry tutor. In: **Proceedings of the Ninth International Joint conference on Artificial intelligence**. [s.n.], 1985. v. 1, p. 1–7. ISBN 0-934613-02-8. Available from Internet: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.7590&rep=rep1&ty>>.

BAKER, R. S. J. D.; CORBETT, A. T.; ALEVEN, V. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 5091 LNCS, p. 406–415, 2008. ISSN 03029743.

BECK, J. E. Difficulties in inferring student knowledge from observations ( and why you should care ). **Proceedings of the AIED2007 Workshop on Educational Data Mining**, p. 21–30, 2007.

BLOOM, B. S. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. **Educational Researcher**, v. 13, n. 6, p. 4–16, 1984. ISSN 0013-189X.

CARNOY, M. Education for All and the quality of education : a reanalysis. **Education**, 2005.

CEN, H.; KOEDINGER, K. R.; JUNKER, B. Is over practice necessary? Improving learning efficiency with the Cognitive Tutor through educational data mining. **Frontiers in Artificial Intelligence and Applications**, v. 158, p. 511–518, 2007. ISSN 0922-6389. Available from Internet: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.7340&rep=rep1&ty>>.

CHANG, K.-m. et al. Does Help Help? A Bayes Net Approach to Modeling Tutor Interventions. **Artificial Intelligence**, 2006.

CORBETT, A.; KOEDINGER, K. Intelligent Tutoring Systems. **Handbook of Human-Computer Interaction**, v. 228, n. 4698, p. 2–5, 1997. ISSN 0036-8075.

CORBETT, A. T.; ANDERSON, J. R. Student Modeling and Mastery Learning in a Computer-Based Programming Tutor. **Its '92**, n. November, p. 413–420, 1992.

JANSEN, F. V.; NIELSEN, T. D. **Bayesian Networks and Decision Graphs**. 2<sup>a</sup>. ed. [S.l.: s.n.], 2007. ISBN 9780387682815.

LINDSEY, R. V.; MOZER, M. C. How Deep is Knowledge Tracing ? 2016.

PARDOS, Z. A.; HEFFERNAN, N. T. Modeling individualization in a Bayesian networks implementation of knowledge tracing. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 6075 LNCS, p. 255–266, 2010. ISSN 03029743.

PIECH, C. et al. Deep Knowledge Tracing. p. 1–13, 2015.

REDDY, S.; LABUTOV, I.; JOACHIMS, T. Latent Skill Embedding for Personalized Lesson Sequence Recommendation. 2016.

REYE, J. **A Belief Net Backbone for Student Modelling**. 1996. 596–604 p. Available from Internet: <[http://link.springer.com/chapter/10.1007/3-540-61327-7{\\\_}](http://link.springer.com/chapter/10.1007/3-540-61327-7{\_})>.

REYE, J. Student modelling based on belief networks. **International Journal of Artificial Intelligence in Education**, v. 14, p. 1–33, 2004. ISSN 1560-4292. Available from Internet: <<http://dl.acm.org/citation.cfm?id=1434856>>.

SHANNON, C. E. C. A mathematical theory of communication. **The Bell System Technical Journal**, v. 27, n. July 1928, p. 379–423, 623–656, 1948. ISSN 15591662. Available from Internet: <<http://dl.acm.org/citation.cfm?id=584093>>.

SPOEL, E. van der et al. Association analysis of insulin-like growth factor-1 axis parameters with survival and functional status in nonagenarians of the Leiden Longevity Study. **Aging**, v. 7, n. 11, p. 956–963, 2015. ISSN 19454589.