EDUARDO DE MORAES FURTADO

REPRESENTAÇÃO ALGORÍTMICA DA LÓGICA DE DIÁLOGOS

Trabalho de conclusão de curso apresentado ao Curso de Ciência da Computação.

UNIVERSIDADE PRESIDENTE ANTÔNIO CARLOS

Orientadora: Profa. Lorena Sophia Campos de Oliveira

BARBACENA
2004

EDUARDO DE MORAES FURTADO

REPRESENTAÇÃO ALGORÍTMICA DA LÓGICA DE DIÁLOGOS

Este trabalho de conclusão de curso foi julgado adequado à obtenção do grau de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação da Universidade Presidente Antônio Carlos.

Barbacena – MG, 14 de junho de 2004.

Profa. Lorena Sophia Campos de Oliveira - Orientadora do Trabalho

Prof. Cesário José Ferreira - Membro da Banca Examinadora

Prof. Luís Augusto Mattos Mendes - Membro da Banca Examinadora

AGRADECIMENTOS

Agradeço à minha orientadora pela dedicação, à minha família pela compreensão, e aos amigos pelo apoio dado no transcorrer do curso.

RESUMO

Este trabalho tem como proposta descrever a aplicabilidade do algoritmo, já que o mesmo possibilita uma vasta aplicação nas diversas áreas do conhecimento com a utilização da Lógica Proposicional e da Lógica de Diálogos. Para tal foi realizado um estudo sobre o algoritmo constante na tese de Mestrado da Professora Lorena Campos de Oliveira, conforme anexo A., visando aprimorar e embasar as idéias apresentadas na mesma.

Palavras-chave: Lógica Proposiocional, Lógica de Diálogos, Algoritmo de Diálogos.

SUMÁRIO

LISTAS

1 INTRODUÇÃO

A matemática, por sua beleza e coerência lógica tem sido o campo preferencial de experimentos envolvendo a representação de conhecimentos e de raciocínio humano. Estudos sobre a resolução de problemas matemáticos muito têm contribuído para a compreensão da forma humana de raciocinar e, por conseqüência, serviram como suportes a diversos modelos de sistemas computacionais. Para tal, será utilizado os fundamentos lógicos visando embasar o referencial teórico do Algoritmo de Diálogos.

No capítulo 2, “Lógica Proposicional”, serão definidas as propriedades e aplicações, bem como, os conceitos fundamentais de conectivos lógicos para a validação de um argumento.

Em seguida, no capítulo 3, “Lógica de Diálogos”, os integrantes, Oponente ou Proponente, conhecem todas as regras e critérios do jogo. Com isso, o computador (Proponente) lança uma pergunta (Argumento), e o adversário (Oponente) decide se irá defendê-la ou atacá-la. Quando um dos integrantes não tiver mais como argumentar, vence quem der a “última palavra”. Com isso, será demonstrado as regras de criação e manipulação de um *tableau* de jogos.

Finalmente, será apresentado o estudo realizado no algoritmo existente, conforme anexo A, para uma futura implementação.

.

2 LÓGICA PROPOSICIONAL

Através da aplicação da Lógica Proposicional, tem-se a verificação do conceito semântico de validade a partir de expressões sintáticas da linguagem.

2.1 INTRODUÇÃO

Lógica, de um modo geral, consiste em ser a arte de pensar corretamente. Nem sempre se raciocina da maneira correta, às vezes, é tomada uma decisão ao invés de outra, agindo de maneira ilógica. Na verdade se deve raciocinar de forma tranqüila, paciente e principalmente correta, e é neste ponto que é encontrado a lógica como:

> "Uma ciência que estuda as leis e critérios de validade que regem o pensamento e a demonstração, ou seja, ciências dos princípios formais do raciocínio". [GAB84]

2.2 CÁLCULO PROPOSICIONAL

A lógica proposicional é o estudo de argumentos. Um argumento é uma seqüência de enunciados, na qual um dos enunciados é a conclusão, derivado a partir dos outros enunciados, chamados premissas.

O objetivo do cálculo proposicional é a verificação da validade de certas formas de argumento. Um argumento é válido se todas as suas instâncias são válidas, ou seja, é impossível que sua conclusão seja falsa quando suas premissas são verdadeiras; e inválido se pelo menos uma de suas instâncias é inválida.

O silogismo disjuntivo, por exemplo, é uma forma de argumento válido. Se as premissas do argumento forem verdadeiras, então a conclusão será obrigatoriamente verdadeira. Se as premissas não forem verdadeiras, contudo, mesmo se a forma de argumento for válida, nada se pode afirmar a respeito da veracidade da conclusão. Assim os argumentos podem ser representados simbolicamente como:

Todo amigo de Carlos é amigo de Jonas.
<u>Pedro não é amigo de Jonas.</u>
Logo, Pedro não é amigo de Carlos.

$(P(x,c) \rightarrow P(x,j))$

$\neg \underline{P(p,j)}$

$\neg P(p,c)$

Onde P(x,y) significa que x é amigo de y e c, p, j são constantes que representam Carlos, Pedro e Jonas respectivamente.

O Cálculo Proposicional fornece um modo de provar a validade de qualquer forma de argumento válido. Entretanto, não fornece métodos para se determinar a validade das premissas, estes aspectos da avaliação de um argumento devem ser tratados por outros meios.

O processo de formalização de um argumento consiste na conversão de uma sentença declarativa em uma estrutura composta de letras sentenciais e de operadores lógicos.

As letras sentenciais não têm, por si próprias, significado; mas no contexto de um problema podem ser interpretadas como proposições ou enunciados definidos. Todas as fórmulas do cálculo proposicional são construídas a partir de três conjuntos de símbolos, que formam o vocabulário do cálculo proposicional:

1. **Letras sentenciais**

Qualquer letra maiúscula é considerada uma letra sentencial. Uma letra sentencial representa uma afirmativa que tem um valor-verdade, ou seja, pode ser verdadeira ou falsa.

2. **Operadores lógicos**

Os operadores lógicos, são os seguintes: $\neg$ (negação), $\wedge$ (conjunção), $\vee$ (disjunção), $\rightarrow$ (condicional) e $\leftrightarrow$ (bicondicional).

3. **Parênteses**

(e), que servem para agrupar fórmulas do cálculo proposicional de maneira a formar outras fórmulas.

Uma fórmula do cálculo proposicional é uma seqüência de elementos do seu vocabulário.

As fórmulas da linguagem são descritas pelas seguintes regras:

- Todo símbolo proposicional de Φ é uma fórmula, chamada fórmula atômica.
- Se Φ é uma fórmula, então $\neg\,\Phi$ também é uma fórmula.
- Se Φ e Ψ são duas fórmulas, então $(\Phi \wedge \Psi)$, $(\Phi \vee \Psi)$, $(\Phi \rightarrow \Psi)$ também são fórmulas.

Nada mais é uma fórmula, a não ser que seja atendida por um dos itens acima.

2.3 PRECEDÊNCIA E OPERAÇÕES LÓGICAS

A utilização da ordem de precedência dos conectivos proposicionais permite a simplificação das fórmulas, sendo a seguinte:

- Maior precedência: $\neg$ (negação)
- Média precedência: $\rightarrow$ (condicional) , $\leftrightarrow$ (bicondicional)
- Menor precedência: $\wedge$ (conjunção) , $\vee$ (disjunção)

> "Na lógica proposicional não se tem o significado dos conectivos considerados isoladamente. A interpretação das fórmulas é feita a partir de um conjunto de regras semânticas, obtidas dos significados semânticos dos símbolos, de verdade e dos conectivos proposicionais". [SOU02]

2.4 OPERAÇÕES LÓGICAS SOBRE PROPOSIÇÕES

Quando se pensa são efetuadas certas operações sobre proposições chamadas operações lógicas. Estas, obedecem as regras do cálculo proposicional semelhante ao da aritmética sobre números. Desta forma, são definidas as seguintes operações lógicas fundamentais:

Negação ($\neg$): chama-se negação de uma proposição **p** a proposição representada por "**não p**", cujo valor lógico é a verdade (V) quando **p** é falsa e a falsidade (F) quando **p** é verdadeira. Assim, "**não p**" tem o valor lógico oposto daquele de **p**. Simbolicamente, a negação de **p** indica-se com a notação "**$\neg$ p**", que se lê: "**não p**".

O valor lógico da negação de uma proposição é, portanto, definido pela seguinte tabela-verdade:

Tabela 1 – Negação:

p	**¬ p**
V	F
F	V

ou seja, pelas igualdades: ¬V = F, ¬F = V

Exemplos*:*

p: 2 + 3 = 5 (V) e **¬ p**: 2 + 3 ≠ 5 (F)

¬ **V** = F

q: 7 < 3 (F) e ¬ **q**: 7 < 3 (V)

¬ **F** = V

r: Roma é a capital da França (F) e ¬ **r**: Roma não é a capital da França (V)

Na linguagem comum a negação antepõe o advérbio "**não**" ao verbo da proposição dada. Assim, por exemplo, a negação da proposição:

p: O Sol é uma estrela é ¬ **p**: O Sol **não** é uma estrela.

Outra maneira de efetuar a negação consiste em antepor à proposição dada expressões tais como " **não é verdade que**", " **é falso que**".

Assim, por exemplo, a negação da seguinte proposição:

q: Carlos é mecânico.

É demonstrada conforme abaixo:

¬ **q**: **Não é verdade** que Carlos é mecânico.

ou

¬ **q**: **É falso** que Carlos é mecânico.

Observe, entretanto, que a negação de “**Todos os homens são elegantes**” é “**Nem todos os homens são elegantes**” e a de “**Nenhum homem é elegante**” é “**Algum homem é elegante**”.

Conjunção ($\wedge$) : chama-se conjunção de duas proposições **p** e **q** a proposição representada por “**p e q**”, cujo valor lógico é a verdade (V) quando as proposições **p** e **q** são ambas verdadeiras e a falsidade (F) nos demais casos. Simbolicamente, a conjunção de duas proposições **p** e **q** são indicadas com a notação: “**p $\wedge$ q**”, que se lê: “**p e q**”.

O valor lógico da conjunção de duas proposições é, portanto, definido pela seguinte tabela-verdade:

Tabela 2 – Conjunção:

p	**q**	**p $\wedge$ q**
V	V	V
V	F	F
F	V	F
F	F	F

ou seja, pelas igualdades:

V $\wedge$ V = V,   V $\wedge$ F = F,   F $\wedge$ V = F, F $\wedge$ F = F   e V(**p** $\wedge$ **q**) = V(**p**) $\wedge$ V(**q**).

Exemplos:

p: A neve é branca (V)

q: 2 < 5 (V)

p $\wedge$ q: A neve é branca e 2 < 5 (V)

V(**p**$\wedge$**q**) = V(**p**) $\wedge$ V(**q**) = V $\wedge$ V = V

p: Viena é capital do Paraná (F)

q: 7 é um número primo (V)

$\mathbf{p} \wedge \mathbf{q}$: Viena é capital do Paraná e 7 é um número primo (F)

$V(\mathbf{p}\wedge\mathbf{q}) = V(\mathbf{p}) \wedge V(\mathbf{q}) = F \wedge V = F$

Disjunção ($\vee$) : chama-se disjunção de duas proposições **p** e **q** a proposição representada por "**p ou q**", cujo valor lógico é verdade (V) quando ao menos uma das proposições **p** e **q** é verdadeira e a falsidade (F) quando as proposições **p** e **q** são ambas falsas. Simbolicamente, a disjunção de duas proposições **p** e **q** são indicadas com a notação: "$\mathbf{p} \vee \mathbf{q}$", que se lê: "**p ou q**".

O valor lógico da disjunção de duas proposições é, portanto, definido pela seguinte tabela-verdade:

Tabela 3 – Disjunção:

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

ou seja, pelas igualdades:

$V \vee V = V$, $V \vee F = V$, $F \vee V = V$, $F \vee F = F$ e $V(\mathbf{p} \vee \mathbf{q}) = V(\mathbf{p}) \vee V(\mathbf{q})$.

Exemplos:

p: Paris é a capital da França (V)

q: 9 – 4 = 5 (V)

$\mathbf{p} \vee \mathbf{q}$: Paris é a capital da França ou 9 – 4=5 (V)

$V(\mathbf{p}\vee\mathbf{q}) = V(\mathbf{p}) \vee V(\mathbf{q}) = V \vee V = V$

p: Camões escreveu os Lusíadas (V)

q: $\pi = 3$ (F)

$\mathbf{p} \vee \mathbf{q}$: Camões escreveu os Lusíadas ou π=3 (V)

$V(\mathbf{p}\vee\mathbf{q}) = V(\mathbf{p}) \vee V(\mathbf{q}) = V \vee F = V$

Condicional (→) : chama-se proposição condicional ou apenas condicional uma proposição representada por "**se p então q**", cujo valor lógico é a falsidade (F) no caso em que **p** é verdadeira e **q** é falsa, e verdadeira (V) nos demais casos. Simbolicamente, a condicional de duas proposições **p** e **q** são indicadas com a notação: "$\mathbf{p} \rightarrow \mathbf{q}$", que também se lê das seguintes maneiras:

(**i**) **p** é condição suficiente para **q**;

(**ii**) **q** é condição necessária para **p**.

Na condicional "$\mathbf{p} \rightarrow \mathbf{q}$", diz-se que **p** é o antecedente e **q** o conseqüente. O símbolo "→" é chamado símbolo de implicação.

O valor lógico da condicional de duas proposições é, portanto, definido pela seguinte tabela-verdade:

Tabela 4 – Condicional:

p	q	p → q
V	V	V
V	F	F
F	V	V
F	F	V

ou seja, pelas igualdades:

$V \rightarrow V = V$, $V \rightarrow F = F$, $F \rightarrow V = V$, $F \rightarrow F = V$ e $V(\mathbf{p} \rightarrow \mathbf{q}) = V(\mathbf{p}) \rightarrow V(\mathbf{q})$.

Portanto, uma condicional é verdadeira todas as vezes que o seu antecedente é uma proposição falsa.

Exemplos:

p: O mês de Maio tem 31 dias (V)

q: A Terra é plana (F)

p → q: Se o mês de Maio tem 31 dias, então a Terra é plana (F)

V(**p → q**) = V(**p**) → V(**q**) = V → F = F

p: Santos Dumont nasceu no Ceará (F)

q: O ano tem 9 meses (F)

p → q: Se Santos Dumont nasceu no Ceará, então o ano tem 9 meses (V)

V(**p → q**) = V(**p**) → V(**q**) = F → F = V

Bicondicional (↔) : chama-se proposição bicondicional ou apenas bicondicional uma proposição representada por "**p se e somente se q**", cujo valor lógico é a verdade (V) quando **p** e **q** são ambas verdadeiras ou ambas falsas, e a falsidade (F) nos demais casos. Simbolicamente, a bicondicional de duas proposições **p** e **q** são indicadas com a notação**:** **p ↔ q**, que também se lê das seguintes maneiras:

(**i**) **p** é condição necessária e suficiente para **q**;

(**ii**) **q** é condição necessária e suficiente para **p**.

O valor lógico da bicondicional de duas proposições é, portanto, definido pela seguinte tabela-verdade:

Tabela 5 – Bicondicional:

p	q	p ↔ q
V	V	V
V	F	F
F	V	F
F	F	V

ou seja, pelas igualdades:

V $\leftrightarrow$ V = V, V $\leftrightarrow$ F = F, F $\leftrightarrow$ V = F, F $\leftrightarrow$ F = V e V(**p** $\leftrightarrow$**q**) = V(**p**) $\leftrightarrow$ V(**q**)

Portanto, uma bicondicional é verdadeira quando também são duas condicionais: **p** $\rightarrow$ **q** e **q** $\rightarrow$ **p**.

Exemplos:

p: Roma fica na Europa (V)

q: A neve é branca (V)

p $\leftrightarrow$ **q**: Roma fica na Europa se e somente se a neve é branca (V)

V(**p** $\leftrightarrow$ **q**) = V(**p**) $\leftrightarrow$ V(**q**) = V $\leftrightarrow$ V = V

p: Vasco da Gama descobriu o Brasil (F)

q: Tiradentes foi enforcado (V)

p $\leftrightarrow$ **q**: Vasco da Gama descobriu o Brasil se e somente se Tiradentes foi enforcado (F)

V(**p** $\leftrightarrow$ **q**) = V(**p**) $\leftrightarrow$ V(**q**) = F $\leftrightarrow$ V = F

As premissas e a conclusão de um argumento, formuladas em uma linguagem estruturada, permitem que o argumento possa ter uma análise lógica apropriada para a verificação de sua validade. Quando se verifica a validade de um argumento, se está verificando que no caso das premissas serem verdadeiras elas inferem uma determinada conclusão. Isto é possível através dos métodos utilizados no Cálculo Proposicional.

3 LÓGICA DE DIÁLOGOS

As origens da lógica moderna encontram-se na Grécia antiga, onde a lógica constituía uma disciplina básica praticada ao lado da Oratória e da arte discussiva, confundindo-se às vezes com esta e recebendo a denominação de "Dialética". A Dialética não era uma arte solipsista, pelo contrário, consistia não apenas em encontrar princípios, razões e explicações para algumas preposições, mas principalmente em encontrar argumentos para convencer os demais participantes do diálogo.[PEQ89]

3.1 INTRODUÇÃO

Pode-se aplicar a lógica em jogos como futebol, xadrez, estratégias de guerra, dentre outros, desde que, forneça um modelo coerente às dinâmicas lógicas do fluxo de informações. As argumentações lógicas de uma declaração são como um jogo. Pode-se "vencer um argumento" através de um posicionamento contrário em relação aos argumentos de seu adversário.

3.2 CONCEITO DE LÓGICA DE DIÁLOGOS

A tradição dialética da lógica foi redescoberta por Lorenz [LOR82] que apresenta propostas formais de representação da lógica em termos de diálogos. Ao mesmo tempo, Hamblin [HAM71] e Mackenzie [MAC96] apresentaram modelos matemáticos para os diálogos em linguagem natural, que englobam os diálogos lógicos. [PEQ89]

3.3 JOGOS DE DIÁLOGOS

O debate racional tem sido uma grande fonte de inspiração para a Lógica. Os Jogos Lógicos modelam interações entre dois jogadores, o "**Proponente**" e o "**Oponente**", que debatem com argumentações. O jogo deve ter regras lógicas que regulam defesa e ataque das afirmações, negações e disjunções. Neste, os participantes seguem rigorosas características como:

- Falar sempre a verdade.
- Afirmar algo admissível e relevante.
- Evitar circularidade.

Finalmente, será utilizada uma convenção para determinar quem venceu. O jogador que não tem mais nada a argumentar será declarado perdedor, ou seja, um diálogo é ganho pelo **Proponente** (**P**) se este for finito, isto é, não ocorrer repetições infinitas (**circularidade**); e se não permitir que o **Oponente** (**O**) continue com outro ataque ou defesa.

3.4 REGRAS LÓGICAS DE ATAQUES E DEFESAS

Símbolos:

Serão utilizados os símbolos estudados na Lógica Proposicional, a saber:

- Negação “¬”
- Conjunção “∧”
- Disjunção “∨”
- Condicinal “→”

Fórmulas:

Seguem conforme as definições descritas no capítulo 2 - Lógica Proposicional.

Regras:

- A conjunção **A ∧ B**, pode ser atacada em conjunto, mas o argumento de A ou B servirá em separado à defesa correspondente;
- A disjunção **A ∨ B**, é atacada e possui uma chance de resposta. A defesa consiste na escolha da afirmação de A ou B, e defende a esta;
- A negação **¬ A**, é atacada com a declaração de somente A, ou seja, não há defesa;
- A condicional **A → B**, o ataque consiste em defender A, e sua correspondente defesa é escolher a afirmação B.

Em resumo:

Tabela 6 – Estratégias:

	Ataque	**Defesa**
A ∧ B	?L	A
	?R	B
A ∨ B	?	A ou B
¬ A	A	
A → B	A	B

Neste diálogo lógico é necessário um contador de movimentos e um registro do que é dito e para qual finalidade. Tal contexto é utilizado para um melhor entendimento das argumentações (**diálogos**). O contador de movimento tem a finalidade de contabilizar o número de "rounds" efetuados pelos jogadores. No registro será armazenado os dados referentes as jogadas, expressões e tipos de estratégias.

3.4.1 REGRAS DO JOGO

Para a realização de um diálogo organizado e bem modelado, os jogadores devem seguir certas regras funcionais, sendo:

1ª) Os jogadores se movimentam em revezamento, isto é, alternando os jogadores e respeitando as regras.

2ª) O **Proponente** (**P)** somente poderá afirmar uma fórmula atômica, depois que esta tiver sido afirmada pelo **Oponente** (**O).**

3ª) As fases nos diálogos lógicos, *rounds*, são compostas de um ataque e de uma defesa. Esta regra determina que o jogador tem que responder aos ataques em aberto, isto é, terminar primeiramente os *rounds* mais recentes e só então os mais antigos.

4ª) Um ataque só pode ser respondido uma única vez pelo **Oponente** (**O**).

5ª) Uma fórmula afirmada pelo jogador, só poderá ter um único ataque.

3.5 INICIANDO O JOGO

Para começar a jogada são utilizados alguns exemplos possíveis que obedecem a um formato padrão de exibição para um melhor entendimento de cada passo. Em cada linha será registrado o Ataque ou Defesa da afirmação anterior. Inicia-se na linha 0 com o **Proponente** afirmando uma expressão, logo em seguida, é a vez do **Oponente** atacar a afirmação [A (ataque), n° da linha]. Os jogadores vão se revezando até o fim das argumentações conforme as regras do jogo.

"Como em todos os jogos, há uma diferença entre poder ganhar e realmente ganhar". [OLI04]

Exemplos:

Defender $p \land \neg(p \land q) \rightarrow \neg q$

0	**P**	$p \land \neg(p \land q) \rightarrow \neg q$	
1	**O**	$p \land \neg(p \land q)$	[A, 0]
2	**P**	(**Proponente** pode responder ao primeiro ataque ou contra atacar)	
	P	?L	[A, 1]
3	**O**	p	[D, 2]
		(Esta é a única coisa que o **Oponente** pode fazer)	
4	**P**	?R	[A, 1]
5	**O**	$\neg(p \land q)$	[D, 4]
6	**P**	$\neg q$	[D, 1]
7	**O**	(**Oponente** não tem mais nenhum ataque adicional a responder, então ataca)	
	O	q	[A, 6]
8	**P**	$p \land q$	[A, 5]
9	**O**	?L	[A, 8]
10	**P**	p	[D, 9]

(Admissível, pois p foi afirmado antes pelo **Oponente**)

11 **O** ?R [A, 9]

12 **P** q [D,11]

(Admissível, pois q foi afirmado antes pelo **Oponente**)

13 **O** Não tem nada para dizer, então perde.

Este exemplo exibe um diálogo, onde a proposição $\mathbf{p \wedge \neg(p \wedge q) \rightarrow \neg\, q}$ é verdadeira, pois o **Proponente** (**P**) venceu o debate, e ele estava defendendo a mesma, enquanto o **Oponente** (**O**) que a estava atacando, perdeu. Todas as regras do jogo foram respeitadas pelos jogadores.

Por exemplo, ao defender $\mathbf{A \rightarrow A \vee B}$, o **Proponente P** poderá obviamente ganhar escolhendo A quando for atacado. Mas poderá também perder se ele quiser afirmar B, conforme demonstrado a seguir:

0 **P** $A \rightarrow A \vee B$

1 **O** A [A, 0]

2 **P** $A \vee B$ [D, 1]

3 **O** ? [A, 2]

(O **Oponente** está forçando o **Proponente** a afirmar A ou B)

4 **P** (Se o **P** quiser afirmar A ele poderá ganhar, pois o **O** já afirmou; mas se **P** quiser afirmar B, não poderá, pois B ainda não foi afirmado por **O**, neste caso **P** perderá, pois não tem mais nenhum ataque ou defesa).

P A [D, 3]

5 **O** Não tem mais nada a dizer, então perde.

Os seguintes casos ilustram alguns efeitos destas convenções. Primeiramente, considere a classificação válida de implicação **q → ¬¬ q**, onde **P** tem uma estratégia de ganho:

0	**P**	$q \rightarrow \neg\neg q$	
1	**O**	q	[A, 1]
2	**P**	$\neg\neg q$	[D, 2]
3	**O**	$\neg q$	[A, 3]
4	**P**	q	[A, 4]
5	**O**	Não tem mais nada a argumentar, então perde.	

Agora, considere o inverso **¬¬ q → q**

0	**P**	$\neg\neg q \rightarrow q$	
1	**O**	$\neg\neg q$	[A, 0]

2 **P** Não poderá responder este a ataque, porque q ainda não foi afirmado, então ele contra-ataca.

	P	$\neg q$	[A, 1]
3	**O**	q	[A, 2]

4 **P** O **Proponente** gostaria de usar esta resposta para responder o primeiro ataque, mas tem que responder ao último ataque, pois deve obedecer a **regra 2** do item 3.4.1, que é impossível.

Observou-se que o **Proponente** (**P**) não pode ganhar este jogo! Assim é visto que a lógica dos jogos de diálogos não obedece à mesma inferência clássica.

Querendo que o **Proponente** (**P**) ganhe, será preciso mudar as regras, alterando a **regra 2** do item 3.4.1, permitindo que **P** e somente **P**, deixe o último *round* em aberto e responda a um anterior.

4 **P** q [D, 1]

5 **O** Não tem mais nada a dizer, então perde.

O próximo exemplo ilustra esta diferença, entre a lógica clássica e a lógica intuicionista um pouco melhor.

0 **P** $q \vee \neg q$

1 **O** ? [A, 0]

2 **P** $\neg q$ [D, 1]

3 **O** q [A, 2]

4 **P** Não tem mais nada a dizer, então perde.

Querendo que o **Proponente** (**P**) ganhe, será preciso mudar as regras, alterando a **regra 3** do item 3.4.1, e permitindo que **P** e somente **P**, responda outra vez a um mesmo ataque.

5 **P** q [D, 2]

6 **O** Agora não tem mais nada a dizer, então perde.

3.6 O PROBLEMA DA CIRCULARIDADE

Na lógica de Diálogos podem ocorrer repetições infinitas durante o debate fazendo com que não ocorra o término do jogo. Como mencionado no item 3.3 um debate deve ser **finito**, caso contrário, nenhum jogador ganha. Ao aplicar a **regra 4** do item 3.4.1 em que um ataque deve ser respondido uma única vez e a **regra 5** do mesmo item em que uma fórmula afirmada pelo jogador, só poderá ter um único ataque; a **circularidade** será evitada pois não há como voltar a um ataque realizado ou permanecer na defesa de um ataque continuamente.

3.7 DIFERENÇA ENTRE LÓGICA INTUICIONISTA E LÓGICA CLÁSSICA

Historicamente, há várias formas de “raciocínio válido” partindo-se de um conjunto de premissas em direção a uma conclusão. A visão *de teoria de modelos* define esta idéia como a transmissão de premissas válidas para conclusões. A *teoria de prova* vê esta idéia como prova por derivação elementar em algum sistema natural. Mas o que se tem é um terceiro tipo, *teoria dos jogos*, que “objetiva-se em vencer através de argumentações”.

Para apreciar resultados deste tipo de jogo baseado em Lorenzen, há necessidade de compreender a diferença entre lógica clássica e lógica intuitiva. A última tem uma interpretação construtivista do raciocínio em termos de prova. [OLI04]

Prova clássica de:

$\mathbf{A} \wedge \mathbf{B}$ a partir das provas: de A e de B.

$\mathbf{A} \vee \mathbf{B}$ prova A , ou prova B.

$\mathbf{A} \rightarrow \mathbf{B}$ provando A então se prova B.

$\neg \mathbf{A}$ prova a contradição de A.

A implicação $\mathbf{A}\rightarrow\neg\neg\mathbf{A}$ é construtivamente provável. Tendo uma prova de **A**, constrói-se uma para $\neg\neg\mathbf{A}$.

A fórmula $\neg\neg\mathbf{A}\rightarrow\mathbf{A}$ é igualmente válida de um ponto de vista clássico, mas falha construtivamente: não há caminho óbvio de transformar provas de $\neg\neg\mathbf{A}$ provando diretamente **A**. Com isso, em função do exemplo citado, pode-se verificar uma diferença entre a lógica clássica e a lógica intuicionista:

1- Classicamente para provar **¬A** assume **A** e induz a contradição;

2- Classicamente para provar **A**, assume **¬A** e induz a contradição.

Nestes exemplos, o 1º será construtivamente válido, mas o 2º não. Isto poderia produzir uma prova de **¬¬A**, que não é uma prova para **A** em si mesmo.

3.7.1 PROVANDO DIALOGICAMENTE QUALQUER TAUTOLOGIA

Considere a exclusão **A∨¬A** e a implicação **¬¬A→A**, mais uma vez. Como já foi visto anteriormente, o **Proponente** pode vencer qualquer um destes jogos de diálogos, se lhe é permitido mudar algumas regras. Assim, manipulando as regras procedurais, isto é, direitos e deveres dos jogadores, isto influenciará nos jogos de diálogos. Permitindo-se estas alterações e revendo a análise acima, será possível que os jogos de diálogos permitam também provar dialogicamente qualquer tautologia da lógica clássica.

O Proponente prova igualmente a Lógica Clássica ao deixar as restrições descritas no item 3.4.1 unicamente aplicada ao **Oponente** !

O **Proponente** pode agora repetir ataques, e mover o foco de jogo para estágios (*rounds*) mais antecipados, isto é, respondendo a ataques anteriores. Usando tais variações procedurais, as versões modernas de jogos de diálogos podem modelar uma vasta variedade de lógicas: intuicionista, clássica, modal, ou linear. [KRA98]

Exemplificando esta idéia, tem-se:

Considere as leis clássicas de Morgan:

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

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

A seguinte condicional **¬(A∧B)→ ¬A∨¬B** é classicamente válida, mas não construtivamente. Pode-se visualizar melhor no desenrolar do debate:

Primeira tentativa:

0 **P** $\neg (A \wedge B) \rightarrow \neg A \vee \neg B$

1 **O** $\neg (A \wedge B)$ [A, 0]

2 **P** (Pode defender ou atacar)

$A \wedge B$ [A, 1]

3 **O** ?L [A, 2]

4 **P** Se o jogador for defender, ele terá que defender ao último ataque, que não é possível, então perde.

Como na linha 1 o **Proponente** poderia também defender. No próximo debate, **P**, ao invés de atacar, irá defender.

Segunda tentativa:

0 **P** $\neg (A \wedge B) \rightarrow \neg A \vee \neg B$

1 **O** $\neg (A \wedge B)$ [A, 0]

2 **P** (Pode defender ou atacar)

$\neg A \vee \neg B$ [D, 1]

3 **O** ? [A, 2]

4 **P** $\neg A$ [D, 3]

5 **O** A [A, 4]

6 **P** $A \wedge B$ [A, 1]

7 **O** ?L [A, 6]

8 **P** A [D, 7]

9 **O** ?R [A, 6]

10 **P** Não tem mais nada a dizer, então perde.

A implicação não foi provada construtivamente de nenhuma maneira. Mas, permitindo-se que as **regras 3** e **4** do item 3.4.1, sejam aplicadas exclusivamente ao **Oponente** (**O**), o mesmo debate (nas duas tentativas) seria realizado da seguinte forma:

Primeira tentativa:

0 **P** $\neg (A \wedge B) \rightarrow \neg A \vee \neg B$

1 **O** $\neg (A \wedge B)$ [A, 0]

2 **P** (Pode defender ou atacar)

$A \wedge B$ [A, 1]

3 **O** ?L [A, 2]

4 **P** Permitindo-se que o **Proponente** (**P**) não respeite a **regra 3** do item 3.4.1, ou seja, primeiramente deve-se responder aos últimos ataques para então defender aos anteriores. **P** defende ao ataque da linha 1 antes de responder ao da linha 3.

$\neg A \vee \neg B$ [D, 1]

5 **O** ? [A, 4]

6 **P** $\neg A$ [D, 5]

(Pode defender $\neg B$ também)

7 **O** A [A, 6]

8 **P** A [D, 3]

9 **O** ?R [A, 2]

10 **P** Permitindo-se que o **Proponente** (**P)** não respeite a **regra 4** do item 3.4.1, ou seja, um ataque só pode ser respondido uma única vez. **P** responde novamente ao ataque da linha 5, mudando desta forma sua defesa.

$\neg B$ [D, 5]

11 **O** B [A,10]

12 **P** B [D, 9]

13 **O** Não tem mais nada a dizer, então perde.

Segunda tentativa:

0 **P** $\neg (A \wedge B) \rightarrow \neg A \vee \neg B$

1 **O** $\neg (A \wedge B)$ [A, 0]

2 **P** (Pode defender ou atacar)

$\neg A \vee \neg B$ [D, 1]

3 **O** ? [A, 2]

4 **P** $\neg A$ [D, 3]

5 **O** A [A, 4]

6 **P** $A \wedge B$ [A, 1]

7 **O** ?L [A, 6]

8 **P** A [D, 7]

9 **O** ?R [A, 6]

10 **P** Permitindo-se que o **Proponente** (**P**) não respeite a **regra 4** do item 3.4.1, ou seja, um ataque só pode ser respondido uma única vez. O **Proponente** responde novamente ao ataque da linha 3, mudando desta forma, sua defesa.

	$\neg$B	[D, 3]
11 **O**	B	[A,10]
12 **P**	B	[D, 9]
13 **O**	Não tem mais nada a dizer, então perde.	

Desta forma, as leis de Morgan são provadas tanto classicamente como construtivamente.

3.8 CONSTRUÇÃO DE *TABLEAU* DE JOGOS

O objetivo de um *tableau* é tentar provar as proposições a serem argumentadas, independentes da consistência, verdadeiras ou não. A proposição será considerada provada, isto é, o *tableau* será fechado, quando atingir êxito “contra” qualquer tentativa de ataque em um diálogo.

Para o desenvolvimento do diálogo é necessário seguir certas regras , algumas já descritas anteriormente, para serem utilizadas no decorrer do jogo:

1ª) Um diálogo é ganho por aquele a quem pertence a “última palavra”.

2ª) O **Proponente** (**P**) só pode usar uma proposição logicamente simples, isto é, sem conectores nem qualificadores, quando esta já tiver sido empregada pelo **Oponente** (**O**).

3ª) Uma fórmula afirmada pelo jogador, só pode ser atacada uma única vez. O jogador não pode atacar uma fórmula que tenha sido defendida em um momento anterior.

4ª) Um ataque só pode ser respondido uma única vez, esta regra se aplica somente ao **Oponente** (**O**).

5ª) Nos diálogos lógicos existem certas fases denominadas *rounds* compostas de um ataque e de uma defesa. A regra consiste em que o jogador tem que responder aos ataques em aberto, isto é, fechar primeiramente os *rounds* mais recentes e só então os mais antigos; esta regra se aplica somente ao **Oponente** (**O**). [OLI04]

3.8.1 REGRAS ESTRUTURAIS

Há ainda os seguintes aspectos a preservar:

- O ataque é contra uma parte da proposição.
- O esquema do diálogo lê-se em linhas, que serão enumeradas, sendo sempre revezadas pelo **Proponente** e **Oponente.**
- É necessário que haja um número finito de *rounds*, para que se possa provar uma proposição, pois caso contrário ganhará o **Oponente**, já que o **Proponente** não foi capaz de deixar o seu interlocutor sem argumentos.
- A negação não tem defesa, apenas permite um contra-ataque ou um novo ataque, abrindo-se assim um novo *round*, ficando o anterior em aberto.[OLI04]

3.8.2 REGRAS UTILIZADAS NO *TABLEAU*

Sejam A e B duas fórmulas. Os jogadores **P (Proponente)** e **O (Oponente)** assumem as posições X e Y, de acordo com a ordem das jogadas, lembrando-se que elas são alternadas entre cada jogador. As regras de defesa e ataque de cada jogador serão expressas no *tableau*, conforme tabela 6, da seguinte forma: [OLI04]

Ataque e Defesa da conjunção "**A ∧ B**". Neste caso são possíveis dois ataques, como demonstrados a seguir:

$A_{\wedge 1}$ $\begin{array}{c} X: A \wedge B \\ \downarrow \\ Y: ?L \end{array}$ $A_{\wedge 2}$ $\begin{array}{c} X: A \wedge B \\ \downarrow \\ Y: ?R \end{array}$

$D_{\wedge 1}$ $\begin{array}{c} X: ?L \\ \downarrow \\ Y: A \end{array}$ $D_{\wedge 2}$ $\begin{array}{c} X: ?R \\ \downarrow \\ Y: B \end{array}$

Ataque e Defesa da disjunção "**A ∨ B**". Neste caso é possível apenas um ataque e a defesa pode escolher qual fórmula defender, como demonstrado a seguir:

$A_{\vee}$ $\begin{array}{c} X: A \vee B \\ \downarrow \\ Y: ? \end{array}$ $D_{\vee}$ $\begin{array}{ccc} & X: ? & \\ \swarrow & & \searrow \\ Y: A & & Y: B \end{array}$

Ataque e Defesa da regra de implicação "**A → B**". Neste caso é possível apenas um ataque e uma defesa, como demonstrado a seguir:

$A_{\rightarrow}$ $\begin{array}{c} X: A \rightarrow B \\ \downarrow \\ Y: A \end{array}$ $D_{\rightarrow}$ $\begin{array}{c} X: A \rightarrow B \\ \downarrow \\ Y: B \end{array}$

Ataque e Defesa negação "**¬A**". Neste caso é possível apenas atacar, e não existe defesa, como demonstrado a seguir:

$A_{\neg}$ X: ¬A

$\downarrow$

Y: A

3.9 CONSTRUÇÃO DA ÁRVORE DO *TABLEAU* DE JOGOS

Uma jogada no *tableau* é representada por uma árvore começando com o ataque do Oponente à fórmula que o Proponente deseja defender. E cada nó desta árvore é composto por:

Expressões de diálogos rotuladas (**Exp**).

Definição 1: Seja **Prop** um conjunto de letras proposicionais. O conjunto das fórmulas **Form** é o menor conjunto que satisfaz a linguagem de primeira ordem com seus símbolos ($\neg$, $\wedge$, $\vee$, $\rightarrow$).

Fórmulas**:** As fórmulas **Form** da linguagem são descritas pelas seguintes regras:

- ✓ Todo símbolo proposicional de Φ é uma fórmula, chamada fórmula atômica.
- ✓ Se $p \in$ **Prop** então $p \in$ **Form.**
- ✓ Se $\varphi \in$ **Form**, então $(\neg\varphi)$ também $\in$ **Form.**
- ✓ Se φ e $\psi \in$ **Form**, então $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$ e $(\varphi \rightarrow \psi)$ também $\in$ **Form.**

Nada é uma fórmula, a não ser que seja forçado por um dos itens acima.

Definição 2: O conjunto de expressões de diálogos **Exp** é definido como:

Exp = Form U {?, ?L, ?R}

Definição 3: As expressões de diálogos rotuladas são compostas pelo seguinte registro que possui cinco campos: [OLI04]

Tabela 7- Registro:

n	X	E	T	m

Onde:

n – é a posição do registro no diálogo, sabendo-se que o número de linhas que fará parte do total de ataques ou defesas, varia de 1 a n, onde n é o número total de jogadas de ambos os jogadores;

X – assume a indicação do jogador; que seria o Oponente representado por "**O**" ou o Proponente representado por "**P**";

E – é a expressão de diálogo (**Exp**);

T – é a indicação do tipo da jogada, representada por qualquer uma das regras de ataque ou defesa, isto é: $A_{\wedge 1}$, $A_{\wedge 2}$, $D_{\wedge 1}$, $D_{\wedge 1}$, $A_{\vee}$, $D_{\vee}$, $A_{\rightarrow}$, $D_{\rightarrow}$, $A_{\neg}$ ou $A_{A;}$

m – é a indicação da linha que foi atacada ou defendida; estas linhas são enumeradas de 1 a n.

Exemplo:

16	P	$A \wedge B$	$A_{\rightarrow}$	10

Isto significa que na linha 16 é a vez de **P** que atacou alguma **Exp** que o **O** defendeu na linha 10, resultando este ataque à seguinte **Exp** $A \wedge B$.

Definição 4: Um diálogo (para uma determinada fórmula φ) é uma seqüência de repetições $r_0, ..., r_n$, onde $r_0.E = \varphi$, $r_0.X = P$ e $r_0.n = 0$.

Algumas observações são representadas a seguir para a precisão de um diálogo:

D1 – Se i é par $r_i .X = P$, caso contrário $r_i .X = O$;

D2 – se r_i .T = A qualquer, e r_i .m = j; então r_j .E é uma fórmula e r_i é um ataque a $r_{j;}$

D3 – se r_i .T = A qualquer, e r_i .m = j; então r_j .T = A qualquer e r_i é uma defesa ao ataque de $r_{j;}$

D4 – se r_i .X = P e r_i .E = A, onde A é uma fórmula atômica, então existe um r_j que j < i tal que r_i .X = O e r_j .E = A;

D5 – um ataque pode ser defendido no máximo uma vez;

D6 – um registro r_i com r_i .X = P pode ser atacado no máximo uma vez;

D7 – se r_i .T = D e r_i .m = j então não existe nenhum r_k onde i < K < j tal que r_k é um ataque que ainda não foi respondido até a posição i;

Definição 5: Uma estratégia S para uma fórmula φ é uma sub-árvore de diálogo para φ tal que:

S1 – Todo o ramo de S é um diálogo para φ ganho por P.

S2 – Para todo o nó r:

i – se r .X = O, então r só possui um filho em S;

ii – se r.X = P, então r tem tantos filhos quanto forem as possibilidades de **O** jogar no ramo de r até a raiz.

Teorema da Equivalência: Uma fórmula φ pode ser provada se e somente se existe uma estratégia para φ. [OLI04]

Exemplo 1: Construção de um tableau semântico.

Conforme a definição anterior, a árvore a seguir, é um tableau iniciado com a fórmula que o **Proponente** deseja defender.

0 P $((A \rightarrow B) \rightarrow B) \rightarrow (A \vee B)$

1 O $(A \rightarrow B) \rightarrow B$ $[A_{\rightarrow},0]$

2	P	$A \rightarrow B$	$[A_{\rightarrow},1]$	2	P	$A \vee B$	$[D_{\rightarrow},1]$
3	O	B	$[D_{\rightarrow},2]$	3	O	?	$[A_{\vee}, 2]$
4	P	$A \vee B$	$[D_{\rightarrow},1]$	4	P	$A \rightarrow B$	$[A_{\rightarrow},1]$
5	O	?	$[A_{\vee}, 4]$	5	O	B	$[D_{\rightarrow},4]$
6	P	B	$[D_{\vee}, 5]$	6	P	B	$[D_{\vee}, 3]$
7	O	Não tem nada a dizer, perde.		7	O	Nada a dizer, perde.	

Pode-se observar que o *tableau* contém 2 ramos. Isto significa que existe a possibilidade do jogador seguir dois caminhos diferentes, isto é, o jogador pode optar entre duas jogadas. Em todos os caminhos seguidos, o *tableau* foi **fechado**, porque o **Proponente** foi o vencedor.

Exemplo 2:

0 P $(((A \text{ v } B) \wedge (A \rightarrow C)) \wedge (B \rightarrow C)) \rightarrow C$

1 O $((A \text{ v } B) \wedge (A \rightarrow> C)) \wedge (B \rightarrow C)$ $[A_{\rightarrow},1]$

2 P (Ele não pode defender, então ataca.)

?L $[A_{\wedge 1},2]$

3 O $(A \text{ v } B) \wedge (A \rightarrow C)$ $[D_{\wedge 1},3]$

4	P	?R	$[A_{\wedge 2}, 2]$
5	O	$B \rightarrow C$	$[D_{\wedge 2}, 5]$
6	P	?L	$[A_{\wedge 1}, 4]$
7	O	A v B	$[D_{\wedge 1}, 7]$
8	P	?R	$[A_{\wedge 2}, 4]$
9	O	$A \rightarrow C$	$[D_{\wedge 2}, 9]$
10	P	?	$[A_{\vee}, 8]$
11	O	(Ele tem dois caminhos, pode escolher afirmar A ou B.)	

		A	$[D_{\vee}, 11]$			B	$[D_{\vee}, 11]$
12	P	A	$[A_{\rightarrow}, 10]$	12	P	B	$[A_{\rightarrow}, 6]$
13	O	C	$[D_{\rightarrow}, 13]$	13	O	C	$[D_{\rightarrow}, 13]$
14	P	C	$[D_{\rightarrow}, 2]$	14	P	C	$[D_{\rightarrow}, 2]$
15	O	Não tem nada a dizer, perde.		15	O	Nada a dizer, perde.	

Constatou-se que se deve provar um dos dois lados da disjunção (**v**). Caso contrário, seria possível provar (A v B) $\wedge$ ($A \rightarrow C$) $\rightarrow$ C, o que não e verdade. Como demonstrado:

0	P	$(A \vee B) \wedge (A \rightarrow C) \rightarrow C$	
1	O	$(A \vee B) \wedge (A \rightarrow C)$	[A, 0]
2	P	?L	[A, 1]

3	O	A v B	[D, 2]
4	P	?R	[A, 1]
5	O	$A \rightarrow C$	[D, 4]
6	P	?	[A, 3]
7	O		

Se o **Oponente** escolher A, ele perde, mas se escolher B, ele ganha. Como já explicado anteriormente, existe uma diferença entre ganhar e poder ganhar.

3.10 ALGORITMO DE DIÁLOGOS

Para a solução de problemas utilizando a Lógica de Diálogos, foi melhorado o algoritmo existente, apresentado no anexo A, para uma implementação futura. Desta forma a referida complementação estrutural foi:

Procedimento Estrategia – P (φ, S)

Cria_no (r_0);

r_0.n := 0;

r_0.X := P;

r_0.E := φ; // Criação do nó raiz da árvore.

retorne Expande (S, r_0) // S é o tipo de jogada, ataque ou defesa.

Procedimento Expande (S, r_i)

R := 0;

Para cada possibilidade de jogada faça

// Crie um registro r_j

Rj.N := Rj.N+1; // n.

Se j é par então // x.

Rj.X := P

Senão Rj.X := O

1

Rj.E:= φj // φi = nova proposição usada pelo jogador.

Rj.T := S;

Rj.m:= LinhaAberta(); // Função que retorna ultima linha em aberto da pilha do jogador.

R:= R U {r_j}

Se R = 0 e rj.X = P então // É a vez do **O** jogar.

Retorna Sucesso

Senão

Retorna Falha

Se r_0.X = O então // É a vez de **P** jogar.

Para cada $r_j \in$ R faça

Se Expande (S, r_j) então

aloca (r_j, S)

```
                EmpilhaP(rj)
                                          } 2
                DesempilhaO(rj,lista)

                Retorna Sucesso.

        Senão   // isto é se for a vez de O jogar ( r0.X=P ).

            P_perde := False

            Para cada rj ∈ R faça

                Se Expande (S, rj) então

                    P_perde := True

                    EmpilhaO(rj)                } 3

                    DesempilhaP(rj,lista)

                Se P_perde então

                    Retorna Falha

            Senão

                Para cada rj ∈ R faça

                    Aloca (rj,S)

                    Retorna Sucesso
```

No algoritmo apresentado foi acrescentado algumas melhorias, sendo que, no 1º bloco (**1**) foi detalhado a criação do registro conforme a possibilidade de jogada, ou seja, para cada possibilidade será criado um registro seguindo algumas regras para a precisão de um diálogo conforme item 3.9. Foi implementado um contador (Rj.N := Rj.N+1), onde é possível determinar qual o jogador, ou seja, se a linha for par será o **Proponente** e se for ímpar o **Oponente** (Se j é par então / Rj.X := P / Senão Rj.X := O). De acordo com a Tabela de Estratégias (Tabela 6) são montadas todas as proposições possíveis a serem usadas pelo jogador (Rj.E:= φ_j) e também o tipo de jogada, ataque ou defesa, que será definido de acordo com o parâmetro recebido pelo procedimento **Expande** proveniente do procedimento **Estratégia – P**. A linha a ser atacada ou defendida será definida pela função **LinhaAberta ( )** que irá retornar a posição da linha em aberto de acordo com o jogador ((Rj.m:= LinhaAberta( )). No 2º Bloco (**2**), foram inseridas a função **EmpilhaP**, com a finalidade de se montar uma pilha com as jogadas realizadas por **P** e a função **DesempilhaO**, com o objetivo de desempilhar o registro correspondente a uma jogada, caso a jogada efetuada por **P** seja uma resposta correspondente ao registro na pilha de **O**. No 3º bloco (**3**) foram inseridas as funções **EmpilhaO** e **DesempilhaP**, sendo que a primeira, monta uma pilha das jogadas realizadas pelo **Oponente O** e a segunda, desempilha o registro caso seja uma resposta a última defesa.

3.11 CONSIDERAÇÕES FINAIS

A abordagem dialogal da lógica é bastante promissora, pois ao se resgatar as tradições clássicas da lógica, aproxima-se da maneira como a lógica é utilizada em nossas atividades epistêmicas corriqueiras. A inteligência Artificial, em especial, tem muito do que se beneficiar com esta abordagem, principalmente nas áreas de formalização e automatização do raciocínio e processamento de linguagem natural.

4 CONCLUSÃO

Muitos são os desdobramentos possíveis desta monografia. O foco principal foi o melhoramento de um algoritmo básico utilizando-se a Lógica de Diálogos para demonstrar a sua aplicabilidade na formalização do raciocínio do senso comum e na confecção de sistemas de manipulação do conhecimento.

Foi utilizada a Lógica Proposicional para demonstrar a validade de um argumento através das tabelas verdades dos conectivos lógicos, bem como, a conversão de um enunciado em uma estrutura composta de letras e operadores lógicos.

Nas expressões geradas pela Lógica Proposicional foi aplicada a Lógica de Diálogos com intuito de construir um *tableau* onde serão utilizadas regras e estratégias, conforme descrito no capítulo 3, para o que o **Proponente** consiga quase sempre vencer um diálogo (argumentação). Com isso, foi estudado o algoritmo, presente na tese de Mestrado da Professora Lorena Campos de Oliveira, conforme anexo A, que têm como objetivo simular os jogos de diálogos (*tableau*). Para tal, foram efetuadas as complementações (detalhamentos) conforme citadas no item 3.10, ficando o mesmo disponível para uma futura implementação prática.

Desta forma, no campo da Inteligência Artificial (IA), procura-se entender as entidades inteligentes focalizando seus estudos no aprendizado sobre o próprio Homem, com o intuito de poder simular as estruturas inteligentes deste em uma máquina.

REFERÊNCIAS BIBLIOGRÁFICAS

[GAB84] GABBAY, D. M.. **Elementary Logic: A Procedural Perspective**; Lecture Notes, Logic 141, draft 1984, Department of Computing, Imperial College of Science and Technology, London.

[HAM71] HAMBLIN, C. **Mathematical models of dialogue**; Theoria, v. 2, p. 130-155, 1971.

[KRA98] KRABBE, Erick C. W. **The Dialectic of Quasi-Logical Argument**; Fourth ISSA Conference on Argumentation, Amsterdam, 16-19 june 1998.

[LOR82] LORENZ, K. **On the Criteria for choice of Rules of Dialogic Logic**; Argumentation: aproaches to theory formation, SLCS v. 8, John Benjamins, Amsterdam, 1982.

[MAC96] MACKENZIE, Jim D. **Logic and Argumentation**; North Holland, p. 129-141, Amsterdam, 1996.

[OLI04] OLIVEIRA, Lorena Sophia Campos de. **Especificação de Software Educacionais Utilizando Lógica de Diálogos e Sistema Multiagentes Baseados em Conhecimento;** Instituto de Matemática – NCE – UFRJ, Rio de Janeiro, 2004.

[PEQ89] PEQUENO, Marcelino. **Um Raciocinador Automático por Jogos Semânticos**; Anais do IV Simpósio Brasileiro de Inteligência Artificial; Rio de Janeiro, 1989.

[SOU02] SOUZA, João Nunes. **Lógica para Ciência da Computação Fundamentos de Linguagem, Semântica e Sistema de Dedução**; São Paulo, SP: Editora Campus, 2002.

ANEXO A – ALGORÍTMO DE DIÁLOGOS

Conforme descrito na tese de Mestrado da Professora Lorena Sophia Campos de Oliveira. [OLI04]

Procedimento Estrategia – P (φ, S) **// Sucesso ou Falha**

Cria_no (r_0);

r_0.n := 0;

r_0.X := P;

r_0.E := φ; // Foi criado o nó raiz da árvore

retorne Expande (S, r_0) // S é o tipo de jogada, ataque ou defesa.

Procedimento Expande (S, r_i) **// Sucesso ou falha**

R := 0;

Para cada possibilidade de jogada obedecendo **D1,...,D7** faça

Crie um registro r_j

R:= R U {r_j}

Se R = 0 e r_i.X = P então // é a vez do Oponente jogar, e ele não tem nenhum nó a explorar.

Retorna Sucesso

Senão

Retorna Falha

Se r_0.X = O então // se for a vez do P jogar

Para cada $r_j \in$ R faça

Se Expande (S, r_j) então aloca (r_j, S)

Retorna Sucesso.

Senão // isto é se for a vez de O jogar (r_0.X=P)

P_pede := False

Para cada rj $\in$ R faça

Se Expande (S, r_j) então

P_perde := True

Se P_perde então

Retorna Falha

Senão

Para cada $r_j \in$ R faça

Aloca (r_j,S)

Retorna Sucesso