

M.A.LOPES* e B.M.ACIOLY**

SUMÁRIO

Neste trabalho é apresentado um esboço teórico de uma lógica de esquema de soluções funcionais e também um esboço de uma lógica de esquema de problemas. A ligação entre os dois sistemas formais (lógicas) é feita através da noção de satisfação apresentada aqui, através dos conceitos de realização e solubilidade. A partir da noção da satisfação é dado um esboço para uma lógica de resolução de problemas.

ABSTRACT

This paper presents a framework of a functional solution scheme logic and a framework of a problem scheme logic. The connection of these two formal systems (logics) is done through the notion of satisfiability, which in turn is presented using the concepts of realization and solubility. Finally, from the notion of satisfiability a framework of a problem solving logic is derived.

* Bacharel em Matemática (UFPE, 1969), Mestre em Matemática (UFPE, 1971), Doutor em Informática (PUC/RJ, 1981), teoria de problemas, inteligência artificial, tipos abstratos de dados, Sociedade Brasileira de Computação, Rua Ramiro Costa, 79 - Prado - Recife, 50.000 - (081) 228.2922

** Bacharel em Matemática (UFPE, 1970), Mestre em Matemática (UFPE, 1980) lógica, teoria de problemas, inteligência artificial, Rua Gal. Polidoro, 425/201 - Iputinga - Recife, 50.000 - (081) 271.2740.

Uma lógica de problemas, como sugere Kolmogoroff em [1], com sua interpretação para o cálculo proposicional intuicionista, leva em conta fortemente a noção de construção de solução, sem necessariamente, se referir ao problema em si. Em seu trabalho, Kolmogoroff apresentou a sintaxe da lógica intuicionista e dá a ela uma semântica de "problema" sem levar em conta a natureza estrutural de problemas. Considerando um problema como uma estrutura relacional, de modo como faz Lopes em [2], não é possível encontrar o similar à interpretação dada por Kolmogoroff para $\neg P$, que significa: "se da uma solução para o problema P , obtenha uma contradição". Isto porque, em [2], um problema é um construto, onde sua negação, do ponto de vista da solubilidade, não é definível univocamente, uma vez que envolve simultaneamente a noção de intencionalidade e de admissibilidade, enquanto em [1] a noção de problema é pragmática, sem uma estruturação prévia. O aproveitamento da sintaxe da lógica intuicionista para uma interpretação de problemas, no sentido de Lopes, parece inadequado porque a interpretação da negação está prejudicada como veremos na seção 3.1.

A idéia básica deste trabalho é tratar de problemas e solução como entidades sintaticamente distintas e consequentemente de lógicas (sistemas formais) distintas. Desse modo, uma lógica de solução tratará de soluções como esquemas (funcionais) abstratos, que estarão posteriormente sujeitos a realizações, ou seja, dado um esquema de solução e um problema, esse esquema é realizável nesse problema se ele pode ser interpretado como uma certa função do conjunto de dados no conjunto de resultados desse problema.

A ligação entre os dois sistemas lógicos será feita através de uma noção de satisfação (não tarskiana) onde um esquema α satisfaz um problema P se e somente se α realizado em P implica em α resolve P .

2. PROBLEMAS

A palavra *problema* é empregada correntemente em várias acepções. No nosso caso, estamos interessados nas situações em que podemos distinguir claramente uma intenção de associar situações iniciais e situações finais, ou em outras palavras, em todo problema distinguimos três elementos importantes: o *domínio de dados*, um *domínio de resultados*, que representam a parte substantiva do problema, e um *predicado intencional* que representa a intenção (subjativa). Numa questão, como por exemplo: "calcular o comprimento de listas" identificamos: o conjunto L de listas como o domínio de dados, os conjuntos $\mathbb{N}$ dos naturais como o domínio de resultados e o predicado intencional r expresso do seguinte modo: para todo elemento $\ell \in L$ e para todo $n \in \mathbb{N}$, ℓ está relacionado com n por meio de r se e somente se, o comprimento de ℓ for igual a n .

Embora os três elementos listados acima sejam fundamentais em qualquer problema, eles não são os únicos a aparecerem. No exemplo dado, poderíamos estar interessados somente em listas de tamanho finito, ou seja, impomos uma condição sobre os dados, ou poderíamos só admitir soluções algorítmicas. Podemos, portanto, dar a seguinte definição

Def. 2.1. - Um *problema* é uma estrutura do tipo $P = \langle D, R, d, r, A \rangle$, onde D e R são conjuntos não vazios, chamados *domínios de dados* e de *resultados*, respectivamente, $d \subseteq D$, representa a *condição sobre os dados*, $r \subseteq D \times R$ (produto cartesiano de D por R) representa a *condição do problema* e $A \subseteq R^D$ (conjunto das funções de D em R), representa a *condição de admissibilidade*.

Os conjuntos D e R representam a parte substantiva (concreta), ou seja, os universos do problema, enquanto d , r e A representam a parte intencional (subjettiva). A idéia é que r reflita a condição principal ou intencionalidade principal, segundo a qual, os elementos $f \in D$, $s \in R$ se relacionam, isto é, $(f, s) \in r$; d reflita a parte de D relevante no problema e, A seja a restrição as muitas possíveis maneiras de mapear D em R (funções), que satisfazem r e que serão consideradas admissíveis.

Def. 2.2. - Um problema $P = \langle D, R, d, r, A \rangle$ é *solúvel* se existe uma função $a \in R^D$ tal que $a \in A$ e para todo $f \in d$ tem-se $(f, a(f)) \in r$. Uma função que satisfaça estas condições chama-se *solução* do problema P e usamos a notação $a \models P$ para expressar que " a resolve P " ou " a é uma solução de P ".

A partir dos problemas $P = \langle D_1, R_1, d_1, r_1, A_1 \rangle$ e $Q = \langle D_2, R_2, d_2, r_2, A_2 \rangle$, podemos construir novos problemas como segue. Ver [4]

$P \wedge Q = \langle D_1 \times D_2, R_1 \times R_2, d_1 \times d_2, r_1 \times r_2, A_1 \times A_2 \rangle$, onde " $\times$ ", indica o produto cartesiano, e temos o "problema P e Q ".

$P \vee Q = \langle D_1 \times D_2, R_1 \times R_2, d_1 \times d_2, r_1 \cup r_2, A_1 \times A_2 \rangle$, onde " $\cup$ " é o produto cartesiano R

$r_1 \cup r_2 = \{(x, y) / x \in r_1 \text{ ou } y \in r_2\}$ (união cartesiana) $P \vee Q$ será lido " P ou Q "

$P \longrightarrow Q = \langle D_1 \times D_2, R_1 \times R_2, d_1 \times d_2, \bar{r}_1 \cup r_2, A_1 \times A_2 \rangle$, onde $\bar{r}_1$, é o complementar de r_1 em relação a $D_1 \times D_1$. O problema $P \longrightarrow Q$ será lido: " P implica Q ".

Como podemos ter aprioristicamente soluções funcionais, isto é, esquemas abstratos que poderão ser interpretados como funções através de atribuições de entradas (argumentos) e saídas (valor da função) dentro de domínios definidos, devemos ter um mecanismo formal que permita essa atribuição. No caso de funcionais e problemas isso será feito com a seguinte definição:

Def. 2.4. - Uma solução funcional (ou simplesmente funcional) é *realizável* num problema $P = \langle D, R, d, r, A \rangle$ se a interpretação a^P de a em P é uma função $a^P : D \longrightarrow R$. Quando não houver motivo de confusão a^P será simplesmente denotado por a . A notação $a \models P$ será utilizada para expressar a é realizável em P .

Def. 2.5 - Um funcional a *satisfaz* um problema P (denotado por: $a \models P$) se e so

mente se, $a \vdash P$ implica $a \vdash P$, isto é $a \in R^D$ implica $a \in A$ e $\forall f \in d(f, a(f)) \in r$.

A definição 2.5 será utilizada mais tarde para fazer a ligação entre os sistemas formais de soluções funcionais e esquema de problemas.

3. LÓGICA DE PROBLEMAS

3.1. A QUESTÃO DA NEGAÇÃO

É natural que a idéia de negação de um problema seja obtida através da negação de uma intenção, isto é, se $P = \langle D, R, d, r, A \rangle$, então $\neg P = \langle D, R, d, \bar{r}, A \rangle$ onde $\bar{r}$ é a complementação (com relação a $D \times R$) da intenção r . Isso significa que r e $\bar{r}$ são disjuntas ou em outras palavras, independentes, no sentido que as intenções descritas em r e $\bar{r}$ são independentes. Por exemplo: Dada uma sequência de números naturais, sem repetição, ordene-a! Claramente a intenção no problema acima é a ordenação. A negação desta intenção pode ser dada por "não ordene" ou "desordene". Não ordene parece ser a alternativa lógica da negativa que consistiria em acrescentar o conectivo "não" antes da ação "ordene". Por outro lado, isso não chega a ser uma ação em si, uma vez que, "não aja", não tem valor de procedimentos. Desse modo, "desordene" seria uma alternativa para a negação da intenção do problema acima.

Considerando "desordene" como negativa da intenção acima, poderemos apontar o seguinte inconveniente técnico: se descrevermos ordene através da conjunção de dois predicados "mesma" e "boa", onde *mesma* (s, s') significa que as sequências s e s' possuem os mesmos elementos e *boa* (s') significa que s' está ordenada, temos que ordene (s, s') (ou seja, s' é a sequência s ordenada) é equivalente a *mesma* (s, s') $\wedge$ *boa* (s'). assim "desordene" seria descrita por $\neg$ ordene (s, s') ou seja $\neg((\text{mesma}(s, s') \wedge \text{boa}(s')))$ que é equivalente a $\neg \text{mesma}(s, s') \vee \neg \text{boa}(s')$ (1) que não corresponde a intuição de "desordene". Na realidade a fórmula predicativa que refletiria a intuição de "desordene" seria *mesma*(s, s') $\wedge$ $\neg$ *boa*(s') (2) que não é equivalente a fórmula acima. Isto significa que para estabelecermos (2) em vez de (1), para expressarmos "desordene" teríamos que fazer considerações extralógica e isto requer uma análise semântica a priori da estrutura sintática da intenção do problema.

Além do acima exposto, a negação de uma ação com valor de procedimento parece adquirir o "status" de uma ação em si, independente da ação positiva, isto é, ela também é uma ação positiva. Se encarado assim, os problemas P e $\neg P$ são problemas diferentes e independentes. É claro que a negação de um problema, do ponto de vista da intenção, não dá a essa negação, o caráter de negativa lógica (conectivo lógico).

Pelas considerações acima parece-nos adequado que um sistema formal de problemas sirva de interpretação para o cálculo proposicional positivo clássico, como está em [3].

3.2. SISTEMA FORMAL DE ESQUEMAS DE PROBLEMAS

Seja E o conjunto das expressões de uma linguagem L formadas conforme as regras abaixo.

As letras $A, B, C, \dots, P, Q, \dots, Z$, são variáveis e os símbolos $\wedge, \vee$ e $\longrightarrow$, são conectivos lógicos. Na linguagem L as fórmulas bem formadas, aqui chamadas de *esquemas de problemas* ou simplesmente problemas, são obtidas através das seguintes regras:

P_1 - As variáveis $A, B, C, \dots$ etc são (esquemas de) problemas

P_2 - Se A e B são problemas, então $A \wedge B$, $A \vee B$ e $A \longrightarrow B$ são problemas.

P_3 - Os problemas são as sentenças obtidas através de P_1 e P_2 .

A igualdade entre dois problemas será introduzida aqui, pela definição tradicional:

$$A = B \stackrel{\text{def.}}{=} (A \longrightarrow B) \wedge (B \longrightarrow A)$$

O *cálculo de problemas* será definido através do seguinte conjunto de esquemas de axiomas e regras de inferência:

$$A_1 - A \longrightarrow B \longrightarrow A$$

$$A_2 - (A \longrightarrow (B \longrightarrow C)) \longrightarrow (A \longrightarrow B) \longrightarrow (A \longrightarrow C)$$

$$A_3 - ((A \longrightarrow B) \longrightarrow A) \longrightarrow A$$

$$A_4 - A \longrightarrow (A \vee B)$$

$$A_5 - A \longrightarrow (B \vee A)$$

$$A_6 - (A \longrightarrow C) \longrightarrow (B \longrightarrow C) \longrightarrow (A \vee B) \longrightarrow A$$

$$A_7 - A \wedge B \longrightarrow A$$

$$A_8 - (A \wedge B) \longrightarrow B$$

$$A_9 - (A \longrightarrow B) \longrightarrow (A \wedge B)$$

$$\text{MP} - \text{Se } A \text{ e } A \longrightarrow B, \text{ então } B.$$

Denotando "é dedutível de" por " $\vdash$ ", e assumindo as definições clássicas de "prova", "dedutível" e "teorema" temos os seguintes resultados que são consequências diretas dos axiomas e da regra de inferência acima.

1 - Seja Γ um conjunto de problemas e A, B são problemas. Se $\Gamma, A \vdash B$, então $\Gamma \vdash A \longrightarrow B$ (teorema da dedução).

2 - $A \longrightarrow B, B \longrightarrow C \vdash A \longrightarrow C$.

3 - $\vdash A \longrightarrow (A \longrightarrow B) \longrightarrow B$

4 - $\vdash (A \longrightarrow (A \longrightarrow B)) \longrightarrow A \longrightarrow B$

5 - $(A \longrightarrow B) \longrightarrow B, A \longrightarrow C \vdash (C \longrightarrow B) \longrightarrow B$

6 - $\vdash A \vee (A \longrightarrow B)$

7 - $A \vee B, A \longrightarrow C, B \longrightarrow D \vdash C \vee D$

8 - $((A_1 \vee A_2 \vee \dots \vee A_n) \vee B) = (A_1 \vee \dots \vee A_n \vee B)$

- 9 - $A_1 \vee \dots \vee A_n \vdash B_1 \vee \dots \vee B_k$ se cada dos $A_1, \dots, A_n$ é um dos $B_1, \dots, B_k$.
- 10 - $A \vee B, C \longrightarrow A \vdash (B \longrightarrow C) \longrightarrow A$.
- 11 - $A \vee B, A \longrightarrow B \vdash B$
- 12 - $\vdash (A \longrightarrow (B \wedge C)) = (A \longrightarrow B) \wedge (A \longrightarrow C)$
- 13 - $\vdash ((A \wedge B) \longrightarrow C) = (A \longrightarrow B \longrightarrow C)$
- 14 - $\vdash (C \vee (A \wedge B)) = ((C \vee A) \wedge (C \vee B))$

3.3. SISTEMA FORMAL DE SOLUÇÕES FUNCIONAIS

De modo análogo ao Sistema Formal de Esquemas de Problemas, usando as letras minúsculas $a, b, c, \dots$ como variáveis e os símbolos $\wedge, \vee, \longrightarrow$ como conectivos lógicos, podemos definir para um Sistema Formal de soluções funcionais, as fórmulas bem formadas, aqui chamadas de *soluções funcionais* ou simplesmente funcionais, como sendo obtidas através de regras similares a da seção 3.2:

- S_1 - As variáveis $a, b, c, \dots$ são (soluções funcionais) funcionais
- S_2 - Se a e b são funcionais então $a \wedge b, a \vee b$ e $a \longrightarrow b$ são funcionais.

Consideremos, de modo similar ao da seção 3.2, um esquema de axiomas e regras de inferências para o cálculo de soluções funcionais

- S_1 - $a \longrightarrow b \longrightarrow a$
- S_2 - $a \longrightarrow (b \longrightarrow c) \longrightarrow (a \longrightarrow b) \longrightarrow (a \longrightarrow c)$
- S_3 - $((a \longrightarrow b) \longrightarrow a) \longrightarrow a$
- S_4 - $a \longrightarrow (a \vee b)$
- S_5 - $a \longrightarrow (b \vee a)$
- S_6 - $(a \longrightarrow c) \longrightarrow (b \longrightarrow c) \longrightarrow (a \vee b) \longrightarrow a$.
- S_7 - $a \wedge b \longrightarrow a$
- S_8 - $a \wedge b \longrightarrow b$
- S_9 - $(a \longrightarrow b) \longrightarrow (a \wedge b)$

MP Se a e $a \longrightarrow b$, então b .

Como no cálculo de problemas, temos também aqui, resultados similares aqueles, em decorrência dos axiomas e regra de inferência acima.

Devemos observar que um sistema formal de soluções funcionais foi utilizado para interpretar também o cálculo proposicional positivo clássico, uma vez, que a negação de funcionais não tem sentido lógico, isto é, se a é um funcional, $\neg a$ para ter sentido, deva ser interpretado (por ex: $\neg a, a^{-1}$, etc) e isso tira a condição de conectivo lógico que gostaríamos de dar a negação ($\neg$). Desse modo, nos parece razoável imaginar

a formação de funcionais através das regras P_1 , P_2 e P_3 , do início desta secção.

4. LÓGICA DE RESOLUÇÃO DE PROBLEMAS

Nesta secção faremos a junção dos dois sistemas formais vistos na secção 3, utilizando para isto, a definição 2.5 de satisfação. Usando expressões do tipo $a \Vdash A$, onde a e A são sentenças da lógica de soluções funcionais e de problemas, respectivamente, e as letras gregas $\alpha, \beta, \gamma, \dots$ para simbolizar as sentenças de um novo sistema formal, vamos construir as *fórmulas bem formadas* para o sistema formal de resolução de problemas do seguinte modo:

- 1 - As expressões do tipo $a \Vdash A$ e $a \nVdash A$ (a não satisfaz o problema A) são fórmulas bem formadas;
- 2 - Se α e β são fórmulas bem formadas então $\neg \alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$ e $\alpha \implies \beta$ também o são;
- 3 - As fórmulas bem formadas são somente as obtidas pelas regras acima.

O cálculo de resolução de problemas será definida através dos seguintes esquemas de axiomas e regras de inferência:

$$\text{Ax.1 - } \alpha \implies \beta \implies \alpha$$

$$\text{Ax.2 - } (\alpha \implies (\beta \implies \gamma)) \implies ((\alpha \implies \beta) \implies (\alpha \implies \gamma))$$

$$\text{Ax.3 - } (\alpha \implies \beta) \implies (\neg \beta \implies \neg \alpha)$$

e

$$\text{Rf.1 - } \frac{\alpha, \alpha \implies \beta}{\beta}$$

$$\text{Rf.2 - } \frac{a \Vdash A, b \Vdash B}{a \longrightarrow b \Vdash A \longrightarrow B} \quad \text{regra implicativa}$$

$$\text{Rf.3 - } \frac{a \Vdash A, b \Vdash B}{a \wedge b \Vdash A \wedge B} \quad \text{regra conjuntiva}$$

$$\text{Rf.4 - } \frac{a \Vdash A, b \Vdash B}{a \vee b \Vdash A \vee B} \quad \text{regra disjuntiva}$$

Se observarmos que os esquemas de axiomas acima e a regra Rf.1 são os mesmos para o cálculo proposicional clássico, podemos concluir que para as sentenças construídas a partir de sentenças do tipo $\alpha \Vdash A$, são válidos os resultados similares aos daquele cálculo clássico. Por exemplo, uma versão restrita do teorema da dedução como segue:

se

$$\frac{\alpha_1 \Vdash A_1, \alpha_2 \Vdash A_2, \dots, \alpha_n \Vdash A_n; \alpha \Vdash A}{b \Vdash B}$$

então

$$\frac{a_1 \mid \vdash A_1, a_2 \mid \vdash A_2, \dots, a_n \mid \vdash A_n}{a \mid \vdash A \Rightarrow b \mid \vdash B}.$$

O resultado que vem a seguir estabelece de modo mais completo a ligação, que pretendemos, entre os cálculos de soluções funcionais e problemas.

Teorema 4.1 - Se $a \mid \vdash A$, $b \mid \vdash B$ e $c \mid \vdash C$ então

- 1 - $a \rightarrow (b \rightarrow a) \mid \vdash A \rightarrow (B \rightarrow A)$
- 2 - $a \rightarrow (b \rightarrow c) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c)) \mid \vdash A \rightarrow (B \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
- 3 - $((a \rightarrow b)) \rightarrow a \rightarrow a \mid \vdash ((A \rightarrow B) \rightarrow A) \rightarrow A$
- 4 - $a \rightarrow a \vee b \mid \vdash A \rightarrow A \vee B$
- 5 - $a \rightarrow b \vee a \mid \vdash A \rightarrow B \vee A$
- 6 - $(a \rightarrow c) \rightarrow ((b \rightarrow c) \rightarrow (a \vee b \rightarrow a)) \mid \vdash (A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow (A \vee B \rightarrow A))$
- 7 - $a \wedge b \rightarrow a \mid \vdash A \wedge B \rightarrow A$
- 8 - $a \wedge b \rightarrow a \mid \vdash A \wedge B \rightarrow B$
- 9 - $(a \rightarrow b) \rightarrow (a \wedge b) \mid \vdash (A \rightarrow B) \rightarrow (A \wedge B)$

Faremos a prova do item 6 e deixamos ao leitor, as provas dos demais itens, que podem facilmente ser feitas de modo análogo.

Prova do item 6:

- 1 - $a \mid \vdash A$
- 2 - $b \mid \vdash B$ hipóteses
- 3 - $c \mid \vdash C$
- 4 - $a \vee b \mid \vdash A \vee B$ usando (1,2 e RF.4)
- 5 - $a \vee b \rightarrow a \mid \vdash A \vee B \rightarrow A$ (4,1 e RF.2)
- 6 - $b \rightarrow c \mid \vdash B \rightarrow C$ (2,3 e RF.2)
- 7 - $(b \rightarrow c) \rightarrow (a \wedge b \rightarrow a) \mid \vdash (B \rightarrow C) \rightarrow (A \wedge B \rightarrow A)$ (6,5 e RF.2)
- 8 - $a \rightarrow c \mid \vdash A \rightarrow C$ (1,3 e RF.2)
- 9 - $((a \rightarrow c) \rightarrow (b \rightarrow c) \rightarrow (a \vee b \rightarrow a)) \mid \vdash ((A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow (A \vee B \rightarrow A))$ (8,7 e RF.2) $\square$

Teorema 4.2 - Se $a_1, a_2, \dots, a_n$ é uma prova de a no cálculo de soluções funcionais e

$A_1, A_2, \dots, A_n$ é, correspondentemente, a prova de A no cálculo de problemas então $a \mid \vdash A$

Prova: Decorre diretamente da definição de prova para o cálculo proposicional positivo, o teorema 4.1 e a regra (modus ponens) RF. 1 $\square$

Corolário 4.3 - Se a e A são teoremas dos cálculos de solução funcionais e de problemas então $a \mid \vdash A$.

Prova: óbvia $\square$

Devemos observar que o teorema 4.1 compatibiliza o conceito de satisfação, do

cálculo de resolução de problemas, com as axiomáticas dos cálculos de soluções funcionais e de problemas, no sentido de preservar as estruturas sentenciais similares dos dois cálculos. O Teorema 4.2, decorrente fortemente do Teorema 4.1, expressa que a satisfação também preserva os mecanismos de derivação dos dois cálculos, o que era desejável!

Algumas sentenças do cálculo de resolução de problemas merecem especial atenção. Por exemplo, a sentença $a \Vdash A \rightarrow B$ deve ser entendida como $a \Vdash A \Rightarrow a \Vdash B$. Por outro lado, $a \rightarrow b \Vdash A$ deve ser entendida como $a \Vdash A \Rightarrow b \Vdash A$. Isso, entretanto, pode, no caso da sentença $a \rightarrow b \Vdash A \rightarrow B$, gerar sentenças não necessariamente equivalentes, quando o desdobramento for feito primeiramente pelo lado direito do sinal $\Vdash$ (1º caso) ou pelo lado esquerdo (2º caso). Porém, vale o seguinte resultado:

Proposição 4.4 - Se $a \Vdash B$ e $b \Vdash B$ então são logicamente equivalentes as sentenças abaixo.

- 1 - $(a \Vdash A \Rightarrow a \Vdash B) \Rightarrow (b \Vdash A \Rightarrow b \Vdash B)$
- 2 - $(a \Vdash A \Rightarrow b \Vdash A) \Rightarrow (a \Vdash B \Rightarrow b \Vdash B)$

Prova: fácil $\square$

O significado da proposição acima, é, que debaixo das hipóteses dadas, é indiferente que o desdobramento da sentença $a \rightarrow b \Vdash A \rightarrow B$ seja feito pela direita ou pela esquerda.

5. CONCLUSÃO

Este esboço tem o objetivo de fornecer indicações da viabilidade técnica do desenvolvimento de uma lógica de resolução de problemas baseada na conceituação de problemas, proposto em [2]. O estabelecimento de um cálculo de resolução de problemas constitui uma etapa imprescindível para o desenvolvimento posterior de máquinas de inferência de solução de problemas. Além do preenchimento das lacunas aqui deixadas e a exploração de muitos resultados desejáveis para esse cálculo, como por exemplo, uma versão mais geral do teorema de dedução, teoremas de compactação, a extensão para um cálculo de 1ª ordem etc, é natural que seja tentada uma modelação dentro de ambientes contextuais de problemas, ainda a nível de cálculo proposicional.

Gostaríamos de chamar a atenção sobre a idéia da ligação de dois sistemas formais aqui apresentada, sem que para isso tenha sido utilizada a noção de interpretação entre Teorias. Eventualmente, essa idéia poderia ser também usada para ligar outros sistemas formais, por exemplo, aritmética de Zermelo-Fraenkel e a lógica booleana.

REFERÊNCIAS

- [1] KOLMOGOROFF, A.N., Zur Deutung der Intuitionistischen Logik, Mathematische Zeitschrift, vol. 35, pp. 58-65, 1932.

- [2] LOPES, M.A., Introdução a uma Teoria Geral de Problemas, Tese de Doutorado, Departamento de Informática - PUC/RJ - maio/81.
- [3] RASIOWA, H., An Algebraic Approach to Non-classical Logics, North-Holland Publishing Company, 1974.
- [4] LOPES, M.A. & VELOSO, P.A.S., Operations on Problems and their Solution Spaces, Cybernetics and Systems Research, R. Trappe (ed.) North-Holland Publishing Company, 1982.