Vera Lúcia Prudência dos Santos

Desenvolvimento de uma Metodologia para Resolução de Problemas do Tipo Determine

Dissertação submetida ao Curso de Pós-Gradução em Informática do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba, como requisito parcial para a obtenção do grau de Mestre em Informática.

Manoel Agamemnon Lopes
(Orientador)

Campina Grande, Paraíba, Brasil

S237d Santos, Vera Lucia Prudencia dos
Desenvolvimento de uma metodologia para resolucao de problemas do Tipo Determine / Vera Lucia Prudencia dos Santos. - Campina Grande, 1997.
116 f.

Dissertaca (Mestrado em Informatica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Inteligencia Artificial 2. Teoria Geral de Problemas 3. Logica de Primeira Ordem 4. Dissertacao - Informatica I. Lopes, Manoel Agamemnon II. Universidade Federal da Paraiba - Campina Grande (PB)

CDU 004.8(043)

DESENVOLVIMENTO DE UMA METODOLOGIA DE RESOLUÇÃO PARA PROBLEMAS "TIPO DETERMINE"

VERA LUCIA PRUDÊNCIA DOS SANTOS

DISSERTAÇÃO APROVADA EM 03.07.1997

PROF. MANOEL AGAMEMNON LOPES, Dr.
Presidente

PROF. BERNARDO LULA JÚNIOR, Dr.
Examinador

Evandro de Barros Costa
PROF. EVANDRO DE BARROS COSTA, M.Sc
Examinador

PROF. BENJAMIN RENÉ CALLEJAS BEDREGAL, D.Sc
Examinador

CAMPINA GRANDE - PB

Este trabalho é dedicado

aos meus pais, João e Helena,

a Rita, Valdeci e família

Agradecimentos

A Deus, por tudo.

Aos meus pais, João e Helena, pelo amor, paciência e dedicação, às minhas irmãs, Eliana, Cau, Riso e Marcela e ao meu irmão, Luciano, pelo apoio e torcida.

Aos meus pais "postiços", Rita e Valdeci, e meus irmãozinhos, Valcir e Valnir, pela acolhida e apoio fundamental na minha estada em Campina Grande, pelo amor e paciência.

Ao meu orientador, Manuel Agamemnon Lopes, pela orientação acadêmico cientifíca e intelectual, pelo empenho com que tratou minhas limitações e pelo estímulo que deu aos meus esforços.

Ao amor da minha vida, Adriano, pelo carinho, principalmente nas horas difíceis, pelo companheirismo, pelo amor, por tudo.

À minha grande amiga Zane, obrigada pela sua amizade verdadeira e especial.

Ao amigo Haroldo, pela amizade, pelo incentivo e impulso inicial para realização deste, obrigada por tudo.

À Antônio Carlos, pelas discurssões e contribuições para este trabalho, obrigada pela força.

Aos professores, Edilson, Bernardo Lula, Jacques, Hattori (in memorian), Isabel, Ulrich e Hélio.

Aos amigos, Ricardo Lima e Gilson, pelo apoio durante o curso, pela amizade sincera e incentivo.

Aos amigos que conquistei em Campina Grande, Luciênio, Edberto, Paty, Mila, Malty, Rute, Ed, Cenidalva, Almir, Salete, Norma, Michelle, Leonardo e Ana Lígia, Mária Miranda, Ismênia, Arthur, Marcão, Daniele, Maurício, Liliane, Algeir, Walfredo, Fernanda e Dálmer, Cassandra (susto!!)..., pelos chopps no Willy Hauss, pelas festas, pelos risos...

Aos funcionários, Manuela, Zeneide, Teresinha, Josenilda, Lili, Agda, especialmente à Aninha e Vera, pela amizade, paciência e atenção.

À Inês e Jô, pela amizade e pelos lanches maravilhosos.

Aos amigos especiais de Feira de Santana, Cristóvão, Perpétua, Cláudio, Dja, Josete, Renato Taboca, Paulo Cueca, Washington, Kiko e Ângela, Enéas... obrigada pelo incentivo e amizade.

A todos os amigos que não foram aqui citados.

Resumo

Este trabalho apresenta o Desenvolvimento de uma Metodologia para Resolução de Problemas do "Tipo Determine", usando como processo de solução a *Skolemização*. O esboço da metodologia está dividido em duas etapas. Na primeira etapa temos a caracterização do problema, onde a partir do enunciado do problema verificamos se o objetivo deste é buscar resultados, que é a característica principal dos problemas da classe "tipo determine". Na segunda etapa trabalhamos a resolução do problema propriamente dita, descrevemos o problema estruturalmente, fazemos uma especificação formal do problema e, como último passo, usamos como processo de solução a *Skolemização*, chegando uma função solução para o problema, tornando mais simples o processo de resolução de um problema da classe. Para o desenvolvimento deste trabalho, foram utilizados os elementos da Teoria Geral de Problemas (TGP) e as técnicas de *Skolemização* baseada na Lógica de Primeira Ordem. Finalmente todo o mecanismo proposto permite que seja dado um passo fundamental no caminho de desenvolvimento de uma metodologia que resolva problemas de uma classe relativamente ampla.

Abstract

This work presents the Development of a Methodology for Solving "Determine Type" Problems, using *Skolemization* process as solution. The methodology draft is divided in two steps. On first step there is the problem caracterization, where regarding the problem definition, we verify if its target is seaching for results, principal caracteristc of problems a "Determine Type". On the second step we work on solving the problem, structurally describe it, formally describe it, and as the last task *Skolemization* solution process, find a function for the problem what makes even more simple the solution of such problem. General Problem Theory (GPT) and *Skolemization* technics based on the First Order Logic were used for developing this work. At the end, the proposed mechanism leads as to a fundamental step towards the development of a methodology which is able to solve problems of a rather big class.

Sumário

Lista de Ilustrações

Capítulo 1

Introdução

1.1 Motivação

Normalmente é difícil imaginar um momento em que a noção de problema e solução não esteja ligada ao homem, e isto se dá porque ele tem consciência das suas necessidades, ou seja, dos problemas a resolver e da sua capacidade de criar soluções ou estratégias para resolução destes.

A partir daí, surge então um sistema composto pelo problema, o solucionador e a solução. Para analisar tal sistema é preciso compreender cada uma dessas partes e conhecer as relações existentes entre elas.

Ao nos depararmos com um problema em nenhum momento fazemos uma reflexão sobre este e sua resolução. Isto acontece porque há um grande inconveniente de natureza metodológica quando é feita uma inferência a respeito da solubilidade de uma classe de problemas.

Em [Lop81], na Teoria Geral de Problemas (TGP), tem-se um formalismo das noções básicas de problemas e suas soluções, e também algumas relações entre problemas e soluções, o que permite definir formalmente alguns métodos para resolver classes relativamente amplas de problemas.

Em [Alm90], é apresentado um estudo das classes para as quais os domínios de dados e de resultados de seus problemas são estruturas relacionais do mesmo tipo e também outras classes onde todos os seus problemas são descritos através de um conjunto de problemas básicos, de modo que, um problema qualquer da classe é obtido operando-se sobre os problemas básicos, por meio de um determinado conjunto de operações. A solubilidade para este caso, acontece quando solucionamos os problemas da classe, "combinando-se" as soluções dos problemas básicos.

Neste trabalho tentaremos mostrar uma fundamentação teórica acerca de uma técnica de resolução para a classe de problemas do "tipo determine" através do estudo da especificação, tendo como base a TGP.

O objetivo principal deste estudo é desenvolver uma metodologia de resolução para a classe de problemas do "tipo determine" a partir de uma especificação contendo maior clareza e organização, levando o solucionador a encontrar uma solução do problema de forma mais rápida. Isso fará com que o processo de resolução de tais problemas se torne mais simples e eficiente.

O desenvolvimento de uma metodologia de resolução para a classe de problemas do "tipo determine", será feito utilizando as técnicas da *Skolemização* baseada no lógica de primeira ordem. Logo, a resolução para esta classe de problemas será descrita via tratamento lógico.

Acreditamos com este trabalho poder contribuir para o desenvolvimento da Teoria Geral de Problemas, através de uma análise formal da especificação da classe de problemas do "tipo determine" e o desenvolvimento de uma metodologia de resolução para esta classe de problemas.

1.2 Apresentação

O presente trabalho está disposto em oito capítulos, dos quais este é o primeiro. O leitor com um conhecimento a *priori* da Teoria Geral de Problemas, pode sem perda de continuidade, passar para o capítulo três e seguir para os demais. Porém, os mesmos devem ser lidos de forma sequencial.

No capítulo 2, introduzimos as noções básicas de 'algebra universal, os elementos da l'ogica de primeira ordem, as noções de tipos abstratos de dados e os elementos da Teoria Geral de Problemas e das estruturas relacionais empregadas nesta dissertação.

No capítulo 3, apresentamos os métodos de soluções de problemas. Mostramos que a redução de um problema a outro tem como objetivo reduzir as dificuldades para a obtenção da solução e tratamos a decomposição como uma estratégia geral para resolver problemas.

No capítulo 4, apresentamos uma análise sintática da intencionalidade de um problema, permitindo assim, decompor estruturalmente um problema em diversos subproblemas dos quais se conhece formalmente sua intenção e se tem uma idéia vaga e informal do resto dos componentes da estrutura.

No capítulo 5, caracterizamos a classe de problemas do "tipo determine", definindo o que vem a ser um problema desta classe e como se apresenta formalmente a sua intenção. E utilizamos a *Skolemização* como uma forma de se obter símbolos funcionais mais simples no processo de resolução de problemas da classe.

No capítulo 6, utilizamos os resultados do capítulo 5, para fornecer o desenvolvimento de uma metodologia de especificação de problemas do "tipo determine" e, para isto, usamos técnicas de especificação algébrica de tipos abstratos de dados, as quais supomos de conhecimento do leitor.

No capítulo 7, construimos o esboço da metodologia de resolução para a classe de problemas do "tipo determine", mostrando as etapas da análise do problema, partindo da apresentação de enunciado até chegar no processo de solução chamado *Skolemização*.

No capítulo 8, fazemos as considerações finais, apontamos também algumas linhas de pesquisa para futuros trabalhos que permitam dar uma continuidade ao que foi feito nesta dissertação.

Capítulo 2

Elementos da Teoria Geral de Problemas (TGP)

2.1 Introdução

O objetivo deste capítulo é o estudo das noções preliminares de Álgebra Universal, Elementos da Lógica de Primeira Ordem, Noções sobre Tipos Abstratos de Dados (TAD), Elementos da Teoria Geral de Problemas, que servem para ajudar no entendimento dos outros capítulos que aparecem neste trabalho.

As noções apresentadas aqui não serão textos de leitura independente, pois os resultados que aparecem são de breve explanação e por isso, não são mostradas as respectivas demonstrações. Com isso, fica atribuído ao leitor um pouco de conhecimento prévio a respeito desses assuntos, de maneira que esta visualização sirva apenas como uma revisão.

2.2 Noções de Álgebra Universal

Nesta seção partimos do ponto em que o leitor tem um certo conhecimento da Teoria Elementar dos Conjuntos. Os símbolos $\in$, $\cup$, $\cap$, $\neg$ e $\subseteq$ indicarão a *Pertinência*, a *União*, a *Interseção*, a *Complementação* e a relação de *Contido*. Para simbolizar o conjunto vazio usaremos $\emptyset$ e **U** para o conjunto universo.

Definição 2.2.1 *[Lop81] : Sejam A_1 e A_2 dois conjuntos não vazios. O produto cartesiano $A_1 \times A_2$ é definido como segue:*

$$A_1 \times A_2 = \{x \mid \exists a\ \exists b\ (a \in A_1 \land b \in A_2 \land x = (a,b))\}$$

Se $A_1 = A_2 = A$ denotamos por A^2 o produto $A_1 \times A_2$.

Definição 2.2.2 *[Lop81] : Uma relação n-ária $\mathcal{R}$ entre $A_1, A_2, A_3, ..., A_n$ é qualquer subconjunto $A_1 \times A_2 \times ... \times A_n$, isto é, $\mathcal{R}$ é uma relação n-ária, se e somente se, $\forall x \in R \Longrightarrow \exists (x_1, x_2, ..., x_{n-1})\ \exists x_n\ (x_1, x_2, ..., x_n) \in A_1 \times A_2 \times ... \times A_n \land x = (x_1, x_2, ..., x_n)$.*

Definição 2.2.3 *[Lop81] : Seja $\mathcal{R}$ uma relação binária. Então*

$dom\ \mathcal{R} = \{a \mid \exists b\ (a,b) \in R\}$ - *domínio de* $\mathcal{R}$

$Im\ \mathcal{R} = \{b \mid \exists a\ (a,b) \in R\}$ - *imagem de* $\mathcal{R}$

$\mathcal{R}^{-1} = \{(b,a) \mid (a,b) \in \mathcal{R}\ \}$ - *inversa de* $\mathcal{R}$

$\mathcal{R} \mid A' = \{(a,b) \mid (a,b) \in \mathcal{R} \land a \in A'\}$ - *restrição de* $\mathcal{R}$ *com respeito a* A'.

Definição 2.2.4 *[Lop81] : Uma relação binária $\mathcal{R}$ sobre A é dita uma relação de equivalência, se e somente se, são satisfeitas ás propriedades abaixo:*

- $\forall a \in A\ (a,a) \in \mathcal{R}$ *(reflexividade)*
- $\forall a, b \in A\ (a,b) \in \mathcal{R} \Longrightarrow (b,a) \in \mathcal{R}$ *(simétrica)*
- $\forall a, b, c \in A\ (a,b) \in \mathcal{R} \land (b,c) \in \mathcal{R} \Longrightarrow (a,c) \in \mathcal{R}$ *(transitividade)*

Definição 2.2.5 *[Lop81] : Uma relação binária $\mathcal{R}$ sobre A é dita uma relação de ordem parcial sobre A, se e somente se, são satisfeitas as propriedades 1 e 3 acima e mais a seguinte:*

- $\forall a, b \in A \ (a, b) \in \mathcal{R} \wedge (b, a) \in \mathcal{R} \Longrightarrow a = b$ *(antisimétrica)*

Definição 2.2.6 *[Lop81] : Uma relação entre A_1 e A_2, é dita uma aplicação de A_1 em A_2, se e somente se, satisfaz às seguintes propriedades:*

- $\forall x \in A_1 \ \exists y \in A_2 \ (x, y) \in R$ *(totalidade)*
- $\forall x \in A_1 \ \exists y, z \in A_2 \ (x, y) \in R \wedge (x, z) \in R \Longrightarrow y = z$ *(univocidade)*
- *Sejam A e B dois conjuntos. Chamamos de função de A em B, denotado por $f : A \longrightarrow B$, a uma regra que a cada elemento de A associa um único elemento de B, de tal modo que: $\forall x \in A \ \exists y \in B$ tal que $f(x) = y$.*
- *Uma função de A em B é dita injetiva, se e somente se, $\forall x, y \in A \ f(x) = f(y) \Longrightarrow x = y$; é dita sobrejetiva, se e somente se, $\forall y \in B \ \exists x \in A \ f(x) = y$. Uma função que é simultaneamente injetiva e sobrejetiva é dita bijetiva.*

Definição 2.2.7 *[Lop81] : Sejam $f : A \longrightarrow B$, $g : B \longrightarrow C$ e $h : A \longrightarrow C$ três funções. A função h é dita a composta de f e g, se e somente se, $\forall x \in A \ h(x) = g(f(x))$. Denota-se por $h = g \circ f$.*

Uma *relação de equivalência* sobre um conjunto A é definida como uma relação entre pares de elementos de A. Se $x, y \in A$, então escrevemos $xRy \in A$ se x estiver relacionado com y, e $x \not{R} y$ se x não estiver relacionado com y.

Definição 2.2.8 *[Lop81] : Uma estrutura relacional $\mathcal{A}$ é um par $< A, R >$ onde A é um conjunto não vazio chamado base e R é uma família finita de relações sobre A. Uma álgebra é uma estrutura relacional, onde a igualdade (=) está em R e todas as demais relações de R são aplicações sobre A.*

Consideremos $B \subseteq A$, a estrutura $B =< B, R >$ é chamada de subestrutura de $\mathcal{A}$. Deve-se observar que numa subestrutura as operações são mantidas sobre um subdomínio da estrutura original.

Se $\leq$ é uma relação de ordem sobre o conjunto A, dizemos que $\leq$ *ordena* A, ou que $\leq$ *ordena parcialmente* A, ou que é uma *ordem* sobre A.

Exemplo 2.2.0.1 : *Dizemos que o par* $< A, \leq >$ *é um sistema ordenado ou um sitema parcialmente ordenado. Dizemos também que* A *é um conjunto ordenado pela ordem* $\leq$. *Se para todo* $a, b \in A$ *temos* $a \leq b$ *ou* $b \leq a$ *o sistema parcialmente ordenado* A *é chamado de caminho ou sistema totalmente ordenado.*

Definição 2.2.9 *[Lop81] : Um reticulado é uma álgebra* $\mathcal{A} = \langle A, =, \wedge, \vee \rangle$, *onde* $=$ *é a igualdade em* A, $\wedge$ *e* $\vee$ *são operações binárias sobre* A, *chamadas "e" e "ou" satisfazendo as seguintes condições:*

1. $\forall a \in A \; a \wedge a = a$ e $a \vee a = a$
2. $\forall a, b \in A \; a \wedge b = b \wedge a$ e $a \vee b = b \vee a$
3. $\forall a, b, c \in A \; (a \wedge b) \wedge c = a \wedge (b \wedge c)$ e $(a \vee b) \vee c = a \vee (b \vee c)$
4. $\forall a, b \in A \; a \wedge (a \vee b) = a$ e $a \vee (a \wedge b) = a$

Um reticulado $\langle A, \leq \rangle$ diz-se *completo* se todo subconjunto não vazio e majorado admitir um supremo e um ínfimo.

Definição 2.2.10 *[Lop81] : Um reticulado* $\langle A, \leq \rangle$ *diz-se distributivo se satisfaz ás seguintes condições:*

1. $\forall a, b, c \in A$, *tem-se que* $a \wedge (b \vee c) \leq (a \wedge b) \vee (a \wedge c)$.
2. $\forall a, b, c \in A$, *tem-se que* $a \vee (b \wedge c) \geq (a \vee b) \wedge (a \vee c)$.

Definição 2.2.11 *[Lop81] : Uma álgebra booleana é uma estrutura tipo* $\mathcal{A} = \langle A, =, \wedge, \vee, ', 0, 1\rangle$, *onde* $'$ *é uma operação unária sobre A e, 0 e 1 operações nulárias sobre A (constantes de A) satisfazendo ás propriedades abaixo:*

1. $\langle A, =, \wedge, \vee\rangle$ *é um reticulado distributivo*
2. $\forall a \in A\ 0 \vee a = a$ e $1 \wedge a = a$
3. $\forall a \in A\ a \vee a' = 1$ e $a \wedge a' = 0$

Uma questão muito importante e polêmica na teoria dos conjuntos, é o Axioma da Escolha , que afirma que, dado qualquer conjunto de conjuntos mutuamente disjuntos, não-vazios, existe pelo menos um conjunto que contém um e um só elemento em comum com cada um dos conjuntos não-vazios (axioma devido a Zermelo).

2.3 Elementos da Lógica de Primeira Ordem

Esta seção apresenta os principais conceitos da Lógica de Primeira Ordem. Esta lógica pode ser vista como um sistema formal apropriado à definição de teorias do universo de discurso da Matemática.

Definição 2.3.1 *[Cas87] : Um alfabeto de primeira ordem* $\mathcal{A}$ *consiste de:*

símbolos lógicos:

: (,)

: $\{\wedge, \vee, \neg, \Longrightarrow, \Longleftrightarrow, \}$

onde:

$\wedge$ = *conjunção*

$\vee$ = *disjunção*

$\neg$ = *negação*

$\Longrightarrow$ = *implicação*

$\Longleftrightarrow$ = *bi-implicação*

: { $\forall$ *(universal)*, $\exists$ *(existencial)* }

: $V = \{x_1, x_2, ..., x_n\}$

: (=)

símbolos não lógicos:

$\mathcal{C} = \{c_1, c_2, ...\}$ *(símbolos constantes)*

$\mathcal{F} = \{f_1, f_2, ...\}$ *(símbolos funcionais)*

$\mathcal{P} = \{p_1, p_2, ...\}$ *(símbolos predicativos)*

onde para cada símbolo de $\mathcal{F}$ e $\mathcal{P}$ é associado um número $n \in N$ $n \geq 1$ chamado aridade. Sendo que, a aridade dos símbolos constantes é igual a zero.

Definiremos a seguir uma sintaxe da linguagem de primeira ordem.

Definição 2.3.2 *[Cas87] : O conjunto dos termos de primeira ordem sobre um alfabeto de primeira ordem $\mathcal{A}$ é o menor conjunto satisfazendo às seguintes condições:*

1. *toda variável em $\mathcal{A}$ é um termo sobre $\mathcal{A}$;*
2. *toda constante em $\mathcal{A}$ é um termo sobre $\mathcal{A}$;*
3. *se $t_1, ..., t_n$ são termos sobre $\mathcal{A}$ e f é um símbolo funcional n-ário de $\mathcal{A}$, então $f(t_1, ..., t_n)$ também é um termo sobre $\mathcal{A}$.*

Definição 2.3.3 *[Cas87] : O conjunto das fórmulas sobre um alfabeto de primeira ordem $\mathcal{A}$ é o menor conjunto satisfazendo às seguintes condições:*

1. *se $t_1, ..., t_n$ são termos sobre $\mathcal{A}$ e p é um símbolo relacional n-ário de $\mathcal{A}$, então $p(t_1, ..., t_n)$ é uma fórmula sobre $\mathcal{A}$, denominada fórmula atômica.*

2. *se* t_1 *e* t_2 *são termos sobre* $\mathcal{A}$ *e* $=$ *é um símbolo de* $\mathcal{A}$*, então* $(t_1 = t_2)$ *é uma fórmula sobre* $\mathcal{A}$*, também denominada fórmula atômica.*

3. *se* α *e* β *são fórmulas sobre* $\mathcal{A}$*, então* $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \Longrightarrow \beta)$, $(\alpha \Longleftrightarrow \beta)$ *também são fórmulas sobre* $\mathcal{A}$.

4. *se* α *é uma fórmula sobre* $\mathcal{A}$ *e* x *é uma variável de* $\mathcal{A}$*, então* $\forall x\ (\alpha)$ *e* $\exists x\ (\alpha)$ *também são fórmulas sobre* $\mathcal{A}$.

Definição 2.3.4 *[Cas87] : A linguagem de primeira ordem* $\mathcal{L}$ *sobre um alfabeto de primeira ordem* $\mathcal{A}$ *é um terno* $\langle \mathcal{P}_L, \mathcal{F}_L, \mathcal{C}_L \rangle$*, onde* $\mathcal{P}_L \subset \mathcal{P}$, $\mathcal{F}_L \subset \mathcal{F}$, $\mathcal{C}_L \subset \mathcal{C}$.

Definição 2.3.5 *[Cas87] :*

- *Em uma fórmula da forma* $\forall x\ \beta$ *ou* $\exists x\ \beta$, β *é o escopo do quantificador* $\forall x$ *ou de* $\exists x$ *e* x *é a variável do quantificador.*
- *Uma ocorrência de uma variável* x *em uma fórmula* α *é ligada em* α*, se a ocorrência se dá em uma subfórmula de* α *da forma* $\forall x\ (\beta)$ *ou da forma* $\exists x\ (\beta)$*. Caso contrário, a ocorrência de* x *é livre.*

Definição 2.3.6 *[Peq77] : Seja* α *uma fórmula e* x *uma variável, dizemos que uma certa ocorrência de* x *em* α *é livre em* α*, se e somente se:*

1. α *é uma fórmula atômica, ou*

2. α *é da forma* $(\beta \wedge \gamma)$, $(\beta \vee \gamma)$, $(\beta \Longrightarrow \gamma)$ *ou* $(\beta \Longleftrightarrow \gamma)$*, e a ocorrência considerada de* x *em* α *é uma ocorrência livre de* x *em* β *ou* γ*, ou*

3. α *é da forma* $\neg\beta$ *e a ocorrência de* x *é uma ocorrência livre em* β*, ou*

4. α *é da forma* $\forall y\ \beta$ *ou* $\exists y\ \beta$ *, e a considerada ocorrência de* x *é uma ocorrência livre em* β*, e* y *e* x *são variáveis distintas.*

Definição 2.3.7 *[Peq77] : Uma estrutura é uma tripla $\mathcal{E} = \langle \mathcal{D}, \mathcal{R}, \phi \rangle$, onde:*

$\mathcal{D} \longrightarrow$ *é um conjunto não-vazio chamado de universo ou domínio da estrutura $\mathcal{E}$.*

$\mathcal{R} \longrightarrow$ *é uma família de relações $\mathcal{R} \subseteq \mathcal{D}^j$, $1 < j < \infty$*

$\phi \longrightarrow$ *é uma família de funções $\phi : \mathcal{D}^j \longrightarrow \mathcal{D}$, $j \geq 0$.*

Definição 2.3.8 *[Peq77] : Seja $\mathcal{L}$ uma linguagem com notação $\langle \mathcal{P}_L, \mathcal{F}_L, \mathcal{C}_L \rangle$ e $\mathcal{D}$ um conjunto não-vazio, uma interpretação $\mathcal{I}$ de $\mathcal{L}$ em $\mathcal{D}$ é uma estrutura $\mathcal{I} = \langle \mathcal{D}, \mathcal{R}, \phi \rangle$.*

1. *Para cada $p \in \mathcal{P}_L$, define-se uma relação $p^{\mathcal{I}} \subseteq \mathcal{D}^n$ e $\mathcal{R} = \{p^{\mathcal{I}} \mid p \in \mathcal{P}_L\}$*

2. *Para cada $f \in \mathcal{F}_L$, define-se uma função $f^{\mathcal{I}} : \mathcal{D}^n \longrightarrow \mathcal{D}$, e $\phi = \{f^{\mathcal{I}} \mid f \in \mathcal{F}_L\}$*

3. *Para cada $a \in \mathcal{C}_L$ seleciona-se um particular elemento $a^{\mathcal{I}} \in \mathcal{D}$.*

Definição 2.3.9 *[Peq77] : Seja $\mathcal{L}$ uma linguagem de primeira ordem e $\mathcal{I}$ uma interpretação para $\mathcal{L}$ com universo $\mathcal{D}$. Uma função valorização para $\mathcal{L}$ sobre $\mathcal{I}$ é qualquer função s que associa a cada variável de $\mathcal{L}$ um elemento de $\mathcal{D}$, denotado por $s : \mathcal{V}_L \longrightarrow \mathcal{D}$, onde:*

$\mathcal{V}_L$ é o conjunto das variáveis de $\mathcal{L}$ e s tem como extensão ao conjunto $\mathcal{T}_L$ dos termos de $\mathcal{L}$, a seguinte função: $\overline{s} = \mathcal{T}_L \longrightarrow \mathcal{D}$, tal que:

$$\overline{s}(x) = s(x)$$

$$\overline{s}(c) = C^I$$

$$\overline{s}(f(t_1, ..., t_n)) = f^{\mathcal{I}}(\overline{s}(t_1), ..., \overline{s}(t_n)).$$

A função s chama-se função semântica de avaliação de $\mathcal{L}$ em $\mathcal{I}$. Vamos denotar s^{c_x} a substituição em s de x por c observando que:

Definição 2.3.10 *: Uma interpretação $\mathcal{I}$ satisfaz a uma fórmula α, com avaliação s, denotada por $\mathcal{I} \models^s \alpha$, como segue abaixo:*

- $\alpha \equiv t_1 = t_2$ $\mathcal{I} \models^s (t_1 = t_2)$ sse $t_1^{\mathcal{I}} = t_2^{\mathcal{I}}$
- $\alpha \equiv r(t_1, t_2, ..., t_n)$ $\mathcal{I} \models^s r(t_1, t_2, ..., t_n)$ sse $s(t_1), ..., s(t_n) \in r^{\mathcal{I}}$
- $\alpha \equiv (\neg\beta)$ $\mathcal{I} \models^s (\neg\beta)$ sse $\mathcal{I} \not\models^s \beta$
- $\alpha \equiv (\beta \wedge \gamma)$ $\mathcal{I} \models^s (\beta \wedge \gamma)$ sse $\mathcal{I} \models^s \beta$ e $\mathcal{I} \models^s \gamma$
- $\alpha \equiv (\beta \vee \gamma)$ $\mathcal{I} \models^s (\beta \vee \gamma)$ sse $\mathcal{I} \models^s \beta$ ou $\mathcal{I} \models^s \gamma$
- $\alpha \equiv (\beta \longrightarrow \gamma)$ $\mathcal{I} \models^s (\beta \longrightarrow \gamma)$ sse $\mathcal{I} \not\models^s \beta$ ou $\mathcal{I} \models^s \gamma$
- $\alpha \equiv (\forall x\ \beta)$ $\mathcal{I} \models^s (\forall x\ \beta)$ sse $\forall d \in \mathcal{D}$ $\mathcal{I} \models^{s_x^d} \beta$
- $\alpha \equiv (\exists x\ \beta)$ $\mathcal{I} \models^s (\exists x\ \beta)$ sse $\exists d \in \mathcal{D}$ $\mathcal{I} \models^{s_x^d} \beta$

Definição 2.3.11 : *Sejam $\mathcal{L}$ uma linguagem e $\mathcal{I}$ uma interpretação de $\mathcal{L}$:*

- *Uma fórmula α é dita* válida *se e somente se, para qualquer avaliação s tem-se $\mathcal{I} \models^s \alpha$. Denotado por $\mathcal{I} \models \alpha$.*
- *Uma fórmula α é dita* verdadeira *se e somente se α é válida para qualquer interpretação $\mathcal{I}$. Denotado por $\models \alpha$.*

Definição 2.3.12 : *Uma interpretação $\mathcal{I}$ é denominada um* modelo *para um conjunto Γ de sentença de $\mathcal{L}$, se para todo $\alpha \in \Gamma$, $\mathcal{I} \models \alpha$.*

Definição 2.3.13 : *Uma fórmula $\alpha \in \mathcal{L}$ é uma* consequência lógica *de um conjunto Γ de fórmula de $\mathcal{L}$, denotado por $\Gamma \models \alpha$, se e somente se, para qualquer avaliação s tem-se $\mathcal{I} \models^s \Gamma$ então $\mathcal{I} \models^s \alpha$.*

Definição 2.3.14 : *Uma* teoria *de uma linguagem $\mathcal{L}$ é qualquer conjunto $\mathcal{T}$ de sentenças de $\mathcal{L}$ fechado sob consequência lógica.*

Definição 2.3.15 : *Um sistema dedutivo de primeira ordem $\mathcal{S}$ é uma tripla $\mathcal{S} = \langle \mathcal{L}, \mathcal{A}, \mathcal{R} \rangle$, onde:*

> $\mathcal{L}$ *é uma linguagem de primeira ordem,*
>
> $\mathcal{A}$ *é um conjunto de fórmulas especiais chamadas axiomas,*
>
> $\mathcal{R}$ *é um conjunto de regras operacionais chamadas de inferência.*

Se $\langle \alpha_1, \alpha_2, ..., \alpha_r, \alpha_{r+1} \rangle \in \mathcal{R}$, onde $\mathcal{R}$ é uma regra de inferência, dizemos que $\alpha_1, \alpha_2, ..., \alpha_r$ são *premissas* de α_{r+1} e que α_{r+1} é uma *conclusão* de $\alpha_1, \alpha_2, ..., \alpha_r$.

Definição 2.3.16 : *Uma fórmula α é um* teorema *de um conjunto de fórmulas P em $\mathcal{A}$, denotado por $P \vdash \mathcal{A}$, se e somente se existir uma sequência $(\alpha_0, ..., \alpha_n)$ de fórmulas de $\mathcal{L}$ tal que $\alpha_n = \alpha$ e, para todo $i \leq n$, um dos casos abaixo é satisfeito:*

1. *α_i é um axioma de $\mathcal{A}$ ou α_i ocorre em P;*
2. *existem $j, k < i$ tais que α_i pode ser obtida por Modus Ponens de α_j e α_k.*

Definição 2.3.17 : *Seja α uma fórmula de $\mathcal{L}$. Uma* prova *ou dedução de α, a partir de um conjunto P de fórmulas de $\mathcal{L}$, é uma sequência $\alpha_1, \alpha_2, ..., \alpha_n$ tal que:*

1. *$\alpha_n = \alpha$*
2. *$\alpha_i \in v \cup \mathcal{A}$ ou α_i é obtida por alguma regra de inferência das fórmulas $\alpha_{i_1}, \alpha_{i_2}, ..., \alpha_{i_n}$, onde $1 \leq j \leq n$ e $i_j < i$.*

Definição 2.3.18 :

1. *Dada uma fórmula α, com variáveis livres $x_1, ..., x_n$, o* fecho universal *de α é a fórmula $\forall x_1 ... \forall x_n\ (\alpha)$ e o* fecho existencial *de α é a fórmula $\exists x_1 ... \exists x_n\ (\alpha)$.*

2. *Uma fórmula α é uma conjunção se e somente se, $\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n$.*

3. *Uma fórmula α é uma disjunção se e somente se, for da forma $\alpha_1 \vee \alpha_2 \vee ... \vee \alpha_n$.*

Definição 2.3.19 *[Peq77] : Seja $\mathcal{L}$ uma linguagem e α uma fórmula. Dizemos que α é uma forma normal conjuntiva (f.n.c.) se α é da forma $(\alpha_1 \wedge (\alpha_2 \wedge (... \wedge (\alpha_{n-1} \wedge \alpha_n)...)))$ onde cada α_i é da forma $(\alpha_{i_1} \vee (\alpha_{i_2} \vee (...((\alpha_{i_{n_{i-1}}} \vee \alpha_{i_{n_i}})...))$, e cada α_{i_j} é uma fórmula atômica ou a negação de uma fórmula atômica. Dizemos também que cada α_{i_j} é um literal e que cada α_i é uma cláusula. Assim uma f.n.c. é uma conjunção de cláusulas e uma cláusula é uma disjunção de literais. Uma cláusula que não contém literal é chamada cláusula vazia.*

Definição 2.3.20 *[Peq77] : Seja $\mathcal{L}$ uma linguagem e α uma fórmula. Dizemos que α está na forma normal prenex, se e somente se α é livre de quantificadores ou α é da forma $Q_1 x_{i_1} Q_2 x_{i_2} ... Q_n x_{i_n}\ \beta$, onde β é livre de quantificadores e cada Q_i é $\forall$ ou $\exists$.*

Definição 2.3.21 *[Peq77] : Seja α uma fórmula na forma normal prenex e $\mathcal{L}$ uma linguagem, sendo $\alpha \in \mathcal{L}$. A Skolemização de α, denotada por $Skl(\alpha)$, é a wff obtida pelo procedimento recursivo visto abaixo:*

- *Se α é $\exists y\ \beta$, então $Skl(\alpha) = Skl(\beta^a_y)$, onde a é a primeira constante nova.*

- *Se α é $\forall x_{i_1}\ \forall x_{i_2} ...\ \forall x_{i_n}\ \exists y\ \beta$, então $Skl(\alpha) = \forall x_{i_1}\ \forall x_{i_2}\ ...\ \forall x_{i_n}\ Skl(\beta^{f(x_{i_1},...,x_n)}_y)$ onde f é o primeiro símbolo funcional novo.*

- *Se nenhum quantificador existencial ocorre em α, então $Skl(\alpha) = \alpha$.*

2.4 Noções Sobre Tipos Abstratos de Dados

Um tipo de dado consiste basicamente de um ou mais domínios de objetos munidos de operações e relações. Por exemplo, o tipo de dados dos números naturais tem como domínio o conjunto N dos naturais, como operações soma e produto e relação de menor.

Quando trabalhamos em um nível próximo ao problema implica numa abstração de dados. Tal nível desempenha para os dados, de certa forma, papel análogo ao que procedimentos desempenham para o fluxo de controle. Devemos classificar abstrações de dados não por suas similaridades de estrutura de representação, mas sim por seu comportamento exibido. A uma tal abstração denominamos *tipo abstrato de dados* (TAD), e o adjetivo abstrato enfatiza o fato de que estamos nos concentrando no comportamento dos objetos, e nos abstraimos dos destalhes de sua representação.

Veremos a idéia de Tipos Abstratos de Dados do ponto de vista estrutural, pois se torna mais adequada para especificação de problemas.

As estruturas relacionais para linguagens de primeira ordem são utilizadas para especificar tipos de dados. Os tipos abstratos de dados são especificados através de estruturas iniciais de uma classe de estrutura de mesmo tipo. A inicialidade é baseada na idéia de estruturas nomeáveis ou atingíveis e modelos mínimos. Uma estrutura para uma linguagem $\mathcal{L}$ é definida da seguinte forma $\mathcal{I} = \langle D, \mathcal{P}, \mathcal{F}, \mathcal{C} \rangle$, cujas componentes estão definidas em 2.3.8.

Dizemos que uma estrutura é atingível quando seu domínio é constituído somente pelos termos constantes de $\mathcal{L}$, isto é, pelas constantes de $\mathcal{L}$ e por todos os termos obtidos das constantes por aplicação das funções $f \in \mathcal{F}$. A importância dessas estruturas é justificada por algumas propriedades utilizadas no estudo de TAD, vistas abaixo:

- Se $\mathcal{I} = \langle D, \mathcal{P}, \mathcal{F}, \mathcal{C} \rangle$ é uma estrutura atingível para $\mathcal{L}$ e $\mathcal{I}'$ é uma outra estrutura também para linguagem $\mathcal{L}$, então existem, no máximo, um homomorfismo de $\mathcal{I}$ em $\mathcal{I}'$ e que será sobrejetivo, se e somente se, $\mathcal{I}'$ é também atingível.
- Se $\mathcal{I}$ é uma estrutura atingível para $\mathcal{L}$ e $\equiv$ é uma congruência em $\mathcal{I}$, então a estrutura obtida pela classificação de $\mathcal{I}$ através de $\equiv$, $(\mathcal{I}/\equiv)$, é também atingível.
- Se $\mathcal{I}$ é uma estrutura atingível sobre $\mathcal{L}$, então existe uma interpretação de Herbrand única $\mathcal{H}(\mathcal{I})$ tal que $\mathcal{I}$ é isomorfa a $\mathcal{H}(\mathcal{I})/\mathrm{Ker}(\phi)$, onde ϕ é o único homomorfismo de $\mathcal{H}(\mathcal{I})$ em $\mathcal{I}$ e $\mathrm{Ker}(\phi)$ é o núcleo de ϕ.
- Se duas estruturas atingíveis $\mathcal{I}$ e $\mathcal{I}'$ são iniciais na mesma classe, então elas são isomorfas.

A especificação pode ser vista como uma maneira de procurar propriedades do tipo $\mathcal{I}$ e expressá-las através de sentenças axiomáticas com validade verificada no modelo interpretativo $\mathcal{I}$ da linguagem e correspondentes. Se esse modelo é inicial o conjunto de axiomas é bastante para caracterizar o tipo.

Exemplo 2.4.0.2 : *Seja a estrutura* $\mathcal{N} = \langle N, \emptyset, S \rangle$ *dos naturais com zero e sucessor, onde uma axiomatização possível é a seguinte:*

- $\forall x \ (S(x) \neq 0)$
- $\forall x \ \forall y (S(x) = S(y) \Longrightarrow x = y)$
- $\forall y \ (y \neq 0 \Longrightarrow \exists x (y = S(x))$
- $\forall x \ (S(x) \neq x)$
- $\forall x \ (S(S(x) \neq x) \ldots\ldots$
- $\forall x \ (S^n(x) \neq x)$

Observamos que mesmo $\mathcal{N}$ sendo uma estrutura que satisfaz aos axiomas mostrados acima, ela não é única, pois qualquer outra estrutura que seja uma cópia de N seguida de cópias de inteiros, também satisfaz. Com isso, há uma evidência da limitação referida do poder da caracterização do método, que pode ser contornada, eventualmente, através da interseção de todas essas estruturas, visando a obtenção do modelo mínimo.

2.5 Elementos da Teoria Geral de Problemas

2.5.1 Introdução

Desde os primeiros tempos o homem tem enfrentado e resolvido problemas, tentando assim chegar a uma metodologia que lhe permita solucionar qualquer problema parecido.

Descartes (1596-1650), com seu método analítico, quando introduziu as coordenadas cartesianas na geometria, fez a primeira formalização matemática da questão de resolução de problemas, reconhecendo a possibilidade de um estudo sobre solubilidade de um problema, independente dos recursos utilizados na solução. Outros grandes pensadores como Leibnitz (1646-1716), também contribuíram com a questão do estabelecimento de uma metodologia de resolução de problemas. Kolmogoroff em 1932, descreveu uma sintaxe da lógica intuicionista, dando a ela uma semântica de problemas [LA85].

Encontrar um método seguro e completo para resolver qualquer tipo de problema, ou pelo menos uma classe de problemas que seja relativamente ampla e distingüível, é uma idéia que ainda se encontra em aberto. A Teoria Geral de Problemas (TGP) foi desenvolvida com a intenção de se tentar formalizar o conceito, que era intuitivo, de problemas desvinculando-o de qualquer contexto, estudando as relações entre problemas e fazendo uma análise e crítica das soluções, o que leva aos primeiros passos nesta direção.

Nesta seção, veremos alguns conceitos e elementos básicos da TGP [Lop81] que servem de apoio aos outros capítulos que se desenvolveram neste trabalho.

2.5.2 Elementos Básicos

A Teoria Geral de Problemas (TGP) propõe, a partir de uma análise crítica de noções usuais, definir precisamente o conceito de problema como uma estrutura matemática, onde suas soluções são funções. Fazendo isso, cria-se um ambiente adequado para se examinar rigorosamente os questionamentos relacionados a problemas, como por exemplo, solubilidade, métodos de transformação (homomorfismos, isomorfismos, analogias, etc.), resoluções de problemas, etc..

A maioria dos enfoques dados a problemas são voltados para as seguintes questões: "Qual é o problema?" e "Como resolvê-lo?". A TGP trata das questões menos usualmente abordadas, tais como: "O que é um problema?" e "O que é uma solução?", com uma postura mais de análise e crítica.

A formulação adotada pela TGP, tenta mostrar os elementos que são observados intuitivamente, captando alguns elementos estruturais, os quais distinguem invariavelmente o que se conhece como "tipo problema", como por exemplo:

> "Transformar um número hexadecimal em um número binário correspondente."

A partir do enunciado pode-se distinguir três elementos básicos: o conjunto dos números hexadecimais (conjunto dos dados do problema). O conjunto dos números binários (conjunto dos resultados) e a intenção de transformar um elemento no outro, que podemos expressar assim: dado um número hexadecimal x e um número binário y então, $(x, y) \in$ intenção, se e somente se , y é a representação binária de x.

A figura abaixo mostra uma estrutura que capta os elementos essenciais à abstração de problemas, onde os dados serão representados por D, os resultados por R e a intenção por r.

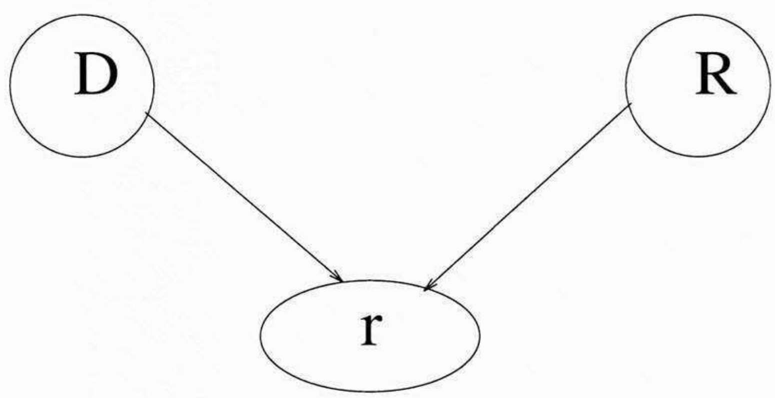

Figura 2.1: *Estrutura Abstrata de um pré-problema*

O domínio de dados (D) e o domínio de resultados (R), representam a parte substantiva do pré-problema definido a seguir, enquanto que a intenção, representa a parte subjetiva deste.

Então, podemos formalizar a definição de pré-problema como mostrado abaixo:

Definição 2.5.1 *[Lop81] : Um pré-problema P, é uma estrutura do tipo $P = \langle D, R, r \rangle$, onde D e R são conjuntos não-vazios, chamados de domínio dos dados e domínio dos resultados respectivamente, e r é um predicado intencional, cuja interpretação é um subconjunto de $D \times R$.*

Exemplo 2.5.2.1 *: Seja P um problema de "Achar o maior elemento de uma sequência de números naturais."*

Uma estruturação para P, pode ser feita através da interpretação abaixo:

$$P = \langle Seq, \mathcal{N}, Maior \rangle,$$

onde,

Seq = conjunto das sequências de números naturais

$\mathcal{N}$ = conjunto dos números naturais

$(s, n) \in Maior \iff n \in s \wedge \forall s \in Seq,\ n \geq s$

tal que,

$\geq$ é relação binária de ordem "maior ou igual" usual no conjunto dos números naturais, com $s \in Seq$ e $n \in \mathcal{N}$.

Observando o exemplo mostrado acima vemos que a definição de pré-problema é muito ampla e envolve elementos não relevantes ao problema (quais as seqüências que interessam? Qualquer solução serve?). Notamos que não são necessárias a utilização de seqüências infinitas e com isso vemos que a solução não será qualquer uma. Isso nos leva a perceber que é preciso um enriquecimento da parte intencional do problema, a qual tem a responsabilidade de dar sentido ao problema. A elaboração de uma definição mais completa é vista a seguir:

Definição 2.5.2 *[Lop81] : Um problema é uma estrutura do tipo $P = \langle D, R, d, r, A \rangle$, onde D e R são conjuntos não vazios, denominados respectivamente como domínio dos dados e domínio dos resultados, $d \subseteq D$ é um predicado unário sobre D que denota a* *restrição sobre os dados, $r \subseteq D \times R$ um predicado binário sobre $D \times R$, que denota a forma como se relacionam os dados com os resultados, e $A \subseteq R^D$ um predicado unário sobre R^D que denota o predicado de admissibilidade.*

As componentes D e R representam a parte substantiva do problema, enquanto que d, r e A representam a parte intencional e são chamados predicado dos dados fonte, predicado intencional e predicado de admissibilidade, respectivamente.

Exemplo 2.5.2.2 :

Seja o problema de ordenação de uma seqüência de números inteiros.

Uma estruturação para o problema é a seguinte:

$P = \langle Seq, Seq, Seq', Ord, Alg \rangle$

onde:

Seq = *conjunto das seqüências de números inteiros*

Seq' = *conjunto das seqüências finitas de números inteiros*

$(s, s') \in Ord = ordenada(s') \wedge mesma(s, s')$

Alg = *funções algorítmicas*

tal que,

ordenada - é um predicado que diz se s' é uma seqüência de Seq ordenada.

mesma - é um predicado que indica que s e s' possuem os mesmos elementos.

Definição 2.5.3 *[Lop81] : Seja P um problema, uma solução para P é uma função de D em R, tal que $a \in A$ e $\forall x \in d$ $(x, a(x)) \in r$. Indica-se que " resolve P" ou " é uma solução para P" por $a \parallel\!- P$.*

Definição 2.5.4 *[Lop81] : Uma instância de um problema P é um problema $P_t = \langle D, R, (t), r, A \rangle$, tal que $t \in d$.*

A partir da definição acima, podemos concluir que qualquer solução para o problema P também é solução para qualquer uma instância sua. Dizendo de outra maneira, temos:

Se $\forall t \in d \; a \Vdash P_t \Longrightarrow a \Vdash P$

Ao definirmos solução de problemas, vemos a necessidade de pensarmos na criação de uma definição do espaço das soluções de um problema.

Definição 2.5.5 *[Lop81] : O Espaço de Soluções de um Problema P denotado por $\sum(P)$, é dado pelo seguinte conjunto:*

$$\sum(P) = \{a \in R^D \mid a \Vdash P\}$$

A seguir veremos uma análise do comportamento do espaço das soluções de um problema.

Proposição 2.5.1 *[Lop81] : Sejam $P = \langle D, R, d, r, A \rangle$, $P_1 = \langle D, R, d_1, r, A \rangle$ e $P_2 = \langle D, R, d_2, r, A \rangle$ três problemas. Então as seguintes implicações são válidas:*

1. $d \subseteq d_1 \Longrightarrow \sum(P) \subseteq \sum(P_1)$
2. $d = d_1 \cup d_2 \Longrightarrow \sum(P) = \sum(P_1) \cap \sum(P_2)$
3. $d = d_1 \cap d_2 \Longrightarrow \sum(P) = \sum(P_1) \cup \sum(P_2)$

Se $d = \emptyset$, então o $\sum(P) = \emptyset$ e se $d = D$, então o $\sum(P)$ é o menor espaço de soluções de P. Portanto, podemos verificar intuitivamente a proposição 1 e em decorrência as proposições 2 e 3.

Proposição 2.5.2 : *Sejam* $P = \langle D, R, d, r, A\rangle$, $P_1 = \langle D, R, d, r_1, A\rangle$ *e* $P_2 = \langle D, R, d, r_2, A\rangle$ *três problemas. Então as seguintes implicações são válidas:*

1. $r \subseteq r_1 \Longrightarrow \sum(P) \subseteq \sum(P_1)$
2. $r = r_1 \cup r_2 \Longrightarrow \sum(P) = \sum(P_1) \cup \sum(P_2)$
3. $r = r_1 \cap r_2 \Longrightarrow \sum(P) = \sum(P_1) \cap \sum(P_2)$

Se $r = \emptyset$, então $\sum(P) = \emptyset$ com $d = D$ e se $r = D \times R$, temos que $\sum(P) = A$. Assim, intuitivamente verificamos a proposição 1, e em decorrência as proposições 2 e 3.

Proposição 2.5.3 : *Sejam* $P = \langle D, R, d, r, A\rangle$, $P_1 = \langle D, R, d, r, A_1\rangle$ *e* $P_2 = \langle D, R, d, r, A_2\rangle$ *três problemas. Então as seguintes implicações são válidas:*

1. $A \subseteq A_1 \Longrightarrow \sum(P) \subseteq \sum(P_1)$
2. $A = A_1 \cup A_2 \Longrightarrow \sum(P) = \sum(P_1) \cup \sum(P_2)$
3. $A = A_1 \cap A_2 \Longrightarrow \sum(P) = \sum(P_1) \cap \sum(P_2)$

Decorrem imediatamente das definições de $\subseteq$, $\cup$, $\cap$ da teoria dos conjuntos.

2.5.3 Relações e Transformações de Problemas

Geralmente é conveniente transformar um problema em outro que tem alguma relação com o problema em estudo. Esta relação existente entre o problema original e o problema relacionado é vista sob dois aspectos: o aspecto estrutural, o qual busca relações entre os espaços das soluções, que leva a um estudo das propriedades que são preservadas através das transformações estruturais; e o aspecto comportamental, que busca analogias ou semelhanças com o problema original, que vai depender da experiência do solucionador.

Homomorfismos entre Problemas

Um homomorfismo entre problemas, é uma relação obtida através de duas funções definidas entre os conjuntos bases (domínio de dados e domínio de resultados) correspondentes dos problemas, e que preservam a parte intencional do primeiro. Formalmente temos:

Definição 2.5.6 *[Lop81] : Sejam $P = \langle D, R, d, r, A\rangle$ e $P' = \langle D', R', d', r', A'\rangle$ dois problemas. Um Homomorfismo $\mathcal{H} = (\phi, \psi)$ de P em P' é um par de funções ϕ e ψ, com $\phi : D \longrightarrow D'$ e $\psi : R \longrightarrow R'$, tal que:*

1. $\forall x \in D,\ x \in d \Longrightarrow \phi(x) \in d'$
2. $\forall x \in D,\ \forall y \in R\ (x, y) \in r \Longrightarrow (\phi(x), \psi(y)) \in r'$
3. $\forall a \in R^D$ e $\forall b \in R'^{D'},\ b \circ \phi = \psi \circ a \Longrightarrow (a \in A \Longrightarrow b \in A')$

Denotaremos por $P \Longrightarrow^H P'$, para indicar que P é homomorfo a P'. A figura 2.2 mostra um esquema de um homomorfismo entre dois problemas.

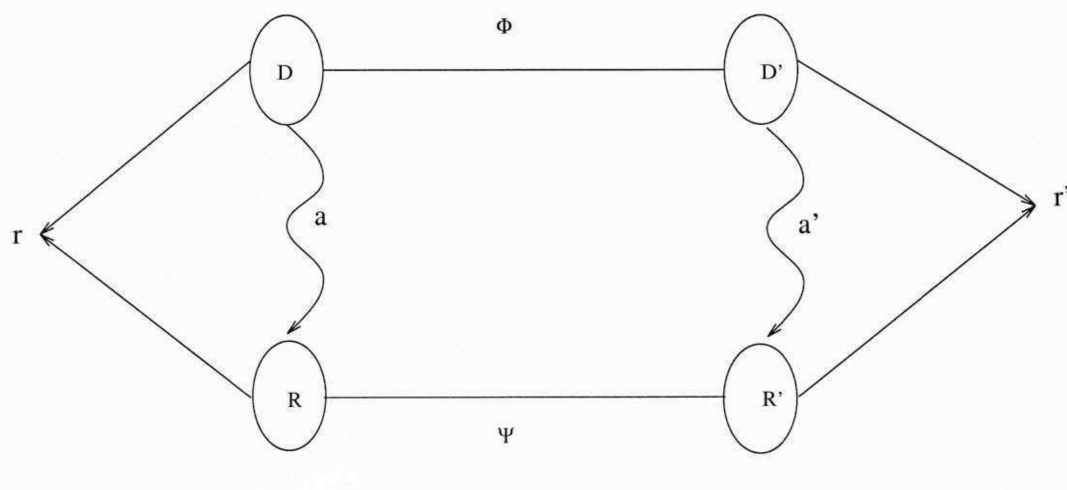

Figura 2.2: *Homomorfismo entre Problemas*

Definição 2.5.7 *[Lop81] : Um homomorfismo $\mathcal{H}$ é dito:*

1. *forte em relação a d* se $\forall x \in d\ [\ \phi(x) \in d' \Longleftrightarrow x \in d\ ]$

2. *forte em relação a r* se $\forall x \in D\ \forall y \in R\ [\ (\phi(x), \psi(y)) \in r' \Longrightarrow (x, y) \in r\ ]$

3. *forte em relação a A* se $b \circ \phi = \psi \circ a \Longrightarrow (b \in A' \Longrightarrow a \in A \vee a' \in A$ *tal que* $a/d = a'/d)$

4. *forte* se é forte em relação a *d*, *r* e *A*

5. *estrutural* se $\forall a \in A\ \exists b \in A$ *tal que* $b \circ \phi = \psi \circ a$

Subproblemas

O estudo de homomorfismo nos conduz ao conceito de subproblema. Tal conceito tem um sentido de refletir a idéia de particularização. A particularização da solução de um problema tanto pode ser uma solução de um dos seus subproblemas como uma solução do próprio problema.

Definição 2.5.8 *[Lop81] : Um problema* $P' = \langle D', R', d', r', A' \rangle$ *é um subproblema de um problema* $P = \langle D, R, d, r, A \rangle$*, que denotaremos por* $P' \leq P$ *se, e somente se,* $D' \subseteq D$*,* $R' \subseteq R$ *e as inserções* $i : D' \longrightarrow D$ *e* $j : R' \longrightarrow R$*, constituem um homomorfismo de* P' *em* P.

A seguir veremos uma proposição que estabelece condições necessárias e suficientes para verificar se um problema P' é um subproblema de um problema P.

Proposição 2.5.4 *[Alm90] : Sejam* P *e* P' *dois problemas* $P' \leq P$ *se, e somente se,* $D' \subseteq D$*,* $R' \subseteq R$*,* $d' \subseteq d$*,* $r' \subseteq r$ *e* $A' = \{b \in R^D / b(D') \subseteq R'$ *e* $b/D' \in A'\} \subseteq A$.

PROVA: Veja [Alm90]

Isomorfismo entre Problemas

Definição 2.5.9 *[Lop81] : Um isomorfismo entre dois problemas P e P', é um homomorfismo de P em P', que satisfaz as seguintes condições:*

1. *O homomorfismo (ϕ, ψ) de P em P' é forte, e*
2. *ϕ e ψ são funções bijetivas*

Do ponto vista estrutural e de solubilidade, dois problemas isomorfos são indistingüíveis. Assim, obter efetivamente uma solução para um problema significa, na prática, obter uma solução para todos os problemas de sua classe de isomorfismo.

Analogias e Semelhanças entre Problemas

Intuitivamente dois problemas são análogos ou similares se possuem a mesma intencionalidade (d, r e A), porém com interpretações diferentes. A seguir formalizaremos esse conceito.

Definição 2.5.10 *[Lop81] : Sejam P e P' dois problemas. E sejam $\alpha : d \longrightarrow d'$, $\beta : r \longrightarrow r'$ e $\gamma : A \longrightarrow A'$ três funções. O terno (α, β, γ) é dito uma analogia de P em P', com denotação $P \Longrightarrow^{A} P'$, se e somente se, existe uma função $\psi : R \longrightarrow R'$, tal que as propriedades abaixo são satisfeitas:*

1. *$\forall a \in A$, $\exists b \in A'$, tal que $b = \gamma a$ e se $x \in d \Longrightarrow b(\alpha(x)) = \psi(a(x))$;*
2. *$\forall x \in D$, $\forall y \in R$, $(x, y) \in r \Longrightarrow \beta(x, y) = (\alpha(x), \psi(y)) \in r'$.*

Para estabelecer as condições necessárias e suficientes que garantem que uma analogia entre problemas preserva solubilidade, temos o seguinte teorema:

Teorema 2.5.1 *[Lop81] : Se (α, β, γ) é uma analogia de P em P', e α é sobrejetiva, então P solúvel implica P' solúvel.*

PROVA: Veja [Lop81]

Operações sobre Problemas

Definição 2.5.11 *[Lop81] : Sejam* $P = \langle D, R, d, r, A\rangle$ *e* $P' = \langle D', R', d', r', A'\rangle$ *dois problemas, tal que:*

1. *A* $P_{combinacão}$ *de* P *e* P' *é o problema* $P \times P' = \langle D \times D', R \times R', d \times d', r \times r', A \times A'\rangle$, *onde "$\times$" é a operação do produto cartesiano natural entre conjuntos.*

2. *Se* P *e* P' *são tais que* $D' = R$, *então, a* $P_{composicão}$ *de* P *com* P', *é o problema* $P' \circ P = \langle D, R', d, r' \circ r, A' \circ A\rangle$, *onde* $r' \circ r$ *é a composição usual entre relações, e* $A' \circ A = \{h' \circ h : D \longrightarrow R / h \in A \text{ e } h' \in A'\}$.

3. *Se em* P, $D = R$, *a* $P_{exponenciacão}$ *de* P *é o problema* $P^{\#} = \langle D \times N, D, d \times N, r^{\#}, A^{\#}\rangle$, *onde* $(x, 0, y) \in r^{\#} \Longleftrightarrow x = y$, $(x, n+1, y) \in r^{\#} \Longleftrightarrow (x, n, z) \in r^{\#}$ *e* $(z, y) \in r$, $A^{\#} = \{f^{\#} / f \in A\}$ *e* $f^{\#}(x, 0) = x$, $f^{\#}(x, n+1) = f(f^{\#}(x, n))$.

Em [Lop81] podem ser encontradas outras operações tais como: $P_{união}$, $P_{intersecão}$, $P_{iteracão}$, P_{soma}, etc.

Capítulo 3

Métodos de Solução de Problemas

3.1 Introdução

No decorrer da história, o homem tem se defrontado com vários problemas , e estes tem sido resolvidos de diversas maneiras, utilizando os mais variados métodos e critérios para chegar a sua solução.

É muito difícil tratar a questão problema sem que haja o desejo de se estabelecer metodologias seguras e completas que influenciem na sua resolução.

Neste capítulo são apresentados dois métodos que visam diretamente a solução de problemas: a Redução e a Decomposição.

3.2 Redução de Problemas

A idéia de redução de um problema a outro tem a finalidade de reduzir as dificuldades para obtenção de solução. Muitas vezes ela ocorre com nomes diferentes: analogia, particularização, generalização, etc.[Pol78].

Intuitivamente, dois problemas são análogos ou similares, se eles possuem a mesma parte intencional, mas com conotações diferentes. Isto é, em não se podendo resolver um problema, operamos uma transformação do seu domínio fonte para o domínio fonte de outro problema que podemos resolver. Determinamos uma solução deste e operamos uma transformação do codomínio de soluções do segundo problema para o codomínio de soluções do problema inicial, esperando dessa maneira, encontrar a solução do problema dado.

Por exemplo, "ordenar sequências de naturais, sem repetições" e "ordenar sequencias de inteiros, sem repetições", são dois problemas diferentes que possuem a mesma parte intencional.

Uma redução pode ser vista como uma transformação de problemas ou como um método de resolução de problemas, dependendo dos seguintes casos:

- Se na impossibilidade de resolver um dado problema, quando o reduzimos a outro na tentativa de solucionar o segundo, possivelmente mais simples, então, para este caso, podemos dizer que redução é uma transformação de problemas.
- Se a redução é feita a um problema que conhecemos sua solução, podemos dizer que a redução é um método de resolução de problemas.

Expressando formalmente teremos:

Definição 3.2.1 *[Lop81]: Uma* $\mathcal{R}$ *de um problema* $\mathcal{P}$ *para outro* $\mathcal{P}'$*, consiste de duas funções Ins e Ret, onde* $Ins : D \longrightarrow D'$ *e* $Ret : R' \longrightarrow R$ *tais que* $\forall x \in D$ $\forall y' \in R'(Ins(x), y') \in r' \Longrightarrow (x, Ret(y')) \in r$.

Se $\mathcal{R}$ *é tal que para todo* $a' \in A'$ *temos que* $Ret \circ a' \circ Ins \in A$*, então* $\mathcal{R}$ *é chamada de uma* .

Se $\mathcal{R}$ *é tal que para todo* x *em* d $Ins(x) \in d'$*, então* $\mathcal{R}$ *é chamada de*

.

A idéia de redução pode ser ilustrada na figura 3.1.

Uma condição suficiente para que uma boa redução conduza a solubilidade de $\mathcal{P}$, através de uma solução de $\mathcal{P}'$, é dada pelo teorema abaixo.

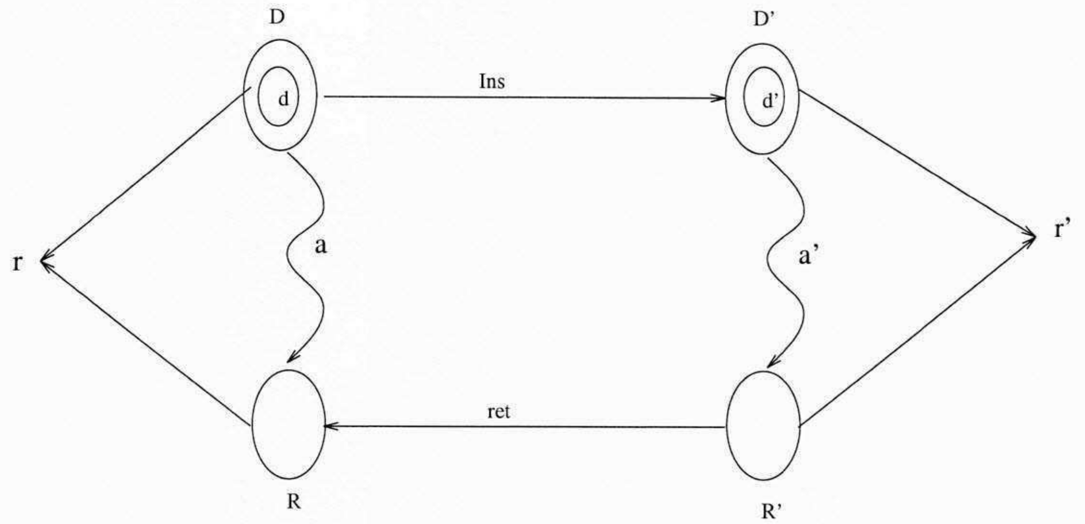

Figura 3.1: *Redução de Problemas*

Teorema 3.2.1 *[Lop81] : Seja $\mathcal{R}$ uma boa redução de $\mathcal{P}$ em $\mathcal{P}'$ tal que $\forall f \in d$ $Ins(f) \in d'$ então $\| - \mathcal{P}' \Longrightarrow \|- \mathcal{P}$.*

PROVA: Sejam $a' \in \sum(P')$ e $f \in d$. Pela condição do estabelecido para Ins temos que $Ins(f) \in d'$. Como $a' \;\|- \; P'$ então $a' \in A'$ e $(Ins(f), a'(ins(f))) \in r'$. Desde que $\mathcal{R} = (Ins, Ret)$ é uma boa redução temos $(f, Ret(a'(Ins(f)))) \in r$ com $Ret.a'.Ins(f)) \in A$, logo $a = Ret \circ a' \circ Ins \;\|- \; P.\bullet$

Um estudo mais aprofundado dos resultados sobre preservação de solubilidade, existência de homomorfismo, etc., podem ser encontrados em [Lop81].

3.3 Decomposição de Problemas

Uma das estratégias de solução que tem uma maior divulgação, tem sido aquela baseada no velho ditado da cultura popular "dividir para conquistar". Isso acontece porque este método pode ser aplicado nos mais variados campos da sociedade humana como, por exemplo: política (Ex: O ditador que divide a oposição com tolices, para manter-se no poder.), administração (Ex: A divisão de uma empresa em departamentos e seções.), saúde (Ex: Há médicos especialistas em determinadas áreas do corpo humano.), computação (Ex: Os estilos de programação modular, recursiva, top-down, etc.), etc.

A idéia de decomposição utilizada na Teoria Geral de Problemas pode ser mostrada do seguinte modo: os elementos do predicado fonte de um problema são partidos repetidamente através de uma função de partição **part** até chegarem a uma forma **simples**. Operam-se soluções **diretas** sobre esses elementos simples, encontrando-se os elementos correspondentes no codomínio de solução e constroem-se, através de uma função **recombinação**, os elementos do codomínio de solução correspondentes aos elementos do predicado fonte. Deve-se ter o cuidado de que a função resultante dessa associação, satisfaça o predicado de admissibilidade do problema.

Existem duas maneiras de se decompor um problema: uma Local e a outra Estrutural.

3.3.1 Decomposição Local

Este tipo de decomposição baseia-se na idéia de recursividade. E recursividade para o caso de problemas, é a propriedade que tem alguns deles de estarem compostos de diversas partes com propriedades de se converterem em totalidades, ou seja, em problemas independentes. Porém mantendo a mesma estrutura do problema, pelo qual estas também podem ser decompostas. Para que haja uma resolução desse tipo, será necessário determinar um limite na recursividade, e para isso é preciso distinguir um subconjunto das possíveis partes do problema recursivo que sejam suficientemente simples, para aplicar diretamente uma solução. Então recombinam-se os resultados assim obtidos para chegar no resultado do problema original. Deve-se ter o cuidado de que a condição de admissibilidade seja satisfeita.

Formalmente teremos a seguinte definição:

Definição 3.3.1 *[Bed87] : Seja $P = \langle D, R, d, r, A \rangle$ um problema. Uma decomposição $\mathcal{D}$ de P consiste de:*

- *uma função unária* $: D \longrightarrow D$

- *uma função unária* $: R \times D \longrightarrow R$
- *uma função unária* $: D \longrightarrow R$
- *um predicado unário* $: \subseteq D$

Seja h a função canônica associada à decomposição $\mathcal{D}$, definida por:

$$h(x) = \begin{cases} direta(x), & \text{se } x \in simples \\ recomb(h(part(x)), x), & \text{senão} \end{cases}$$

Como a função h é recursiva, então para que seja uma solução para P ($h \; \| {-} \; P$), deve-se ter o cuidado de que h seja uma função total que satisfaça r. O teorema abaixo nos mostra condições necessárias e suficientes para que a função h seja uma solução para P.

Teorema 3.3.1 *[Bed87] : Sejam P um problema, e $\mathcal{D}$ uma decomposição local sobre P representada pela quádrupla $\mathcal{D} = \{part, recomb, direta, simples\}$. A função canônica h é uma solução para P ($h \; \| {-} \; P$), sse $h \in A$ e:*

1. *A relação $<<$ sobre D,*

   $x << y \Longleftrightarrow \exists k > 0,\ part^k(y) = x \wedge y \notin$ *, é uma ordem bem fundada,*
2. $\forall x \in simples \Longrightarrow (x, direta(x)) \in r$,
3. $\forall x \in D,\ x \notin \quad \Longrightarrow [\,(part(x), h(part(x))) \in r \Longrightarrow (x, recomb\,(h(part(x)), x)) \in r\,]$.

PROVA: Deve-se provar que $\forall x \in d$, $(x, h(x)) \in r$. Como D é um conjunto bem-fundado, então $d \subseteq D$ também é bem-fundado. Tem-se dois casos:

1. Se $x \in$ **simples**, então por 2 tem-se que $(x, direta(x)) \in r$. Logo, pela definição de h, para elementos de **simples**, tem-se que $(x, h(x)) \in r$.

2. Se $x \notin$ **simples**, por 3 tem-se que $(x, recomb(h(part(x)), x)) \in r$. Logo, pela definição de h para elementos que não são de **simples**, tem-se que $(x, h(x)) \in r$.

Logo $\forall x \in d$, $(x, h(x)) \in r$, e portanto $h \;\|\!-\; P$.

Por outro lado, se $h \;\|\!-\; P$, ou seja, $\forall x \in d\ (x, h(x)) \in r$, tem-se que provar que a decomposição $\mathcal{D}$ junto com sua função canônica satisfazem 1, 2 e 3. Com este objetivo, tem-se que:

1. Para h ser solução de P, a função **part** deve necessariamente cumprir com a relação $<<$ sobre D, isto é,

$$x << y \iff \exists k > 0\ \mathbf{part}^k(y) = x \wedge y \notin \mathbf{simples}$$

caso contrário h não pararia, pois é definido recursivamente através de **part**. Uma vez que isto é constatado, [OJ85] demonstra que $<<$ é uma ordem bem-fundada. Assim, 1 fica provado.

2. Como $(x, h(x)) \in r$, então $\forall x \in$ **simples** $(x, h(x)) \in r$, isto é, $(x, direta(x)) \in r$. Portanto 2 é satisfeita.

3. E $\forall x \notin simples$, também se tem que $(x, h(x)) \in r$, isto é, $(x, recomb(h(part(x)), x))$ $\in r$, e como por 1 se tem que, quando $k = 1$, $part(x) << x$, então $(x, h(x)) \in r$ $\Longrightarrow (part(x), h(x)) \in r$. Logo, 3 é satisfeita.

Portanto, $h \;\|\!-\; P \iff$ 1, 2, 3 são satisfeitos.•

Geralmente, no uso da decomposição para solucionar um problema, se faz necessário a utilização da função **part** várias vezes, com isso, a intuição de decomposição torna-se mais nítida. A seguir veremos uma definição que considera estes elementos:

Definição 3.3.2 *[Bed87] : Seja P um problema. Uma decomposição N-ária , $\mathcal{D}_n$, sobre P consiste de:*

- *N funções unárias $part_i$:D $\longrightarrow$ D, com i= 1,...,N*

- *Uma função $(N+1)$-ária* $recomb_{N+1}$: $R^N X D \longrightarrow R$
- *Uma função unária* $: D \longrightarrow R$
- *Uma relação unária* $\subseteq D$.

Seja h a função canônica associada à decomposição $\mathcal{D}_n$ definida abaixo:

$$h(x) = \begin{cases} direta(x), & \text{se } x \in simples \\ recomb_{N+1}(h(part_1(x)), ..., h(part_N(x)), x), & \text{senão} \end{cases}$$

Mesmo com a aparência de ser um tipo de decomposição mais geral do que a primeira, ambas são equivalentes (demonstrado em [Lop81]). O teorema a seguir mostra as condições necessárias e suficientes através das quais a função canônica da decomposição $\mathcal{D}^n$ constitui uma solução.

Teorema 3.3.2 *[Bed87] : Sejam P um problema e $\mathcal{D}_n$ uma decomposição N-ária sobre o problema P, representada por $\mathcal{D}^n = \{part_1, ..., part_n, recomb_{n+1}, direta, simples\}$. A função canônica h é uma solução para P ($h \; \| {-} \; P$), sse $h \in A$ e:*

- *A relação $<<$ sobre D*

  $x << y \Longleftrightarrow \exists \; part_i \circ ... \circ part_j(y) = x \wedge y \notin simples$ *é uma ordem bem-fundada.*

- $\forall x \in D \; x \in simples \Longrightarrow (x, direta(x)) \in r$
- $\forall i = 1, ..., N \; (part_i(x), \; h(part_i(x))) \in r \Longrightarrow (x, recomb_{N+1} \; (h(part_1(x)), ..., h(part_N(x)), x)) \in r$.

PROVA: Análoga à do teorema 3.3.1

3.3.2 Decomposição Estrutural

Na seção anterior discutimos problemas que são simples do ponto de vista estrutural.Porém, frequentemente, é comum nos defrontarmos com problemas estruturalmente complexos, que podem ser vistos como vários problemas que se relacionam entre si, de diversas maneiras constituindo o problema em estudo.

Em [Bed87], tem-se que a partir da definição estrutural de um problema, se determinarão as condições sobre as quais é permitido decompor o problema e como se compor a solução deste problema, sendo conhecidas as soluções dos subproblemas. E é definido também os diversos tipos de decomposição estrutural, cada uma tentando abranger uma determinada família de problemas, a partir das operações dada em [Lop81] para construção de problemas.

Decomposição Combinacional de Problemas

Muitas vezes, é comum encontrar problemas nos quais se quer fazer duas coisas diferentes. Neste caso, decompomos estes em dois problemas, em que suas respectivas soluções realizem separadamente essas duas coisas.

Definição 3.3.3 *[Bed87] : Um problema P é Decomponível Combinacionalmente nos problemas P_1 e P_2, denotado por $P = P_1 \times P_2$, se as componentes de P são $D = D' \times D'_1 \times D'_2$, onde $D_1 = D' \times D'_1$ e $D_2 = D' \times D'_2$, $R = R_1 \times R_2$, $d = d' \times d'_1 \times d'_2$, com $d_1 = d' \times d'_1$ e $d_2 = d' \times d'_2$, $r = r_1 \times r_2$, onde $((x', x'_1, x'_2), (y_1, y_2)) \in r \iff ((x', x'_1), y_1) \in r_1 \wedge ((x', x'_2), y_2) \in r_2$, e $A = A_1 \times A_2 = \{(a_1 \times a_2) \in R_1 \times R_2^{D' \times D'_1 \times D'_2} / a_1 \in A_1 \wedge a_2 \in A_2\}$.*

Decompor um problema combinacionalmente não é garantia para sua solubilidade, pois deve-se ter certeza que os subproblemas surgidos desta decomposição, sejam solúveis. O seguinte teorema nos mostra isto:

Teorema 3.3.3 *[Bed87] : Se $P = P_1 \times P_2$, então $\| - P \iff \| - P_1 \wedge \| - P_2$.*

PROVA: Seja $a \in \sum(P)$, logo $\forall(x, x_1, x_2) \in d$ $((x, x_1, x_2, a(x, x_1, x_2)) \in r$, e pela definição de A, se tem que $a = a_1 \times a_2$, com $a_1 \in A$ e $a_2 \in A_2$ e, portanto, $((x, x_1, x_2),$ $(a_1(x, x_1), a_2(x, x_2))) \in r$, e $((x, x_1), a_1(x, x_1)) \in r_1$ e $((x, x_2), a_2(x, x_2)) \in r_2$. Logo a_1 $\|- P_1$ e $a_2 \|- P_2$.

Caso contrário, se $a_1 \in \sum(P_1)$ e $a_2 \in \sum(P_2)$, se cumpre que $((x, x_1), a_1(x, x_1)) \in r_1$, e $((x, x_2), a_2(x, x_2)) \in r_2$ e pela própria definição do r se tem $((x, x_1, x_2), a_1(x, x_1), a_2(x, x_2))$ $\in r$, e portanto, $\| - P$. •

Decomposição Somativa de Problemas

A decomposição somativa é comum em problemas nos quais se quer fazer uma ou outra coisa, dependendo de uma condição ou sem haver distinção. Tal tipo de problema será chamado de problema soma.

Definição 3.3.4 *[Bed87] : Um problema P é Decomponível Somativamente nos problemas P_1 e P_2, denotado por $P = P_1 + P_2$, se as componentes de P são: $D = D_1 \cup D_2$, $R = R_1 \cup R_2$, $d = d_1 \cup d_2$, $r = C_1 \times r_1 \cup C_2 \times r_2$ e $A = \{a \in R^D$ tal que $a/d_1 - d_2$ $\in A_1/d_1 - d_2 \wedge a/d_2 - d_1 \in A_2/d_2 - d_1 \wedge a/d_1 \cap d_2 \in (A_1/d_1 \cap d_2 \cup A_2/d_1 \cap d_2)\}$. Onde C_1 e C_2 são condições sobre os dados que se refletem em d_1 e d_2, respectivamente, e para uma j qualquer, $a \in A_j/D_j' \iff d_j' \subseteq d_j$ e $\exists a_j \in A_j$ tal que $\forall x \in d_j$ $a(x) = a_j(x)$.*

O teorema abaixo formaliza a praticidade desta decomposição ao considerar aspectos de solubilidade.

Teorema 3.3.4 *[Bed87] : Se $P = P_1 + P_2$, então $\| - P \iff \| - P_1 \wedge \| - P_2$.*

PROVA: Seja $a \in \sum(P)$, então $\forall x \in d = d_1 \cup d_2$ $(x, a(x)) \in r$, se $x \in d_1$, então $(x, a(x)) \in r$, logo $(x, a/d_1(x)) \in r$, e , portanto, $\| - P_1$. Analogamente, para P_2 se tem que $(x, a/d_2(x)) \in r_2$ e portanto, $\| - P_2$.

Por outro lado, se $a_1 \|- P_1$ e $a_2 \|- P_2$, então pode-se construir uma solução para P como mostrado abaixo:

$$a(x) = \begin{cases} a_1(x), & \text{se } x \in d_1 \\ a_2(x), & \text{se } x \in d_2 - d_1 \\ H(y), & \text{se } x \in D - (d_1 \cup d_2) \end{cases}$$

onde:

$H(y) \in R$, e como $d_1 \cup d_2 = d$, $(D - d) \cup d = D$, $d_1 \cap (d_2 - d_1) = \emptyset$, então a está bem definida. E como $\forall x \in d_1$ $(x, a_1(x)) \in r$, e $\forall x \in d_2 - d_1$ $(x, a_2(x)) \in r$, então $\forall x$ $\in d$ $(x, a(x)) \in r$, e portanto $a \;\|\!- P$.•

Decomposição em Cadeia de Problemas

A maioria dos problemas são tratáveis por esse tipo de decomposição, pois geralmente quando nos deparamos com problemas grandes ou com uma certa complexidade, devemos encará-los por etapas, analisando primeiro as mais fáceis de serem solucionadas.

Do ponto de vista matemático, a decomposição em cadeia possui semelhança com a operação de composição de funções.

Definição 3.3.5 *[Bed87] : Um problema P é Decomponível Encadeadamente nos problemas* P_1 *e* P_2*, denotado por* $P = P_1 \circ P_2$ *, se* $D_2 = R_1$ *e as componentes de P são* $D = D_1$*,* $R = R_2$*,* $d = d_1$ *e* $(x, y) \in r \iff \exists y' \in D_2 = R_1 / (x, y') \in r_1 \wedge (y', y) \in r_2$*, e* $A_2 \circ A_1 \subseteq A$*, isto é,* $\forall a_2 \in A_2$ *e* $\forall a_1 \in A_1$ $a = a_2 \circ a_1 \in A$.

Podemos nos deparar com um problema P, decomponível encadeadamente , no qual duas soluções de P_1 ou de P_2, levam a saídas diferentes para uma mesma entrada, com isso não há nenhuma garantia que a composição destas soluções, de forma independente, com uma qualquer do outro problema satisfaçam P.

Uma definição para problemas que forneçam um único valor de saída, para qualquer solução e servindo também para estabelecer a condição de suficiência, é dada abaixo:

Definição 3.3.6 *[Bed87] : Um problema P é Identicamente Solúvel a uma problema P', sse suas soluções são equivalentes, isto é, $\forall a_1, a_2 \in \sum(P)$ $a_1 = a_2$, isto é , $\forall x \in d$ $a_1(x) = a_2(x)$.*

Ou seja, um problema é identicamente solúvel sse $(x, y) \in r$ e $(x, y') \in r \iff y = y'$. Assim, para cada dado de entrada só existe um único resultado possível tal que o resultado satisfaça a intenção do problema.

O teorema abaixo mostra as condições necessárias e suficientes e, através delas podemos garantir uma maneira segura de compor a solução de um dado problema, que foi decomposto encadeadamente utilizando as soluções de seus subproblemas.

Teorema 3.3.5 *[Bed87] : Seja $P = P_2 \circ P_1$, tal que P_1 e P_2 são problemas identicamente solúveis, e $\forall a_1 \in \sum(P_1)$, a $Imag(a_1/d_1) \subseteq d_2$. Se $a_2 \,\|\!- P_2$, então $a = a_2 \circ a_1$ $\|\!- P$.*

PROVA: Tem-se que $\forall x \in d_1$ $(x, a_1(x)) \in r_1$, $\forall x \in d_1$ $a_1(x) \in d_2$ e pelas hipóteses de serem identicamente solúveis $\exists y'' \neq a_1(x)/(x, y'') \in r_1$. Logo $(a_1(x), a_2(a_1(x))) \in r_2$ e $\exists y'' \neq a_2(a_1(x))/(a_1(x), y'') \in r_2$. Assim, pela definição de r em 3.3.5, se tem que $y' = a_1(x)$ e $y = a_2(a_1(x))$, e então $(x, y) \in r$, isto é, $(x, a_2(a_1(x))) \in r$. Portanto, $a = a_2 \circ a_1 \,\|\!- P$. •

Decomposição Sequencial de Problemas

Normalmente é muito fácil nos depararmos com problemas em que precisamos fazer algo repetidamente, para uma quantidade varíavel de dados. O que leva a ter que repetir várias vezes um mesmo processo para diferentes dados.

Definição 3.3.7 *[Bed87] : Um problema $P^+ = \langle D^+, R^+, d^+, r^+, A^+ \rangle$, é Decomponível Sequencialmente no problema $P = \langle D, R, d, r, A \rangle$, se D^+, R^+ e d^+ são conjuntos de potenciais dos respectivos D, R e d, e $((x_1, ..., x_N), (y_1, ..., y_M)) \in r^+ \iff N = M$ e $(x_i, y_i) \in r$, com $1 < i \leq N$, e $A^+ = a^+/a \in A$ e $a^+(x_1, ..., x_N) = (a(x_1), ..., a(x_N))$.*

Quando decompomos sequencialmente um problema num outro problema, verificamos que sempre é possível compor uma solução para ele, através da solução do outro problema.

Capítulo 4

Análise Sintática da Intencionalidade de um Problema

4.1 Introdução

No decorrer do tempo, muitas abordagens têm surgido com o interesse de encontrar uma metodologia segura que possa resolver qualquer tipo de problema dentro de uma mesma classe. Várias tentativas já foram feitas no intuito de se estabelecer uma lógica para solucionar problemas que sirva de base a estas metodologias [LA85].

De acordo com [Lop81], um problema pode ser visto como uma estrutura que pode, usualmente, ser complexa, onde são definidas algumas técnicas para decomposição de tal problema em vários outros mais simples estruturalmente.

Na estrutura $P = \langle D, R, d, r, A \rangle$, o r representa a essência do problema em si, isto é, descreve a intenção deste. Então, podemos dizer que a intencionalidade de um problema é o que se quer resolver, e não depende do universo de discurso no qual está inserido.

Segundo [Bed87], uma análise da sintaxe do predicado intencional de um problema permitirá que se busque quais são as condições necessárias e suficientes para que, a partir da composição sintática do r, se quebre o problema em vários problemas com sua intenção mais simples, e como se pode construir uma solução do problema baseada nas soluções deste. A figura abaixo mostra a idéia de r ser composto de vários predicados que correspondem a intenção de algum outro problema.

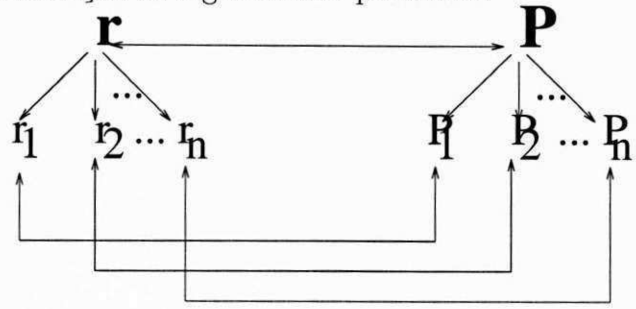

Figura 4.1: *Esquema de Composição do r de um problema*

A intencionalidade de um problema é definida de diversas maneiras. Uma das formas de definição de r, é utilizar uma sentença de linguagem de primeira ordem, e uma padronização adequada seria defini-la numa forma normal disjuntiva [Cas87], onde uma sintaxe para r tem a seguinte forma: $(x, y) \in r \Longleftrightarrow L$, onde:

$\Longleftrightarrow$ é um símbolo metalinguístico de definição. L é uma fórmula que se apresenta como uma disjunção de conjunções. Uma estruturação geral para a sintaxe da definição da intencionalidade é vista a seguir:

$(x, y) \in r \Longleftrightarrow L_1 \vee ... \vee L_n$, onde $n, m, ..., m' \in N$

$L_1 = L_{11} \wedge ... \wedge L_{1m} \;\; . \;\; . \;\; . \;\; L_n = L_{n1} \wedge ... \wedge L_{nm'}$

onde:

$L_{11}, ..., L_{nm'}$ são fórmulas atômicas em alguma linguagem de primeira ordem.

O x e y da estrutura geral de r pode representar uma variável simples ou uma estrutura de variáveis. Portanto, se uma subestrutura x' de x for considerada como um termo numa fórmula atômica L_{ij}, então x' é uma variável de entrada em L_{ij}. Similarmente, se y' uma subestrutura de y e for um termo numa fórmula atômica L_{ij}, então y' é uma variável de saída em L_{ij}.

A composição da intenção de um problema vista acima, vai indicar o caminho a ser percorrido para decompor um problema em subproblemas.

4.2 r Como uma Disjunção de Conjunções

Sabemos que uma disjunção preserva a independência de seus predicados. Assim, quando tivermos o r na forma $(x, y) \in r \iff L_1 \vee ... \vee L_n$, poderemos decompor P nos problemas $P_1, ..., P_n$. O que não implica que estes problemas apresentem a intenção da forma $(x_i, y_i) \in r_i \iff L_i$, isto porque, muitas vezes na composição de L_i, tem restrições sobre os dados que se refletirão ao decompor P, na definição de d_i (restrição dos dados).

Definição 4.2.1 *[Bed87] : Seja P um problema, cuja intenção tem a forma $(x, y) \in r \iff L_1 \vee ... \vee L_n$. Onde $\forall i = 1, ..., n$, L_i é uma sentença conjuntiva. Uma Decomposição Disjuntiva para P, são os problemas $P_1, ..., P_n$, denotada por $P = P_1 \vee ... \vee P_n$, tal que $D = D_1 \cup ... \cup D_n$, $R = R_1 \cup ... \cup R_n$, $d = d_1 \cup ... \cup d_n$, e $\forall i, i = 1, ..., n$ $(x_i, y_i) \in r_i \iff L_i^{'}$, com $L_i^{'}$ sendo igual a L_i sem as fórmulas atômicas que contém só variáveis de entrada, e $A = \{a \in R^D$ tal que $\forall i$, $i = 1, ..., n$ $a/d_i - d_i^{'} \in A_i/d_i - d_i^{'} \wedge a/d_i \cap d_i^{'} \in (A_i/d_i \cap d_i^{'} \vee (A_i/d_i \cap d_i^{'}))\}$, onde $d_i = \bigcup_{i=1}^{n} d_j - d_i$, $A_i = \sum_{j=1}^{n} (A_j) - A_i$, e $a \in A_j/d_j^{'} \iff d_j^{'} \subseteq d_j$ tal que $\forall x \in d_j^{'}$ $a(x) = a_j(x)$.*

Exemplo 4.2.0.1 *[Bed87]: Calcular a área e o perímetro de um hexágono regular ou de uma circunferência, segundo seja o caso, sendo conhecido o raio da figura geométrica.*

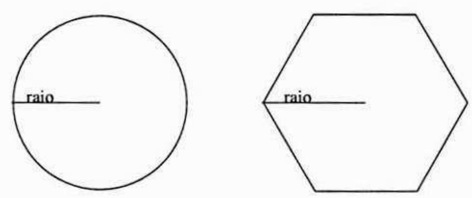

Figura 4.2: Circunferência e hexágono regular com seus respectivos raios

Veremos a seguir uma estruturação para o problema proposto:

$$P = \langle F \times R, A \times P, F' \times R', r, A_d \rangle$$

onde:

F - *É o conjunto dos nomes codificados das figuras geométricas.*

R, A *e* P - *Pertencem ao conjunto dos reais positivos, que denotam o comprimento do raio em cm, a área da figura em* cm^2 *e o perímetro em cm.*

$F' = \{CF, HR\}$

$((fig, ra), (a, p)) \in r \iff FIG(fig, \text{'}HR\text{'}) \wedge AHR(ra, a) \wedge PHR(ra, p) \vee FIG(fig, \text{'}CF\text{'}) \wedge ACF(ra, a) \wedge PCF(ra, p).$

onde:

$FIG(fig, x)$ - *diz se* x *é código* fig *correspondente a uma figura.*

$AHR(ra, a)$ - *é um predicado que diz se a área de um hexágono regular, de raio* ra, *é* a.

$PHR(ra, p)$ - *é um predicado que diz se o perímetro de um hexágono regular, de raio* ra, *é* p.

$ACF(ra, a)$ - *é um predicado que diz se a área de uma circunferência, de raio* ra, *é* a.

$PCF(ra, p)$ - *é um predicado que diz se o perímetro de uma circunferência, de raio* ra, *é* p.

A_d - *sem restrições.*

A especificação do predicado intencional é composto por uma disjunção de conjunções, o que permite aplicar uma decomposição disjuntiva, quebrando o problema nos subproblemas P_1 *e* P_2, *isto é,* $P = P_1 \vee P_2$, *os quais são vistos estruturalmente a seguir:*

$$P_1 = \langle F \times R, A \times P, \{HR\} \times R', r_1, A_1 \rangle \text{ e}$$
$$P_2 = \langle F \times R, A \times P, \{CF\} \times R', r_2, A_2 \rangle$$

onde:

$((fig, ra), (a, p)) \in r_1 \Longleftrightarrow AHR(ra, a) \wedge PHR(ra, p)$

$((fig, ra), (a, p)) \in r_2 \Longleftrightarrow ACF(ra, a) \wedge PCF(ra, p)$

A_1 e A_2 - sem restrições.

A decomposição de um problema P em seu primeiro nível, possibilitará a construção da solução de P a partir das soluções dos subproblemas. Desse modo, uma solução para este problema seria a seguinte função:

$$g(fig, ra) = \begin{cases} a_1(fig, ra), & se\ (fig, ra) \in \{HR\} \times R' \\ a_2(fig, ra), & se\ (fig, ra) \in \{CF\} \times R' \\ (0,\ 0), & se\ (fig, ra) \in (F \times R - (\{HR\} \times R' \cup \{CF\} \times R')) \end{cases}$$

onde:

$a_1 \,\|\!- P_1$ *e* $a_2 \,\|\!- P_2$.

Esta forma de construção de uma solução para P indica que , sempre que se tem a solução dos subproblemas gerados na decomposição disjuntiva do problema, se pode compor uma solução para este. O teorema a seguir expressa formalmente esta afirmação.

Teorema 4.2.1 *[Bed87] : Seja P um problema cuja intenção é $(x, y) \in r \Longleftrightarrow L_1 \vee \ldots \vee L_n$ que é decomponível disjuntivamente nos problemas $P_1, ..., P_n$. Se $\| - P_1, ..., \| - P_n$, então $\| - P$.*

PROVA: Sejam $a_1, ..., a_n$ as respectivas soluções de $P_1, ..., P_n$. Então uma solução para P é a função a definida como segue:

$$a(x) = \begin{cases} a_1(x), & \text{se } x \in d_1 \\ a_2(x), & \text{se } x \in (d_2 - d_1) \\ . & \\ . & \\ . & \\ a_n(x), & \text{se } x \in (d_n - (d_{n-1} \cup ... \cup d_1)) \\ y_1, & \text{se } x \in (D_1 - (d_1 \cup ... \cup d_n)) \\ . & \\ . & \\ . & \\ y_n, & \text{se } x \in (D_n - (d_1 \cup ... \cup d_n)) \end{cases}$$

onde:

$y_1, ..., y_n$ são elementos quaisquer de $R_1, ..., R_n$, respectivamente.

A recíproca do teorema acima não é verdadeira, pois pode-se ter o caso de $P = P_1 \vee P_2$, em que $d = d_1 = d_2$, onde $a_1 \Vdash P_1$, e também $a_1 \Vdash P$, sem haver necessidade de P_2 ser solúvel. Portanto, P_2 não é solúvel e P é solúvel.

4.3 r Como uma Conjunção de Fórmulas Atômicas

Nesta seção estudaremos a composição da intenção dos problemas cujos r tem a seguinte forma: $(x, y) \in r_i \iff L_{i1} \wedge ... \wedge L_{in}$, onde $L_{i1}, ..., L_{in}$ são fórmulas atômicas de alguma linguagem de primeira ordem, tal que todo L_{ij} tem, pelo menos, uma variável de saída.

Definição 4.3.1 *[Bed87] : Sejam $L_1, ..., L_k$ fórmulas atômicas de uma sentença conjuntiva formada a partir destas. Estas fórmulas são chamadas de fórmulas atômicas encadeadas, se $\forall L_i, L_j$, com $1 \leq i$, $j \leq k$:*

1. *L_i e L_j estão encadeadas diretamente, isto é, tem uma mesma variável, ou*

2. L_i e L_j *estão encadeadas indiretamente pelas fórmulas* $L_{i+1}, L_{i+2}, ..., L_{j-2}, L_{j-1}$ *pertencentes a* $L_1, ..., L_k$, *tal que :*
   - L_i *esteja encadeada diretamente com* L_{i+1};
   - *Para qualquer* $L_i^{'} \in L_{i+1}, ..., L_{i-1}, L_i^{'}$ *esteja encadeado também diretamente com* $L_{i+1}^{'}$;
   - L_{j-1} *esteja encadeado diretamente com* L_j.

A princípio, o operador lógico $\wedge$ não preserva a independência entre os predicados, isto por que, usualmente, um predicado amarra-se a outro, ou seja, quando há ocorrência de duas ou mais fórmulas atômicas estarem encadeadas por variáveis de saída.

Exemplo 4.3.0.2 : *Ordenar uma sequência de números naturais.*

Uma estruturação para o problema é:

$P = \langle Seq, Seq, Seq^{'}, r, A \rangle$

onde:

Seq: Conjunto de sequências de naturais.

$Seq^{'}$*: Conjunto de sequências de naturais de comprimento finito.*

$(s, s^{'}) \in r \iff boa(s^{'}) \wedge mesma(s, s^{'})$

onde:

boa diz que s é uma sequência ordenada e mesma indica que s e $s^{'}$ *tem os mesmos elementos.*

A: Sem restrições.

Os predicados *boa* e *mesma* estão amarrados pela variável $s^{'}$, isto é, todas as fórmulas estão amarradas num só grupo. Porém, quando isto não acontece, um primeiro passo seria decompor o problema de acordo com os grupos de encadeamento que apareçam na especificação do r.

Definição 4.3.2 *[Bed87] : Seja um problema, cuja intenção tenha a seguinte forma:* $(x, y) \in r \iff L_1 \wedge ... \wedge L_n$, onde $L_1, ..., L_n$ são fórmulas atômicas. *Uma Decomposição Conjuntiva em Grupos de Cadeias* para P são os problemas $P_1, ..., P_k$, onde: $(x_i, y_i) \in r_i \iff L_{i1} \wedge ... \wedge L_{ij}$, $i = 1, ..., k$, *tal* que $L_{i1}, ..., L_{ij}$ estão amarradas em y, e $\nexists\ L_{ij+1} \in (\{L_1, ..., L_n\} - \{L_{i1}, ..., L_{ij}\})$ *tal* que $\{L_{i1}, ..., L_{ij}, L_{ij+1}\}$ também estejam amarradas em y, para *algum* $1 \leq j \leq (n - K)$.

$\bigcup_{i=1}^{k} L_i = \{L_1, ..., L_n\}$, e $n_{i=1}^{k} L_i = \emptyset\}$

onde: $L_i = \{L_{i1}, ..., L_{ij}\}$

Exemplo 4.3.0.3 *[Bed87]: Calcular para um prisma reto de base hexagonal regular, a área da base, a área da superfície externa e o volume, onde se conhece a medida em cm, da altura e do raio da base, como mostra a figura abaixo:*

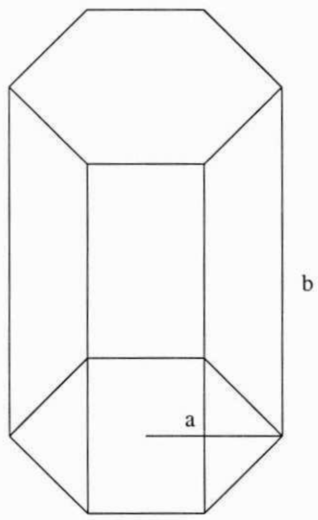

Figura 4.3: Prisma Reto de Base Hexagonal Regular

Uma estruturação para o problema é vista a seguir:

$P = \langle A \times B, AB \times AE \times V, A' \times B', r, A_d \rangle$

onde:

- A e B: *Conjunto dos reais positivos, que representam a medida do raio da base e da altura do prisma, em cm.*
- AB e AE: *Conjunto dos reais positivos, que representam a área da base e da superfície externa do prisma, em cm^2.*
- V: *Conjunto dos reais positivos, que representam o volume do prisma, em cm^3.*
- A' e B': *Conjunto dos reais positivos com limite superior.*
- $((a,b),(x,y,z)) \in r \iff AHR(a,x) \wedge AR(a,b,y) \wedge VPH(x,b,z)$

  onde:

  - $AHR(a,x)$ - *é um predicado que diz se área de um hexágono regular de raio a é x.*
  - $AR(a,b,y)$ - *é um predicado que diz se o volume de um retângulo, cujos os lados são a e b, é y.*
  - $VPH(x,b,z)$ - *é um predicado que diz se o volume de um prisma reto de base hexagonal, com a sendo a área da base e b sua altura, é z.*
- A_d - *sem restrições.*

Na definição da intenção, percebemos a existência de dois grupos de encadeamentos: um que corresponde a primeira e terceira fórmula, no qual existe uma dependência da última com relação a primeira, e o outro é um encadeamento simples correspondente somente à segunda fórmula.

Uma *Decomposição Disjuntiva em Grupos de Encadeamento* para P, pode ser expressa pelos problemas P_1 e P_2 representados estruturalmente abaixo:

$P_1 = \langle A \times B, AB \times V, A' \times B', r_1, A_1 \rangle$ e

$P_2 = \langle A \times B, AE, A' \times B', r_2, A_2 \rangle$

onde:

$((a,b),(x,z)) \in r_1 \Longleftrightarrow AHR(a,x) \wedge VPH(x,b,z)$

$((a,b),y) \in r_2 \Longleftrightarrow AR(a,b,y)$

A_1 e A_2 : Sem restrições.

Em se tratando de dois grupos de encadeamentos, pode-se, então, construir uma solução para o problema, a partir dos subproblemas P_1 e P_2, aplicando um rearranjo sobre o produto cartesiano das duas soluções, as quais dariam o resultado desejado. Sejam as funções $a_1 \Vdash P_1$ e $a_2 \Vdash P_2$, abaixo definidas:

$$a_1(a,b) = (3a^2\sqrt{3}/2, 3a^2b\sqrt{3}/2)$$

$$a_2(a,b) = 6ab$$

Seja a função $a'(a,b) = a_1(a,b) \times a_2(a,b)$, que faz a combinação das duas soluções, e seja a função $g : AB \times V \times AE \longrightarrow AB \times AE \times V$, definida por:

$$g(x,z,y) = (x,y,z)$$

Como as funções a_1 e a_2 satisfazem as fórmulas atômicas que compõem a intenção, e a função g dá ordem adequada, então $((a,b), g \circ a'(a,b)) \in r$. Portanto, $g \circ a' \Vdash P$.

A construção de uma solução usando esta forma, induz o teorema de solubilidade visto a seguir:

Teorema 4.3.1 *[Bed87] : Seja P um problema, cuja intenção tem a forma $(x,y) \in r \Longleftrightarrow L_1 \wedge ... \wedge L_n$, onde existem k grupos de encadeamentos em y. Se $P_1, ..., P_k$ é uma Decomposição Conjuntiva para Grupos de Cadeias em P e se $\Vdash P_1, ..., \Vdash P_k$, então $\Vdash P$.*

PROVA: Sejam $a_1, ..., a_k$ soluções dos respectivos $P_1, ..., P_k$. Como cada uma destas soluções satisfaz um grupo de fórmulas atômicas de $\{L_1, ..., L_n\}$, e em conjunto satisfazem todas, então $a(x) = a_1(x_1) \times ... \times a_k(x_k)$, onde $x_1, ..., x_k$ são subestruturas de x, satisfazendo $L_1, ..., L_n$ em r, pois estas dão o resultado numa outra ordem. Logo, se $g : R_1 \times ... \times R_k \longrightarrow R$, é uma função que muda a ordem de seus parâmetros, tem-se que $g \circ a \; \|\!- \; P$.

Os problemas resultantes quando da aplicação de uma decomposição em grupos de cadeias, tem sua intenção composta de fórmulas atômicas amarradas. Mas, ao decompor este em subproblemas, chega-se ao fenômeno chamado "encadeamento de problemas". Para problemas deste tipo, se procede do seguinte modo:

1. O problema que apresenta a intenção na forma de uma sentença conjuntiva de fórmulas atômicas, que é o menor subconjunto das fórmulas que estão amarradas, e que seja identicamente solúvel, será considerado como base de encadeamento.

2. As variáveis de saídas das fórmulas atômicas que formam os problemas gerados no ítem acima, serão considerados como entradas nas fórmulas restantes, que formarão conjuntamente a intenção de um problema que vai ser encadeado com o problema base do encadeamento.

3. Se o problema do ítem 2, ainda tem fórmulas amarradas na intenção, então se poderá fazer um outro nível de análise , seguindo este mesmo procedimento ou aplicando mais uma vez uma decomposição em grupos de cadeias, se necessário.

Veremos então uma formalização da definição para a decomposição conjuntiva:

Definição 4.3.3 *[Bed87] : Seja P um problema, cuja intenção tem a forma $(x, y) \in r \Longleftrightarrow L_1 \wedge ... \wedge L_n$, onde $L_1, ..., L_n$ são fórmulas atômicas que estão encadeadas por variáveis de saídas. Uma Decomposição Conjuntiva em Cadeia para P são os problemas P_1 e P_2, onde:*

1. *P_1 é um problema identicamente solúvel que tem sua forma $(x_1, y_1) \in r_1 \Longleftrightarrow L_{11}, ..., L_{1j}$, onde $\{L_{i1} \wedge ... \wedge L_{ij}\} \subseteq \{L_1, ..., L_n\}$, $a_1(x_1) = a_2(x_1)$. E $\nexists \{L_{1'1}, ..., L_{1'j'}$*

} com $j' < j$, tal que se P_1' seja um problema identicamente solúvel e cuja intenção seja $(x_1, y_1) \in r_1 \Longleftrightarrow \{L_{1'1}, ..., L_{1'j'}\}$.

2. *As variáveis de saída dos $L_{11}, ..., L_{1j}$, são consideradas como de entrada nas fórmulas restantes, isto é, em $\{L_{21}, ..., L_{2j'}\} = \{L_1, ..., L_n\} - \{L_{i1}, ..., L_{ij}\}$.*

   O problema P_2 tem como intenção $(x_2, y_2) \in r_2 \Longleftrightarrow L_{21} \wedge ... \wedge L_2 j'$.

Observações:

- O problema P_2 ainda pode ser decomposto conjuntivamente, e assim teremos m níveis de decomposição.
- P_1 é o menor problema em P solúvel identicamente, e embora possa ter mais de uma fórmula atômica em sua intenção, não pode ser decomposto, por que não se tem nenhuma forma geral que permita a qualquer solução de cada subproblema surgido, combiná-la para obter a solução requerida.
- Como as variáveis de saída de r_1 são consideradas como entradas em r_2, então $R_1 \cap R_2 = \emptyset$, e existe uma função $g : R_1 \times R_2 \longrightarrow R$, tal que g é um rearranjo.

Teorema 4.3.2 *[Bed87] : Seja P um problema, cuja intenção tem a forma $(x, y) \in r \Longleftrightarrow L_1 \wedge ... \wedge L_n$, onde $L_1, ..., L_n$ estão amarradas. Se P_1 e P_2 é uma decomposição conjuntiva em cadeia para P e $\| - P_1$ e $\| - P_2$, então $\| - P$.*

PROVA: Seja a_1 e a_2 as respectivas soluções de P_1 e P_2. Seja a função $a(x) = a_1(x_1) \times a_2(x_2, y_1)$, onde x_1 e x_2 são subestruturas de x, enquanto y_1 é uma subestrutura de $a_1(x_1)$. Como as funções a_1 e a_2 dão resultados parciais de y em qualquer ordem, não se pode dizer que $a \in A$, mas se pode aplicar um rearranjo sobre esta função de tal maneira que se dê a ordem necessitada. Logo, a função $g : R_1 \times R_2 \longrightarrow R$, muda unicamente a ordem de suas entradas, e como $g \circ a \in A$ e a satisfaz cada um dos L_i, então $g \circ a \| - P$.

4.4 Conclusão

Apresentamos neste capítulo, uma análise da sintaxe do predicado intencional de um problema, definido numa forma normal disjuntiva. Para tanto, foi necessário o estudo das diferentes situações que ocorriam ao definir a intenção, as condições sobre as quais se podia decompor o problema, e em quais subproblemas se podia decompor e como construir uma solução deste baseado nas soluções dos subproblemas assim surgidos. A árvore abaixo, mostra os diferentes níveis de decomposição que são geradas quando é feita uma análise desta natureza.

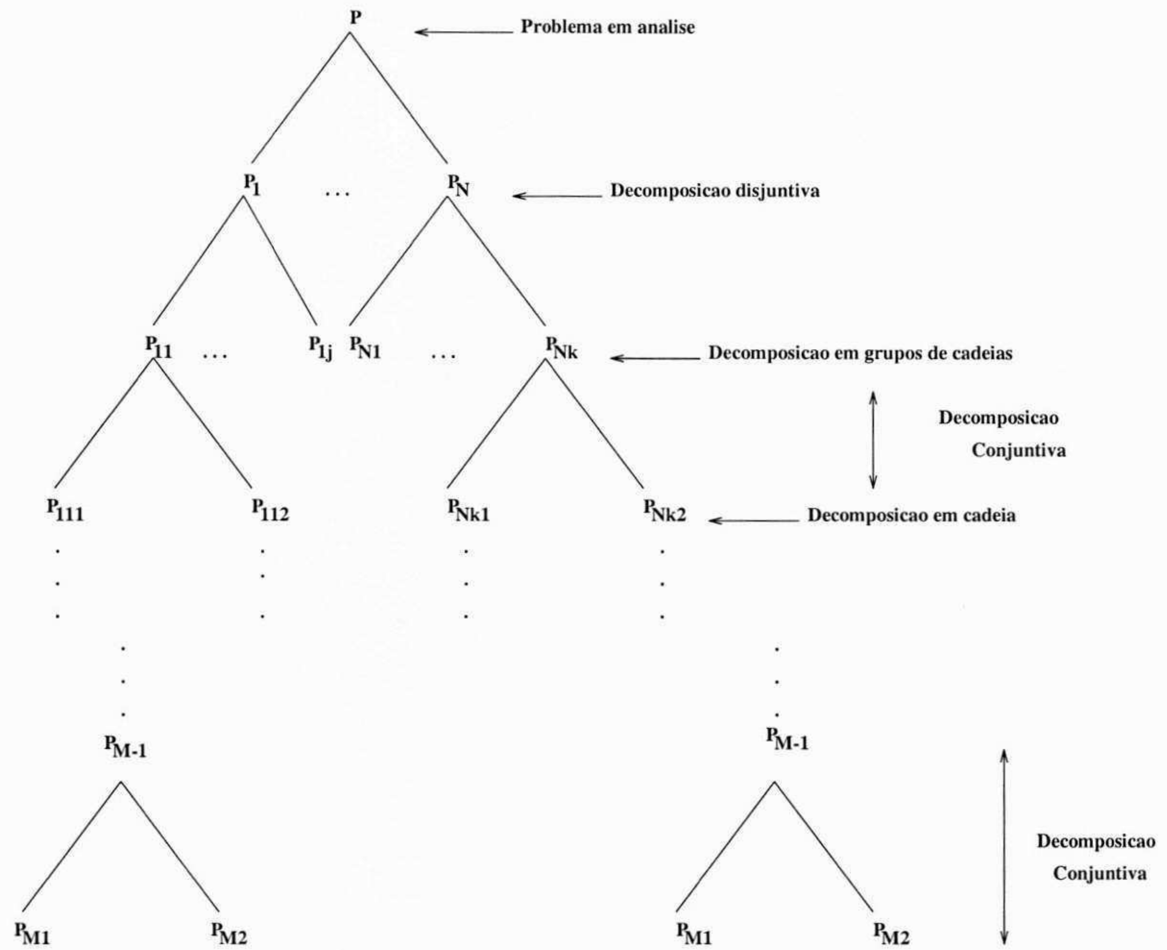

Figura 4.4: *Árvore de decomposição derivada de uma análise sintática de um problema*

Assim, num primeiro nível, temos a decomposição disjuntiva. Portanto, é possível aplicar uma decomposição conjuntiva aos subproblemas que dela são resultantes. Esta decomposição é feita até que se obtenha problemas indivisíveis, seja porque sua intenção é definida por uma fórmula atômica ou porque é determinada por uma cadeia destas, que estão amarradas de tal forma que qualquer quebra da intenção do problema geraria somente problemas que tivessem duas soluções que dêem como resultados valores diferentes para um mesmo valor de entrada.

Os três tipos de decomposição ilustrados na figura quebra o problema estruturalmente. No entanto, pode-se determinar morfismo entre problemas que são decomponíveis via análise da intencionalidade, com problemas decomponíveis estruturalmente.

Capítulo 5

A Classe de Problemas do "Tipo Determine"

5.1 Introdução

Na maioria das vezes, quando tentamos resolver problemas, usamos métodos de soluções utilizados em outros problemas anteriormente. Isto acontece, devido a certas características comuns existentes entre eles. Se houver um agrupamento entre esses problemas, possivelmente poderão constituir uma "classe".

O objetivo deste capítulo é o estudo de uma classe específica de problemas, mais precisamente a classe de problemas "tipo determine". Metodologicamente, o que vamos estudar é de grande importância, pois a partir de um problema, aplicando transformações estruturais entre problemas da mesma classe, poderemos inferir sobre a solubilidade dos problemas da classe e mostrar soluções para esses problemas.

5.2 Paradigmas de Classes de Problemas

De acordo com [Lop81], um questionamento de grande importância na TGP é sobre a existência de problemas que possam ser considerados iniciais (paradigmas) em suas respectivas classes, no sentido de que as solubilidades dos problemas de uma determinada classe possam ser inferidas, utilizando-se a solubilidade de um dado problema, chamado "problema inicial". Tais paradigmas devem ter a propriedade de permitir o acesso a qualquer outro problema da classe onde estão inseridos, por um único caminho. Deste modo, podemos estabelecer metodologias seguras de solução de problemas especiais.

Para o estudo de paradigmas de classes de problemas é necessário a busca de relações entre o paradigma e um problema qualquer da classe. Essas relações são feitas entre o espaço de soluções do problema paradigma e o espaço de soluções de um problema qualquer da classe. Isto nos leva à preservação de propriedades através de transformações estruturais. Usaremos homomorfismos entre problemas, pois esses preservam as estruturas dos paradigmas das classes e são caminhos eficientes para que possamos inferir sobre a solubilidade de problemas da classe, via soluções de paradigmas.

É interessante caracterizar os paradigmas de classes de maneira que, seus problemas tenham a parte substantiva vista como estruturas relacionais. Para definirmos essas classes de problemas, será necessária a caracterização de problemas do mesmo tipo.

Definição 5.2.1 *[Alm90] : Sejam $\mathcal{P}$ e $\mathcal{P}'$ dois problemas cujos componentes D e R de $\mathcal{P}$ e D' e R' de $\mathcal{P}'$ são estruturas relacionais. Diremos que $\mathcal{P}$ e $\mathcal{P}'$ são **problemas do mesmo tipo** se, e somente se, as seguintes condições são satisfeitas:*

- *D e D', R e R' são estruturas correspondentemente do mesmo tipo;*
- *Existe uma semelhança β de $\mathcal{P}$ e $\mathcal{P}'$.*

A exigência de uma semelhança entre dois problemas do mesmo tipo é de fundamental importância, pois podemos ter dois problemas, em que seus componentes da parte substantiva são estruturas do mesmo tipo porém, as partes intencionais são totalmente diferentes. Isto implica, a não possibilidade de dois problemas quaisquer serem do mesmo tipo.

Definição 5.2.2 *[Alm90] : Ao conjunto de todos os problemas do tipo de um problema $\mathcal{P}$, chamaremos de classe de problemas do tipo de $\mathcal{P}$, que denotaremos por:*

$$\mathcal{C}(\mathcal{P}) = \{ \mathcal{P}' | \mathcal{P}' \equiv \mathcal{P} \}$$

Denotaremos por $\mathcal{P}' \equiv \mathcal{P}$, quando os problemas $\mathcal{P}$ e $\mathcal{P}'$ forem do mesmo tipo.

Definição 5.2.3 *[Alm90] : Sejam $\mathcal{P}$ um problema e $\mathcal{C}(\mathcal{P})$ a classe de problemas do tipo de $\mathcal{P}$. Diremos que $\mathcal{P}$ é um paradigma (estrutural) de $\mathcal{C}(\mathcal{P})$ se, e somente se, para todo $\mathcal{P}' \in \mathcal{C}(\mathcal{P})$, existe um único homomorfismo total de $\mathcal{P}$ em $\mathcal{P}'$.*

Teorema 5.2.1 *[Alm90] : Sejam $\mathcal{P}$ um problema e $\mathcal{C}(\mathcal{P})$ a classe de problemas do tipo de $\mathcal{P}$. Suponha que $\mathcal{P}$ tem as seguintes propriedades:*

1. *As componentes D e R de $\mathcal{P}$, são estruturas iniciais nas suas respectivas classes;*

2. *Se $\mathcal{P}'$ é um problema qualquer de $\mathcal{C}(\mathcal{P})$ e $\phi : D \longrightarrow D'$ e $\psi : R \longrightarrow R'$, são homomorfismos entre as respectivas estruturas, então, (ϕ, ψ) é um homomorfismo de $\mathcal{P}$ em $\mathcal{P}'$.*

Então $\mathcal{P}$ é um paradigma de $\mathcal{C}(\mathcal{P})$.

PROVA: Seja $\mathcal{P}$ um problema de uma classe $\mathcal{C}(\mathcal{P})$ tal que, as componentes D e R de $\mathcal{P}$ são estruturas iniciais em suas respectivas classes. Sejam $\mathcal{P}'$ um problema qualquer de $\mathcal{C}(\mathcal{P})$ e $\mathcal{H} = (\phi, \psi)$ um homomorfismo entre $\mathcal{P}$ e $\mathcal{P}'$ tal que, $\phi : D \longrightarrow D'$ e $\psi : R \longrightarrow R'$ são homomorfismo entre as respectivas estruturas. Devemos mostrar que $\mathcal{H}$ é o único homomorfismo total entre $\mathcal{P}$ e $\mathcal{P}'$.

1. $\mathcal{H}$ é um homomorfismo total por hipótese;

2. $\mathcal{H}$ é único.

Seja $\mathcal{H}' = (\phi', \psi')$ outro homomorfismo entre os problemas $\mathcal{P}$ e $\mathcal{P}'$ tal que, as funções ϕ' e ψ' são homomorfismos entre as estruturas D e D', R e R', respectivamente. Como, por hipótese, D e R são estruturas iniciais, ϕ e ψ são os únicos homomorfismos entre as respectivas estruturas. Assim, podemos concluir que $\phi = \phi'$ e $\psi = \psi'$. Logo, $\mathcal{H} = \mathcal{H}'$. Isto mostra que o homomorfismo entre $\mathcal{P}$ e um problema qualquer da classe é único. •

5.3 O que é um Problema do "Tipo Determine"

A visualização da classe de problemas do "tipo determine" tem uma razoável facilidade de compreensão, porém , isso só acontece se forem considerados os questionamentos sugeridos por [Pol78] e seguindo os métodos de solução existentes na Teoria Geral de Problemas (TGP) [Lop81].

De acordo com [Pol78], as partes principais de um problema do "tipo determine" são: a incógnita, os dados e a condicionante. Devemos conhecê-las profundamente. Para isso, fazem-se necessárias algumas indagações e sugestões: Qual é a incógnita? Quais são os dados? Qual é a condicionante? Separe as várias partes da condicionante. Procure a relação entre os dados e a incógnita. Procure pensar num problema conhecido que tenha a mesma incógnita ou outra semelhante.

Intuitivamente, sabemos que um problema do "tipo determine" se caracteriza pela busca de um resultado desconhecido, e que tal resultado é uma decorrência da manipulação dos dados fornecidos. Isto é, a questão que envolve um problema desta classe é determinar um certo objeto, o resultado do problema , satisfazendo a condição do problema a qual relaciona o resultado com os dados do problema.

Os problemas de determinação podem ser práticos ou teóricos, abstratos ou concretos. Nos problemas práticos (Ex.: engenharia - problema de barragem) temos uma grande quantidade de dados e condicionantes. O que não acontece com os problemas teóricos (Ex.: matemáticos) bem formulados, onde todos os dados e todas as condicionantes são essenciais e têm de ser levados em conta.

A incógnita pode ser de todo tipo imaginável, por exemplo:

> Em um problema de construção geométrica a incógnita é a figura (triângulo, losango, etc.)
>
> Quando resolvemos uma equação algébrica, a incógnita é um número, ou seja, uma raiz daquela equação.

Existem dois métodos que visam diretamente as soluções de problemas: Redução e Decomposição. Para o estudo da solução de problemas do "tipo determine", trabalharemos, na maioria das vezes, com o método da Decomposição Estrutural, pois é comum, nesta classe existirem problemas de relativa complexidade, e fica mais fácil encará-lo por etapas, ou seja, solucioná-los aos poucos, primeiro uma parte e com esse resultado solucionar a outra parte, até obter a solução do problema todo.

Ao analisarmos a intenção de um problema da classe "tipo determine", além das fórmulas atômicas, percebemos claramente a presença de quantificadores universais e existenciais, retratando assim a necessidade de se fazer uma descrição lógica para o estudo desta classe de problemas.

No estudo da intencionalidade feita no capítulo 4, não foi considerado os quantificadores universais e existenciais na definição da intenção de um problema, houve então um pouco de perda da naturalidade na definição e um aumento do grau de dificuldade de solução dos problemas indivisíveis.

Segundo [LA85], quando descrevemos problemas via lógica, na verdade estamos tratando-os como entidades distintas sintaticamente e consequentemente lógicas distintas (sistemas formais distintos). Um modo de fazer um tratamento lógico sobre um esquema de solução é a possível realização do processo de *Skolemização*. Assim, uma lógica de solução tratará de soluções como esquemas (funcionais) abstratos, que estarão posteriormente sujeitos a realizações, isto é, 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.

"*Skolemizar* significa eliminar quantificadores existenciais. Isto é, uma fórmula que contém uma variável quantificada em termos existenciais afirma que existe um valor que pode ser substituído pela variável e que faz com que a fórmula seja verdadeira. Podemos eliminar o quantificador substituindo a variável por uma referência a uma função que produza o valor desejado. Uma vez que não sabemos necessariamente como produzir o valor, precisamos criar um novo nome de função para cada substituição" [RK93].

Assim, a eliminação do quantificador existencial de uma dada sentença θ ocorre da seguinte forma:

Seja θ uma fórmula corrente. Cria-se uma nova fórmula substituindo a subfórmula de θ da forma $\exists y\ (\varphi)$ por $Skl(\varphi)[y/f(x_1...x_n)]$, onde $x_1...x_n$ são variáveis livres de $\exists y\ (\varphi)$ e f é qualquer símbolo funcional n-ário que não ocorre em θ e $f \in R^D$.

Na Teoria Geral de Problemas temos que, uma solução para um problema seria associar um determinado dado, restrito ao domínio de dados, a um elemento do domínio de resultados, e isso aconteceria satisfazendo uma intenção do problema. Para a classe dos problemas "tipo determine", teremos no processo de aperfeiçoamento (refinamento) da intenção a possível necessidade da *Skolemização*. Logicamente falando, nesse aperfeiçoamento chega-se a uma etapa em que temos uma fórmula do seguinte modo:

$$(\forall c \exists x \varphi(c,x) \longrightarrow (c,x) \in r)$$

Na sentença $\forall c\ \exists x\ \varphi(c,x)$, c representa todos os dados fornecidos do problema, e x representa o resultado que deverá ser encontrado a partir dos dados c fornecidos, satisfazendo as condicionantes do problema. A presença do quantificador existencial na sentença acima, cujo escopo é x, é uma característica importante para os problemas do "tipo determine", pois indica que existe um resultado para o problema.

Sendo a questão colocada desta maneira, encontrar x significa eliminar o quantificador existencial, o que aponta para a necessidade de se skolemizar a sentença ($\forall c\ \exists x\ \varphi(c,x)$), isto é, x deverá ser substituído por uma função de c devidamente escolhida e que seja admissível, e a sentença ($\forall c\ \exists x\ \varphi(c,x)$) deverá continuar válida. Porém, estamos tratando com problemas, e isto implica que devemos mostrar a sentença acima dentro da estrutura de problema definida na TGP [Lop81].

Definição 5.3.1 : *Um problema $P = \langle D, R, d, r, A\rangle$ é um problema "tipo determine" se existe φ tal que, $(\forall c \in D\ \exists x \in R\ \varphi(c,x) \longrightarrow (c,x) \in r)$.*

Na definição acima, a identificação de um problema como sendo do "tipo determine" somente poderá ser dada a partir da exibição de φ. Se φ for exibida, entao P é do "tipo determine", se φ não for exibida nada se poderá dizer a respeito de P ser ou não do "tipo determine". Desse modo, a classe de problemas do "tipo determine" se caracteriza como sendo recursivamente enumerável. Portanto, a resolução de um problema do "tipo determine" deverá abranger duas questões fundamentais:

1. Exibição de φ nos moldes da definição.
2. Skolemizar a sentença $(\forall c\, \exists x\ \varphi(c,x))$ e verificar se a Skolemização é admissível.

A idéia intuitiva de problema do "tipo determine" é preservada na definição 5.3.1 pela presença do quantificador existencial ligada ao resultado a ser encontrado.

Teorema 5.3.1 : *Seja $P = \langle D, R, d, r, A\rangle$ um problema. Se existe φ tal que $\forall c \in d$ $(c, Skl(\varphi(c))) \in r$ então P é um problema do "tipo determine".*

PROVA: Seja α tal que $\forall c \in d\ r(c, Skl(\alpha(c)))$. Se $\forall c \in d\ r(c, Skl(\alpha(c)))$, então $\forall c\, \exists x$ $r(c,x)$. Logo $(\forall c\, \exists x\ r(c,x) \longrightarrow (c,x) \in r)$. Daí, de acordo com a definição de problema do "tipo determine" podemos tomar $\phi(c,x) = r(c,x)$. E assim, $(c, Skl(\phi(c))) \in r$. •

Corolário 5.3.1 : *Se P é um problema "tipo determine" e φ é tal que $Skl(\varphi) \in A$ então $Skl(\varphi) \Vdash P$.*

PROVA: Pela definição de solução de problema temos que para a função a satisfazer a intencionalidade de um problema P qualquer, ela deve ser admissível, isto é, $a \in A$ e $\forall c \in d\ (c, a(c)) \in r$. Fazendo $a = Skl(\varphi)$, então $Skl(\varphi) \Vdash P$. •

5.3.1 Exemplos

Exemplo 5.3.1.1 :

Determinar as raízes da equação do $2^{\underline{o}}$ grau $ax^2 + bx + c = 0$, com $a \neq 0$, no domínio dos reais.

Intuitivamente, o problema enunciado é do "tipo determine" porque estamos buscando um resultado, neste caso, as raízes da equação do 2º grau.

Uma estruturação para o problema é a seguinte:

$P = \langle D, R, d, r, A \rangle$

$D = \mathcal{R}^3$

$R = \mathcal{R}$

$d = \{(a_0, a_1, a_2) \mid a_1^2 - 4a_0a_2 \geq 0 \}$

$((a_0, a_1, a_2), x) \in r \iff a_0x^2 + a_1x + a_2 = 0$

$A = \{f \mid f : D \longrightarrow R \wedge Adm(f)\}$

onde:

$Adm(f)$ = *propriedade de* f *ser algorítmica e ser descrita por radicais.*

A partir da estruturação apresentada acima para o problema, vamos descrevê-lo formalmente nos moldes da definição 5.3.1. Para simplificar o processo de resolução do problema, usaremos a seguinte notação: $a_0 = a$, $a_1 = b$ *e* $a_2 = c$.

$(\forall a \in D\ \forall b \in D\ \forall c \in D\ \exists y \in R\ (P^2[x], y)) \longrightarrow (P^2[x], y) \in r$

De acordo com a definição 5.3.1, um problema do "tipo determine" é caracterizado pela exibição de φ. *Como foi dito anteriormente com relação a* φ, *a mesma deverá conter os elementos da intencionalidade do problema. Assim, um candidato natural para* φ *é a sentença abaixo:*

$(\forall a \in D\ \forall b \in D\ \forall c \in D\ \exists y \in R\ \varphi((a, b, c, y)) \longrightarrow (P^2[x], y) \in r$

A presença do quantificador existencial na sentença que define o problema como sendo do "tipo determine", nos indica que para chegarmos ao resultado procurado precisaremos fazer a Skolemização da sentença. Neste caso, o processo de Skolemização será o caminho para chegarmos na solução do problema, uma vez que pelo corolário

5.3.1 temos que, se a função $Skl(\varphi)$ é admissível, então ela é solução para o problema. Assim, o estudo do processo da Skolemização é feito sobre a sentença acima:

Seja $(\forall a \in D\ \forall b \in D\ \forall c \in D\ \exists y \in R\ \varphi((a,b,c,y)) \longrightarrow (a,b,c,y) = 0$ *uma fórmula* θ *. Na subfórmula de* θ*,* $\exists y \in R\ \varphi((a,b,c,y)$*, tomemos* $\varphi((a,b,c,y) = \beta$*, passando a subfórmula de* θ *a ser* $\exists y \in R\ (\beta)$*. Agora, cria-se uma nova fórmula substituindo* $\exists y \in R\ (\beta)$ *por* $Skl(\beta)[y/f(a,b,c)]$*, onde* a,b,c *são variáveis livres de* $\exists y \in R(\beta)$ *e* f *é qualquer símbolo funcional n-ário que não ocorre em* θ *e* $f \in R^D$*.*

O novo símbolo funcional f *introduzido na fórmula* θ *é chamado de função de Skolem e o processo de substituição de* y *por* $f(a,b,c)$ *é chamado de Skolemização.*

Nesse exemplo, usaremos a demonstração da fórmula de Bhaskara no processo de Skolemização, verificando uma série de passos:

1. *Dividir a equação* $ax^2 + bx + c = 0$ $(a \neq 0)$ *por* a*:*

   $x^2 + \frac{b}{a}x + \frac{c}{a} = 0$

2. *Passar o* $\frac{c}{a}$ *para o segundo membro:*

   $x^2 + \frac{b}{a}x = -\frac{c}{a}$

3. *Completar o trinômio quadrado perfeito:*

   $x^2 + \frac{b}{a}x + \frac{b^2}{4a^2} = \frac{b^2}{4a^2} - \frac{c}{a}$

4. *Fatorar o primeiro membro:*

   $(x + \frac{b}{2a})^2 = \frac{b^2-4ac}{4a^2}$

5. *Extrair a raiz quadrada (supondo* $b^2 - 4ac \geq 0$*)*

   $x + \frac{b}{2a} = \pm\sqrt{\frac{b^2-4ac}{4a^2}}$

6. *Isolar o* x*:*

$x = -\frac{b}{2a} \pm \sqrt{\frac{b^2-4ac}{4a^2}}$

7. *Definir os valores de x:*

$$x_1 = -\frac{b + \sqrt{b^2 - 4ac}}{2a}$$

e

$$x_2 = -\frac{b - \sqrt{b^2 - 4ac}}{2a}$$

Assim, veremos que a função de Skolem para o problema é dada da seguinte forma:

$f_1(a,b,c) = x_1 = -\frac{b+\sqrt{b^2-4ac}}{2a}$ *ou*

$f_2(a,b,c) = x_2 = -\frac{b-\sqrt{b^2-4ac}}{2a}$

Portanto, temos que as funções $f_1(a,b,c)$ e $f_2(a,b,c)$ são algorítmicas e satisfazem r para todos os elementos de d. Então $f_1(a,b,c)$ e $f_2(a,b,c)$ são as soluções para P, isto é, $((f_1(a,b,c) \Vdash P)$ e $(f_2(a,b,c) \Vdash P))$.

Exemplo 5.3.1.2 :

> *Calcular o volume de um paralelepípedo, do qual se conhece sua largura (x), sua altura (y) e seu comprimento (z).*

Podemos visualizar estruturalmente o problema $P = \langle D \times D \times D, V, d' \times d' \times d', r, A\rangle$:

onde:

D = *Conjunto das medidas em $\mathcal{R}$ (cm)*

V = *Conjunto das medidas em $\mathcal{R}^3$ (cm^3)*

d = *Conjunto das medidas finitas em $\mathcal{R}$ (cm)*

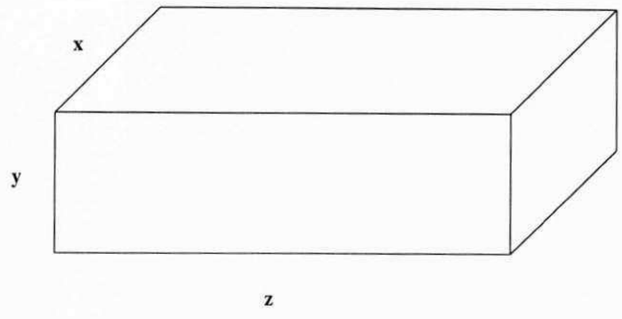

Figura 5.1: Paralelepípedo de largura (x), altura (y) e comprimento (z)

$((x,y,z),v) \in r \Longleftrightarrow v =$ *volume do paralelepípedo*

$A =$ *Sem restrições*

De acordo com a definição 5.3.1, um problema do "tipo determine" é caracterizado pela exibição de φ. Como foi dito anteriormente com relação a φ, a mesma deverá conter os elementos da intencionalidade do problema. Assim, um candidato natural para φ é a sentença abaixo:

$$(\forall x \in D\ \forall y \in D\ \forall z \in D\ \exists v \in V\ \varphi((x,y,z),v))) \longrightarrow ((x,y,z),v) \in r$$

A presença do quantificador existencial na sentença que define o problema como sendo do "tipo determine", nos indica que para chegarmos ao resultado procurado precisaremos fazer a Skolemização da sentença. De acordo com o corolário 5.3.1, teremos que encontrar uma função que seja admissível e que dados x, y, z, ela retorne o volume do paralelepípedo.

Seja $(\forall x \in D\ \forall y \in D\ \forall z \in D\ \exists v \in V\ \varphi((x,y,z),v))) \longrightarrow ((x,y,z),v) \in r$ uma fórmula ρ. Na subfórmula de ρ da forma $\exists v \in V\ \varphi((x,y,z),v)) = \alpha$ passando a subfórmula de ρ a $\exists v \in V\ (\alpha)$. Agora, cria-se uma nova fórmula substituindo $\exists v \in V\ (\alpha)$ por $Skl(\alpha)[v/f(x,y,z)]$,onde x,y,z são variáveis livres de $\exists v \in V\ (\alpha)$ e f é qualquer símbolo funcional n-ário que não ocorre em ρ e $f \in R^D$.

Assim, a função de Skolem procurada é dada da seguinte forma:

$$f(x,y,z) = x.y.z$$

Portanto, temos que a função $f(x,y,z)$ satisfaz r para todos os elementos de d, ou seja, $\forall (x,y,z) \in d \; ((x,y,z), f(x,y,z)) \in r$, então $f(x,y,z)$ é uma solução para P, isto é, $f(x,y,z) \Vdash P$.

Exemplo 5.3.1.3 *:[Bed87]*

> *Calcular o volume e a superfície da parte externa de um cilindro, sendo conhecido o raio da base(a) e sua altura(h) como mostra a figura abaixo:*

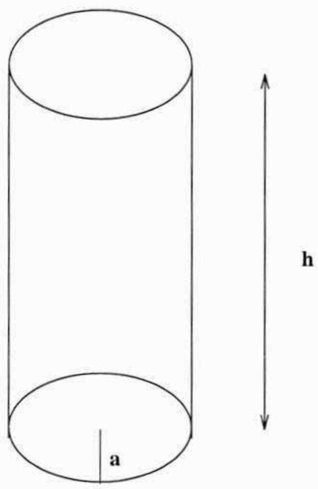

Figura 5.2: Cilindro de raio (a) e altura (h)

Podemos visualizar o problema estruturalmente do seguinte modo:

$$P = \langle D \times D, V \times S, d' \times d', r, A \rangle$$

onde:

D = *Conjunto das medidas em* $\mathcal{R}$ (cm)

V = *Conjunto das medidas em* $\mathcal{R}^3$ (cm^3)

S = *Conjunto das medidas em* $\mathcal{R}^2$ (cm^2)

d = *Conjunto das medidas finitas em* $\mathcal{R}$ (cm)

$((a,h),(v,s)) \in r \iff vol((a,h),v) \wedge sup(v,s)$

A = *Sem restrições*

De acordo com a definição 5.3.1, um problema do "tipo determine" é caracterizado pela exibição de φ. Como foi dito anteriormente com relação a φ, a mesma deverá conter os elementos da intencionalidade do problema. Assim, um candidato natural para φ é a sentença abaixo:

$(\forall a\ \forall h\ \exists v\ \exists s\ \varphi((a,h),(v,s)) \longrightarrow ((a,h),(v,s))) \in r$

Para facilitar a resolução do problema, vamos trabalhar por etapas: 1- calcular o volume de um cilindro dados seu raio e sua altura. 2- calcular a superfície de um cilindro do qual se conhece o seu volume. Então uma interpretação estrutural para o problema seria:

$$P_1 = \langle D \times D, V, d' \times d', r_1, A_1 \rangle$$

onde:

D = *Conjunto das medidas em* $\mathcal{R}$ (*cm*)

V = *Conjunto das medidas em* $\mathcal{R}^3$ (cm^3)

d' = *Conjunto das medidas finitas em* $\mathcal{R}$ (*cm*)

$((a,h),v) \in r_1 \iff vol((a,h),v)$

A_1 = *Sem restrições*

De acordo com a definição 5.3.1, um problema do "tipo determine" é caracterizado pela exibição de φ. Como foi dito anteriormente com relação a φ, a mesma deverá conter os elementos da intencionalidade do problema. Assim, um candidato natural para φ é a sentença abaixo:

$(\forall a \in D\ \forall h \in D\ \exists v \in V\ \varphi((a,h),v))) \longrightarrow ((a,h),v) \in r_1$

A presença do quantificador existencial na sentença que define o problema como sendo do "tipo determine", nos indica que para chegarmos ao resultado procurado precisaremos fazer a Skolemização da sentença. de acordo com o corolário 5.3.1 teremos *que encontrar uma função que seja admissível e que dados a, h, ela retorne o volume do cilindro.*

Seja $(\forall a \in D\ \forall h \in D\ \exists v \in V\ \varphi((a,h),v))) \longrightarrow ((a,h),v) \in r_1$ *uma fórmula* α. *Na subfórmula de* α *da forma* $\exists v \in V\ \varphi((a,h),v)) = \theta$, *passando a subfórmula de* θ *a* $\exists v \in V\ (\theta)$. *Agora, cria-se uma nova fórmula substituindo* $\exists v \in V\ (\theta)$, *por* $Skl(\theta) = [v/g_1(a,h)]$, *onde* a, h *são variáveis livres de* $\exists v \in V\ (\theta)$ *e* g_1 *é qualquer símbolo funcional n-ário que não ocorre em* α *e* $g_1 \in R^D$.

Assim, a função de Skolem procurada é dada da seguinte forma:

$$g_1(a,h) = \pi.a^2.h$$

Portanto, temos que a função $g_1(a,h)$ *é algorítmica e satisfaz* r *para todos os elementos de* d', *ou seja,* $\forall(a,h) \in d'\ ((a,h), g_1(a,h)) \in r_1$, *então* $g_1(a,h)$ *é uma solução para* P_1, *isto é,* $g_1(a,h) \; \| \!- \; P_1$.

$$P_2 = \langle V, S, v', r_2, A_2 \rangle$$

onde:

V = *Conjunto das medidas em* $\mathcal{R}^3$ (cm^3)

S = *Conjunto das medidas em* $\mathcal{R}^2$ (cm^2)

v' = *Conjunto das medidas finitas em* $\mathcal{R}^3$ (cm^3)

$(v,s) \in r_2 \Longleftrightarrow sup(v,s)$

A_2 = *Sem restrições*

De acordo com a definição 5.3.1, um problema do "tipo determine" é caracterizado pela exibição de φ. *Como foi dito anteriormente com relação a* φ, *a mesma deverá conter os elementos da intencionalidade do problema. Assim, um candidato natural para* φ *é a sentença abaixo:*

$$(\forall v \in V\ \exists s \in S\ \varphi((v,s))) \longrightarrow (v,s) \in r_2$$

A presença do quantificador existencial na sentença que define o problema como sendo do "tipo determine", nos indica que para chegarmos ao resultado procurado precisaremos fazer a Skolemização da sentença. Teremos que encontrar uma função que seja admissível e que dado v, ela retorne a superfície do cilindro.

Seja $(\forall v \in V\ \exists s \in S\ \varphi((v,s))) \longrightarrow (v,s) \in r_2$ *uma fórmula* β. *Na subfórmula de* β *da forma* $\exists s \in S\ \varphi(v,s) = \alpha$, *passando a subfórmula de* β *a* $\exists s \in S\ (\alpha)$. *Agora, cria-se uma nova fórmula substituindo* $\exists s \in S\ (\alpha)$, *por* $Skl(\alpha) = [s/g_2(v)]$, *onde* v *é uma variável livre de* $\exists s \in S\ (\alpha)$ *e* g_2 *é qualquer símbolo funcional n-ário que não ocorre em* β *e* $g_2 \in R^D$.

Assim, a função de Skolem procurada é dada da seguinte forma:

$$g_2(v) = 2v/a$$

Portanto, temos que a função $g_2(v)$ *satisfaz* r_2 *para todos os elementos de* v', *ou seja,* $\forall v \in v'\ ((v), g_2(v)) \in r_2$, *então* $g_2(v)$ *é uma solução para* P_2, *isto é,* $g_2(v) \;||\!-\; P_2$.

Assim, temos que as funções g_1 *e* g_2 *são as respectivas soluções de* P_1 *e* P_2, *e como* $P = P_1 \circ P_2$ *então* $g = g_1 \circ g_2$ *é uma solução para* P, *isto é,* $g_1 \circ g_2 \;||\!-\; P$.

Exemplo 5.3.1.4 : *O Problema do Caixeiro Viajante*

> *Um vendedor tem uma lista de cidades que precisa visitar exatamente uma vez. Há estradas diretas entre cada par de cidades da lista. Encontre a rota que o vendedor deverá seguir para que a viagem seja a menor possível, e que comece e termine em uma mesma cidade, que poderá ser qualquer uma da lista.*

Estruturalmente, podemos analisar o problema assim:

$$P = \langle D, R, d, r, A \rangle$$

onde:

D = *Conjunto de listas de cidades* (L_c)

R = *Conjunto das rotas entre as cidades.*

d = *Conjunto de* n *cidades interligadas entre si* (l_c)

$(l_c, r_c) \in r \iff mesma(c_i, c_n) \wedge menor(r_c) \wedge única(c_i)$

A = *Funções algorítmicas.*

De acordo com a definição 5.3.1, um problema do "tipo determine" é caracterizado pela exibição de φ. *Como foi dito anteriormente com relação a* φ, *a mesma deverá conter os elementos da intencionalidade do problema. Assim, um candidato natural para* φ *é a sentença abaixo:*

$(\forall l_c \in D\ \exists r_c \in R\ \varphi(l_c, r_c))) \longrightarrow (l_c, r_c) \in r$

A presença do quantificador existencial na sentença que define o problema como sendo do "tipo determine", nos indica que para chegarmos ao resultado procurado precisaremos fazer a Skolemização da sentença. De acordo com o corolário 5.3.1, teremos que encontrar uma função que seja admissível e que dado um conjunto de n *cidades interligadas entre si, ela retorne uma rota.*

Seja $(\forall l_c \in D\ \exists r_c \in R\ \varphi(l_c, r_c))) \longrightarrow (l_c, r_c) \in r$ *uma fórmula* μ. *Na subfórmula de* μ *da forma* $\exists r_c \in R\ \varphi(l_c, r_c)) = \omega$, *passando a subfórmula de* μ *a* $\exists r_c \in R\ (\omega)$. *Agora, cria-se uma nova fórmula substituindo* $\exists r_c \in R\ (\omega)$ *por* $Skl(\omega)[r_c/a(l_c)]$, *onde* l_c *é uma variável livre de* $\exists r_c \in R\ (\omega)$ *e* a *é qualquer símbolo funcional n-ário que não ocorre em* μ *e* $a \in R^D$.

Assim, a função de Skolem procurada é dada da seguinte forma:

$$a(l_c) = r_c$$

Portanto, temos que a função $a(l_c)$ *é algorítmica* $(a(l_c) \in A)$ *e satisfaz* r *para todos os elementos de* d, *ou seja,* $\forall l_c \in d\ (l_c, a(l_c)) \in r$, *então* $a(l_c)$ *é uma solução para* P $(a(l_c) \ \|\!- P)$.

Analisando os exemplos acima, observamos que nos problemas do "tipo determine" a característica fundamental é buscar **qual é** o resultado para o problema no domínio dos resultados usando como solução, o processo da *Skolemização* e tal solução é uma função admissível. Vejamos então, um contra-exemplo, no qual conhecemos o resultado que estamos buscando, pertencente ao domínio dos resultados, e procuramos o caminho de **como** chegar a este resultado.

5.3.2 Contra-exemplo

Exemplo 5.3.2.1 : *O problema da Torre de Hanói,* ***nos termos abaixo****, não é um problema do "tipo determine".*

O problema requer três pinos, sendo que no primeiro há uma série de n discos concêntricos, cada qual menor em diâmetro que o disco imediatamente abaixo dele. O segundo e terceiro pinos estão inicialmente vazios. O objetivo é determinar a seqüência de movimentos necessários para mover todos os discos do pino 1 para o pino 3.

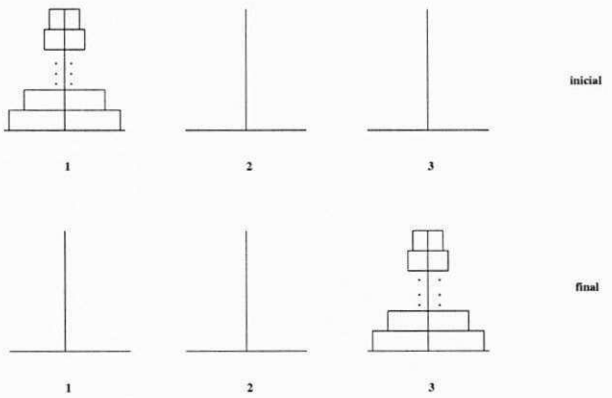

Figura 5.3: Esquema da Torre de Hanói

REGRAS:

- *Somente um disco pode ser movido por vez;*

- *Em nenhum momento um disco maior pode ser colocado sobre um menor;*
- *Um disco pode ser movido de um pino para qualquer outro pino.*

De acordo com o enunciado do problema, sabemos qual é o resultado, o que temos que encontrar é um caminho que, obedecendo as regras, possamos chegar até o resultado desejado. A solução admissível é um conjunto de movimentos que aplicados à configuração inicial chegará a uma configuração final.

Uma estruturação para o problema é a seguinte:

$P = \langle D, R, d, r, A \rangle$

onde:

$D = R = d =$ *Conjunto de configurações para três pinos e n discos*

$(Conf_{inc}, Conf_{fim}) \in r$ *se, e somente se,* $\forall n \in D$ $Conf_{inc} = (x_1...x_n, 0, 0)$ $\exists(C_1, C_2, ..., C_p)$ $\in A$ *tal que,* $C_p \circ ... \circ C_2 \circ C_1(x_n...x_1, 0, 0) = (0, 0, x_1...x_n) = Conf_{fim}$

$A =$ *Conjunto de funções algorítmicas*

Para chegar a uma configuração final deve-se fazer uma análise das operações de mover um disco de um pino para outro pino, analisar também quando uma configuração é considerada ***boa*** *e a partir daí encontrar um caminho que será a solução do problema.*

Primeiro veremos como mover um disco de um pino para outro pino:

Se $n = 1$:

A solução é trivial. Apenas transfere um disco do pino 1 para o pino 3.

Se $n \neq 1$:

A solução será:

1. *Mover* $n - 1$ *discos do pino 1 para o pino 2*

2. *Mover o disco 1 do pino 1 para o pino 3*

3. *Mover* $n-1$ *discos do pino 2 para o pino 3*

As operações necessárias para mover n discos são:

- $A_1 = (x_1...x_n, 0, 0) = (x_n, x_1...x_{n-1}, 0)$
- $A_2 = (0, x_1...x_{n-1}, x_n)$
- $A_3 = (x_1...x_{n-2}, x_{n-1}, x_n)$
- $A_4 = (x_1...x_{n-2}, 0, x_{n-1}x_n)$
- $A_5 = (0, x_1, x_2...x_n)$
- $A_6 = (0, 0, x_1...x_n)$

A análise das configurações é feita da seguinte forma:

$$Conf_{inc} = (x_1...x_n, 0, 0)$$
$$Conf_{fim} = (0, 0, x_1...x_n)$$

onde:

$x_1...x_n$ *são discos ordenados crescentemente de 1 para n.*

Para que uma configuração seja considerada **boa** *precisamos observar se a ordem de tamanho dos discos em cada pino é decresecente:*

- $tam(x_i) < tam(x_{i+1})$ *e*
- $tam(y_j) < tam(y_{j+1})$ *e*
- $tam(z_k) < tam(z_{k+1})$

Se $\forall i\ (1 \leq i \leq n)\ \forall j\ (1 \leq j \leq m)\ \forall k\ (1 \leq k \leq t)\ (Conf = (x_1 ... x_i x_{i+1} ... x_n, y_1 ... y_j y_{j+1} ... y_m, z_1 ... z_k z_{k+1} ... z_t))$ *então* $Conf \in boa$

A solução procurada para o problema é um caminho (C_p) *que será percorrido para mover os* n *discos do pino 1 para o pino 3. Assim,*

$$C_p \bullet ... \bullet C_2 \bullet C_1(Conf_{inc}) = Conf_{fim} \text{ e}$$

$$\forall q\ (1 \leq q \leq p) C_i = A_1 \vee A_2 \vee ... \vee A_6$$

Neste problema o caminho C_p *encontrado é o que conduz ao resultado já conhecido. Nos exemplos anteriores, o resultado não é conhecido, devendo portanto ser determinado.*

Capítulo 6

Especificação de Problemas "Tipo Determine"

6.1 Introdução

Neste capítulo estudaremos a questão da especificação de problemas do "tipo determine". Para isso, iniciaremos apresentando uma metodologia de especificação de problemas em geral e em seguida de problemas do "tipo determine".

Na maioria das vezes, um problema é enunciado utilizando-se uma linguagem sem muita precisão, ou seja, é dada uma descrição informal de um problema em linguagem natural e com isso, nos deparamos com eventuais características, que tornam pouco preciso o enunciado do problema. Algumas dessas características mais comuns são:

1. Presença de elementos no enunciado, que não dão informações importantes a qualquer característica do problema;
2. Existência de uma (ou mais) característica do problema, que não foi abordada por elemento do enunciado;
3. Presença de elementos no enunciado, que correspondem, não a uma característica do problema, e sim, a uma possível solução;

4. Presença de dois ou mais elementos no enunciado, que definem uma característica do problema, de uma forma incompatível (contradições);

5. Presença de elementos no enunciado, que possibilitam a definição de uma característica do problema de, pelo menos, duas maneiras distintas (ambigüidades);

6. Presença de elementos no enunciado, que definem uma característica de um problema, de uma maneira tal que, uma possível solução não possa ser realisticamente validada, com respeito a essa característica.

A partir dessas características pode-se ter certeza que, descrever um problema em linguagem natural leva a certas armadilhas. O papel do especificador é principalmente, transformar o enunciado numa especificação, isto é, o especificador deve identificar as dificuldades. Esse fato torna importante a especificação formal pois, a mesma ajudará ao especificador a não cometer possíveis enganos, como também, irá complementar a descrição informal do problema.

Baseado nesses argumentos, introduziremos na seção (6.2) as seguintes questões:

* O que é uma especificação?

* Para que serve uma especificação?

* Como é feita uma especificação?

Com a explanação de tais questões tem-se uma abordagem geral da necessidade de especificação formal de problemas de classes diversas.

6.2 Especificação de Problemas

6.2.1 O que é uma Especificação de Problemas?

Especificar um problema é descrevê-lo claramente, sem ambigüidades e acima de tudo corretamente. Normalmente quando nos deparamos com uma descrição precisa da realidade, vemos que é difícil o entendimento. Juntamente com essa dificuldade, alguns pressupostos contextuais, em geral tomados quando do enunciado do problema, devem aparecer explicitamente na especificação.

Não obstante as dificuldades, podemos especificar um problema através de técnicas utilizadas para especificação de tipos abstratos de dados. Essas técnicas geralmente dependem da experiência do especificador. É ele, com a sua visão macroscópica da realidade na qual está inserida o problema, que tenta identificar nessa "realidade" comportamentos que possam conduzir a interpretações factuais de modelos conhecidos, expondo ambigüidades e contradições, objetivando descrever as características do problema, de uma forma precisa e rigorosa.

Em se tratando de especificação de problemas, o procedimento acontece da seguinte forma:

- Um primeiro estágio, é responder, de maneira informal, as seguintes questões [Pol78]:

- Quais são os dados? [descrição de D]
- Quais são os resultados? [descrição de R]
- Qual é a intenção do problema? (Qual é a relação entre dados e resultados ou quais são as condições do problema?) [descrição de r]

- Depois de respondidas essas perguntas, de acordo com [Lop81], devemos responder, adicionalmente, segundo o contexto em que está inserido o problema, as seguintes perguntas:

- Quais são as restrições sobre os dados? (Quais são os dados relevantes no problema?) [descrição de d]
- Quais são as maneiras aceitáveis de relacionar dados com resultados? (Como são as soluções?) [descrição de A]

Ao se responder tais questões, na verdade, está se dando uma estruturação do problema, ainda que informal.

Uma vez estruturado informalmente o problema, inicia-se então a fase de especificação propriamente dita. Assim, um problema P será especificado através da especificação de seus componentes estruturais $\langle D, R, d, r, A \rangle$.

6.2.2 Para que serve uma Especificação de Problemas?

Especificar um problema é transformar uma descrição informal e pouco clara em uma descrição precisa e formal. Para isto, se faz necessário um estudo aprofundado a respeito do problema através de uma descrição formal e esta será a ponte de transformação.

Quando queremos resolver um problema que está na linguagem natural (nível abstrato) devemos descrevê-lo até chegar numa linguagem em que haja uma ligação entre a realidade e o solucionador (o homem e/ou máquina), e esta linguagem é a especificação. Esta descrição acontece num nível formal através da interpretação dos componentes do problema. A interpretação parte de uma estrutura genérica até chegar numa estrutura específica.

Com essa visão, tem-se um sentido minimal, pois a intenção do solucionador é chegar num modelo mínimo para aquele problema, através da transformação do nível abstrato para um nível concreto, com um sentido mais rico e consequentemente, de menor número possível de modelos para aquele problema.

6.2.3 Como é feita uma Especificação de Problemas?

Numa especificação de problemas são necessárias três etapas, quais sejam:

- Concretização ou Modelação do Problema

Consiste em estabelecer uma interpretação factual φ da estrutura abstrata do problema $P = \langle D, R, d, r, A \rangle$. Isto é, definir $\varphi = P \longrightarrow \mathcal{M}$.

onde:

$\mathcal{M}$ é um modelo para a estrutura P, ou seja, $\mathcal{M}$ será chamado de problema concreto.

Nessa etapa, podemos dizer que, de um modo geral, a passagem da estrutura abstrata P para uma específica $\mathcal{M}$, obedece a figura 6.1 [Bun76]:

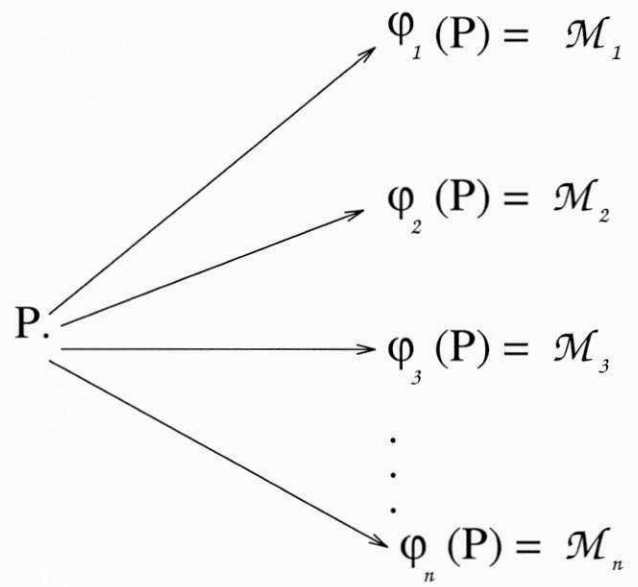

Figura 6.1: *Esquema de passagem da estrutura abstrata P para uma específica* $\mathcal{M}$

onde:

Cada $\mathcal{M}_i$ (i=1,...,n) é um modelo para P, ou seja, $\mathcal{M}_i$ é a imagem de alguma interpretação φ_i, que satisfaz à definição de problema.

- Especificar o Problema Concreto $\mathcal{M}$

Utiliza-se para isto as técnicas de especificação de tipos abstratos de dados. Ou seja, exibe a especificação de cada componente do problema, o que corresponde a especificar as interpretações $\varphi(D)$, $\varphi(R)$, $\varphi(d)$, $\varphi(r)$, $\varphi(A)$. Observe que as duas primeiras interpretações $\varphi(D)$ e $\varphi(R)$ são tipos de dados e como tais, possuem suas linguagens subjacentes, isto é, podem se apresentar como domínios de estruturas complexas, cujos componentes são transparentes a nível de estrutura abstrata do problema. Dessa forma, elas podem ser especificadas usando-se uma estratégia de especificação via descrição axiomática, algébrica, etc.. A parte intencional do problema abstrato (d, r, A) deve ser descrita utilizando a mesma linguagem que foi usada na especificação da parte substantiva (D e R).

É uma escolha do solucionador (especificador), que tipo de estratégia de especificação usar, observando os seus recursos técnicos e teóricos, bem como, a adequação da descrição do problema específico.

- Representação dos Tipos Envolvidos no Problema

A representação dos tipos envolvidos é feita utilizando-se uma linguagem mais formal, por exemplo: uma linguagem de primeira ordem, uma linguagem de programação como PASCAL, C, C++, etc., com a qual se representarão os símbolos das estruturas embutidas no problema.

Esta etapa é conduzida de modo que as soluções $a \in A$ sejam algorítmicas (como forma de descrever funções) e na maneira dos exemplos utilizados na Teoria Geral de Problemas , as soluções são tratadas como funções de D em R, devemos encará-las como maneiras de relacionar elementos de D com elementos de R de um modo razoavelmente seguro. Isso deve ficar claro se pensarmos que sendo $a \in A$ uma função (relação), a nível de terceira etapa da especificação poderíamos representar a mesma função a das maneiras mais diversas (na mesma linguagem) com os recursos mais variados (linguagens diferentes).

A avaliação que fazemos no espaço de soluções, e nesse nível são transparentes as estruturas D e R, é feita segundo as propriedades funcionais das soluções, como por exemplo, algo que reflita que uma solução computável é mais desejável que uma não computável, uma solução explícita é mais desejável que uma implícita, uma metodologia é melhor que uma heurística, um programa sem recursão é mais aceitável que um com recursão, uma função continuamente diferenciável é mais desejável que uma contínua, etc.

Existe o fato de A possuir dois "status" diferentes: na estrutura $P = \langle D, R, d, r, A \rangle$, onde A é um predicado (de segunda ordem) de admissibilidade, isto é, um predicado para funções de D e R, enquanto $\mathcal{M}$ é interpretado como um conjunto de funções (relações). Embora d e r também estejam sujeitos a essa observação, ela está voltada para o componente A devido ao fato de que, eventualmente, poderíamos confundir A como sendo mais um conjunto base para a estrutura abstrata P (tipo problema). Isso significaria um condicionante prévio muito forte na definição de problema, com perda de generalidade que se refletiria nas noções de solubilidade e transformações de problemas (homomorfismos, analogias, etc.).

Um problema abstrato $P = \langle D, R, d, r, A \rangle$, pode ser visto como um tipo abstrato de dados (TAD). A definição abaixo confere este fato:

Definição 6.2.1 *[OJ85] : Um problema abstrato P, é um TAD, cuja linguagem têm os seguintes símbolos:*

- *D, para um sort de dados;*
- *R, para um sort de resultados;*
- *d, para o símbolo predicado unário sobre o sort D;*
- *r, para o símbolo predicado binário sobre o sort D;*
- *A, para o símbolo predicado unário, de segunda ordem sobre $D \longrightarrow R$.*

Podemos observar que, as realizações de um problema abstrato são os problemas concretos correspondentes e também que, a linguagem que descreve os sorts D e R, parte substantiva do problema, deve ser utilizada para descrever d, r, A, a parte intencional do problema. A especificação de d, r e A normalmente é feita através do enriquecimento dos TAD's D e R.

Existem três maneiras de se entender um TAD:

1. Por definições de sorts;
2. Por definições procedurais (por exemplo: programas);
3. Por definições não procedurais (por exemplo: fórmulas lógicas)

Exemplo 6.2.3.1 *[Alm90]:*

Especificação de um problema de ordenação de seqüências finitas de naturais.

Deve-se responder, informalmente, as seguintes perguntas:

- *Quais são os dados? [Seqüências de números naturais].*
- *Quais são os resultados? [Seqüências de números naturais].*
- *Qual a relação entre dados e resultados? [Dado uma seqüência de números naturais, o resultado é esta seqüência ordenada].*
- *Quais são os dados relevantes ao problema? [As seqüências finitas de números naturais].*
- *Quais são as formas admissíveis de associar dados a resultados? [As formas algorítmicas].*

Após responder as questões a respeito da estrutura do problema abstrato, inicia-se a especificação propriamente dita. Para isso, é necessário uma modelação para o problema.

Sejam $\mathcal{M}$ um modelo para P e $\varphi : P \longrightarrow \mathcal{M}$, tais que:

$$\varphi(D) = SEQ(N)$$
$$\varphi(R) = SEQ(N)$$

$$\varphi(d) = SEQ(N)$$
$$\varphi(r) = ord$$
$$\varphi(A) = Alg$$

Observe que, estaremos usando a mesma simbologia, tanto para a estrutura como para o seu conjunto base correspondente, para o caso da parte substantiva do problema (D, R) serem estruturas relacionais.

6.3 Especificação de Problemas do "Tipo Determine"

Na especificação de problema do "tipo determine" utilizaremos a *Skolemização* como processo de solução para chegarmos a uma função (função de Skolem) que dará o resultado do problema em questão. Porém, isso não acontece com todos os problemas da classe, depende da maneira de como é especificado o predicado de admissibilidade, pois, de acordo com o corolário 5.3.1 a função Skolem só será solução para o problema se ela for admissível.

Nesta seção será mostrado como é feita uma especificação de problemas "tipo determine" , para isto serão apresentados alguns exemplos e o desenvolvimento da especificação destes.

Exemplo 6.3.0.2 *[LB88]: Um Problema da Teoria dos Números*

Uma das questões distinguidas na Teoria dos Números são as equações diofantinas e um resultado particular destas questões é o seguinte:

***TEOR.** 1 [Sidki75]: Existem inteiros x_1 e x_2 tais que $x_1^2 - x_2^2 = n$ se e somente se $n = a.b$, para inteiros a e b da mesma paridade.*

Seja L_z a linguagem de primeira ordem sobre os inteiros, L_z=$\langle =, par, +, -, . \rangle$, cujos símbolos predicativos e funcionais tem o significado usual. Então o TEOR.1 pode ser expressado pela seguinte sentença desta linguagem:

$$\forall n(\exists x_1 \exists x_2 \mid x_1^2 + (-x_2^2) = n \Longleftrightarrow \exists a \exists b \mid a.b = n \wedge ((par(a) \wedge par(b)) \vee (ímpar(a) \wedge ímpar(b))) \quad (6.1)$$

OBS: A soma de dois números pares ou dois números ímpares é sempre um par, e a soma de dois números de paridades diferentes é sempre um ímpar, então pode-se abreviar a sentença acima como abaixo:

$$\forall n(\exists x_1 \exists x_2 \mid x_1^2 + (-x_2^2) = \Longleftrightarrow \exists a \exists b \mid a.b = n \wedge par(a+b)) \quad (6.2)$$

A partir daí vemos que o teorema sugere que se solucione o seguinte problema:

- *"Determinar as soluções inteiras da equação $x_1^2 - x_2^2 = n$, para um inteiro n na forma $n = a.b$, com a e b de mesma paridade".*

Assim, podemos descrever o problema pela seguinte sentença:

$$\underbrace{\forall n(\exists a \exists b \mid a.b = n \wedge par(a+b)}_{\alpha} \rightarrow \underbrace{\exists x_1 \exists x_2 \mid x_1^2 + (-x_2^2) = n)}_{\beta}$$

(6.3)

E então o problema fica reduzido ao problema de eliminar os quantificadores existenciais da sentença, o que em outras palavras, significa fazer a Skolemização da sentença.

Pela definição de problema esta situação pode ser associada a uma estrutura do tipo $P = \langle D, R, d, r, A \rangle$, cuja interpretação para este caso é dada por:

$D = Z$

$R = Z \times Z$

Onde a linguagem subjacente a D e a R são descritas nos diagramas abaixo, respectivamente:

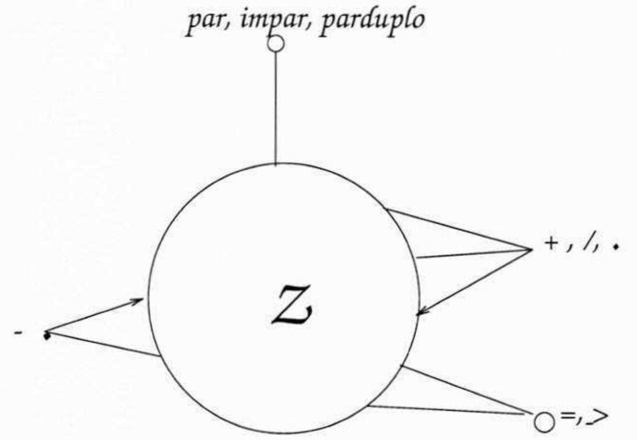

Figura 6.2: $L(D) = \langle \geq, par, ímpar, parduplo, =, -, +, / \rangle$

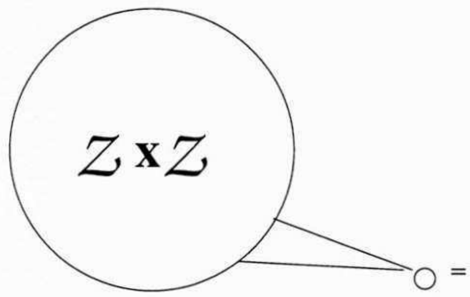

Figura 6.3: $L(R) = \langle = \rangle$

Supondo que dentro do contexto em que se encontra o problema, haja o desejo que as soluções sejam funções algorítmicas, onde ser algorítmica implica satisfazer a propriedade ***Alg*** *abaixo:*

$$Alg(a) \Longleftrightarrow \exists P \; (\forall x \; (x[P]y \longleftrightarrow y = a(x)))$$

onde:

$P=$ é uma fórmula bem formada de uma linguagem de programação.

$x[P]y=$ significa que para uma entrada x no programa P, este dá como saída o valor y.

Logo:

$L(P) = L(D) \cup L(R) \cup Alg$

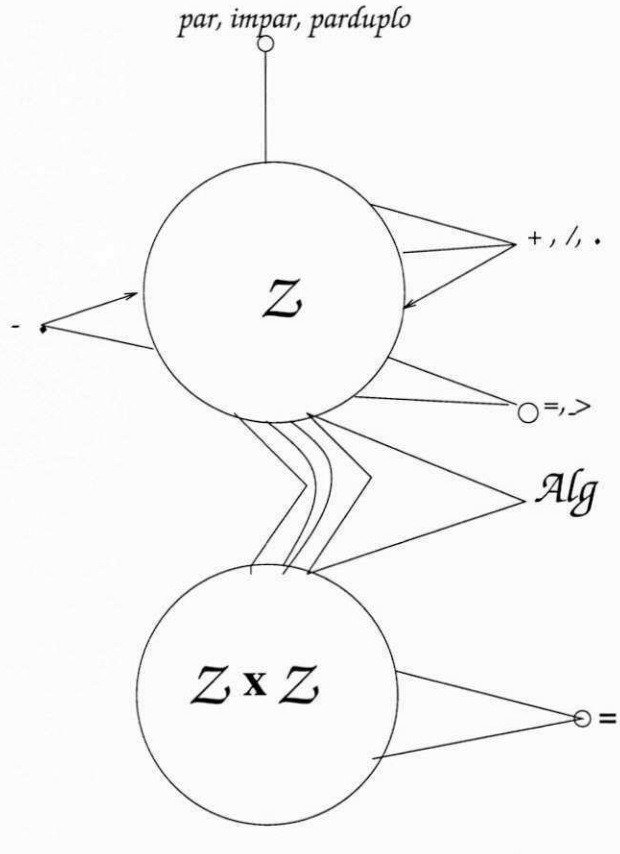

Figura 6.4: $L(P) = L(D) \cup L(R) \cup Alg$

Portanto, pode-se fazer a descrição de d, r e A de P, nesta linguagem.

$n \in d \iff n \geq 0 \wedge (ímpar(n) \vee (parduplo(n))$

$(n, (x_1, x_2)) \in r \longleftrightarrow \exists a \exists b (n = a.b \wedge par(a + b) \wedge x_1^2 + (-x_2^2) = a.b)$

$a \in A \iff Alg(a)$

Em $L(P)$ os símbolos funcionais tem o significado usual. Para fazer mais precisa esta primeira especificação devemos descrever cada símbolo predicativo através deles. Assim, temos:

$\geq$ *é um predicado binário com o significado tradicional.*

$parduplo(x) \Longleftrightarrow \exists y \mid x = 4.y$

$par(x) \Longleftrightarrow \exists y \mid x = 2.y$

$ímpar(x) \Longleftrightarrow \exists y \mid x = 2y + 1$

O problema então está suficientemente especificado, o que significa que sabemos exatamente o que queremos resolver.

Logo, podemos iniciar a etapa de solucionar o problema. O caminho sugerido aqui é verificar a possibilidade de quebrá-lo.

Como a sentença (6.3) é uma implicação, tem-se que β decorre de α, ou seja, se existem inteiros a e b que satisfazem α, então existem inteiros x_1 e x_2 que satisfazem β. O problema então pode ser visto como o encadeamento do problema P_α com o problema P_β.

onde:

P_α = *"Encontrar inteiros a e b que satisfazem α".*

P_β = *"Dados dois inteiros a e b, encontrar inteiros x_1 e x_2 que satisfazem β, onde $n = a.b$".*

Então, temos:

$$P = P_\beta \circ P_\alpha$$

onde P_α é um problema estruturado do seguinte modo:

$$P_\alpha = \langle D_\alpha, R_\alpha, d_\alpha, r_\alpha, A_\alpha \rangle$$

onde:

$D_\alpha = D$

$R_\alpha = Z \times Z$

$d_\alpha = d$

$(n, (a, b)) \in r_\alpha \iff par(a + b) \wedge n = a.b$

$A_\alpha = Her(A, R_\alpha^{D_\alpha})$

A definição de solução de problema diz que uma função admissível é uma solução, se ela satisfaz para todo elemento de d_α o predicado intencional r_α. Neste caso, os elementos de d_α são números inteiros positivos ímpares ou "parduplo". Para o primeiro caso existiria uma solução trivial, que é definir a e b como sendo 1 e n, respectivamente.

OBS: *Um número ímpar mais 1 é sempre um par, e 1.n=n.*

Portanto, satisfaz r_α.

Assim, tem-se a seguinte solução parcial:

$$a_1(n) = (1, n)\ \forall n \geq 0 \wedge ímpar(n)$$

Quando se tem o caso de ser inteiro positivo que seja "parduplo", pode ser dividido por 2 com a segurança de que sempre a soma de 2 com a metade de n dá sempre um número par. Assim, chega-se a seguinte solução parcial:

$$a_1(n) = (2, n/2)\ \forall n \geq 0 \wedge parduplo(n)$$

Como temos a solução para o caso de ímpares e de pares duplos, então podemos construir a solução de P_α. Logo,

Seja a função $a_1 : D_\alpha \longrightarrow R_\alpha$, definida abaixo:

$$a_1(n) = \begin{cases} (1, n), & se\ ímpar\ (n) \\ (2, n/2), & senão \end{cases}$$

A função a_1 *é total e está bem definida, pelo que é algorítmica e portanto* $a_1 \in A_\alpha$. *E como já mostrou-se que satisfaz* r_α *para todos os elementos de* d_α, *isto é,* $\forall n \in d_\alpha$, $(n, a_1(n)) \in r_\alpha$, *então* $a_1 \parallel\!- P_\alpha$.

Uma estruturação para o subproblema P_β, *que faz parte da decomposição em cadeia aplicada a* P, *é a seguinte:*

$$P_\beta = \langle D_\beta, R_\beta, d_\beta, r_\beta, A_\beta \rangle$$

onde:

$D_\beta = d_\beta = R_\alpha$

$R_\beta = R$

$((a,b),(x_1,x_2)) \in r_\beta \longleftrightarrow x_1^2 + (-x_2^2) = a.b$

$A_\beta = Her(A, R_\beta^{D_\beta})$

Seja $a_2 : D_\beta \longrightarrow R_\beta$, *tal que* $a_2(a,b) = (x_1, x_2)$.

- Se supomos $a_2 \parallel\!- P_\beta$, *então* $((a,b), a_2(a,b)) \in r_\beta$. *E* $((a,b), a_2(a,b)) \in r_\beta$ *se e somente se* $((a,b), a_2(x_1,x_2)) \in r_\beta$. *E* $((a,b), a_2(x_1,x_2)) \in r_\beta$ *se e somente se* $x_1^2 + (-x_2^2) = a.b$. *Se* $x_1^2 + (-x_2^2) = a.b$ *se e somente se* $(x_1 + x_2).(x_1 + (-x_2) = a.b$. *Se* $(x_1 + x_2).(x_1 + (-x_2) = a.b$, *podemos ter* $x_1 + x_2 = a$ *e* $x_1 + (-x_2) = b$. *E* $x_1 + x_2 = a$ *e* $x_1 + (-x_2) = b$ *se e somente se* $x_1 = a + b/2$ *e* $x_1 + x_2 = b$ *se e somente se* $x_2 = a + (-b)/2$.

Portanto, uma solução para P_β *é a função* $a_2 : D_\beta \longrightarrow R_\beta$, *definida como:*

$$a_2(a,b) = ((a+b)/2, (a + (-b))/2)$$

Observe que a_2 *é uma função total e bem definida, pelo que é uma função algorítmica logo pode-se dizer que* a_2 *é efetivamente uma solução para* P_β $(a_2 \parallel\!- P_\beta)$.

Com isso, temos que as funções a_1 *e* a_2 *são as respectivas soluções de* P_α *e* P_β, *e como* $P = P_\beta \circ P_\alpha$, *então o* ***TEOR.1*** *nos garante que a função* $a = a_2 \circ a_1$ *é uma solução para* P, *isto é,* $a = a_2 \circ a_1 \parallel\!- P = P_\beta \circ P_\alpha$.

Logo,

$$a(n) = \begin{cases} ((1+n)/2, (1+(-n))/2), & \textit{se ímpar } (n) \\ ((4+n)/4, (4+(-n))/4), & \textit{senão} \end{cases}$$

é uma solução para o problema dado.

Exemplo 6.3.0.3 *:*

Determinar a diagonal de um paralelepípedo retângulo do qual são conhecidos o comprimento, a largura e a altura, como mostra a figura abaixo:

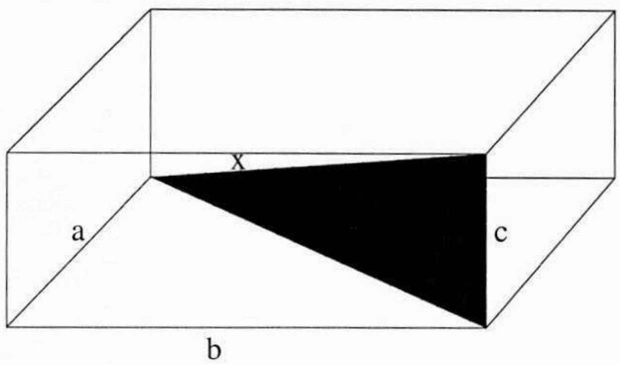

Figura 6.5: Visualização da diagonal do paralelepípedo

- Podemos expressar o problema pela seguinte sentença:

$$\underbrace{\exists y \mid \sqrt{a^2+b^2} = y)}_{\beta} \longrightarrow \underbrace{(\exists x \mid \sqrt{y^2+c^2} = x)}_{\gamma} \qquad (6.4)$$

E então o problema fica reduzido ao problema de eliminar os quantificadores existenciais da sentença, o que em outras palavras, significa fazer a Skolemização da sentença.

Uma estrutura para este problema é como segue:

$P = \langle D \times D, R, d' \times d', r, A_d \rangle$

onde:

D = é um conjunto de medidas em $\mathcal{R}$ (conj. dos reais) que tem como unidade o cm.

R = é um conjunto de medidas em $\mathcal{R}$ (conj. dos reais) que tem como unidade o cm.

- As linguagens subjacentes a D e a R são apresentadas nas figuras a seguir:

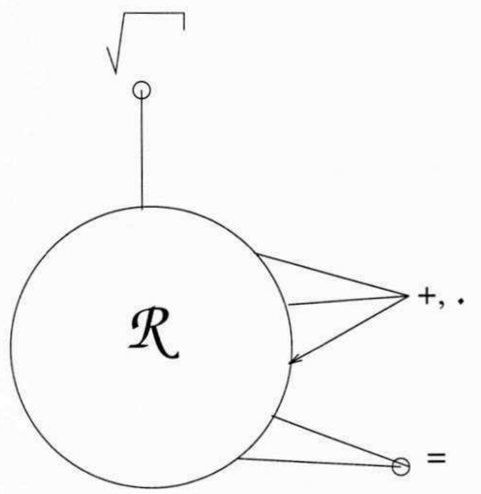

Figura 6.6: $L(D) = \langle =, +, ., \surd \rangle$

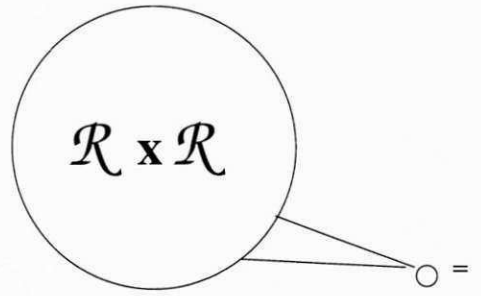

Figura 6.7: $L(R) = \langle = \rangle$

- Supondo que se deseja que as soluções sejam eventuais funções algorítmicas, sabendo-se que ser algorítmica implica em satisfazer a propriedade ***Alg****, descrito abaixo:*

$$Alg(a) \Longleftrightarrow \exists P\ (\forall z\ (z[P]w \longleftrightarrow w = a(z)))$$

onde:

P = *é uma fórmula bem formada de uma linguagem de programação.*

$z[P]w$ = *significa que para o conjunto de entradas z no programa P, este dá como saída o valor w.*

- Dessa forma, a linguagem subjacente ao problema P é:

$$L(P) = L(D) \cup L(R) \cup Alg$$

- Pode-se então fazer uma descrição de d', r, e A_d de P, nesta linguagem.

$d' = D$

$\forall(a, b, c) \in D\ \forall(y, x) \in R\ ((a, b, c) \in d \longrightarrow \exists y \in R\ \exists x \in R\ ((a, b, c), (y, x)) \in r$

$raizq \in A_d \Longleftrightarrow Alg(raizq)$

Nesse ponto chega-se a conclusão que o problema está suficientemente especificado. Inicia-se então a etapa de solução do problema.

Primeiramente, deve-se verificar a possibilidade de quebra do problema P em sub-problemas.

Na sentença 6.4 temos uma implicação, e isso sugere que γ decorre de β, ou seja, se existe um número real y que satisfaz a β, então existe um número real x que satisfaz a γ.

De acordo com o que foi mencionado acima, o problema pode ser visto como o encadeamento do problema P_β com o problema P_γ ($P = P_\gamma \circ P_\beta$).

onde:

P_β= *"Determinar a hipotenusa y do triângulo retângulo com lados a e b".*

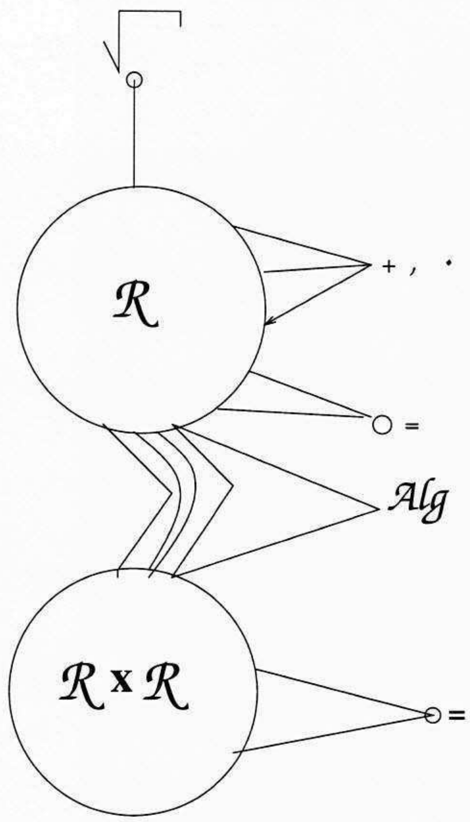

Figura 6.8: $L(P) = L(D) \cup L(R) \cup Alg$

P_γ= *"Determinar a hipotenusa x do triângulo com lados y e c".*

A estruturação para o subproblema P_β é feita do seguinte modo:

$$P_\beta = \langle D_\beta, R_\beta, d_\beta, r_\beta, Ad_\beta \rangle$$

onde:

$D_\beta = D = \mathcal{R} \times \mathcal{R}$

$R_\beta = \mathcal{R}$

$d'_\beta = d'$

$((a, b), y) \in r_\beta \iff \sqrt{a^2 + b^2} = y$

$Ad_\beta = Her(Ad, R_\beta^{D_\beta})$

- Seja a função $raiz_1 : D_\beta \longrightarrow R_\beta$*, definida como abaixo:*

$$raiz_1(a, b) = \sqrt{a^2 + b^2} = y$$

Pelo que $raiz_1$ *é algorítmica, então* $raiz_1 \in Ad_\beta$*. Logo,* $raiz_1$ *satisfaz* r_β*, isto é,* $((a, b), raiz_1(a, b)) \in r_\beta$*, então* $raiz_1 \mid\mid\!- P_\beta$ *(* $raiz_1$ *é solução para* P_β*).*

-Uma estruturação para o subproblema P_γ *é a seguinte:*

$$P_\gamma = \langle D_\gamma, R_\gamma, d'_\gamma, r_\gamma, Ad_\gamma \rangle$$

onde:

$D_\gamma = d'_\gamma$

$R_\gamma = R$

$((y, c), x) \in r_\gamma \Longleftrightarrow \sqrt{y^2 + c^2} = x$

$Ad_\gamma = Her(Ad, R_\gamma^{D_\gamma})$

Seja a função $raiz_2 : D_\gamma \longrightarrow R_\gamma$*, definida tal que* $raiz_2(y, c) = \sqrt{y^2 + c^2} = x$*. Se* $raiz_2$ *é algorítmica, é uma função total e bem definida, então* $raiz_2$ *satisfaz* r_γ*, isto é,* $((y, c), raiz_2(y, c)) \in r_\gamma$*. Logo,* $raiz_2$ *é uma solução para* P_γ *(*$raiz_2 \mid\mid\!- P_\gamma$*).*

Portanto, $raiz_1$ *e* $raiz_2$ *são soluções para* P_β *e* P_γ*, respectivamente, e como* $P = P_\gamma \circ P_\beta$ *, então o teorema 3.3.5 garante que a função* $raiz = raiz_2 \circ raiz_1$ *é solução para* P*, ou seja,* $raiz = raiz_2 \circ raiz_1 \mid\mid\!- P = P_\gamma \circ P_\beta$*.*

Logo,

$$raiz(a, b, c) = raiz_2(raiz_1(a, b, c))$$

é uma solução para o problema P*, a qual é garantida pelo mecanismo de construção das funções recursivas de acordo com [Lop81].*

Exemplo 6.3.0.4 *Determinar a velocidade v, conhecendo-se a força f, a massa m e a distância s. Considerando-se que a velocidade inicial (v_0) é igual a zero e a distância inicial (s_0) também é igual a zero.*

Este problema poderia ser descrito pela seguinte sentença:

$$\underbrace{\forall a(\exists f \exists m \mid f/m = a}_{\eta} \longrightarrow \underbrace{\exists v \mid S = v_0.t + 1/2a.t^2 \wedge v = a.t)}_{\theta} \tag{6.5}$$

-Assim, teremos que eliminar os quantificadores existenciais da sentença, ou seja, faremos a Skolemização.

-Primeiro veremos que o problema pode ser visualizado estruturalmente do seguinte modo:

$P = \langle F, V, f, vel, A \rangle$

- A interpretação para P será:

$F = \mathcal{R}$

$V = \mathcal{R} \times \mathcal{R}$

As linguagens subjacentes a F e V são descritas abaixo:

Figura 6.9: $L(F) = \langle =, +, ., | \rangle$

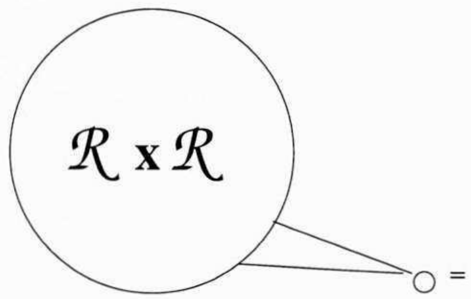

Figura 6.10: $L(V) = \langle = \rangle$

- Supondo que o problema está inserido num contexto onde se deseja que as suas soluções sejam funções algorítmicas, e para ser algorítmica implicará em satisfazer a propriedade ***Alg*** *abaixo:*

$$Alg(g) \Longleftrightarrow \exists P(\forall w(w[P]y \longleftrightarrow y = g(w))) \tag{6.6}$$

onde:

$P=$ *é uma fórmula bem formada.*

$w[P]y=$ *significa que para uma entrada w no programa P, este dá como saída o valor w.*

Logo, a linguagem subjacente para o problema P será:

$\mathcal{L}(P) = \mathcal{L}(F) \cup \mathcal{L}(V) \cup Alg$

- Assim, pode-se descrever as componentes f, vel e A nesta linguagem.

$f = \mathcal{R}$

$\forall (f,m,s) \in D\ \forall (a,v) \in R\ ((f,m,s) \in d \longrightarrow \exists a \in R\ \exists v \in R\ ((f,m,s),(a,v)) \in r$

$g \in A \Longleftrightarrow Alg(g)$

- Portanto, o problema está suficientemente especificado, o que significa que sabemos a que ponto queremos chegar.Ou seja, verificar se existe a possibilidade de quebrar o problema P.

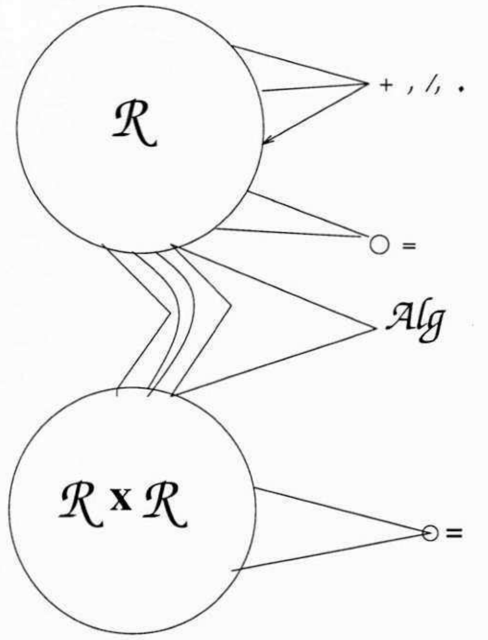

Figura 6.11: $\mathcal{L}(P) = \mathcal{L}(F) \cup \mathcal{L}(V) \cup Alg$

- Pelo fato da existência de uma implicação na sentença 6.5, isso sugere que θ decorre de η.

- O problema P então pode ser visto como o encadeamento do problema P_η com o problema P_θ.

onde:

P_η = *"Determinar a aceleração a, dados a força f e a massa m."*

P_θ = *"Determinar a velocidade v, a partir da solução de P_θ e com a distância s dada."*

- Assim temos que $P = P_\theta \circ P_\eta$.

- E P_η é estruturado da seguinte maneira:

$$P_\eta = \langle F_\eta, V_\eta, f_\eta, vel_\eta, A_\eta \rangle$$

onde:

$F_\eta = \mathcal{R}$ *(com unidades em N e kg)*

$V_\eta = \mathcal{R} \times \mathcal{R}$ *(com unidades em m/s^2)*

$f_\eta = f$ *(satisfazendo a equação $f = m.a$)*

$((f, m), a) \in vel_\eta \iff \exists a \ (a = f/m)$

$A_\eta = Her(A, V_\eta^{F_\eta})$

- Seja a função $g_1 : F_\eta \longrightarrow V_\eta$, definida como segue:

$$g_1(f, m) = a$$

g_1 sendo algorítmica, então $g_1 \in A_\eta$ e g_1 satisfaz vel_η, ou seja, $((f, m), g_1(f, m)) \in vel_\eta$. Logo, $g_1 \Vert\!- P_\eta$.

- Uma estruturação para o subproblema P_θ é dada abaixo:

$$P_\theta = \langle F_\theta, V_\theta, f_\theta, vel_\theta, A_\theta \rangle$$

onde:

$F_\theta = f_\theta = V_\eta$

$V_\theta = V$

$((a, s), v) \in vel_\theta \iff \exists v \ (S = v_0.t + 1/2a.t^2) \wedge (v = a.t)$

$A_\theta = Her(A, V_\theta^{F_\theta})$

-Seja a função $g_2 : F_\theta \longrightarrow V_\theta$, definida como segue:

$$g_2(a, s) = v$$

- Se supomos que $g_2 \Vert\!- P_\theta$, então $((a, s), g_2(a, s)) \in vel_\theta$ se e somente se $((a, s), v) \in vel_\theta$.

- Sendo g_2 uma função algorítmica, então $g_2 \in A_\theta$, e g_2 satisfaz a vel_θ. Logo, g_2 é efetivamente solução para P_θ ($g_2 \Vert\!- P_\theta$).

- Portanto, temos que as funções g_1 e g_2 são as respectivas soluções para P_η e P_θ, e como $P = P_\theta \circ P_\eta$, então a função $g = g_1 \circ g_2$ é uma solução para P, ou seja,

$g = g_1 \circ g_2 \; \| - \; P = P_\theta \circ P_\eta$

Logo,

$g(f, m, s) = g_2(g_1(f, m))$

é uma solução para o problema dado.

Exemplo 6.3.0.5 *:*

Determinar as raízes da equação do $2^{\underline{o}}$ *grau* $ax^2 + bx + c = 0$, *com* $a \neq 0$, *no domínio dos reais.*

Podemos descrever o problema pela seguinte sentença:

$\forall a \; \forall b \; \forall c \; \exists y \mid ay^2 + by + c = 0$

Para eliminarmos o quantificador existencial da sentença, isto é, fazermos a Skolemização da sentença, precisaremos de uma visualização estrutural do problema:

$P = \langle D, R, d, r, A \rangle$

onde:

$D = \{P^n[x] = (a_0, a_1, ..., a_n),\ a_i \in \mathcal{R},\ \wedge\ P^n[y] = a_0 y^n + a_1 y^{n-1} + ... + a_n y^0,\ a_i \in \mathcal{R}.\ a_0 \neq 0 \wedge x = y\}$

$R = \mathcal{R}$

Onde a linguagem subjacente a D e R são descritas nos diagramas 6.12 e 6.13, respectivamente.

- Supondo que o problema está inserido num contexto onde se deseja que as suas soluções sejam funções algorítmicas, e para ser algorítmica implicará em satisfazer a propriedade ***Alg*** *abaixo:*

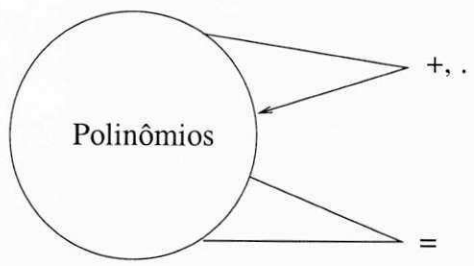

Figura 6.12: $L(D) = \langle =, +, . \rangle$

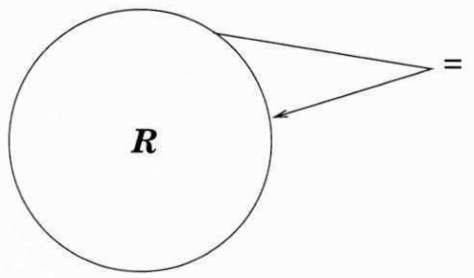

Figura 6.13: $L(R) = \langle = \rangle$

$$Alg(h) \iff \exists P(\forall y(y[P]z \longleftrightarrow z = h(y))) \tag{6.7}$$

onde:

$P=$ *é uma fórmula bem formada.*

$y[P]z=$ *significa que para uma entrada* y *no programa* P*, este dá como saída o valor* z.

Logo, a linguagem subjacente para o problema P *será:*

$\mathcal{L}(P) = \mathcal{L}(D) \cup \mathcal{L}(R) \cup Alg$

Portanto, pode-se fazer a descrição de r*,* d *e* A *de* P*, nesta linguagem.*

$d=$ $\{P^2[x] = (a_0, a_1, a_2)$ *e* $a_1^2 - 4a_0a_2 \geq 0$ $\}$

$(P^2[x], y) \in r \iff P^2(y) = 0$

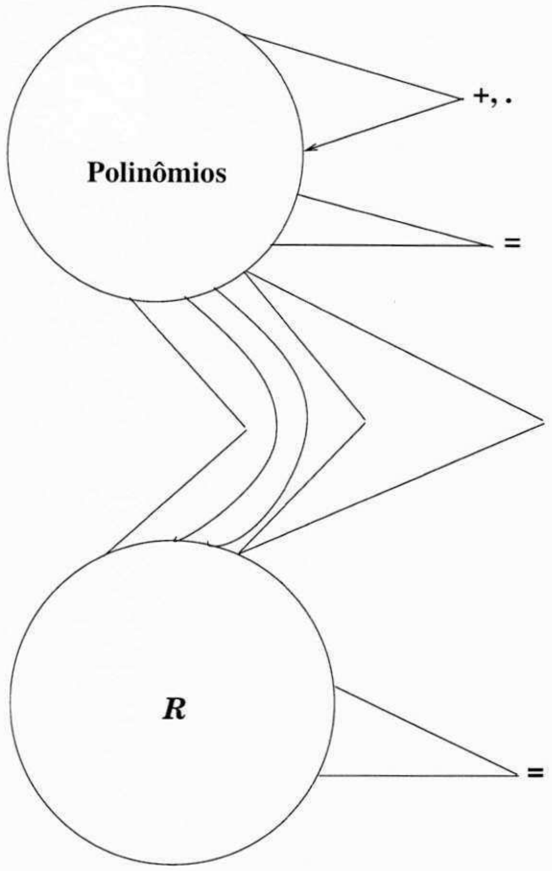

Figura 6.14: $\mathcal{L}(P) = \mathcal{L}(D) \cup \mathcal{L}(R) \cup Alg$

$A = \{f \mid f : D \longrightarrow R \wedge Adm(f)\}$

onde:

$Adm(f)$ = *propriedade de* f *ser algorítmica e ser descrita sem radicais.*

A partir da estruturação apresentada acima para o problema, vamos descrevê-lo formalmente nos moldes da definição 5.3.1. Para simplificar o processo de resolução do problema, usaremos a seguinte notação: $a_0 = a$, $a_1 = b$ *e* $a_2 = c$.

$(\forall a \in D\ \forall b \in D\ \forall c \in D\ \exists y \in R\ \varphi((a, b, c, y)) \longrightarrow (a, b, c, y) = 0$

A presença do quantificador existencial na sentença que define o problema como *sendo do "tipo determine", nos indica que para chegarmos ao resultado procurado precisaremos fazer a Skolemização da sentença. Neste caso, o processo de Skolemização será o caminho para chegarmos na solução do problema, uma vez que pelo corolário 5.3.1 temos que, se a função $Skl(\varphi)$ é admissível, então ela é solução para o problema.*

Nesse caso, o processo da Skolemização não servirá como solução para o problema, pois a função de Skolem encontrada no exemplo 5.3.1.1 para o polinômio dado possui radicais. Na Teoria de Galois [BM77] é provado que uma equação é solúvel por radicais se, e somente se, o grupo de automorfismos associado a ela é solúvel num sentido puramente da teoria dos grupos (os automorfismos em questão são aqueles que pertencem ao corpo de extensão gerado por todas as raízes da equação, que deixam fixados todos os coeficientes da equação). Ainda segundo Galois, existe uma equação de quinta ordem que não é solúvel por radicais. Então, podemos concluir que qualquer equação polinomial de grau $n \leq 4$ com coeficientes reais ou complexos só é solúvel por radicais.

Exemplo 6.3.0.6 : *Um Problema Prático de Engenharia*

Construção de uma barragem sobre um rio [Pol78].

Quais são os dados?

- *Dados topográficos relativos às bacias do rio e dos seus tributários;*
- *Dados geológicos essenciais aos estudos da solidez das fundações , da estanqueidade da barragem e dos materiais de contrução disponíveis;*
- *Dados climatológicos e hidrológicos referentes à precipitação anual e a altura das cheias;*
- *Dados econômicos relativos ao valor das terras que serão inundadas, dos custos dos materiais e da mão-de-obra, etc.*
- *Dados a respeito da investigação da localização da barragem:*

Extensão apropriada de Investigação; Reconhecimento; Investigações preliminares; Investigações finais; Escolha do local; Topografia - Mapeamento áereo; Método de projeção áerea; Equipamento múltiplo; Controle do solo para mapeamento múltiplo; Custos comparativos para mapeamento múltiplo; Utilização de mapas topográficos áereos; Mapas de localização; Investigações geológicas; Exploração superficial; Métodos de Exploração de superfície; Resistividade elétrica e Métodos Sísmicos de Exploração, etc.

- *A escolha do tipo de barragem:*

  Barragens de concreto de gravidade sólida; Barragens de concreto de gravidade profunda; Barragens de concreto arqueado; Dique (aterro); Barragens de madeira; Barragens de aço, etc.

- *Preparação e proteção da fundação:*

  Tratamento das fundações rochosas:

  Verificar a superfície final; Tratamento de defeitos; Vazamento através da fundação rochosa; Rebocar; Consolidar reboco; Drenagem da fundação; Proteção da base para a fundação rochosa.

  Tratamento de fundações de terra:

  Suportar resistência da fundação de terra; Suportar estacas; A rede de fluxo; Infiltração direta nas fundações de terra.

Qual é a incógnita?

- *A localização exata da barragem;*
- *As dimensões e forma geométrica;*
- *Os materiais a utilizar na construção, etc.*

Qual é a condicionante?

- *Para um empreendimento tão amplo é necessário atender a muitas condições econômicas e afetar o menos possível outras;*
- *A barragem deverá proporcionar energia elétrica;*
- *Fornecer água para irrigação e para abastecimento de certas localidades;*
- *Contribuir para o controle de inundações;*
- *Deverá causar o mínimo de prejuízos à navegação, a pesqueiros de importância econômica ou à beleza da paisagem;*
- *Deverá custar o mínimo possível e ser construída no prazo mais curto possível.*

Para resolver tal problema é necessário um certo conjunto de conhecimentos previamente adquiridos. O engenheiro tem a sua disposição uma série de conhecimentos altamente especializados, tais como: uma teoria da resistência dos materiais, a sua própria experiência profissional e a acumulada dos livros técnicos, deve ter conceitos precisos, qualitativos e científicos da pressão dos fluídos e das tensões e deformações que se desenvolvem nos corpos sólidos. Além disso, o engenheiro utiliza também o que ele sabe acerca da erosão causada pela água corrente, do transporte dos sedimentos, etc.

Tomando como base o exemplo acima, vemos que os conhecimentos necessários e os conceitos utilizados são mais complexos e menos nitidamente definidos nos problemas práticos do que nos problemas teóricos. Esta é a diferença mais importante. A motivação fundamental e os processos de solução parecem ser os mesmos para os dois tipos de problemas. Porém, existe a impressão de que os problemas práticos exigem uma maior experiência do solucionador.

Nos problemas práticos, temos sempre um grande número de dados e condicionantes. Devemos considerar tanto quanto for possível, mas muitas vezes somos forçados a desprezar alguns destes. No caso do projeto de uma grande barragem por exemplo, leva-se em consideração o interesse público e importantes interesses econômicos, porém coloca-se de lado pequenas pretensões e reclamações.

Utilizou todos os dados que poderiam contribuir consideravelmente para a solução?

Utilizou todas as condicionantes que poderiam influenciar muito na solução?

Estas são as principais indagações que devemos fazer quando tratamos com problemas práticos. Muitas vezes há um grande excesso de dados que não tem qualquer importância para a forma final da solução.

No caso de problemas práticos de engenharia, como no exemplo acima, o engenheiro não pode contar apenas com o bom senso, principalmente quando estiver tratando com um projeto novo e audacioso; ele tem que calcular a resistência da barragem projetada, prever quantitativamente as tensões e as deformações que se desenvolverão no seu interior. Para isso, ele tem que utilizar a teoria da elasticidade. Em tal utilização ele necessitará de muita matemática, com isso vemos então que um problema prático de engenharia conduz a um problema teórico.

Exemplo 6.3.0.7 : *Um Problema Prático*

Determine uma maneira de construir um edifício num tempo curto e com um custo mínimo. Tem-se os dados geológicos, topográficos, tipos de materiais disponíveis e custo destes.

Fazendo uma primeira análise do problema, temos a s seguintes questões:

Quais são os dados?

- *Dados geológicos relativos ao terreno;*
- *Dados topográficos relativos ao terreno;*

- *Dados a respeito da solidez da fundação;*
- *Dados sobre materiais de construção disponíveis;*
- *Dados a respeito da mão-de-obra;*
- *Dados a respeito do custo dos materiais.*

Qual é a incógnita?

- *Localização exata da construção do edifício;*
- *Suas dimensões;*
- *Forma geométrica (Arquitetônica);*
- *Tipos de materiais a serem utilizados na construção;*
- *Quantidade de material necessária para a construção;*
- *Quantidade de projetos necessários.*

Qual é a condicionante?

- *O edifício deverá proporcionar novas moradias;*
- *Deve ser construído num prazo mais curto possível;*
- *O custo deve ser mínimo;*
- *O material utilizado deve ser da melhor qualidade;*
- *Deve haver controle do expurgo do material;*
- *Deve obedecer as normas citadas pela prefeitura da cidade;*
- *Tem que obedecer os padrões de qualidade exigidos;*
- *A localização não deve ser prejudicial a outras construções.*

Ao analisarmos um problema prático vemos que este é uma abstração de um problema ocorrendo na prática. São diferentes dos problemas puramente matemáticos em diversos aspectos. Quando resolvemos um problema teórico, partimos de conceitos claros razoavelmente ordenados em nossa mente, mas ao resolvermos um problema prático, muitas vezes partimos de idéias obscuras, o que faz com que a clarificação dos conceitos se torne uma parte importante.

Capítulo 7

Esboço de uma Metodologia de Resolução para Problemas do "Tipo Determine"

7.1 Introdução

O conceito de classe de problemas é visto como sendo um conjunto de problemas com uma determinada característica comum a todos eles. Para a classe dos problemas do "tipo determine" a característica comum é a busca de resultados.

A questão "Qual é a classe desse problema?" leva a uma outra "O que pode ser feito dentro dessa classe de problemas?". Responder tais questões pode ser proveitoso, pois só assim distingüimos problemas de vários tipos, e isso pode sugerir o tipo de solução.

Na classe de problemas do "tipo determine" descrevemos o estudo da sua intencionalidade através da lógica de primeira ordem. Consideramos os quantificadores universais e existenciais, e vimos que podemos fazer um tratamento via lógica usando, como processo de solução, a *Skolemização.*

7.2 Apresentação do Esboço da Metodologia

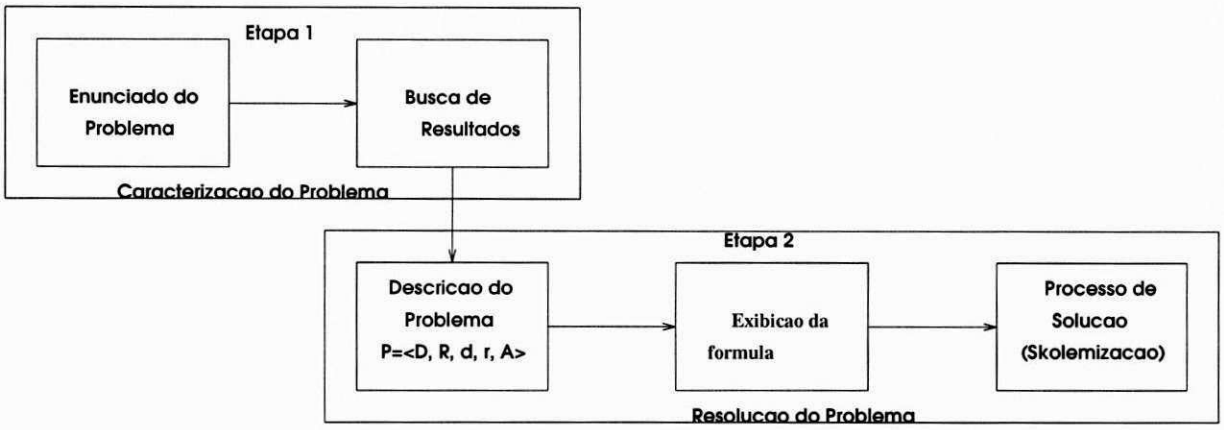

Figura 7.1: *Representação das etapas que envolvem o mecanismo de resolução de problema do "tipo determine"*

O método exposto neste trabalho consiste na análise da resolução dos problemas do "tipo determine" a partir das seguintes etapas:

7.2.1 Apresentação do Enunciado

Nesta etapa verificamos se o objetivo do problema é buscar resultados - característica principal dos problemas da classe "tipo determine". Para isso, primeiramente fazemos uma análise dos componentes estruturais do problema, respondendo as seguintes questões:

$$P = \langle D, R, d, r, A \rangle$$

1. Quais são os dados? (D)
2. Quais são os resultados? (R)
3. Quais as restrições sobre os dados? (d)
4. Qual a intenção do problema? (r)

5. Como são as soluções? (A)

De acordo com a estruturação apresentada para o problema devemos descrevê-la nos moldes da definição 5.3.1, e observar a existência de uma sentença da seguinte forma: $(\forall c \in D\ \exists x \in R\ \varphi(c,x)) \longrightarrow (c,x) \in r$. Se tal sentença existe, então foi possível fazer a exibição de φ, logo concluímos que realmente se trata de um problema do "tipo determine".

7.2.2 Verificação da Solução Adequada

Quando constatamos que realmente o problema em estudo é do "tipo determine", usamos então este fato para fazermos a argumentação que a *Skolemização* é um bom caminho para chegarmos até a solução do problema, uma vez que pelo corolário 5.3.1 temos que se a função Skolem é admissível então ela é solução para o problema.

Na sentença $(\forall c\ \exists x\ \varphi(c,x))$, c representa todos os dados fornecidos do problema, e x representa o resultado que deverá ser encontrado a partir dos dados c fornecidos, satisfazendo as condicionantes do problema. A presença do quantificador existencial na sentença acima, cujo escopo é x, é uma característica importante para os problemas do "tipo determine", pois indica que existe um resultado para o problema.

Sendo a questão colocada desta maneira, encontrar x significa eliminar o quantificador existencial, o que aponta para a necessidade de se skolemizar a sentença $(\forall c\ \exists x\ \varphi(c,x))$, isto é, x deverá ser substituído por uma função de c devidamente escolhida e que seja admissível, e a sentença $(\forall c\ \exists x\ \varphi(c,x))$ deverá continuar válida.

7.2.3 Especificação do Problema

Nesta etapa já temos uma descrição do problema através de uma sentença θ, na qual existe a presença dos quantificadores universais e existenciais e a fórmula φ. O problema então, fica reduzido à eliminação dos quantificadores existenciais da sentença θ. A especificação acontece obedecendo os seguintes passos:

1. Definição da parte substantiva do problema, isto é, dos conjuntos D e R da estrutura.

2. Definição da linguagem de dados $L(D)$ e de resultados $L(R)$ para o problema.

3. Definição da propriedade **Alg**, quando dentro do contexto em que se encontra o problema, haja o desejo que as soluções sejam funções algorítmicas, onde ser algorítmica implica satisfazer a propriedade **Alg** abaixo:

$$Alg(a) \Longleftrightarrow \exists P\ (\forall x\ (x[P]y \longleftrightarrow y = a(x)))$$

onde:

$P=$ é uma fórmula bem formada de uma linguagem de programação.

$x[P]y=$ significa que para uma entrada x no programa P, este dá como saída o valor y.

4. Definição da linguagem do problema $L(P)$:

$$L(P) = L(D) \cup L(R) \cup Alg$$

Em $L(P)$ os símbolos funcionais tem o significado usual. Para fazer mais precisa a especificação deve-se descrever cada símbolo predicativo através deles.

5. Definição da parte intencional do problema, d, r, A, na linguagem do problema $L(P)$.

Finalizadas as etapas acima, o problema deve está suficientemente especificado. Ou seja, sabe-se exatamente o que se quer resolver. Inicia-se então a etapa de solucionar o problema.

7.2.4 Processo de Solução - *Skolemização*

Para o estudo da especificação dos problemas do "tipo determine", precisamos verificar se há necessidade de usar o método de decomposição estrutural pois, muitas vezes nos deparamos com problemas de relativa complexidade. Assim, poderemos encarar o problema por etapas (subproblemas), solucionar uma primeira parte e com esse resultado solucionar a outra parte, até obter a solução do problema todo.

De acordo com [Bed87], na definição estrutural de um problema, são determinadas as condições sobre as quais é permitido decompor o problema e como se compor a solução deste problema, sendo conhecidas as soluções dos subproblemas.

A definição de solução de problema diz que uma função admissível é uma solução, se ela satisfaz para todo elemento de d o predicado intencional r. Pelo corolário 5.3.1 temos que uma função de Skolem é uma solução para o problema se ela é admissível. Assim, a eliminação do quantificador existencial da sentença $(\forall c\ \exists x\ \varphi(c,x))$ ocorre da seguinte forma:

> Seja $(\forall c\ \exists x\ \varphi(c,x))$ uma fórmula corrente θ. Cria-se uma nova fórmula substituindo a subfórmula de θ da forma $\exists x\ \varphi(c,x)$ por $Skl(\varphi)[x/f(c)]$, onde c é uma variável livre de $\exists x\ \varphi(c,x)$ e f é qualquer símbolo funcional n-ário que não ocorre em θ e $f \in R^D$.

Os novos símbolos funcionais introduzidos na fórmula são chamados de funções de Skolem e o processo de substituição é chamado de *Skolemização.* Porém, a função de Skolem nem sempre será a solução para o problema do "tipo determine". E em algumas situações a função pode não satisfazer ao predicado de admissibilidade, como foi o caso do problema do exemplo 6.3.0.5 , onde as funções admissíveis são algorítmicas e sem radicais, e as funções de Skolem encontradas no exemplo 5.3.1.1, para uma equação de segundo grau, são algorítmicas e possuem radicais.

Capítulo 8

Conclusão

Neste capítulo, fazemos algumas considerações finais sobre a dissertação, bem como as perspectivas futuras de dar continuidade ao que aqui foi feito.

Este trabalho foi construído com o objetivo de apresentar o desenvolvimento de uma metodologia de resolução para problemas da classe "tipo determine". Vimos a definição de um problema da classe, os passos para especificação de alguns exemplos desta e o esboço da metodologia para resolução de problemas do "tipo determine", utilizando, muitas vezes, a decomposição estrutural, a qual permite que se quebre um problema em vários subproblemas estruturalmente mais simples. Isto tornou mais nítida a idéia de ser a decomposição uma estratégia segura para uma abordagem de problemas de classes diversas.

A presente dissertação dá continuidade aos diversos trabalhos desenvolvidos por [Lop81], [Bed87], [LB88], procurando consolidar a Teoria Geral de Problemas como uma teoria que trate de problemas independente de um contexto específico e, al'em disso, formalize as relações entre estes e dê uma análise crítica das soluções.

Em [Bed87], foi feita uma análise sintática do predicado intencional, descrito numa forma normal disjuntiva para uma lógica proposicional. No entanto, quando fizemos o estudo da classe de problemas do "tipo determine", elevamos o estudo para definições numa linguagem de primeira ordem, incluindo os quantificadores universal e existencial.

Na especificação de problemas do "tipo determine", no estudo do predicado intencional, trabalhamos com a eliminação de quantificadores existenciais através da *Skolemização*, chegando a uma função solução para o problema, tornando mais simples o processo de resolução de um problema da classe. Assim, demos um passo fundamental no caminho do desenvolvimento de uma metodologia que permita resolver problemas de uma classe relativamente ampla.

Será objeto de estudo para trabalho futuro tentarmos especificar formalmente os problemas práticos presentes nesta classe, já que em tais problemas temos sempre um grande número de dados e condicionantes, e muitas vezes somos forçados a desprezar alguns desses dados e condicionantes, e isso acaba nos levando a uma incerteza da utilização de todos os dados e condicionantes que poderiam contribuir consideravelmente para a solução do problema. Além disso, nos problemas práticos as idéias de resolução são obscuras e é importante evidenciar os conceitos necessários para sua solução.

Uma questão fundamental de trabalho futuro é definir e caracterizar as subclasses que compõem a classe de problemas do "tipo determine", mostrando como se dá a identificação de φ em cada subclasse.

Um outro objeto importante de trabalho futuro é fazer uma comparação formal da classe de problemas "tipo determine" com diferentes classes de problemas. Isso nos daria condições de identificar as principais características presentes em cada classe comparada, assim poderíamos chegar a uma metodologia de resolução para classes diversas.

Com o presente trabalho contribuimos para o desenvolvimento da Teoria Geral de problemas [Lop81], permitindo um melhor entendimento da questão **problema** de torná-la uma teoria mais rica para análise e crítica de problemas e soluções.

Referências

[Alm90] Adriano P. de Almeida. Especificação e solubilidade de classes de problemas com paradigmas e de classes nomeáveis. Dissertação de mestrado, Depto. de Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1990.

[AP91] Jair Minoro Abe and Nelson Papavero. *Teoria Intuitiva dos Conjuntos*. Makron Books, McGraw-Hill, São Paulo (SP)-Brasil, 1991.

[Bed87] Benjamín R. C. Bedregal. Especificação de problemas solúveis por decomposição via análise da intencionalidade. Dissertação de mestrado, Depto. Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1987.

[BJ80] George S. Boolos and Richard C. Jeffrey. *Computability and Logic*. Cambrige University Press, England, 1980.

[BJ82] Dines Bjorner and Cliff B. Jones. *Formal Specification and Software Development*. Prentice-Hall Internacional, Inc., New Jersey, 1982.

[BL74] Walter S. Brainerd and Lawrence H. Landweber. *Theory of Computation*. J. Willey and Sons New York, New York, 1974.

[BL88] Benjamín R. C. Bedregal and Manoel A. Lopes. Decomposição de um problema a partir da análise sintática de seu predicado intencional. Relatório técnico, Depto. Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1988.

[BM77] Garrett Birkhoff and Saunders Maclane. *A Survey of Modern Algebra*. Macmillan Publishing Co., Inc., New York, 1977.

[Bun76] Mario Bunge. *Tratado de Filosofia Básica: Semântica*, volume 2. EPU-EDUSP, São Paulo (SP)-Brasil, 1976.

[Bun80] Mario Bunge. *Epstemologia: Curso de Atualização*. EPU-EDUSP, São Paulo (SP)-Brasil, 1980.

[Cas87] Marco A. et alii Casanova. *Programação em Lógica e a Linguagem Prolog*. Ed. Edgard Blücher Ltda,, São Paulo (SP)-Brasil, 1987.

[CK77] Chang C. C. and H. J. Keisler. *Model Theory*. North-Holland Pub. Co. New York, New York, 2a. edition, 1977.

[CL73] Chin-Liang Chang and R. C. T. Lee. *Symbolic Logic and Mechanical Theorem*. Academic Press. New York, New York, 1973.

[Coh65] P. M. Cohn. *Universal Algebra*. Harper and Row. New York, New York, 1965.

[End72] Hebert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press. New York, New York, 1972.

[Gil76] Arthur Gill. *Applied Algebra for the Computer Sciences*. Prentice-Hall, Inc. New Jersey, New Jersey, 1976.

[Gon79] Adilson Gonçalves. *Introdução à Álgebra*. Instituto de Matemática Aplicada, Rio de Janeiro (RJ)-Brasil, 1979.

[Hae87a] Armando M. et alii Haeberer. Hacia un metamodelo del processo de desarollo de software basado en teoria de problemas. Relatório Técnico 3/87, Depto. Informática/PUC-RIO, Rio de Janeiro (RJ)-Brasil, 1987.

[Hae87b] Armando M. et alli Haeberer. Sobre uma teoria algébrica de problemas y el desarollo de software. Relatório Técnico 2/87, Depto. Informática/PUC-RIO, Rio de Janeiro (RJ)-Brasil, 1987.

[Hau88] Edward H. Hauesler. *Intutionistic Type Theory and General Problem Theory: A Relationship*. Rio de Janeiro (RJ)-Brasil, 1988.

[Heg73] Leônidas Hegenberg. *Lógica: O Cálculo de Predicados.* Ed. Herder. Editora da USP, São Paulo (SP)-Brasil, 1973.

[Kle67] Stephen C. Kleene. *Mathematical Logic.* John Willey and Sons, Inc. New York, New York, 1967.

[LA85] Manoel A. Lopes and Benedito M. Acióly. Lógica de resolução de problemas: Uma introdução. Relatório técnico, Depto. de Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1985.

[Lak78] Imre Lakatos. *A Lógica do Descobrimento Matemático: Provas e Refutações.* Zahar Editores, Rio de Janeiro (RJ)-Brasil, 1978.

[LB88] Manoel A. Lopes and Benjamím R. C. Bedregal. Decomposição de problemas: Uma resolução completa para um problema da teoria dos números. Relatório técnico, Depto. de Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1988.

[LB0a] Manoel A. Lopes and Benjamín R. C. Bedregal. Problemas mentalmente solúveis: Uma extensão dos problemas church computáveis. Relatório técnico, Depto. Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1990a.

[Lop81] Manoel A. Lopes. *Introdução a uma Teoria Geral de Problemas.* Tese de doutoramento, Depto. Informática/PUC-Rio, Rio de Janeiro-Brasil, 1981.

[LV83] Manoel A. Lopes and Paulo A. S. Veloso. Transformação de problemas: Solubilidade via homomorfismo e analogia. Relatório técnico, Depto. Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1983.

[Men87] Elliott Mendelson. *Introductionto Mathematical Logic.* Wads-Worth and Books/Cole Advanced Books and Software, Monterey-Califórnia, 1987.

[Nil71] Nils J. Nilsson. *Problem Solving Methods in Artificial Inteligence.* McGraw-Hill, New York, 1971.

[OJ85] Wilson R. Oliveira Júnior. Especificação e características dos problemas solúveis por decomposição. Dissertação de mestrado, Depto. Informática/CCEN/UFPE, Recife (PE)-Brasil, 1985.

[Peq77] Tarcísio H. C. Pequeno. Lógica aplicada e verificação de programas. Dissertação de mestrado, Depto. Informática/PUC-RIO, Rio de Janeiro (RJ)-Brasil, 1977.

[Pol78] George Polya. *A Arte de Resolver Problemas: Um Novo Aspecto do Método Matemático*. Interciência, Rio de Janeiro, 22a. edition, 1978.

[Pol81] George Polya. *Mathematical Discovery: An Understanding Learning and Teaching*, volume 1 e 2. J. Willey and Sons New York, New York, 1981.

[RK93] Elaine Rich and Kevin Knight. *Inteligência Artificial*. Ed. Makron Books, São Paulo (SP)-Brasil, 2a. edition, 1993.

[Set83] j. S. A. Sette. Sobre a noção algébrica de implementação tipos de abstratos de dados. Relatório técnico, Depto. Informática/CCEN/UFPE, Recife (PE)-Brasil, 1983.

[Sil85] Ivan P. da Silva. Uma visão da teoria de banerji através da teoria geral de problemas. Disssertação de mestrado, Depto. Informática/CCEN/UFPE, Recife (PE)-Brasil, 1985.

[Sim84] G. L. Simons. *Introducing Artificial Inteligence*. NCC the National Centre for Information Technology, 1984.

[Sup72] Patrick Suppes. *Axiomatic Set Theory*. Dover Publications, Inc. New York, New York, 1972.

[Vel82] Paulo A. S. Veloso. Outilines of a mathematical theory of problems. Relatório técnico, Depto. Informática/PUC-RIO, Rio de Janeiro (RJ)-Brasil, 1982.

[Vel86] Paulo A. S. Veloso. *Verificação e Construção de Programas*. Ed. UNICAMP, São Paulo (SP)-Brasil, 1986.