

GUILHERME ALEX DERENIEVICZ

**OTIMIZAÇÃO DE SISTEMAS INTERVALARES NÃO
LINEARES ACÍCLICOS**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Fabiano Silva

CURITIBA

2014

GUILHERME ALEX DERENIEVICZ

**OTIMIZAÇÃO DE SISTEMAS INTERVALARES NÃO
LINEARES ACÍCLICOS**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Fabiano Silva

CURITIBA

2014

D431o Derenievicz, Guilherme Alex
Otimização de sistemas intervalares não lineares acíclicos / Guilherme
Alex Derenievicz. – Curitiba, 2014.
115f. : il. color. ; 30 cm.

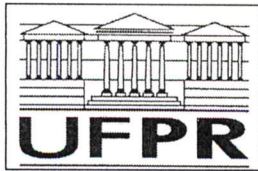
Dissertação (mestrado) - Universidade Federal do Paraná, Setor de
Ciências Exatas, Programa de Pós-graduação em Informática, 2014.

Orientador: Fabiano Silva.

Bibliografia: p. 107-110.

1. Análise de intervalos (Matemática). 2. Otimização matemática. 3.
Lógica simbólica e matemática. I. Universidade Federal do Paraná. II. Silva,
Fabiano. III. Título.

CDD: 004.0151134



Ministério da Educação
Universidade Federal do Paraná
Programa de Pós-Graduação em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, do aluno Guilherme Alex Derenievicz, avaliamos o trabalho intitulado, “*Otimização de sistemas intervalares não lineares acíclicos*”, cuja defesa foi realizada no dia 27 de fevereiro de 2014, às 13:30 horas, no Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela:

aprovação do candidato. () **reprovação** do candidato.

Curitiba, 27 de fevereiro de 2014.

Prof. Dr. Fabiano Silva
DINF/UFPR – Orientador

Prof. Dr. Luiz Carlos Matioli
UFPR – Membro Externo

Prof. Dr. André Luís Vignatti
DINF/UFPR – Membro Interno



AGRADECIMENTOS

Agradeço primeiramente a Deus, por nos permitir a busca pelo conhecimento acerca de um universo cheio de mistérios, paradoxos e incertezas.

A meu pai Aleixo e minha mãe Zuleide, pelo apoio, carinho e motivação incondicionais, sem os quais eu não teria chegado até aqui, e pelos valores que me mostraram ser importantes para a pessoa humana. Também agradeço ao meu irmão Gustavo, pelas vezes que ouviu a curiosidades e problemas envolvendo números e algoritmos, opinando sobre um ou outro parágrafo deste trabalho.

Ao meu amigo e orientador Fabiano Silva, pelas longas e incansáveis reuniões, nas quais compartilhou de forma ilimitada seu conhecimento e experiência; pela disponibilidade, paciência e confiança. Este trabalho foi possível somente graças à sua ótima orientação.

Aos professores Andre Luís Vignatti e Luiz Carlos Matioli, pela disponibilidade e contribuições para esta dissertação. Também agradeço ao professor Marcos Alexandre Castilho, pelo incentivo a uma abordagem teórica sobre o tema escolhido, e ao professor Alexandre Ibrahim Direne, pela grande amizade e partilha de conhecimentos durante todo o meu mestrado. Agradeço aos meus colegas do LIAMF, em especial ao Ricardo, pelas correções efetuadas neste trabalho.

Aos meus amigos e familiares que, de uma forma ou outra, contribuíram para que esse trabalho se tornasse realidade.

*“Donde, pois, vem a sabedoria, e onde está o lugar da inteligência?
Pois está encoberta aos olhos de todo o vivente, e oculta às aves do céu.
Deus entende o seu caminho, e Ele sabe o seu lugar.
Porque Ele vê as extremidades da terra; e vê tudo o que há debaixo dos céus.
Quando deu peso ao vento, e tomou a medida das águas;
Quando prescreveu leis para a chuva e caminho para o relâmpago dos trovões;
Então a viu e relatou; estabeleceu-a, e também a esquadrinhou”.*

(Jó 28:20-27)

SUMÁRIO

LISTA DE ALGORITMOS	iv
LISTA DE SÍMBOLOS E SIGLAS	v
RESUMO	viii
ABSTRACT	ix
1 INTRODUÇÃO	1
2 ÁLGEBRA INTERVALAR	5
2.1 Definições	6
2.2 Operações	9
2.3 Consistência Intervalar	17
2.4 Considerações	25
3 ÁLGEBRA BOOLEANA	27
3.1 Satisfatibilidade Booleana	28
3.2 Satisfatibilidade Módulo Teoria	33
3.3 Considerações	37
4 PROBLEMAS INTERVALARES	38
4.1 Satisfatibilidade Intervalar	42
4.2 Satisfatibilidade Módulo Álgebra Intervalar	48
4.3 Existência de Valoração Degenerada	59
4.4 Considerações	63
5 OTIMIZAÇÃO INTERVALAR	65
5.1 Representação por Grafos e Digrafos	68
5.2 Otimização Exata	77

	iii
5.3 Otimização Aproximada	88
5.4 Considerações	99
6 CONCLUSÃO	104
6.1 Trabalhos Futuros	105
BIBLIOGRAFIA	110
APÊNDICE A	111

LISTA DE ALGORITMOS

1	Davis-Putnam-Logemann-Loveland (DPLL)	32
2	SATII Ramificação e Poda (SATII _{RP})	45
3	Consistência Intervalar (<i>consistencia</i>)	46
4	iSAT	50
5	Encontra Valoração Degenerada (EVD)	75
6	Encontra Valoração Degenerada Exata (exatoEVD)	80
7	Encontra Valoração Degenerada - Híbrido (hibridEVD)	95

LISTA DE SÍMBOLOS E SIGLAS

\subseteq	contido
\subset	estritamente contido
\setminus	diferença de conjuntos
\emptyset	conjunto vazio
\emptyset	fórmula vazia
\diamond	valor indefinido de funções parciais
\equiv	equivalência
\Rightarrow	implicação lógica
\Leftrightarrow	dupla implicação lógica
$\forall x : \dots$	para todo x vale que ...
$\forall x \mid \dots : \dots$	para todo x tal que ..., vale que ...
$\exists x \mid \dots$	existe x tal que ...
$\exists!x \mid \dots$	existe um único x tal que ...
\mathbb{Z}^*	conjunto dos números inteiros excluindo o zero
\mathbb{R}	conjunto dos números reais
$\overline{\mathbb{R}}$	conjunto estendido dos números reais
\mathbb{I}	conjunto dos intervalos em \mathbb{R}
\mathbb{M}	conjunto dos multi-intervalos em \mathbb{R}
$\lfloor x \rfloor$	chão (ou piso) de x
\preceq	relação de otimalidade
\underline{x}	limitante inferior de um intervalo X
\overline{x}	limitante superior de um intervalo X
$\tau_e(X)$	função que define se o intervalo X é aberto ou fechado à esquerda
$\tau_d(X)$	função que define se o intervalo X é aberto ou fechado à direita
$c(X)$	comprimento do intervalo X

$\ X\ $	envoltória do multi-intervalo X
$\ R\ $	restrição intervalar na envoltória
$h(R)$	cabeça da restrição R
$t(R)$	cauda da restrição R
\circ	operador binário
\bullet	inverso de um operador binário
\odot	segundo inverso de um operador binário
\wedge	potenciação
Υ	radiciação
\wr	logaritmo
\oplus	segundo inverso do operador $-$
\otimes	segundo inverso do operador \div
$\hat{\wr}$	segundo inverso do operador Υ
$\hat{\Upsilon}$	segundo inverso do operador $\hat{\wr}$
$\hat{\wr}$	segundo inverso do operador \wr
\boxplus	conjunto dos operadores intervalares algébricos
\boxcup	conjunto de todos os operadores intervalares
\simeq	relação de consistência intervalar
$C(\psi)$	conjunto de cláusulas de ψ
$A(\psi)$	conjunto de átomos (restrições) de ψ
$V(\psi)$	conjunto de variáveis de ψ
$D(\psi)$	conjunto domínio de ψ
v	valoração
v^*	valoração ótima
σ	valoração degenerada
$\varepsilon_{\sigma}^{v_0}$	erro de valoração
ϵ	erro de aproximação
$\mathfrak{I}(\psi, v)$	interpretação da fórmula ψ sob a valoração v
$\mathfrak{I}(\alpha, v)$	interpretação do átomo α sob a valoração v

\mathfrak{c}	consistência intervalar
$\mathfrak{C}_c(R, \nu)$	função que aplica a consistência intervalar \mathfrak{c} em R
\mathfrak{T}	teoria
ω	cláusula
$\omega^{\mathfrak{T}}$	cláusula sobre \mathfrak{T}
φ	fórmula booleana
$\varphi_{\mathfrak{T}}$	fórmula SMT
Φ	fórmula intervalar
$\mathcal{G}(\Phi)$	grafo da fórmula Φ
$\mathcal{D}(\Phi)$	digrafo da fórmula Φ
$\mathcal{A}_{\Phi}(X)$	grafo induzido pelos vértices alcançáveis a partir de X
$\mathcal{R}(\Phi)$	digrafo restante da fórmula Φ
$P(\mathcal{G}(\Phi))$	conjunto de vértices de $\mathcal{G}(\Phi)$
$E(\mathcal{G}(\Phi))$	conjunto de arestas de $\mathcal{G}(\Phi)$
POI	Otimização Intervalar
SAT	Satisfatibilidade Booleana
SATII	Satisfatibilidade Intervalar
SMT	Satisfatibilidade Módulo Teoria
VSIDS	<i>Variable State Independent Decaying Sum</i>

RESUMO

Intervalos permitem uma representação aproximada de números reais, com a qual podemos modelar matematicamente problemas do mundo real de uma forma menos restritiva que a modelagem sobre restrições reais. Assim, podemos definir problemas intervalares de decisão e otimização que são relaxamentos da Programação Não Linear usual. Recentemente, técnicas utilizadas em algoritmos para o problema da Satisfatibilidade Booleana foram aplicadas na solução de problemas intervalares de decisão, utilizando a álgebra intervalar para refinar intervalos e obter soluções que satisfaçam um conjunto de restrições sob uma precisão preestabelecida. Embora essa abordagem não resolva problemas de otimização, ela apresenta um método para extrair uma solução real de uma solução intervalar, se o problema apresentar determinadas características. Neste trabalho, estendemos esse método, definindo uma classe de problemas para os quais é possível a extração de uma solução real mesmo sem a garantia de todas as condições exigidas pelos resolvedores anteriores. Além disso, mostramos que o método estendido pode ser utilizado para resolver algumas classes de problemas de otimização.

Palavras-chave: Álgebra Intervalar. Otimização Intervalar. Satisfatibilidade Módulo Teoria.

ABSTRACT

Intervals allow an approximate representation of real numbers. With them, one can model real word problems in a less restrictive way than using constraints over real numbers. Thus, one can define problems of decision and optimization over intervals that are relaxation of the usual Nonlinear Programming. Recently, techniques that are used to solve the Boolean Satisfiability problem have been implemented on algorithms for interval decision problems, applying interval arithmetics to refine intervals and obtain a solution that satisfy the set of constraints under a precision error. Although this approach does not apply for optimization problems, it introduces a method of extracting a real solution of an interval one, if the problem has some specific features. In this work we extend this method and define a class of problems for which it is possible to extract a real solution even without ensuring all conditions required by the preview solvers. Furthermore, we show that the extended method can also be applied to some set of optimization problems.

Key-words: Interval arithmetics. Interval Optimization. Satisfiability Modulo Theories.

CAPÍTULO 1

INTRODUÇÃO

*“Estamos andando em lendas ou
sobre a terra verde à luz do dia?”*

(J.R.R. Tolkien, As Duas Torres, Cap. 2)

Vários problemas reais, em áreas como roteamento, bioinformática, verificação de *hardware* e *software*, planejamento e logística, podem ser modelados como problemas de satisfação de restrições, nos quais queremos encontrar uma atribuição de valores às variáveis do problema de forma a satisfazer determinadas restrições.

Essa modelagem envolve, dessa forma, a definição de variáveis e constantes que representam valores observados no mundo real. Embora várias dessas variáveis tenham uma correspondência direta com valores reais, como, por exemplo, a quantidade de determinado elemento, muitas vezes é necessário a representação de valores irracionais, como $\sqrt{2}$ ou π .

A representação computacional de valores irracionais é inviável. Mesmo assim, uma aproximação de tais valores pode ser suficiente para o problema que está sendo modelado. Por exemplo, ao se utilizar uma aproximação π' de π com uma precisão de 39 casas decimais é possível calcular a circunferência do universo observável com a precisão da largura de um átomo de hidrogênio¹.

A fim de uma representação mais adequada de tais aproximações, definimos limitantes superior e inferior que nos oferecem uma certa informação sobre o valor real do número representado. No caso de π , podemos representá-lo por

¹*Pi and the size of the Universe - Numberphile.* https://www.youtube.com/watch?v=FpyrF_Ci2TQ.

$$3.14159265358979 \leq \pi \leq 3.14159265358980$$

E, equivalentemente, denotamos que

$$\pi \in [3.14159265358979, 3.14159265358980]$$

Assim, chamamos $\Pi = [3.14159265358979, 3.14159265358980]$ de **intervalo**, isto é, um conjunto de números reais que contém um valor que se deseja representar. Ou ainda, dizemos que Π representa o número π com um determinado erro de precisão. Neste trabalho, vamos utilizar “.” (ponto) ao invés de “,” (vírgula) para representar números racionais, evitando ambiguidade em relação à notação dos intervalos.

No processo de modelagem matemática de qualquer problema real, é comum existir constantes irracionais ou com várias casas decimais, geralmente aproximadas por números de máquina que as representam com o mínimo de perda de informação [30]. Alternativamente, a modelagem intervalar representa tais constantes por intervalos. É importante observar que a partir do momento em que modelamos o problema e tentamos resolvê-lo matematicamente, perdemos contato com o problema real. Assim, não temos mais a informação de que Π representa π , mas algum valor entre 3.14159265358979 e 3.14159265358980. Nesse sentido, Π também é uma aproximação de π , embora mais dinâmica que a aproximação π' .

Nessa abordagem, a solução de um problema de satisfação de restrições não é mais uma atribuição de valores únicos às variáveis definidas, mas de um conjunto de valores reais, ou intervalos, que representam números reais específicos. Equivalentemente, uma solução é um refinamento do conjunto domínio das variáveis do problema, resultando em subconjuntos que podem conter a solução real.

Essa modelagem intervalar é, portanto, um relaxamento da modelagem matemática sobre os números reais [2]. Nesse sentido, estamos em um cenário mais otimista, “sobre a terra verde à luz do dia”. Mesmo assim, o problema de encontrar uma atribuição de intervalos às variáveis de um conjunto de restrições é \mathcal{NP} -difícil no caso geral [2],

tornando a existência de um algoritmo eficiente para todas as instâncias desse problema pouco provável.

Um algoritmo para resolver esse problema foi proposto em 1997, aplicando uma estratégia de ramificação e poda sobre variáveis intervalares [32]. Mais recentemente, o resolvidor iSAT [17] aplicou técnicas utilizadas em algoritmos para o problema da Satisfatibilidade Booleana também no cenário intervalar, obtendo resultados superiores aos demais resolvidores da época.

Neste trabalho, vamos formalizar a álgebra intervalar e alguns problemas de satisfação de restrições sobre intervalos. Além disso, vamos apresentar os principais algoritmos para resolver os problemas definidos e mostrar como podemos utilizar suas propriedades para descrever métodos para outros problemas intervalares.

O objetivo deste trabalho é definir uma classe de problemas intervalares para os quais é possível efetuar uma atribuição de valores reais às variáveis do problema dentro do domínio restringido pela solução da modelagem intervalar. Isto é, queremos mostrar que é possível, para alguns problemas intervalares, extrair uma solução nos números reais de uma solução intervalar. Além disso, vamos apresentar um método que efetua essa extração de maneira eficiente.

Adicionalmente, existem problemas reais para os quais a modelagem de restrições não é suficiente para representar todas as suas características. Mais especificamente, em problemas de otimização, não queremos como solução uma atribuição de valores que apenas satisfaça todas as restrições, mas que também seja ótima sob algum critério. Usualmente, problemas com essas características são modelados como problemas de Programação Não Linear, nos quais temos, além do conjunto de restrições, uma função objetivo a ser minimizada.

Da mesma forma, podemos definir uma versão intervalar desse problema, a Otimização Intervalar Não Linear, cuja solução é um conjunto de intervalos que satisfazem as restrições minimizando uma função objetivo intervalar. Neste trabalho também vamos apresentar formalmente esse problema e mostrar que os métodos propostos para a extração de solução real podem ser utilizados de forma direta para resolver problemas de otimização.

A estrutura deste texto é dividida em capítulos que abordam conceitos e problemas distintos e complementares. No capítulo 2, apresentamos a álgebra intervalar e suas principais definições e operações, necessárias para a definição de problemas intervalares e suas soluções. Também apresentamos o conceito de consistência intervalar, imprescindível para a eficiência e corretude dos algoritmos posteriores.

No capítulo 3, mudamos de contexto e apresentamos as principais definições da álgebra booleana e do problema da Satisfatibilidade Booleana. Embora esse problema não tenha relação direta com a álgebra intervalar, no capítulo seguinte vamos utilizar alguns conceitos definidos no contexto booleano sobre os intervalos e mostrar que o principal algoritmo para resolver problemas intervalares é muito semelhante ao algoritmo base dos resolvidores booleanos.

Ainda no capítulo 4, apresentamos três problemas intervalares e suas respectivas soluções, sendo um deles a respeito da existência de solução real dentro do domínio restringido pela modelagem intervalar. Vamos focar principalmente no algoritmo iSAT, proposto em [17], que soluciona problemas de satisfação de restrições intervalares e define uma classe de problemas para os quais, sob determinadas condições, é possível extrair solução nos reais.

No capítulo 5, definimos o problema da Otimização Intervalar e estendemos a classe de problemas apresentadas em [17] de forma a excluir algumas das condições necessárias para a existência de solução real, tornando sua aplicação mais realista. Também apresentamos um método eficiente para encontrar tal solução minimizando uma dada função objetivo, resolvendo também problemas de otimização.

Por fim, no capítulo 6, temos a conclusão e a proposta de trabalhos futuros relacionados ao tema.

CAPÍTULO 2

ÁLGEBRA INTERVALAR

*“Vasculhou com as pontas dos dedos mais uma vez
dentro da toalha, mas não havia mais letras.*

– Quer dizer que é isso? – disse Ford.

– É isso.

– Seis vezes nove. Quarenta e dois.

– É isso. Está tudo aí.”

(Douglas Adams, O Restaurante no Fim do Universo, Cap. 33)

Intervalos podem ser interpretados como conjuntos de valores possíveis à uma variável real. Nesse sentido, eles representam possibilidades ou aproximações. Para o primeiro caso, suponha, por exemplo, que queiramos representar o tempo de execução t de um processo qualquer. Embora saibamos os limites mínimo 0 e máximo *timeout* do tempo de execução, não existe um valor real para essa informação, mas um conjunto infinito de possibilidades, dependendo de cada execução. Como o tempo é contínuo, podemos representá-lo pela faixa de valores reais $0 \leq t \leq \textit{timeout}$, isto é, pelo intervalo $T = [0, \textit{timeout}]$.

Para o segundo caso, temos o exemplo do intervalo Π , apresentado na introdução, que representa o número transcendental π . Como $\pi \in \Pi$, então existe um valor em Π que representa exatamente π , mas todos os outros valores são aproximações.

Como apontado na introdução deste trabalho, quando estamos trabalhando com uma modelagem matemática, não temos acesso a valores reais representados pelos intervalos. O número π é um ótimo exemplo dessa limitação; embora possamos representá-lo por intervalos com precisão de trilhões de casas decimais, não sabemos qual o seu valor exato. Assim, o segundo caso de interpretação de um intervalo se reduz ao primeiro: um conjunto

de aproximações possíveis à uma variável real.

Dessa forma, todas as operações e relações sobre intervalos devem considerar a combinação de todas as aproximações contidas nos intervalos de interesse, sem excluir possibilidades de aproximação permitidas. Neste capítulo, vamos apresentar algumas definições sobre intervalos e formalizar suas operações sobre essa abordagem combinatória. Por fim, vamos apresentar o conceito de **consistência intervalar**, imprescindível para a eficiência e correte de métodos intervalares.

2.1 Definições

Sobre dois conjuntos A e B quaisquer, utilizamos a seguinte notação:

- $A \subseteq B$ se, e somente se, $\forall x \in A : x \in B$;
- $A \subset B$ se, e somente se, $A \neq B$ e $\forall x \in A : x \in B$;
- $A \setminus B = \{x \in A \mid x \notin B\}$.

As demais operações e relações sobre conjuntos são definidas da forma usual [33].

Definição 2.1 *Vamos denotar por $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ o conjunto estendido dos números reais, tal que $-\infty < x < \infty$, para todo $x \in \mathbb{R}$.*

Vamos utilizar as operações sobre $\overline{\mathbb{R}}$ usuais [7], exceto pela indeterminante $0 \cdot \pm\infty$, que vamos definir como $0 \cdot \pm\infty = 0$, comumente utilizado em probabilidade e teoria da medida.

Definição 2.2 *Um **intervalo fechado** $X = [\underline{x}, \overline{x}]$ é o conjunto de números reais $X = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\}$, onde $\underline{x}, \overline{x} \in \overline{\mathbb{R}}$ são chamados de limitantes.*

Para fins de notação, os intervalos serão representados por letras maiúsculas e seus limitantes inferior e superior pela respectiva letra em minúsculo, sublinhada e sobrelinhada.

Definição 2.3 *Um **intervalo aberto** $X = (\underline{x}, \overline{x})$ é o conjunto de números reais $X = \{x \in \mathbb{R} \mid \underline{x} < x < \overline{x}\}$, onde $\underline{x}, \overline{x} \in \overline{\mathbb{R}}$.*

É direto que $[-\infty, \infty] = (-\infty, \infty) = \mathbb{R}$.

Definição 2.4 Um *intervalo fechado à direita* $X = (\underline{x}, \bar{x}]$ é o conjunto de números reais $X = \{x \in \mathbb{R} \mid \underline{x} < x \leq \bar{x}\}$, onde $\underline{x}, \bar{x} \in \overline{\mathbb{R}}$.

Definição 2.5 Um *intervalo fechado à esquerda* $X = [\underline{x}, \bar{x})$ é o conjunto de números reais $X = \{x \in \mathbb{R} \mid \underline{x} \leq x < \bar{x}\}$, onde $\underline{x}, \bar{x} \in \overline{\mathbb{R}}$.

Definição 2.6 Dizemos que um intervalo X é *semi-aberto*, ou *semi-fechado*, se X é fechado à esquerda ou fechado à direita.

Definição 2.7 Denotamos por $\mathbb{I} = \{[\underline{x}, \bar{x}], (\underline{x}, \bar{x}], [\underline{x}, \bar{x}), (\underline{x}, \bar{x}) \mid \underline{x}, \bar{x} \in \overline{\mathbb{R}}\}$ o conjunto de todos os intervalos em \mathbb{R} .

Definição 2.8 Uma *variável intervalar*, ou *intervalo*, $X \in \mathbb{I}$ é um intervalo fechado, aberto ou semi-aberto.

É importante observar que, se $\underline{x} = \infty$ ou $\bar{x} = -\infty$ ou $\bar{x} < \underline{x}$, para algum intervalo $X \in \mathbb{I}$, então $X = \emptyset$.

Definição 2.9 As funções $\tau_e : \mathbb{I} \mapsto \{0, 1\}$ e $\tau_d : \mathbb{I} \mapsto \{0, 1\}$ associam dois valores $\tau_e(X)$ e $\tau_d(X)$ à um intervalo $X \in \mathbb{I}$ de forma que

$$\tau_e(X) = \begin{cases} 0 & \text{se } X \text{ é um intervalo aberto ou fechado à direita} \\ 1 & \text{caso contrário} \end{cases}$$

$$\tau_d(X) = \begin{cases} 0 & \text{se } X \text{ é um intervalo aberto ou fechado à esquerda} \\ 1 & \text{caso contrário} \end{cases}$$

Dessa forma, sabemos se um intervalo X é aberto, fechado ou semi-aberto apenas observando os valores de $\tau_e(X)$ e $\tau_d(X)$. A saber, se $\tau_e(X) + \tau_d(X) = 2$, então X é um intervalo fechado; se $\tau_e(X) + \tau_d(X) = 1$, então X é um intervalo semi-aberto e, finalmente, se $\tau_e(X) + \tau_d(X) = 0$, então X é um intervalo aberto. No caso de X ser um intervalo semi-aberto, se $\tau_e(X) = 1$, então X é fechado à esquerda; do contrário, X é fechado à direita.

Definição 2.10 O *comprimento* de um intervalo $X \in \mathbb{I}$ é dado por $c(X) = \bar{x} - \underline{x}$.

Representações computacionais ou medidas tomadas de processos naturais não são contínuas, impossibilitando uma representação exata dos números reais. Nesse caso, os intervalos permitem uma representação mais adequada, cuja precisão é dada pelo comprimento de cada intervalo. Assim, podemos representar $\sqrt{2}$ pelo intervalo $X = [1.4, 1.5]$, onde $c(X) = 0.1$, ou pelo intervalo mais aproximado $Y = [1.414, 1.415]$, onde $c(Y) = 0.001$.

Na literatura existem diversas definições sobre a comparação entre intervalos. Em [28] temos uma análise detalhada das definições mais comuns e suas possíveis aplicações. No entanto, para dois intervalos X e Y quaisquer, as relações $X = Y$, $X \neq Y$, $X \subset Y$, $X \subseteq Y$ e as operações $X \cap Y$, $X \cup Y$, $X \setminus Y$ são definidas como as respectivas relações e operações sobre conjuntos. Além disso, como intervalos são um conjunto contínuo de valores reais, podemos verificar quaisquer das relações acima apenas observando os limitantes dos intervalos. Por exemplo, $X = Y$ se, e somente se, $\underline{x} = \underline{y}$, $\bar{x} = \bar{y}$, $\tau_e(X) = \tau_e(Y)$ e $\tau_d(X) = \tau_d(Y)$.

Definição 2.11 O intervalo X é um *refinamento* do intervalo Y se $X \subseteq Y$. Nesse caso, também dizemos que Y é uma *expansão* de X .

Definição 2.12 Um intervalo X é dito *degenerado* se $X = [x, x]$.

Um intervalo degenerado é um conjunto de tamanho unitário que representa um número real com erro nulo, isto é, o único elemento do intervalo é o próprio número real representado. Na literatura, intervalos degenerados e números reais são tratados de forma equivalente ($[x, x] = x$), pois mesmo que um intervalo degenerado ainda seja um conjunto, ele representa um número real exato.

A seguir, vamos definir algumas operações intervalares. Como $[x, x] \circ [y, y] = [x \circ y, x \circ y]$, para toda operação \circ que $x \circ y$ esteja definida, então \circ pode ser considerada uma extensão da respectiva operação sobre os reais.

2.2 Operações

Em [24], as operações básicas $+$, $-$, \times , \div aplicadas sobre dois intervalos X e Y não vazios são definidas como funções que geram um intervalo que contém o resultado da operação aplicada à todo par de números tomados em X e Y . Isto é,

$$X \circ Y = \{x \circ y \mid x \in X \text{ e } y \in Y\}, \text{ para qualquer } \circ \in \{+, -, \times, \div\}$$

Novamente, podemos definir essas quatro operações apenas calculando seus limitantes. Sejam $X = [\underline{x}, \bar{x}]$ e $Y = [\underline{y}, \bar{y}]$ duas variáveis intervalares tais que $X \neq \emptyset$ e $Y \neq \emptyset$:

- **Adição:**

$$X + Y = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$$

A adição é comutativa e associativa e o intervalo $[0, 0]$ é o elemento neutro da adição.

Por exemplo, sejam $X = [-1, 3]$, $Y = [2, \infty]$ e $Z = [0, 0]$. Pela definição acima temos que $X + Y + Z = X + Y = [1, \infty]$.

- **Subtração:**

$$X - Y = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$$

A subtração é associativa e também podemos escrevê-la definindo a operação unária

$$-Y = [-\bar{y}, -\underline{y}], \text{ de forma que } X - Y = X + (-Y) = [\underline{x}, \bar{x}] + [-\bar{y}, -\underline{y}] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}].$$

É importante observar que $X - X = [\underline{x} - \bar{x}, \bar{x} - \underline{x}]$, e portanto, $X - X = [0, 0]$ se, e somente se, $c(X) = 0$. Do contrário, é direto que $[0, 0] \subset X - X$.

Por exemplo, sejam $X = [0, \infty]$ e $Y = [0, \infty]$. Pela definição da subtração, $X - Y = [-\infty, \infty]$.

- **Multiplicação:**

$$X \times Y = [\min\{\underline{x} \cdot \underline{y}, \underline{x} \cdot \bar{y}, \bar{x} \cdot \underline{y}, \bar{x} \cdot \bar{y}\}, \max\{\underline{x} \cdot \underline{y}, \underline{x} \cdot \bar{y}, \bar{x} \cdot \underline{y}, \bar{x} \cdot \bar{y}\}]$$

A multiplicação é comutativa, associativa e subdistributiva em relação à adição e subtração, isto é, $X \times (Y \pm Z) \subset X \times Y \pm X \times Z$. O intervalo $[1, 1]$ é o elemento neutro da multiplicação.

Vale lembrar que definimos $0 \cdot \pm\infty = 0$. Assim, para $X = [-\infty, \infty]$ e $Y = [0, 0]$, obtemos $X \times Y = [0, 0]$, o que convém com a intuição de que $\forall x \in X, x \cdot 0 = 0$.

- **Divisão** (denominador não contém zero):

Se $\underline{y} > 0$ ou $\bar{y} < 0$, definimos

$$X \div Y = \begin{cases} [\underline{x}, \bar{x}] \times [\frac{1}{\bar{y}}, 0) & \text{se } \underline{y} = -\infty \\ [\underline{x}, \bar{x}] \times (0, \frac{1}{\underline{y}}] & \text{se } \bar{y} = \infty \\ [\underline{x}, \bar{x}] \times [\frac{1}{\bar{y}}, \frac{1}{\underline{y}}] & \text{caso contrário} \end{cases}$$

Por exemplo, sejam $X = [-\infty, \infty]$ e $Y = [-\infty, -1]$.

$$\begin{aligned} X \div Y &= [-\infty, \infty] \times [\frac{1}{-1}, 0) = [-\infty, \infty] \times [-1, 0) \\ &= [\min\{\infty, 0, -\infty, 0\}, \max\{\infty, 0, -\infty, 0\}] \\ &= [-\infty, \infty] \end{aligned}$$

Como $(\min\{\frac{\underline{x}}{\underline{x}}, \frac{\bar{x}}{\underline{x}}\}, \max\{\frac{\underline{x}}{\bar{x}}, \frac{\bar{x}}{\bar{x}}\}) \subseteq X \div X$, então $X \div X = [1, 1]$ se, e somente se, $c(X) = 0$. Do contrário, é direto que $[1, 1] \subset X \div X$.

- **Potenciação:**

Vamos definir a potenciação de forma que ela satisfaça a mesma propriedade das operações anteriores, isto é, $X^Y = \{x^y \mid x \in X \text{ e } y \in Y\}$. No entanto, estamos interessado apenas quando o expoente é um intervalo degenerado inteiro. Portanto, vamos defini-la para $Y = [y, y]$ tal que $y \in \mathbb{Z}^*$.

$$X \wedge Y = X^Y = \begin{cases} [1, 1] \div (X \wedge -Y) & \text{se } y < 0 \\ [\underline{x}^y, \bar{x}^y] & \text{se } y > 0 \text{ é ímpar} \\ [\underline{x}^y, \bar{x}^y] & \text{se } y > 0 \text{ é par e } \underline{x} \geq 0 \\ [\bar{x}^y, \underline{x}^y] & \text{se } y > 0 \text{ é par e } \bar{x} < 0 \\ [0, \max\{\underline{x}^y, \bar{x}^y\}] & \text{caso contrário} \end{cases}$$

Por exemplo, para $X = [-\infty, -2]$ e $Y = [-3, -3]$, temos que

$$\begin{aligned} X \wedge Y &= [1, 1] \div ([-\infty, -2] \wedge [3, 3]) = [1, 1] \div [(-\infty)^3, (-2)^3] \\ &= [1, 1] \div [-\infty, -8] = [1, 1] \times [-\frac{1}{8}, 0] = [-\frac{1}{8}, 0] \end{aligned}$$

Na álgebra intervalar, $X \times X \neq X \wedge [2, 2]$.

Também queremos definir a radiciação e a divisão por um intervalo que contém zero. Para isso, vamos apresentar o conceito de multi-intervalo e utilizá-lo no conjunto imagem dessas operações.

Definição 2.13 Um *multi-intervalo* $\mathbf{X} = \langle X_1, X_2, \dots, X_n \rangle$ é o conjunto de números reais $\mathbf{X} = \bigcup_{i=1}^n X_i$, onde $X_i \in \mathbb{I}$.

Definição 2.14 Denotamos por $\mathbb{M} = \{ \langle X_1, X_2, \dots, X_n \rangle \mid n \in \mathbb{N} \text{ e } X_i \in \mathbb{I} \}$ o conjunto de todos os multi-intervalos em \mathbb{R} .

O conceito de multi-intervalo é uma extensão do conceito de intervalo. Eles também representam conjuntos de valores reais e podemos compará-los, assim como os intervalos, da mesma forma que comparamos conjuntos: $\mathbf{X} = \mathbf{Y}$, $\mathbf{X} \neq \mathbf{Y}$, $\mathbf{X} \subset \mathbf{Y}$ ou $\mathbf{X} \subseteq \mathbf{Y}$. No entanto, não basta comparar apenas os limitantes dos intervalos que o compõe, tornando a computação de tais relações um pouco mais específica.

Como qualquer intervalo X pode ser representado por um multi-intervalo $\mathbf{X} = \langle X \rangle$, todas as definições e resultados posteriores sobre multi-intervalos também valem para intervalos.

Finalmente, vamos definir as novas operações sobre dois intervalos $X = [\underline{x}, \bar{x}]$ e $Y = [\underline{y}, \bar{y}]$ não vazios.

- **Divisão** (denominador contém zero):

Se $\underline{y} \leq 0 \leq \bar{y}$ e $c(Y) > 0$, definimos

$$X \div Y = \begin{cases} X \times Z_1 & \text{se } \bar{y} = 0 \\ X \times Z_2 & \text{se } \underline{y} = 0 \\ X \times \langle Z_1, Z_2 \rangle & \text{caso contrário} \end{cases}$$

onde $Z_1, Z_2 \in \mathbb{I}$ são dois intervalos tais que

$$Z_1 = \begin{cases} [-\infty, 0) & \text{se } \bar{y} = \infty \\ [-\infty, \frac{1}{\underline{y}}] & \text{caso contrário} \end{cases}$$

$$Z_2 = \begin{cases} (0, \infty] & \text{se } \underline{y} = \infty \\ [\frac{1}{\bar{y}}, \infty] & \text{caso contrário} \end{cases}$$

Intuitivamente, como $c(Y) > 0$, temos 3 casos possíveis:

- se $\bar{y} = 0$, então $\underline{y} < 0$ e $\max\{\frac{1}{y} \mid \underline{y} \leq y < \bar{y}\} = \frac{1}{\underline{y}}$, pois $\frac{1}{y}$ é estritamente decrescente. Por outro lado, $\lim_{y \rightarrow 0^-} \frac{1}{y} = -\infty$;
- se $\underline{y} = 0$, então $\bar{y} > 0$ e $\min\{\frac{1}{y} \mid \underline{y} < y \leq \bar{y}\} = \frac{1}{\bar{y}}$, pois $\frac{1}{y}$ é estritamente decrescente. Por outro lado, $\lim_{y \rightarrow 0^+} \frac{1}{y} = \infty$;
- se $\underline{y} < 0$ e $\bar{y} > 0$, então, da mesma forma, $\lim_{y \rightarrow 0^-} \frac{1}{y} = -\infty$ e $\lim_{y \rightarrow 0^+} \frac{1}{y} = \infty$.
No entanto, se $\underline{y} > -\infty$ ou $\bar{y} < \infty$, então $\frac{1}{\underline{y}} < \frac{1}{\bar{y}}$ e $\nexists y \in Y \mid \frac{1}{\underline{y}} < \frac{1}{y} < \frac{1}{\bar{y}}$.
Portanto, temos dois intervalos disjuntos.

• **Radiciação:**

Utilizando a mesma definição das demais operações, temos que $\sqrt[y]{X} = \{\sqrt[y]{x} \mid x \in X \text{ e } y \in Y\}$. Assim como na potenciação, estamos interessados apenas quando $Y = [y, y]$ e $y \in \mathbb{Z}^*$.

$$X \curlywedge Y = \sqrt[y]{X} = \begin{cases} [1, 1] \div (X \curlywedge -Y) & \text{se } y < 0 \\ [\sqrt[y]{\underline{x}}, \sqrt[y]{\overline{x}}] & \text{se } y > 0 \text{ é ímpar} \\ \langle [-\sqrt[y]{\underline{x}}, -\sqrt[y]{\overline{x}}], [\sqrt[y]{\underline{x}}, \sqrt[y]{\overline{x}}] \rangle & \text{se } y > 0 \text{ é par e } \underline{x} \geq 0 \\ \emptyset & \text{se } y > 0 \text{ é par e } \overline{x} < 0 \\ [0, \overline{x}] \curlywedge Y & \text{caso contrário} \end{cases}$$

Como estamos trabalhando apenas com números reais, definimos que a raiz par de intervalos que contém apenas números negativos é o conjunto vazio.

Por exemplo, para $X = [1, \infty]$ e $Y = [-3, -3]$, temos que

$$\begin{aligned} X \curlywedge Y &= [1, 1] \div ([1, \infty] \curlywedge [3, 3]) = [1, 1] \div [\sqrt[3]{1}, \sqrt[3]{\infty}] \\ &= [1, 1] \div [1, \infty] = [1, 1] \times (0, 1] = (0, 1] \end{aligned}$$

Definição 2.15 O conjunto $\uplus = \{ +, -, \times, \div, \wedge, \curlywedge \}$ é o conjunto dos operadores intervalares algébricos.

Teorema 2.1 Sejam três intervalos não vazios X, Y e Z tais que $X \subseteq Y \circ Z$, para algum $\circ \in \uplus$, então $\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y \circ z$.

Prova. Segue no apêndice A.

□

Todas as operações aqui apresentadas se estendem à intervalos não fechados. Sejam $Y, Z \in \mathbb{I}$, onde pelo menos um deles é um intervalo não fechado, e $X \in \mathbb{I}$ tal que $X = Y \circ Z$, para algum $\circ \in \uplus$. Os limitantes \underline{x} e \overline{x} são calculados da mesma forma como foram definidos acima, no entanto, $\tau_e(X)$ e $\tau_d(X)$ variam conforme os operandos sejam abertos ou semi-abertos.

Por exemplo, para $Y = (1, 2]$ e $Z = [3, 4]$, temos que $X = Y \times Z = (3, 8]$, pois $\underline{x} = \min\{\underline{y} \cdot \underline{z}, \underline{y} \cdot \bar{z}, \bar{y} \cdot \underline{z}, \bar{y} \cdot \bar{z}\} = 3$ e $\bar{x} = \max\{\underline{y} \cdot \underline{z}, \underline{y} \cdot \bar{z}, \bar{y} \cdot \underline{z}, \bar{y} \cdot \bar{z}\} = 8$. Como $\underline{x} = \underline{y} \cdot \underline{z}$, então $\tau_e(X) = \tau_e(Y) \cdot \tau_e(Z)$, e, como $\bar{x} = \bar{y} \cdot \bar{z}$, então $\tau_d(X) = \tau_d(Y) \cdot \tau_d(Z)$.

Assim, para $X = Y \circ Z$ qualquer:

$$\tau_e(X) = \begin{cases} \tau_e(Y) \cdot \tau_e(Z) & \text{se } \underline{x} = \underline{y} \circ \underline{z} \\ \tau_e(Y) \cdot \tau_d(Z) & \text{se } \underline{x} = \underline{y} \circ \bar{z} \\ \tau_d(Y) \cdot \tau_e(Z) & \text{se } \underline{x} = \bar{y} \circ \underline{z} \\ \tau_d(Y) \cdot \tau_d(Z) & \text{se } \underline{x} = \bar{y} \circ \bar{z} \end{cases}$$

$$\tau_d(X) = \begin{cases} \tau_e(Y) \cdot \tau_e(Z) & \text{se } \bar{x} = \underline{y} \circ \underline{z} \\ \tau_e(Y) \cdot \tau_d(Z) & \text{se } \bar{x} = \underline{y} \circ \bar{z} \\ \tau_d(Y) \cdot \tau_e(Z) & \text{se } \bar{x} = \bar{y} \circ \underline{z} \\ \tau_d(Y) \cdot \tau_d(Z) & \text{se } \bar{x} = \bar{y} \circ \bar{z} \end{cases}$$

Por fim, para manter as operações consistentes, vamos definir mais duas operações:

- **Divisão por zero** (denominador é intervalo degenerado que contém zero):

$$X \div [0, 0] = [-\infty, \infty]$$

- **Logaritmo:**

$$X \wr Y = \log_Y X = [-\infty, \infty]$$

As operações acima serão utilizadas neste trabalho apenas como operadores inversos, não sendo necessário defini-las com as mesmas propriedades das demais operações. Como estamos interessados em sistemas de equações sobre operações algébricas, a operação $X \wr Y$ não aparecerá na modelagem e, portanto, basta defini-la para que não torne alguma equação da forma $Z = X \wr Y$ falsa. Como definimos $X \wr Y = [-\infty, \infty]$, a equação $Z = X \wr Y$ será verdadeira para quaisquer intervalos X, Y e Z .

Associamos a cada operador \circ um operador inverso \bullet correspondente. O operador

inverso \bullet de um operador \circ será definido como o operador inverso usual. Por exemplo, para $\circ = +$ temos que $\bullet = -$ e para $\circ = \times$ temos que $\bullet = \div$.

Além disso, vamos definir para cada operador \circ um segundo inverso não usual \odot . Para tanto, vamos definir mais cinco operações intervalares.

Definição 2.16 *Os operadores binários \oplus , \otimes , $\hat{\Upsilon}$, $\hat{\lambda}$ e $\hat{\wr}$ sobre dois intervalos não vazios X e Y , são dados por*

$$X \oplus Y = Y - X$$

$$X \otimes Y = Y \div X$$

$$X \hat{\Upsilon} Y = Y \wedge X$$

$$X \hat{\lambda} Y = Y \Upsilon X$$

$$X \hat{\wr} Y = Y \wr X$$

Assim, se $\circ = -$, então o seu segundo inverso é o operador $\odot = \oplus$ e, se $\circ = \div$, então $\odot = \otimes$. O segundo inverso do operador \wedge é o operador \wr , representando o usual logaritmo, e o segundo inverso de Υ será definido como $\hat{\wr}$. Na tabela 2.1 apresentamos o inverso e o segundo inverso de cada operador. Na primeira coluna temos os operadores, na segunda seus respectivos inversos e na terceira os segundos inversos.

\circ	\bullet	\odot
+	-	-
-	+	\oplus
\oplus	\oplus	+
\times	\div	\div
\div	\times	\otimes
\otimes	\otimes	\times
\wedge	Υ	\wr
Υ	\wedge	$\hat{\wr}$
\wr	$\hat{\Upsilon}$	$\hat{\lambda}$
$\hat{\Upsilon}$	\wr	Υ
$\hat{\wr}$	$\hat{\lambda}$	$\hat{\Upsilon}$
$\hat{\lambda}$	$\hat{\wr}$	\wedge

Tabela 2.1: Relações entre operadores

Vamos utilizar esses operadores para tornar padrão as manipulações algébricas sobre

equações intervalares e reais. Por exemplo, a igualdade $x = y + z$ pode ser escrita como $y = x - z$ e $z = x - y$. Para a subtração, no entanto, esse padrão não vale, pois $x = y - z$ pode ser escrita como $y = x + z$ mas não como $z = x + y$, forçando a utilização do operador \oplus : $z = x \oplus y$.

Assim, podemos representar as três formas equivalentes da equação $x = y \circ z$ por

$$x = y \circ z$$

$$y = x \bullet z$$

$$z = x \odot y$$

Definição 2.17 O conjunto $\mathfrak{U} = \{ +, -, \times, \div, \wedge, \vee, \lrcorner, \oplus, \otimes, \hat{\vee}, \hat{\wedge}, \hat{\lrcorner} \}$ é o conjunto de todos os operadores intervalares.

Definição 2.18 Sejam dois multi-intervalos $\mathbf{X} = \langle X_1, X_2, \dots, X_n \rangle$ e $\mathbf{Y} = \langle Y_1, Y_2, \dots, Y_m \rangle$ e uma operação $\circ \in \mathfrak{U}$ qualquer, definimos $\mathbf{X} \circ \mathbf{Y}$ como uma extensão da respectiva operação sobre os intervalos, de forma que $\mathbf{X} \circ \mathbf{Y} = \langle X_1 \circ Y_1, X_1 \circ Y_2, \dots, X_1 \circ Y_m, X_2 \circ Y_1, X_2 \circ Y_2, \dots, X_2 \circ Y_m, \dots, X_n \circ Y_1, X_n \circ Y_2, \dots, X_n \circ Y_m \rangle$.

Definição 2.19 O **comprimento** de um multi-intervalo $\mathbf{X} = \langle X_1, \dots, X_n \rangle$ é dado por $c(\mathbf{X}) = \sum_{i=1}^n c(X_i)$.

Teorema 2.2 Sejam três multi-intervalos não vazios \mathbf{X} , \mathbf{Y} e \mathbf{Z} tais que $\mathbf{X} \subseteq \mathbf{Y} \circ \mathbf{Z}$, para algum $\circ \in \mathfrak{U}$, então $\forall x \in \mathbf{X} : \exists y \in \mathbf{Y}, \exists z \in \mathbf{Z} \mid x = y \circ z$.

Prova. Suponha três multi-intervalos $\mathbf{X} = \langle X_1, \dots, X_l \rangle$, $\mathbf{Y} = \langle Y_1, \dots, Y_m \rangle$ e $\mathbf{Z} = \langle Z_1, \dots, Z_n \rangle$ tais que $\mathbf{X} \subseteq \mathbf{Y} \circ \mathbf{Z}$, para algum $\circ \in \mathfrak{U}$. Suponha $x \in \mathbf{X}$ qualquer. Como $\mathbf{X} \subseteq \mathbf{Y} \circ \mathbf{Z}$, então $[x, x] \subseteq (\mathbf{Y} \circ \mathbf{Z})$.

Pela definição, $\mathbf{Y} \circ \mathbf{Z} = \langle Y_1 \circ Z_1, \dots, Y_1 \circ Z_n, \dots, Y_m \circ Z_1, \dots, Y_m \circ Z_n \rangle$ e, portanto,

$$[x, x] \subseteq (Y_1 \circ Z_1) \text{ ou } [x, x] \subseteq (Y_1 \circ Z_2) \text{ ou } \dots \text{ ou } [x, x] \subseteq (Y_m \circ Z_n)$$

Pelo teorema 2.1, é direto que $\exists y \in \mathbf{Y}, \exists z \in \mathbf{Z} \mid x = y \circ z$. Como essa propriedade vale

para todo $x \in X$, então

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y \circ z \quad \square$$

Teorema 2.3 *Sejam $X, Y, Z \in \mathbb{M}$ tais que $X \subseteq Y \circ_1 Z$, para algum $\circ_1 \in \mathfrak{U}$:*

i) Se $Y \subseteq A \circ_2 B$ então $X \subseteq (A \circ_2 B) \circ_1 Z$, para quaisquer $A, B \in \mathbb{M}$ e $\circ_2 \in \mathfrak{U}$;

ii) Se $Z \subseteq A \circ_2 B$ então $X \subseteq Y \circ_1 (A \circ_2 B)$, para quaisquer $A, B \in \mathbb{M}$ e $\circ_2 \in \mathfrak{U}$;

Prova. Sejam $X, Y, Z \in \mathbb{M}$ tais que $X \subseteq Y \circ_1 Z$. Pelo teorema 2.2, $\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y \circ_1 z$.

i) Suponha que $Y \subseteq A \circ_2 B$, para $A, B \in \mathbb{M}$ e $\circ_2 \in \mathfrak{U}$. Pelo mesmo teorema, $\forall y \in Y : \exists a \in A, \exists b \in B \mid y = a \circ_2 b$.

Portanto, temos que

$$\forall x \in X : \exists y \in Y, \exists z \in Z, \exists a \in A, \exists b \in B \mid x = y \circ_1 z \text{ e } y = a \circ_2 b$$

E assim, $\forall x \in X : \exists a \in A, \exists b \in B, \exists z \in Z \mid x = (a \circ_2 b) \circ_1 z$.

Logo, $X \subseteq (A \circ_2 B) \circ_1 Z$.

ii) Segue de forma análoga ao item *i*).

□

Definição 2.20 *Se $X = \emptyset$ ou $Y = \emptyset$, então $X \circ Y = \emptyset$ para qualquer $\circ \in \mathfrak{U}$.*

2.3 Consistência Intervalar

A modelagem intervalar consiste em definir um conjunto de restrições sobre intervalos que representam valores em problemas reais. Tais restrições podem ser sobre cada intervalo de forma isolada ou sobre relações algébricas entre eles.

No primeiro caso, temos especificações sobre os limitantes ou comprimentos dos intervalos. As dimensões comprimento, altura e largura de um objeto, por exemplo, são sempre positivos e podemos representá-las pelas restrições $C \subseteq [0, \infty]$, $A \subseteq [0, \infty]$ e $L \subseteq [0, \infty]$. Outro exemplo é quando representamos um número irracional com precisão preestabelecida: o intervalo Π tal que $\Pi \subseteq [3.14, 3.15]$ e $c(\Pi) < 0.0001$ pode representar o número π com precisão de 4 casas decimais.

Definição 2.21 *Uma restrição sobre um intervalo é uma relação que associa um multi-intervalo X a um intervalo constante $[a, b]$ ou o seu comprimento $c(X)$ a um erro $\epsilon \in \overline{\mathbb{R}}$ tais que*

$$\begin{aligned} X &\subseteq [a, b] \\ c(X) &\leq \epsilon \end{aligned}$$

Além disso, os intervalos e multi-intervalos devem conter valores que satisfaçam determinadas restrições independente de qual seja o valor real representado pelos demais intervalos.

Em um exemplo que tem como contexto a epígrafe deste capítulo, observemos que dados dois intervalos $X = [\frac{14}{3}, 6]$ e $Y = [7, 9]$ que representam, respectivamente, dois números reais x e y tais que $x \cdot y = 42$, embora $X \times Y = [\frac{98}{3}, 54] \neq [42, 42]$, temos que $\forall x \in X : \exists y \in Y \mid x \cdot y = 42$ e, da mesma forma, $\forall y \in Y : \exists x \in X \mid x \cdot y = 42$. Nesse sentido, as restrições intervalares não são uma extensão direta das restrições reais, mas apresentam uma acepção distinta, com relação aos valores reais de fato representados pelos intervalos relacionados. Nesse exemplo, os intervalos X e Y são consistentes com a restrição dada, mas não garantem que quaisquer valores $x' \in X$ e $y' \in Y$ satisfazem $x' \cdot y' = 42$. Assim, para se obter uma solução real desse problema, temos primeiramente que definir x' e, como $\forall x' \in X : \exists y' \in Y \mid x' \cdot y' = 42$, inferir o valor y' ; do contrário, poderíamos obter uma resposta incorreta (por exemplo, $6 \cdot 9 = 42$).

Enquanto obter a solução real de um conjunto de intervalos consistentes seja o principal objetivo deste trabalho (e será discutido detalhadamente no capítulo 5), manter tal

consistência é um problema já discutido em diversos trabalhos [16, 18, 31, 26]. Tanto as comparações definidas na seção 1.1 quanto as operações definidas na seção 1.2 mantêm essa consistência; ambas garantem que existem valores em todos os intervalos capazes de satisfazer as restrições (podemos verificar isso pelos teoremas 2.1 e 2.2). Por exemplo, o valor real representado por $Y \circ Z$ deve ser exatamente o resultado de $y \circ z$, onde $y \in \mathbb{R}$ e $z \in \mathbb{R}$ são os valores representados, respectivamente, pelos intervalos Y e Z . A partir do momento em que trabalhamos com o modelo intervalar, e não com o problema real, perdemos informações sobre os números y e z e, conseqüentemente, sobre o intervalo $Y \circ Z$. Portanto, para todo $y \in Y$ e $z \in Z$ deve existir $x \in Y \circ Z$ tal que $x = y \circ z$.

Conseqüentemente, manipulações algébricas sobre intervalos apresentam um comportamento diferente da álgebra elementar. Por exemplo, a igualdade $X = Y \circ Z$ não implica $Y = X \bullet Z$ e nem $Z = X \odot Y$, para qualquer $\circ \in \uplus$.

Suponha três multi-intervalos X , Y e Z tais que $X \subseteq Y \circ Z$. Pelo teorema 2.2, $\forall x \in X, \exists y \in Y, \exists z \in Z \mid x = y \circ z$. Isto é, independente de qual seja o valor representado por X , existem valores em Y e em Z capazes de satisfazer a restrição $x = y \circ z$. Nesse caso, a restrição $X \subseteq Y \circ Z$ está relacionada com a restrição real $x = y \circ z$.

Como multi-intervalos são uma generalização dos intervalos, essa relação também vale para intervalos. Assim, podemos associar uma modelagem intervalar à uma modelagem real algébrica. Para tanto, repare que $(x = y \circ z) \equiv (y = x \bullet z)$, enquanto que $(X \subseteq Y \circ Z) \not\equiv (Y \subseteq X \bullet Z)$. Por exemplo, para $X = [2, 2]$, $Y = [1, 2]$ e $Z = [1, 2]$ temos que $X \subseteq Y + Z$ e $Y \not\subseteq X - Z$, pois $[2, 2] \subseteq [2, 4]$ e $[1, 2] \not\subseteq [0, 1]$. Portanto, um conjunto de intervalos cujos reais representados satisfazem a restrição $x = y \circ z$ pode não satisfazer a restrição $y = x \bullet z$. Como o mesmo ocorre com o segundo inverso \odot , temos que representar uma restrição real utilizando três restrições intervalares.

Vamos denotar por $X \asymp Y \circ Z$ a representação intervalar da restrição $x = y \circ z$, de forma que $Y \asymp X \bullet Z$ representa $y = x \bullet z$ e $Z \asymp X \odot Y$ representa $z = x \odot y$.

Definição 2.22 *Uma restrição intervalar R relaciona um multi-intervalo X com uma função dada por $Y \circ Z$ de forma que $R = (X \asymp Y \circ Z)$, onde*

$$X \asymp Y \circ Z \text{ se, e somente se, } (X \subseteq Y \circ Z) \wedge (Y \subseteq X \bullet Z) \wedge (Z \subseteq X \odot Y)$$

Teorema 2.4

$$(X \asymp Y \circ Z) \equiv (Y \asymp X \bullet Z) \equiv (Z \asymp X \odot Y)$$

Prova. Primeiramente, vamos estender a tabela 2.1 adicionando quatro novas colunas, obtendo a tabela 2.2. Na quarta coluna temos o operador \bullet' , que é o inverso do inverso \bullet do operador \circ . Na próxima coluna, temos o operador \bullet'' , que é o segundo inverso do inverso \bullet do operador \circ . Nas duas últimas colunas, temos os operadores \odot' e \odot'' , que são, respectivamente, o inverso do segundo inverso \odot do operador \circ e o segundo inverso de \odot .

\circ	\bullet	\odot	\bullet'	\bullet''	\odot'	\odot''
+	-	-	+	\oplus	+	\oplus
-	+	\oplus	-	-	\oplus	+
\oplus	\oplus	+	\oplus	+	-	-
\times	\div	\div	\times	\otimes	\times	\otimes
\div	\times	\otimes	\div	\div	\otimes	\times
\otimes	\otimes	\times	\otimes	\times	\div	\div
λ	γ	\wr	λ	$\hat{\wr}$	$\hat{\gamma}$	$\hat{\lambda}$
γ	λ	$\hat{\wr}$	γ	\wr	$\hat{\lambda}$	$\hat{\gamma}$
\wr	$\hat{\gamma}$	$\hat{\lambda}$	\wr	γ	$\hat{\wr}$	λ
$\hat{\gamma}$	\wr	γ	$\hat{\gamma}$	$\hat{\lambda}$	λ	$\hat{\wr}$
$\hat{\wr}$	$\hat{\lambda}$	$\hat{\gamma}$	$\hat{\wr}$	λ	\wr	γ
$\hat{\lambda}$	$\hat{\wr}$	λ	$\hat{\lambda}$	$\hat{\gamma}$	γ	\wr

Tabela 2.2: Relações estendidas entre operadores

Vamos utilizar a tabela 2.2 para mostrar que os três conjuntos de restrições abaixo são equivalentes:

$$\begin{aligned}
 X \subseteq Y \circ Z & \quad Y \subseteq X \bullet Z & \quad Z \subseteq X \odot Y \\
 Y \subseteq X \bullet Z & \equiv X \subseteq Y \bullet' Z & \equiv X \subseteq Z \odot' Y \\
 Z \subseteq X \odot Y & \quad Z \subseteq Y \bullet'' X & \quad Y \subseteq Z \odot'' X
 \end{aligned}$$

Iniciando pelos dois primeiros conjuntos, temos que $Y \subseteq X \bullet Z$ está presente em ambos. Além disso, pela tabela, $Y \circ Z = Y \bullet' Z$ e, pela definição 2.16, $X \odot Y = Y \bullet'' X$, para todo $\circ \in \mathbb{U}$. Portanto, $(X \subseteq Y \circ Z) \equiv (X \subseteq Y \bullet' Z)$ e $(Z \subseteq X \odot Y) \equiv (Z \subseteq Y \bullet'' X)$. Logo, os dois primeiros conjuntos de restrições são equivalentes.

Agora, comparando o segundo e o terceiro conjuntos, temos que $(Z \subseteq X \odot Y) \equiv (Z \subseteq Y \bullet'' X)$ e, pela tabela, $Z \odot' Y = Y \bullet' Z$ e $Z \odot'' X = X \bullet Z$, para todo $\circ \in \mathfrak{U}$. Portanto, $(X \subseteq Z \odot' Y) \equiv (X \subseteq Y \bullet' Z)$ e $(Y \subseteq Z \odot'' X) \equiv (Y \subseteq X \bullet Z)$. Logo, os dois últimos conjuntos de restrições também são equivalentes e, portanto, o primeiro com o último também o são.

Então, pela definição 2.22, temos que

$$(X \asymp Y \circ Z) \equiv (Y \asymp X \bullet Z) \equiv (Z \asymp X \odot Y)$$

□

Mais especificamente, essa restrição é definida como uma **restrição primitiva**. Restrições não primitivas são aquelas cuja função à ser comparada tem aridade maior que dois. Podemos facilmente reescrever essas restrições por uma conjunção de restrições primitivas [17].

Exemplo 2.1 *Podemos reescrever a restrição não primitiva $X \asymp Y + (Z \times W)$ substituindo a operação $(Z \times W)$ por um novo multi-intervalo A e criando uma nova restrição que relaciona A com a operação $(Z \times W)$:*

$$(X \asymp Y + A) \wedge (A \asymp Z \times W)$$

Pela definição de restrição intervalar,

$$\begin{aligned} X \asymp Y + (Z \times W) \text{ se, e somente se } & X \subseteq Y + (Z \times W) \wedge \\ & Y \subseteq X - (Z \times W) \wedge \\ & Z \subseteq (X - Y) \div W \wedge \\ & W \subseteq (X - Y) \div Z \wedge \end{aligned}$$

Enquanto a conjunção de restrições primitivas é tal que

$$\begin{aligned}
 (X \asymp Y + A) \wedge (A \asymp Z \times W) \quad \text{se, e somente se} \quad & X \subseteq Y + A \quad \wedge \\
 & Y \subseteq X - A \quad \wedge \\
 & A \subseteq X - Y \quad \wedge \\
 & A \subseteq Z \times W \quad \wedge \\
 & Z \subseteq A \div W \quad \wedge \\
 & W \subseteq A \div Z \quad \wedge
 \end{aligned}$$

Pelo teorema 2.3, é direto que as duas representações são equivalentes, pois

$$\begin{aligned}
 (X \subseteq Y + A) \wedge (A \subseteq Z \times W) &\Rightarrow X \subseteq Y + (Z \times W) \\
 (Y \subseteq X - A) \wedge (A \subseteq Z \times W) &\Rightarrow Y \subseteq X - (Z \times W) \\
 (Z \subseteq A \div W) \wedge (A \subseteq X - Y) &\Rightarrow Z \subseteq (X - Y) \div W \\
 (W \subseteq A \div Z) \wedge (A \subseteq X - Y) &\Rightarrow W \subseteq (X - Y) \div Z
 \end{aligned}$$

Onde o símbolo \Rightarrow representa a implicação usual ($p \Rightarrow q \equiv$ “se p então q ”).

□

Na literatura, esse conceito de restrição intervalar é comumente definido como **consistência no arco** [2, 18, 31].

Definição 2.23 *Um conjunto de intervalos $\{X, Y, Z\}$ é **consistente no arco** de uma restrição R se R é satisfeita por $\{X, Y, Z\}$, isto é, se R é verdadeira para os intervalos X , Y e Z .*

Na álgebra intervalar, inequações podem ser transformadas em equações adicionando novas variáveis à fórmula, como ocorre na transformação em restrições primitivas. Por exemplo, uma restrição da forma $X \leq Y \circ Z$ pode ser reescrita pela conjunção $(A \asymp Y \circ Z) \wedge (X \asymp A - [0, \infty])$.

Se um intervalo X aparece duas vezes ou mais numa restrição ou conjunto de restrições, essa álgebra considera que cada instância de X pode representar um número real distinto. Isto é, não existe uma relação que garante que todas as instâncias de X representam o mesmo valor real.

Por exemplo, a restrição $X \asymp X \times X$ é satisfeita para $X = (0, 1)$. Isto é, $\forall x \in X : \exists x' \in X, \exists x'' \in X \mid x = x' \cdot x''$. No entanto, queremos que $x = x' = x''$, ou ainda, que se X representa um valor real x , então esse mesmo valor deve ser representado por todas as instâncias de X . Contudo, $\nexists x \in X \mid x = x \cdot x$.

Assim, a consistência intervalar não garante a existência de um conjunto de valores reais capaz de satisfazer todas as restrições, salvo quando todos os intervalos são degenerados. Posteriormente vamos apresentar as propriedades que um conjunto de restrições deve manter para que os valores reais representados pelos respectivos intervalos sejam únicos.

Alternativamente à consistência no arco, temos o conceito de **consistência na envoltória**: um relaxamento da primeira que visa evitar a definição de multi-intervalos. Assim, podemos restringir o conjunto de *multi-intervalos* consistentes com uma dada restrição R à um conjunto de *intervalos*.

Exemplo 2.2 *Dados dois intervalos $Y = [1, 2]$ e $Z = [2, 2]$, a operação $Y \vee Z = \langle [-\sqrt{2}, -1], [1, \sqrt{2}] \rangle$ resulta em um multi-intervalo. Assim, a restrição $R = (X \asymp Y \vee Z)$ só pode ser satisfeita se $X = \langle X_1, \dots, X_n \rangle$ e, para todo $1 \leq i \leq n$, $X_i \subseteq [-\sqrt{2}, -1]$ ou $X_i \subseteq [1, \sqrt{2}]$. O multi-intervalo de comprimento máximo consistente com R , nesse caso, é $X = \langle [-\sqrt{2}, -1], [1, \sqrt{2}] \rangle$, pois*

$$X \subseteq Y \vee Z \quad \Rightarrow \quad \langle [-\sqrt{2}, -1], [1, \sqrt{2}] \rangle \subseteq \langle [-\sqrt{2}, -1], [1, \sqrt{2}] \rangle$$

$$Y \subseteq X \wedge Z \quad \Rightarrow \quad [1, 2] \subseteq \langle [1, 2], [1, 2] \rangle$$

$$Z \subseteq X \hat{\vee} Y \quad \Rightarrow \quad [2, 2] \subseteq [-\infty, \infty]$$

A fim de que o multi-intervalo de comprimento máximo possa ser escrito como um

intervalo, relaxamos a restrição R obtendo uma restrição $\|R\| = (\mathbf{X} \asymp \|\mathbf{Y} \wedge \mathbf{Z}\|)$, onde $\|\mathbf{Y} \wedge \mathbf{Z}\| = [-\sqrt{2}, \sqrt{2}]$. Assim, $X = [-\sqrt{2}, \sqrt{2}]$ é o intervalo de comprimento máximo capaz de satisfazer a restrição $\|R\|$.

□

Definição 2.24 A envoltória $\|\mathbf{X}\|$ de um multi-intervalo $\mathbf{X} = \langle X_1, \dots, X_n \rangle$ é um intervalo tal que

- $\underline{x} = \min\{\underline{x}_i \mid 1 \leq i \leq n\}$ é o limitante inferior de $\|\mathbf{X}\|$, tal que \underline{x}_i é o limitante inferior do intervalo X_i ;
- $\bar{x} = \max\{\bar{x}_i \mid 1 \leq i \leq n\}$ é o limitante superior de $\|\mathbf{X}\|$, tal que \bar{x}_i é o limitante superior do intervalo X_i ;
- $\tau_e(\|\mathbf{X}\|) = \max\{\tau_e(X_k) \mid \underline{x}_k = \underline{x}, \text{ para } 1 \leq k \leq n\}$ define se $\|\mathbf{X}\|$ é fechado à esquerda ou não, tal que \underline{x}_k é o limitante inferior do intervalo X_k ;
- $\tau_d(\|\mathbf{X}\|) = \max\{\tau_d(X_k) \mid \bar{x}_k = \bar{x}, \text{ para } 1 \leq k \leq n\}$ define se $\|\mathbf{X}\|$ é fechado à direita ou não, tal que \bar{x}_k é o limitante superior do intervalo X_k .

Definição 2.25 Uma *restrição intervalar na envoltória* $\|R\|$ relaciona um multi-intervalo \mathbf{X} com a envoltória de uma função dada por $\mathbf{Y} \circ \mathbf{Z}$ de forma que $\|R\| = (\mathbf{X} \asymp \|\mathbf{Y} \circ \mathbf{Z}\|)$, onde

$$\mathbf{X} \asymp \|\mathbf{Y} \circ \mathbf{Z}\| \text{ se, e somente se, } (\mathbf{X} \subseteq \|\mathbf{Y} \circ \mathbf{Z}\|) \wedge (\mathbf{Y} \subseteq \|\mathbf{X} \bullet \mathbf{Z}\|) \wedge (\mathbf{Z} \subseteq \|\mathbf{X} \odot \mathbf{Y}\|)$$

Teorema 2.5

$$(\mathbf{X} \asymp \|\mathbf{Y} \circ \mathbf{Z}\|) \equiv (\mathbf{Y} \asymp \|\mathbf{X} \bullet \mathbf{Z}\|) \equiv (\mathbf{Z} \asymp \|\mathbf{X} \odot \mathbf{Y}\|)$$

Prova. Direto da definição de restrição intervalar na envoltória e da tabela 2.2 (teorema 2.4).

□

O teorema 2.2 não vale para $X = \|\mathbf{Y} \circ \mathbf{Z}\|$. Por exemplo, se $\mathbf{Y} = \langle [4, 9] \rangle$ e $\mathbf{Z} = \langle [2, 2] \rangle$, então $\mathbf{Y} \vee \mathbf{Z} = [4, 9] \vee [2, 2] = \langle [-3, -2], [2, 3] \rangle$. Por outro lado, $X = \|\mathbf{Y} \vee \mathbf{Z}\| = [-3, 3]$ e

$\exists x \in X \mid x \neq \sqrt{y}$, para todo $y \in Y$; a saber, $\forall x \in (-2, 2)$. Mesmo assim, a consistência na envoltória é bastante utilizada e aceita na prática.

Definição 2.26 *Um conjunto de variáveis $\{X, Y, Z\}$ é consistente na envoltória de uma restrição $R = (X \asymp Y \circ Z)$ se a restrição $\|R\| = (X \asymp \|Y \circ Z\|)$ é satisfeita por $\{X, Y, Z\}$.*

Teorema 2.6 *Se um conjunto de variáveis $\{X, Y, Z\}$ é consistente no arco de uma restrição R , então ele também é consistente na envoltória de R .*

Prova. Direto do fato que $X \subseteq \|X\|$ para todo multi-intervalo X .

□

Na literatura, existem ainda outras consistências intervalares. São, em geral, relaxamentos da consistência no arco ou consistências globais, que consideram todo um conjunto de restrições na verificação da consistência e não cada restrição isoladamente, permitindo, inclusive, restrições não primitivas. Entre tais consistências, podemos citar a *consistência no caminho*, *consistência de caixa* e *consistência kB* [2, 13, 3].

2.4 Considerações

Neste capítulo apresentamos alguns exemplos da utilização da modelagem intervalar na representação de problemas reais. Também apresentamos as principais definições e teoremas da álgebra intervalar e suas operações: adição, subtração, multiplicação, divisão, potenciação e radiciação inteiras, necessárias para a representação de restrições intervalares algébricas. Essas restrições foram definidas como primitivas e mostramos que elas não são expressadas por equações ou inequações, como nas restrições sobre os números reais, mas fazem uso do operador relacional \asymp de forma a garantir a consistência intervalar.

Em seguida, apresentamos o conceito de consistência intervalar. Neste trabalho, esse conceito define a aceitação das restrições intervalares e será utilizado, de maneira mais aprofundada, nos algoritmos apresentados no capítulo 4. Nesse mesmo contexto, introduzimos dois problemas comumente encontrados no cenário intervalar: a extração de uma

solução real de um conjunto de intervalos consistentes e a existência de tal solução, não garantida pela consistência intervalar. Vamos abordar ambos os problemas no final do capítulo 4 e durante todo o capítulo 5, apresentando a nossa solução para algumas classes de sistemas intervalares.

CAPÍTULO 3

ÁLGEBRA BOOLEANA

“Alguns mais curiosos, atraídos pelo nome chamativo, compraram a obra e leram, assombrados, a única sentença do tomo, formada por apenas quatro caracteres, sendo um deles um ponto final: $a = a$. Foi um espanto! Popô havia, simplesmente, descoberto a Tautologia.”

(Joshua, Triste Fim de Popô Joquempô)

A lógica proposicional clássica é uma importante ferramenta da matemática e da filosofia [5] que permite descrever raciocínios capazes de inferir *valores verdade* de sentenças proposicionais e gerar contradições ou provas de teoremas e argumentos. Ao mesmo tempo, calcular a *satisfatibilidade* de um raciocínio descrito em lógica proposicional é fundamental para definir sua validade em um determinado estado ou mesmo descobrir tautologias (verdades incontestáveis).

A formalização da lógica proposicional como álgebra booleana foi acompanhada pelo reconhecimento de que vários problemas combinatoriais que existem no mundo real, como verificação de *software* e *hardware*, planejamento e otimização, podem ser expressados por fórmulas booleanas (proposicionais) [5], motivando a implementação de diversas técnicas para analisar propriedades de tais fórmulas [11, 27, 25, 23].

Neste capítulo definimos alguns conceitos e notação da Satisfatibilidade Booleana, bem como algoritmos e técnicas atualmente utilizadas para resolvê-la. Em seguida, apresentamos algumas variantes da álgebra booleana capazes de representar domínios específicos de uma maneira mais direta e eficiente.

3.1 Satisfatibilidade Booleana

No decorrer deste trabalho definimos algumas classes de fórmulas lógicas. Tais fórmulas estão sempre na *Forma Normal Conjuntiva*, isto é, possuem uma estrutura composta por uma conjunção de *cláusulas*. Intuitivamente, uma cláusula é uma disjunção de variáveis ou restrições (neste trabalho também chamados de *átomos*) sobre alguma teoria bem definida. Assim, uma cláusula é composta por variáveis cujo domínio é específico a cada classe definida. Essa forma facilita a definição de problemas e a concepção de algoritmos. É importante observar que qualquer fórmula pode ser reduzida à sua forma normal conjuntiva em tempo linear [29].

Dada uma fórmula lógica ψ na forma normal conjuntiva, definimos os seguintes conjuntos:

- $C(\psi)$ é o conjunto de cláusulas de ψ ;
- $A(\psi)$ é o conjunto de átomos de ψ ;
- $A(\omega)$ é o conjunto de átomos da cláusula ω ;
- $V(\omega)$ é o conjunto de variáveis da cláusula ω ;
- $V(\alpha)$ é o conjunto de variáveis do átomo α ;
- $V(\psi)$ é o conjunto de variáveis da fórmula ψ ;
- $D(\psi)$ é o conjunto domínio das variáveis da fórmula ψ .

Além disso, as seguintes definições são estendidas a todas as fórmulas lógicas definidas neste trabalho.

Definição 3.1 *Uma **valoração** $v : V(\psi) \mapsto D(\psi)$ é uma função parcial que associa a cada variável $x \in V(\psi)$ no máximo um valor de seu domínio.*

Como v é uma função parcial, denotamos por $v(x) = \diamond$ se x não possui valor definido em v . Adicionalmente, dada uma função parcial $v : V(\psi) \mapsto D(\psi)$ tal que $v(y) = \diamond$ para

algum $y \in V(\psi)$, denotamos por $v' = v \cup (y \mapsto a)$ a extensão de v tal que $a \in D(\psi)$, $v'(y) = a$ e, para todo $x \in V(\psi) \setminus \{y\}$, $v'(x) = v(x)$.

Definição 3.2 Uma *interpretação* $\mathfrak{I} : \psi \times v \mapsto \{0, 1\}$ é uma função parcial que associa um valor verdade à fórmula ψ de acordo com a valoração v .

Definição 3.3 Uma valoração v é um **modelo** da fórmula ψ se $\mathfrak{I}(\psi, v) = 1$.

Definição 3.4 Dizemos que ψ é **satisfatível** se possui pelo menos um modelo. Caso contrário, dizemos que ψ é **insatisfatível**.

No contexto da álgebra booleana propriamente dita, vamos apresentar algumas definições necessárias para descrever o problema da Satisfatibilidade Booleana e, em seguida, o algoritmo DPLL, comumente utilizado para resolver tal problema, especificando técnicas de implementação e otimização de estruturas de dados.

Definição 3.5 Uma *variável booleana* $x \in \{0, 1\}$ assume apenas os valores 0 (falso) ou 1 (verdadeiro).

Na álgebra booleana, os operadores **conjunção** (\wedge), **disjunção** (\vee) e **negação** (\neg) operam sobre variáveis booleanas e são definidos como na lógica proposicional clássica [9].

Definição 3.6 Um *literal* \dot{x} representa a variável booleana x ou sua negação $\neg x$.

Definição 3.7 Uma *cláusula* ω é uma disjunção de literais.

Definição 3.8 Uma *fórmula booleana* φ é uma conjunção de cláusulas.

Vamos denotar por $L(\omega)$ o conjunto de literais da cláusula ω e $L(\varphi) = \bigcup_{\omega \in C(\varphi)} L(\omega)$ o conjunto de literais da fórmula φ .

Dados um literal \dot{x} e uma cláusula ω que contém \dot{x} , a função $sign : C(\varphi) \times L(\varphi) \mapsto \{0, 1\}$ indica o sinal da variável representada por \dot{x} em ω :

$$sign_{\omega}(\dot{x}) = \begin{cases} 1 & \text{se } \dot{x} \equiv x \text{ em } \omega \\ 0 & \text{se } \dot{x} \equiv \neg x \text{ em } \omega \end{cases}$$

Também definimos a função $var(\dot{x}) = x$ que retorna a variável representada pelo literal \dot{x} .

Para qualquer fórmula booleana φ , temos que $D(\varphi) = \{0, 1\}$ e

$$\mathfrak{J}(\varphi, v) = \begin{cases} 1 & \text{se } \forall \omega \in C(\varphi), \exists \dot{x} \in L(\omega) \mid v(var(\dot{x})) = sign_{\omega}(\dot{x}) \\ 0 & \text{se } \exists \omega \in C(\varphi) \mid \forall \dot{x} \in L(\omega), v(var(\dot{x})) = \neg sign_{\omega}(\dot{x}) \\ \diamond & \text{c.c.} \end{cases}$$

Dessa forma, podemos definir um problema computacional de decisão no qual queremos saber se uma dada fórmula booleana possui ou não um modelo.

Definição 3.9 *O problema computacional da **Satisfatibilidade Booleana** é dado por:*

Satisfatibilidade Booleana (SAT)

Entrada: fórmula booleana φ .

Saída: modelo v de φ , se tal modelo existir, ou ***insatisfável***.

Não se conhece algoritmo de tempo polinomial no número de variáveis para esse problema. De fato, é o primeiro problema provado ser \mathcal{NP} -Completo [8], e diversas técnicas de busca e poda foram implementadas na tentativa de contornar o pior caso, tirando proveito de estruturas e propriedades das fórmulas booleanas.

O algoritmo DPLL (Davis-Putnam-Logemann-Loveland) foi proposto em [11] como um procedimento de ramificação e inferência de *valores verdade* para resolver SAT. Esse método, descrito no algoritmo 1, simplesmente escolhe variáveis ainda não valoradas e divide o problema em dois subproblemas (com a variável valorada *verdadeira* e *falsa*) e o resolve recursivamente, verificando se a cada passo alguma das subfórmulas geradas é satisfável ou insatisfável (decidível) sob a valoração corrente.

Além disso, apresenta dois métodos para inferir valores de variáveis, tais que a fórmula apenas pode ser satisfável se essas valorações forem respeitadas.

- **Inferência por cláusula unitária:** uma cláusula ω é dita **unitária** se possui apenas uma variável ainda não valorada. A função $unit(\omega, v)$ retorna um literal \dot{x}

tal que $var(\dot{x})$ é a única variável ainda não valorada pela valoração v em ω . Se existirem mais variáveis não valoradas, ou todas já tiverem sido valoradas, a função retorna \diamond .

$$unit(\omega, v) = \begin{cases} \dot{x} & \text{se } \exists \dot{x} \in L(\omega) \mid v(var(\dot{x})) = \diamond \text{ e} \\ & \forall \dot{y} \in L(\omega) \setminus \{\dot{x}\} : v(var(\dot{y})) \neq \diamond \\ \diamond & \text{c.c.} \end{cases}$$

- **Inferência por literal puro:** um literal \dot{x} é dito **puro** se aparece em φ apenas em sua forma positiva ou negada (nunca ambos). A função $puro(\dot{x}, \varphi, v)$ retorna 1 caso \dot{x} represente x em todas as cláusulas em que aparece, 0 caso represente a negação $\neg x$ e \diamond caso represente x em algumas cláusulas e $\neg x$ em outras (indicando que \dot{x} não é um literal puro). O valor retornado por $puro(\dot{x}, \varphi, v)$ deve ser usado para valorar $var(\dot{x})$.

$$puro(\dot{x}, \varphi, v) = \begin{cases} 1 & \text{se } sign_{\omega}(\dot{x}) = 1, \text{ para todo } \omega \in C(\varphi) \mid \dot{x} \in L(\omega) \\ & \text{e } \nexists \dot{y} \in L(\omega) \mid v(var(\dot{y})) = 1 \\ 0 & \text{se } sign_{\omega}(\dot{x}) = 0, \text{ para todo } \omega \in C(\varphi) \mid \dot{x} \in L(\omega) \\ & \text{e } \nexists \dot{y} \in L(\omega) \mid v(var(\dot{y})) = 1 \\ \diamond & \text{c.c.} \end{cases}$$

Algoritmo 1: Davis-Putnam-Logemann-Loveland (DPLL)

Entrada: (φ, v)

Saída: modelo v de φ ou *insatisfatível*

```

1 para todo  $\omega \in C(\varphi) \mid \text{unit}(\omega, v) = \dot{x}$  faça
2    $v \leftarrow v \cup (\text{var}(\dot{x}) \mapsto \text{sign}_\omega(\dot{x}))$ 
3 fim
4 para todo  $\dot{x} \in L(\varphi) \mid \text{puro}(\dot{x}, \varphi, v) \neq \diamond$  faça
5    $v \leftarrow v \cup (\text{var}(\dot{x}) \mapsto \text{puro}(\dot{x}, \varphi, v))$ 
6 fim
7 se  $\mathfrak{J}(\varphi, v) = 1$  então
8   retorna  $v$ 
9 fim
10 se  $\mathfrak{J}(\varphi, v) = 0$  então
11   retorna insatisfatível
12 fim
13  $x \leftarrow \text{escolhe}(V(\varphi), v)$ 
14  $\phi \leftarrow \text{DPLL}(\varphi, v \cup (x \mapsto 1))$ 
15 se  $\phi \neq \text{insatisfatível}$  então
16   retorna  $\phi$ 
17 fim
18 retorna  $\text{DPLL}(\varphi, v \cup (x \mapsto 0))$ 

```

O procedimento $\text{escolhe}(V(\varphi), v)$ seleciona uma variável sem valor definido em v segundo alguma heurística, como VSIDS [25], por exemplo. Logo em seguida, v é estendida, definindo um valor do domínio $\{0,1\}$ para a variável selecionada.

Na prática, o DPLL é implementado por uma busca em profundidade, de forma iterativa, com inferência por cláusula unitária (geralmente não implementa literal puro) e *retrocesso não cronológico*. Utiliza uma pilha de literais decididos (que são selecionados pelo procedimento $\text{escolhe}(V(\varphi), v)$) e constrói um *grafo de implicação* [23] que relaciona cada literal inferido aos literais decididos que ocasionaram sua valoração. Quando uma

valoração que torna a fórmula falsa é encontrada, a cláusula conflitante é analisada e uma nova cláusula é *aprendida* (anexada à fórmula), utilizando o grafo de implicação e garantindo que o subconjunto conflitante da valoração corrente não se repita durante a busca. Então, todos os ramos que valoraram variáveis após as valorações que geraram o conflito são descartados do espaço de busca, garantindo que a valoração corrente, mesmo incompleta, não pode ser modelo para a fórmula.

Essas técnicas têm grande impacto no desempenho dos resolvidores SAT. Além disso, várias outras estruturas de dados utilizadas em rotinas como detecção de cláusulas unitárias ou gerenciamento de cláusulas aprendidas são de fundamental importância num resolvidor eficiente.

3.2 Satisfatibilidade Módulo Teoria

Várias modelagens que utilizam a lógica de primeira ordem em sua representação dispõem de uma ou mais teorias base para representar seus átomos. Isso facilita a implementação de resolvidores de primeira ordem, que podem ser específicos a algumas determinadas teorias, sem se preocupar com a interpretação de predicados e símbolos não pertencentes às teorias base [5].

A partir do interesse em se modelar problemas de uma forma mais expressiva e natural, utilizando a lógica de primeira ordem, e da viabilidade de sua implementação para algumas teorias, concebeu-se um novo modelo de satisfatibilidade, no qual os literais não representam apenas variáveis booleanas, mas também átomos sobre qualquer teoria \mathfrak{T} bem definida. Entre tais teorias, Aritmética (Não Linear) Inteira ou Real, Vetores de Bits, Equações de Funções Indefinidas, entre outras, são implementadas em resolvidores SMT [6, 12].

Definição 3.10 *Um **átomo** α sobre uma teoria \mathfrak{T} é uma relação entre variáveis e/ou operadores de \mathfrak{T} que assume um valor verdade.*

Por exemplo, $\alpha = (x + y = 2)$ é um átomo sobre a Aritmética Linear Inteira e $\neg\alpha = (x + y \neq 2)$ é sua negação.

Definição 3.11 Uma *cláusula sobre* \mathfrak{T} $\omega^{\mathfrak{T}}$ é uma disjunção de átomos sobre uma teoria \mathfrak{T} .

Definição 3.12 Uma *fórmula SMT* $\varphi_{\mathfrak{T}}$ é uma conjunção de cláusulas sobre \mathfrak{T} .

A interpretação de uma fórmula SMT $\varphi_{\mathfrak{T}}$ sob uma valoração v é dada por:

$$\mathfrak{I}(\varphi_{\mathfrak{T}}, v) = \begin{cases} 1 & \text{se } \forall \omega^{\mathfrak{T}} \in C(\varphi_{\mathfrak{T}}), \exists \alpha \in A(\omega^{\mathfrak{T}}) \mid \mathfrak{I}(\alpha, v) = 1 \\ 0 & \text{se } \exists \omega^{\mathfrak{T}} \in C(\varphi_{\mathfrak{T}}) \mid \forall \alpha \in A(\omega^{\mathfrak{T}}), \mathfrak{I}(\alpha, v) = 0 \\ \diamond & \text{c.c.} \end{cases}$$

onde $\mathfrak{I}(\alpha, v)$ é a interpretação do átomo α sob a valoração v e depende da teoria \mathfrak{T} .

Da mesma forma que no contexto booleano, podemos definir um problema computacional de decisão no qual queremos saber se uma dada fórmula SMT possui ou não um modelo.

Definição 3.13 O problema computacional da *Satisfatibilidade Módulo Teoria* é dado por:

Satisfatibilidade Módulo Teoria (SMT)

Entrada: fórmula SMT $\varphi_{\mathfrak{T}}$.

Saída: modelo v de $\varphi_{\mathfrak{T}}$, se tal modelo existir, ou *insatisfável*.

Neste trabalho vamos apresentar três abordagens para resolver SMT. Duas delas, definidas a seguir, são as mais utilizadas pelos resolvedores SMT e fazem uso de um resolvedor SAT para resolver parte ou toda a fórmula. A terceira abordagem é apresentada no capítulo 4, implementada no método iSAT [17] sob a teoria Álgebra Intervalar.

- **Abordagem Ansiosa:**

Nessa abordagem transformamos a fórmula SMT de entrada em uma fórmula booleana equivalente e a resolvemos utilizando um resolvedor SAT. Essa transformação é altamente dependente das teorias que estão sendo utilizadas e sua implementação deve ser específica a cada teoria de interesse.

Não é utilizada na prática, pois mesmo para teorias mais simples é mais eficiente fazer uso das próprias características das cláusulas (cardinalidade, eliminação de variáveis, cláusula unitária, entre outras) do que gerar a fórmula booleana equivalente.

- **Abordagem Preguiçosa:**

Seja $\varphi_{\mathfrak{T}}$ uma fórmula SMT da forma

$$\varphi_{\mathfrak{T}} = \bigwedge_{j=1}^m \bigvee_{i=1}^{n_j} \alpha_{ij}$$

Podemos associar a cada átomo α_{ij} um literal \hat{x}_{ij} e reescrever $\varphi_{\mathfrak{T}}$ da forma

$$\varphi_{\mathfrak{T}} = \underbrace{\bigwedge_{j=1}^m \bigvee_{i=1}^{n_j} \hat{x}_{ij}}_{\varphi} \wedge \underbrace{\bigwedge_{j=1}^m \bigwedge_{i=1}^{n_j} \hat{x}_{ij} \Leftrightarrow \alpha_{ij}}_{\varphi'_{\mathfrak{T}}}$$

Assim, temos duas subfórmulas φ e $\varphi'_{\mathfrak{T}}$, onde φ é uma fórmula booleana e $\varphi'_{\mathfrak{T}}$ é uma conjunção de átomos sobre \mathfrak{T} . Então, podemos resolver o problema utilizando dois métodos complementares: um resolvidor SAT, responsável pela estrutura booleana, enumerando átomos cuja conjunção deve ser satisfatível, e um resolvidor específico da teoria \mathfrak{T} , que verificará a satisfatibilidade da conjunção de átomos selecionados pelo resolvidor SAT.

Os dois resolvidores se comunicam, indicando a consistência booleana e teórica da fórmula.

Exemplo 3.1 *Seja φ_{EFI} uma fórmula SMT sobre a teoria Equações de Funções Indefinidas:*

$$\varphi_{EFI} = \underbrace{(g(a) = c)}_{x_1} \wedge \underbrace{(f(g(a)) \neq f(c))}_{\neg x_2} \vee \underbrace{(g(a) = d)}_{x_3} \wedge \underbrace{(c \neq d)}_{\neg x_4}$$

Pela abordagem preguiçosa, extraímos uma fórmula booleana φ de φ_{EFI} tal que

$$\varphi = (x_1) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_4)$$

Dada φ como entrada, um resolvidor SAT retorna, por exemplo, o modelo $v(x_1) = 1$, $v(x_2) = 0$ e $v(x_4) = 0$. O resolvidor específico da teoria, então, analisa a conjunção de átomos referentes ao modelo encontrado pelo resolvidor SAT, $g(a) = c \wedge f(g(a)) \neq f(c) \wedge c \neq d$, e retorna *inconsistente*, forçando o resolvidor SAT a procurar outro modelo para φ . Este encontra um novo modelo $v(x_1) = 1$, $v(x_2) = 1$, $v(x_3) = 1$ e $v(x_4) = 0$ e o retorna ao resolvidor da teoria. No entanto, a conjunção $g(a) = c \wedge f(g(a)) = f(c) \wedge g(a) = d \wedge c \neq d$ também é inconsistente. Por fim, não existem mais modelos para φ e, portanto, φ_{EFI} é *insatisfável*.

□

Inferências booleanas ou teóricas também ocorrem nessa abordagem. Por exemplo, o resolvidor da teoria pode informar ao resolvidor SAT que um determinado átomo (literal) deve ser verdadeiro para garantir a satisfatibilidade da fórmula. Aprendizado de novos átomos também pode ser implementado.

Além disso, a combinação de diversas teorias é comumente utilizada na construção de uma fórmula. Isto exige resolvidores específicos a todas as teorias adjacentes, contudo, permitindo modelagens mais expressivas e a utilização de técnicas como o *fortalecimento*, que adiciona à fórmula átomos sobre novas teorias durante sua execução. Essa técnica pode ser utilizada, por exemplo, na representação de funções objetivo e limitantes globais de valores ótimos, úteis nos problemas de otimização.

Alguns exemplos de resolvidores para a teoria Aritmética Linear são apresentados em [15] e [19]. No primeiro, um método Simplex modificado é implementado, utilizando um *tableau* que é atualizado em cada interação com o resolvidor SAT, adicionando e removendo restrições lineares. Já em [19], é apresentado um resolvidor para o domínio dos números inteiros, relaxando a fórmula de entrada e utilizando um resolvidor de Aritmética Linear, como o Simplex, e, por fim, aplicando um método

de ramificação e poda sobre as variáveis valoradas com números não inteiros.

3.3 Considerações

Neste capítulo apresentamos dois problemas de satisfação de restrições sobre fórmulas lógicas: SAT e SMT. Ambas possuem soluções inteligentes que visam contornar o pior caso, que demanda tempo de processamento exponencial. No próximo capítulo, vamos apresentar alguns problemas relacionados a intervalos cuja estrutura e solução possui várias semelhanças com os dois problemas aqui definidos.

Dessa forma, tanto as técnicas implementadas em resolvedores SAT baseados no DPLL quanto os próprios conceitos de valoração, interpretação, modelo e satisfatibilidade podem ser estendidos de forma natural à álgebra intervalar, facilitando e otimizando os algoritmos intervalares.

CAPÍTULO 4

PROBLEMAS INTERVALARES

*“If knowledge can create problems, it is not
through ignorance that we can solve them.”*

(Isaac Asimov)

Uma vez definidos os intervalos, suas operações e a consistência intervalar, somos capazes de construir sistemas de restrições sobre números aproximados, adequando as representações matemáticas dos problemas reais à um formalismo no qual os números têm aproximações dinâmicas. Dessa forma, podemos obter uma versão intervalar de problemas como a Programação Não Linear ou a Satisfatibilidade Módulo Teoria, garantindo que os intervalos dos problemas são sempre consistentes sob alguma consistência intervalar definida.

Neste capítulo, vamos mostrar como a consistência intervalar é capaz de garantir que os intervalos de um conjunto de restrições representem aproximações que satisfaçam essas restrições também no contexto dos números reais e como podemos utilizá-la para excluir aproximações que não pertencem às soluções possíveis. Em seguida, vamos apresentar três problemas intervalares: a Satisfatibilidade Intervalar, na qual queremos encontrar um conjunto de variáveis que satisfaçam uma conjunção de restrições; a Satisfatibilidade Módulo Álgebra Intervalar, na qual a combinação de restrições a ser satisfeita é uma conjunção de disjunções de restrições e, finalmente, a Existência de Valoração Degenerada, na qual queremos garantir que os intervalos definidos contêm aproximações que compõe, de fato, uma solução para o problema nos números reais.

Independente da consistência intervalar aplicada ao problema, podemos facilmente encontrar um conjunto de variáveis que satisfaça uma dada restrição intervalar R . Observe

que se os intervalos não têm restrições sobre seus limitantes ou comprimento, encontrar tal conjunto é trivial. Por exemplo, a restrição $R = (X \asymp Y + Z)$ é satisfeita para $X = Y = Z = \langle [-\infty, \infty] \rangle$.

Na maioria dos casos, estamos interessados na solução com intervalos de comprimento máximo, para não excluir quaisquer valores reais capazes de satisfazer as restrições dadas. No exemplo acima, os intervalos $X = \langle [2, 2] \rangle$, $Y = \langle [1, 1] \rangle$ e $Z = \langle [1, 1] \rangle$ também satisfazem R , no entanto, existe uma infinidade de soluções não representadas por esses intervalos. Nesse caso, a solução $X = Y = Z = \langle [-\infty, \infty] \rangle$ é realmente a mais correta.

Quando a modelagem apresenta restrições sobre os intervalos de forma isolada, não basta encontrar um conjunto qualquer de variáveis que satisfaça R , mas também que satisfaça as restrições sobre cada intervalo.

Exemplo 4.1 *Suponha que, além da restrição $R = (X \asymp Y + Z)$, existam também as seguintes restrições sobre os intervalos:*

$$X \subseteq [2, 5]$$

$$Y \subseteq [-1, 2]$$

$$Z \subseteq [-1, 2]$$

Os multi-intervalos de comprimento máximo que satisfazem as restrições acima são $X = \langle [2, 5] \rangle$, $Y = \langle [-1, 2] \rangle$ e $Z = \langle [-1, 2] \rangle$. No entanto, a restrição $X \asymp Y + Z$ não é satisfeita, pois X não é consistente com o multi-intervalo $Y + Z$ ($[2, 5] \not\subseteq [-2, 4]$). Se X “assumir” qualquer valor maior que 4, não existirão y e z tais que $x = y + z$. Portanto, $x \in X \mid x > 4$ não faz parte do conjunto solução.

Como não podemos expandir o intervalo $Y + Z$, pois Y e Z já são os multi-intervalos de comprimento máximo que satisfazem as restrições sobre os intervalos, temos que refinar X , obtendo $X = \langle [2, 4] \rangle$. Repare que esse é o menor refinamento de X que podemos obter; do contrário, estaríamos perdendo soluções (por exemplo, se $X = \langle [2, 4] \rangle$ perdemos a solução na qual X representa o valor real 4, que satisfaz todas as restrições).

Ainda assim, a restrição R não é satisfeita, pois, embora $X \subseteq Y + Z$, a condição

$Y \subseteq X - Z$ não é verificada ($\langle [-1, 2] \rangle \not\subseteq [0, 5]$). Da mesma forma, temos que escolher um refinamento de Y consistente com $X - Z$. Para $Y = \langle [0, 2] \rangle$ essa restrição é satisfeita, no entanto, $Z \subseteq X - Y$ não o é, forçando o refinamento de Z para $Z = \langle [0, 2] \rangle$.

$$X \subseteq Y + Z \Leftrightarrow \langle [2, 4] \rangle \subseteq \langle [0, 2] + [0, 2] \rangle \Leftrightarrow \langle [2, 4] \rangle \subseteq \langle [0, 4] \rangle$$

$$Y \subseteq X - Z \Leftrightarrow \langle [0, 2] \rangle \subseteq \langle [2, 4] - [0, 2] \rangle \Leftrightarrow \langle [0, 2] \rangle \subseteq \langle [0, 4] \rangle$$

$$Z \subseteq X - Y \Leftrightarrow \langle [0, 2] \rangle \subseteq \langle [2, 4] - [0, 2] \rangle \Leftrightarrow \langle [0, 2] \rangle \subseteq \langle [0, 4] \rangle$$

Portanto, para $X = \langle [2, 4] \rangle$, $Y = \langle [0, 2] \rangle$ e $Z = \langle [0, 2] \rangle$:

$$(X \subseteq [2, 5]) \wedge (Y \subseteq [-1, 2]) \wedge (Z \subseteq [-1, 2]) \wedge (X \asymp Y + Z)$$

□

Pela construção do exemplo, notamos que nenhuma solução foi excluída, pois obtemos cada multi-intervalo excluindo os valores que não pertenciam ao resultado das operações sobre os intervalos restantes. Mais especificamente, sejam $X_0 = [2, 5]$, $Y_0 = [-1, 2]$ e $Z_0 = [-1, 2]$ as restrições sobre os multi-intervalos X , Y e Z (ou os intervalos iniciais). No exemplo anterior, obtemos

$$X = X_0 \setminus (X_0 \setminus (Y_0 + Z_0)) = X_0 \cap (Y_0 + Z_0)$$

$$Y = Y_0 \setminus (Y_0 \setminus (X - Z_0)) = Y_0 \cap (X - Z_0)$$

$$Z = Z_0 \setminus (Z_0 \setminus (X - Y)) = Z_0 \cap (X - Y)$$

Definição 4.1 A função $\mathfrak{C}_c(R, X_0, Y_0, Z_0) = (X, Y, Z)$ associa a cada restrição intervalar $R = (X \asymp Y \circ Z)$ três multi-intervalos $X \subseteq X_0$, $Y \subseteq Y_0$ e $Z \subseteq Z_0$ tais que

$$\begin{aligned}
X &= \begin{cases} X_0 \cap (Y_0 \circ Z_0) & \text{se } \mathfrak{c} = \text{consist\^encia no arco} \\ X_0 \cap (\|Y_0 \circ Z_0\|) & \text{se } \mathfrak{c} = \text{consist\^encia na envolt\^oria} \end{cases} \\
Y &= \begin{cases} Y_0 \cap (X \bullet Z_0) & \text{se } \mathfrak{c} = \text{consist\^encia no arco} \\ Y_0 \cap (\|X \bullet Z_0\|) & \text{se } \mathfrak{c} = \text{consist\^encia na envolt\^oria} \end{cases} \\
Z &= \begin{cases} Z_0 \cap (X \odot Y) & \text{se } \mathfrak{c} = \text{consist\^encia no arco} \\ Z_0 \cap (\|X \odot Y\|) & \text{se } \mathfrak{c} = \text{consist\^encia na envolt\^oria} \end{cases}
\end{aligned}$$

Note que se $\mathfrak{c} = \text{consist\^encia na envolt\^oria}$, ent\^ao os multi-intervalos resultantes (X, Y, Z) s\~ao, de fato, intervalos.

Teorema 4.1 *Dada uma restri\~cao intervalar $R = (X \asymp Y \circ Z)$ e tr\^es multi-intervalos X_0, Y_0 e Z_0 , os intervalos $\mathfrak{C}_{\mathfrak{c}}(R, X_0, Y_0, Z_0) = (X, Y, Z)$ s\~ao tais que*

- *Se $\mathfrak{c} = \text{consist\^encia no arco}$, ent\~ao $X \asymp Y \circ Z$, $A \subseteq X \subseteq X_0$, $B \subseteq Y \subseteq Y_0$ e $C \subseteq Z \subseteq Z_0$, para todo $A \subseteq X_0$, $B \subseteq Y_0$ e $C \subseteq Z_0$ tais que $A \asymp B \circ C$. Isto \^e, X, Y e Z s\~ao os refinamentos de X_0, Y_0 e Z_0 maximais em R .*
- *Se $\mathfrak{c} = \text{consist\^encia na envolt\^oria}$, ent\~ao $X \asymp \|Y \circ Z\|$, $A \subseteq X \subseteq X_0$, $B \subseteq Y \subseteq Y_0$ e $C \subseteq Z \subseteq Z_0$, para todo $A \subseteq X_0$, $B \subseteq Y_0$ e $C \subseteq Z_0$ tais que $A \asymp \|B \circ C\|$. Isto \^e, X, Y e Z s\~ao os refinamentos de X_0, Y_0 e Z_0 maximais em $\|R\|$.*

Prova: A fun\~cao $\mathfrak{C}_{\mathfrak{c}}$ \^e essencialmente equivalente ao usual *contractor* para restri\~coes primitivas [17], que permite restringir os dom\^nios das vari\~aveis da restri\~cao sem excluir solu\~coes, isto \^e, de forma correta e completa [2, 3].

□

No caso espec\^ifico de $X = [0, 0]$, $Y = [-\infty, \infty]$ e $Z = [0, 0]$, a restri\~cao $X \asymp Y \times Z$ expressa, intuitivamente, que $0 = y \cdot 0$, para algum $y \in \mathbb{R}$. No entanto, nos reais, n\~ao existe $y = \frac{0}{0}$, mas nem por isso $0 = y \cdot 0$ deixa de ser verdadeiro. Essa propriedade deve ser mantida nos intervalos e, portanto, como $X \subseteq Y \times Z$, as rela\~coes $Y \subseteq X \div Z$ e $Z \subseteq X \div Y$ devem ser verdadeiras (do contr\~ario, ter\~amos que refinar Z , gerando um intervalo vazio,

ou Y , excluindo valores de um conjunto de soluções possíveis, pois a restrição é claramente verdadeira para todo real). Como $Y \subseteq X \div Z$ deve ser verdadeira para todo Y , então $X \div Z = [0, 0] \div [0, 0] = [-\infty, \infty]$, justificando a definição da divisão por zero, apresentada no capítulo 2.

4.1 Satisfatibilidade Intervalar

No problema SAT queremos encontrar um conjunto de valores booleanos que satisfaça uma dada fórmula também booleana. Podemos interpretar esse problema como uma conjunção de restrições (cláusulas) sobre variáveis booleanas que deve ser satisfeita pelas variáveis definidas. De modo geral, nos problemas de satisfação de restrições, queremos verificar se uma dada conjunção de restrições pode ser satisfeita por alguma valoração às variáveis do problema.

Do mesmo modo, vamos definir um problema no qual queremos encontrar intervalos que satisfaçam uma conjunção de restrições intervalares algébricas e primitivas. Adicionalmente, os intervalos devem ser refinamentos de intervalos iniciais dados e apresentar uma precisão mínima preestabelecida.

Definição 4.2 *Uma fórmula intervalar Φ é uma conjunção de restrições intervalares.*

$$\Phi = \bigwedge_{i=1}^n R_i$$

Assim como nas fórmulas booleanas e SMT, vamos definir que:

- $A(\Phi)$ é o conjunto de restrições (ou átomos) da fórmula Φ ;
- $V(\Phi) = \bigcup_{R \in A(\Phi)} V(R)$ é o conjunto de variáveis da fórmula Φ .

Definição 4.3 *A fórmula $\Phi = \emptyset$ é a fórmula vazia, tal que $A(\Phi) = V(\Phi) = \emptyset$.*

Quando representamos um problema com restrições intervalares, temos que especificar três argumentos principais:

- *conjunto de restrições intervalares*: uma fórmula intervalar Φ que modela as restrições entre os intervalos do problema;
- *conjunto de restrições sobre os intervalos*: uma conjunção de restrições da forma $X \subseteq [a_X, b_X]$ ou $c(X) \leq \epsilon$, onde $X \in V(\Phi)$, $[a_X, b_X]$ é um intervalo inicial e $\epsilon \in \overline{\mathbb{R}}$;
- *consistência intervalar*: definição da satisfação das restrições, que pode ser consistência no arco, se as restrições de Φ devem ser satisfeitas, ou consistência na envoltória, se um relaxamento das restrições de Φ devem ser satisfeitas.

Vamos estender os conceitos de valoração, interpretação e modelo também para variáveis intervalares, afim de facilitar a descrição dos algoritmos posteriores. Assim, temos que uma valoração $v : V(\Phi) \mapsto \mathbb{M}$ é uma função parcial que associa a cada variável de uma dada fórmula Φ no máximo um valor de seu domínio.

Podemos especificar a conjunção de restrições da forma $X \subseteq [a_X, b_X]$ por uma valoração inicial v_0 que associa a cada variável $X \in V(\Phi)$ no máximo um intervalo constante $[a_X, b_X] \in \mathbb{I}$. Além disso, vamos utilizar uma valoração corrente $v : V(\Phi) \mapsto \mathbb{M}$ como resposta da modelagem intervalar, isto é, a definição de cada multi-intervalo $X \in V(\Phi)$ sobre os conjuntos de restrições dados.

Definição 4.4 *Uma valoração v é dita um **refinamento** da valoração v_0 , denotada por $v \subseteq v_0$ se $\forall X \in V(\Phi), v(X) \subseteq v_0(X)$. Nesse caso, também dizemos que v_0 é uma **expansão** de v .*

Para fins de notação, utilizando o conceito de valoração, vamos abreviar por $\mathfrak{C}_\epsilon(R, v)$ a função $\mathfrak{C}_\epsilon(R, v(X), v(Y), v(Z))$ tal que $R = (X \asymp Y \circ Z)$.

A interpretação de uma fórmula intervalar Φ sob um erro ϵ , uma consistência intervalar \mathfrak{c} , uma valoração corrente v e uma inicial v_0 é dada por

$$\mathfrak{I}(\Phi, v, v_0, \epsilon, \mathbf{c}) = \begin{cases} 0 & \text{se } \exists \mathbf{X} \in V(\Phi) \mid v(\mathbf{X}) = \emptyset \\ 1 & \text{se } \forall \mathbf{X} \in V(\Phi) : c(v(\mathbf{X})) \leq \epsilon \text{ e} \\ & \forall R \in A(\Phi) : \mathfrak{I}(R, v, \mathbf{c}) = 1 \text{ e} \\ & v \subseteq v_0 \\ \diamond & \text{c.c.} \end{cases}$$

A interpretação de uma restrição $R = (\mathbf{X} \asymp \mathbf{Y} \circ \mathbf{Z})$ sob uma consistência intervalar \mathbf{c} é dada por

$$\mathfrak{I}(R, v, \mathbf{c}) = \begin{cases} v(\mathbf{X}) \asymp v(\mathbf{Y}) \circ v(\mathbf{Z}) & \text{se } \mathbf{c} = \text{consistência no arco} \\ v(\mathbf{X}) \asymp \|v(\mathbf{Y}) \circ v(\mathbf{Z})\| & \text{se } \mathbf{c} = \text{consistência na envoltória} \end{cases}$$

E, da mesma maneira, uma valoração v é dita um modelo da fórmula Φ com erro ϵ , valoração inicial v_0 e consistência \mathbf{c} se $\mathfrak{I}(\Phi, v, v_0, \epsilon, \mathbf{c}) = 1$.

Definição 4.5 *Uma valoração v é dita **consistente** em \mathbf{c} com a fórmula Φ se $\forall R \in A(\Phi) : \mathfrak{I}(R, v, \mathbf{c}) = 1$.*

Definição 4.6 *O problema computacional da **Satisfatibilidade Intervalar** é dado por:*

Satisfatibilidade Intervalar (SAT \mathbb{I})

Entrada: fórmula intervalar Φ , valoração inicial v_0 , erro de aproximação ϵ e consistência intervalar \mathbf{c} .

Saída: modelo v de Φ , se tal modelo existir, ou **insatisfável**.

O erro $\epsilon \geq 0$ impede que alguma valoração que satisfaça todas as restrições, embora com uma precisão muito pequena, seja uma resposta para o problema. A valoração inicial v_0 serve tanto para restringir o espaço de possíveis soluções para o problema quanto para modelar restrições de limites superior e inferior de cada variável. Por este motivo, a interpretação foi definida de forma que apenas os refinamentos dos intervalos dados como entrada podem ser modelos para a fórmula, pois, do contrário, essas restrições de limitantes não seriam satisfeitas.

Em 1997, van Hentenryck propôs um método de ramificação e poda para o problema SATII [32]. Esse método, descrito pelo algoritmo 2, aplica a consistência intervalar sobre o conjunto de restrições, obtendo uma valoração consistente. Se a valoração obtida não for suficiente para a solução do problema, isto é, algum intervalo tiver comprimento maior que o mínimo estabelecido, então esse intervalo é particionado, gerando dois subproblemas com novas valorações iniciais.

Algoritmo 2: SATII Ramificação e Poda (SATII_{RP})

Entrada: $(\Phi, v_0, \epsilon, \mathbf{c})$

Saída: modelo v de Φ ou *insatisfável*

```

1  para todo  $X \in V(\Phi) \mid v_0(X) = \diamond$  faça
2  |    $v_0 \leftarrow v_0 \cup (X \mapsto [-\infty, \infty])$ 
3  fim
4   $v \leftarrow \text{consistencia}(\Phi, v_0, \mathbf{c})$ 
5  se  $\mathfrak{J}(\Phi, v, v_0, \epsilon, \mathbf{c}) = 1$  então
6  |   retorna  $v$ 
7  fim
8  se  $\mathfrak{J}(\Phi, v, v_0, \epsilon, \mathbf{c}) = 0$  então
9  |   retorna insatisfável
10 fim
11  $X \leftarrow \text{escolhe}(V(\Phi), v, \epsilon)$ 
12  $(v', v'') \leftarrow \text{ramifica}(X, v)$ 
13  $\phi \leftarrow \text{SATII}_{RP}(\Phi, v', \epsilon, \mathbf{c})$ 
14 se  $\phi \neq \text{insatisfável}$  então
15 |   retorna  $\phi$ 
16 fim
17 retorna  $\text{SATII}_{RP}(\Phi, v'', \epsilon, \mathbf{c})$ 

```

Primeiramente, todos os intervalos não valorados são valorados com $[-\infty, \infty]$, repre-

sentando qualquer valor real. Em seguida, obtemos um refinamento v da valoração inicial v_0 aplicando a consistência intervalar sobre a fórmula Φ . A valoração v é obtida pelo método $consistencia(\Phi, v_0, \mathfrak{c})$, descrito simplificadamente pelo algoritmo 3. Esse método aplica a função $\mathfrak{C}_\mathfrak{c}$ em toda restrição $R \in A(\Phi)$. Como o refinamento de um multi-intervalo X pode tornar as demais restrições insatisfatíveis, temos que repetir o processo continuamente, até obter uma valoração consistente sobre todas as restrições. Embora o algoritmo 3 demande um tempo de processamento linear no tamanho da fórmula se cada variável aparecer em apenas uma restrição [2], sua execução pode não terminar para algumas classes de fórmulas intervalares [10]. Em [16] foram propostas algumas regras de propagação de intervalos capazes de evitar o tempo de processamento infinito para a grande maioria das fórmulas e em [18] encontramos uma nova consistência intervalar (*consistência de caixa*), ainda mais forte que a consistência no arco, implementada através de ramificações e validada em diversos problemas amostra (*toy problems*).

Algoritmo 3: Consistência Intervalar (*consistencia*)

Entrada: $(\Phi, v_0, \mathfrak{c})$

Saída: valoração v consistente com Φ

```

1  $v \leftarrow v_0$ 
2 para todo  $R = (X \asymp Y \circ Z) \in A(\Phi) \mid \mathfrak{J}(R, v, \mathfrak{c}) = 0$  faça
3    $(X', Y', Z') \leftarrow \mathfrak{C}_\mathfrak{c}(R, v)$ 
4    $v \leftarrow v \cup (X \mapsto X')$ 
5    $v \leftarrow v \cup (Y \mapsto Y')$ 
6    $v \leftarrow v \cup (Z \mapsto Z')$ 
7 fim
8 retorna  $v$ 

```

Em seguida verificamos se a valoração encontrada v é um modelo para Φ . Caso não seja, e nem torne a fórmula insatisfatível, escolhemos uma variável $X \in V(\Phi)$ tal que $c(v(X)) \geq \epsilon$, pelo procedimento $escolhe(V(\Phi), v, \epsilon)$.

Finalmente, pelo procedimento $ramifica(X, v)$, geramos duas valorações v' e v'' tais

que, se $v(\mathbf{X}) \in \mathbb{I}$ e seus limitantes inferior e superior são, respectivamente, \underline{x} e \bar{x} , então

$$v' = \begin{cases} v \cup (\mathbf{X} \mapsto [\underline{x}, k]) & \text{se } \tau_e(v(\mathbf{X})) = 1 \\ v \cup (\mathbf{X} \mapsto (\underline{x}, k]) & \text{se } \tau_e(v(\mathbf{X})) = 0 \end{cases}$$

$$v'' = \begin{cases} v \cup (\mathbf{X} \mapsto (k, \bar{x}]) & \text{se } \tau_d(v(\mathbf{X})) = 1 \\ v \cup (\mathbf{X} \mapsto (k, \bar{x})) & \text{se } \tau_d(v(\mathbf{X})) = 0 \end{cases}$$

Geralmente, k é obtido calculando-se a mediana do intervalo $v(\mathbf{X})$ ou utilizando o método de Newton [2].

Se $v(\mathbf{X}) \notin \mathbb{I}$, então $v(\mathbf{X}) = \langle X_1, \dots, X_n \rangle$ e as valorações geradas são tais que

$$v' = v \cup (\mathbf{X} \mapsto \langle X_1, \dots, X_k \rangle)$$

$$v'' = v \cup (\mathbf{X} \mapsto \langle X_{k+1}, \dots, X_n \rangle)$$

onde $k = \lfloor \frac{n}{2} \rfloor$.

O algoritmo 2 compartilha a mesma estrutura do DPLL (algoritmo 1), diferenciando nos seguintes aspectos:

- Enquanto no DPLL uma nova valoração pode ser inferida por cláusula unitária e literal puro, no $\text{SAT}\mathbb{I}_{RP}$ ela pode ser inferida aplicando-se a consistência intervalar;
- No $\text{SAT}\mathbb{I}_{RP}$, mesmo as variáveis já valoradas podem ser escolhidas para realizar a ramificação, o que não ocorre no DPLL;
- O método $\text{ramifica}(\mathbf{X}, v)$ decide qual o conjunto imagem para a variável \mathbf{X} sobre a valoração v . No DPLL, esse método não se faz necessário, pois, para qualquer variável booleana, o conjunto imagem é fixo (0 ou 1).

Como o algoritmo 3 é completo, pela completude da função \mathfrak{C}_c (teorema 4.1), e uma árvore de busca é construída pela ramificação de forma que todos os ramos são executados recursivamente, então o algoritmo proposto por van Hentenryck também é completo.

No entanto, se o erro ϵ dado na entrada for nulo, não é garantido que a execução de $\text{SAT}\mathbb{I}_{RP}$ termine, pois o método de ramificação, geralmente pelo ponto médio do intervalo, é incapaz de gerar intervalos degenerados.

Além deste algoritmo, em [28] é apresentada uma versão do método Simplex capaz de trabalhar com variáveis intervalares. Ainda em [24], temos um detalhamento mais conciso do método de ramificação e poda aplicado ao problema da Satisfatibilidade Intervalar.

4.2 Satisfatibilidade Módulo Álgebra Intervalar

Além da satisfatibilidade de fórmulas intervalares, existe a necessidade da representação de determinados problemas, como sistemas híbridos ou máquinas de estados, por uma combinação complexa de restrições intervalares. Tais fórmulas podem ser escritas como uma conjunção de disjunções de restrições por métodos de transformação na forma normal clausal [29]. Como já definido, essa representação é chamada Satisfatibilidade Módulo Teoria; nesse caso, a teoria Álgebra Intervalar.

Existem resolvedores SMT que implementam a abordagem preguiçosa e são capazes de encontrar modelos de tais fórmulas [6, 12]. Técnicas como a *ramificação em demanda* [1] ou *ramificação dinâmica* [22] são úteis na resolução de tais problemas. No entanto, além de que tais resolvedores são especialistas em problemas no domínio dos reais (e não especificamente intervalares), o método iSAT, apresentado em [17], obteve melhores resultados em comparação com alguns resolvedores SMT da época. Essa terceira abordagem para resolver SMT consiste na integração completa do resolvedor SAT com o resolvedor da teoria, gerenciando átomos e intervalos.

Inspirados pela semelhança entre os algoritmos DPLL e $\text{SAT}\mathbb{I}_{RP}$, Fränzle, Herde e Teige propuseram, em [17], um método para encontrar modelos de fórmulas SMT sobre a teoria Álgebra Intervalar, incluindo funções transcendentais.

Dada uma fórmula SMT sobre restrições intervalares $\varphi_{\mathbb{I}}$, o algoritmo 4 implementa uma ramificação e poda baseada no DPLL, propagando os intervalos de variáveis decididas e inferidas. Esse processo integra uma enumeração de possíveis problemas de Satisfatibilidade Intervalar (semelhante ao que os resolvedores SAT fazem) e a verificação

da satisfatibilidade de cada problema enumerado (como no algoritmo 2).

Como se trata de uma fórmula SMT, a interpretação de $\varphi_{\mathbb{I}}$ é diferente da apresentada neste capítulo para fórmulas intervalares, assemelhando-se mais à interpretação de fórmulas SMT:

$$\mathfrak{I}(\varphi_{\mathbb{I}}, v, v_0, \epsilon, \mathbf{c}) = \begin{cases} 0 & \text{se } \exists \omega^{\mathbb{I}} \in C(\varphi_{\mathbb{I}}) \mid \forall R \in A(\omega^{\mathbb{I}}) : \exists \mathbf{X} \in V(R) \text{ tal que } v(\mathbf{X}) = \emptyset \\ 1 & \text{se } \forall \omega^{\mathbb{I}} \in C(\varphi_{\mathbb{I}}) : \exists R \in \omega^{\mathbb{I}} \mid \mathfrak{I}(R, v, \mathbf{c}) = 1 \text{ e } \forall \mathbf{X} \in V(R) : \\ & c(v(\mathbf{X})) \leq \epsilon \text{ e } v(\mathbf{X}) \subseteq v_0(\mathbf{X}) \\ \diamond & \text{c.c.} \end{cases}$$

A consistência intervalar \mathbf{c} utilizada no iSAT é a consistência na envoltória e, portanto, podemos definir todas as variáveis como intervalos, não sendo necessário a utilização de multi-intervalos.

Algoritmo 4: iSAT

Entrada: $(\varphi_{\mathbb{I}}, v_0, \epsilon)$
Saída: modelo v de $\varphi_{\mathbb{I}}$ ou *insatisfatível*

```

1  para todo  $X \in V(\varphi_{\mathbb{I}}) \mid v_0(X) = \diamond$  faça
2  |    $v_0 \leftarrow v_0 \cup (X \mapsto [-\infty, \infty])$ 
3  fim
4   $\mathbf{c} \leftarrow$  consistência na envoltória
5   $v \leftarrow v_0$ 
6  para todo  $\omega^{\mathbb{I}} \in C(\varphi_{\mathbb{I}}) \mid$  unitaria $(\omega^{\mathbb{I}}, v, \mathbf{c}) = R = (X \succcurlyeq Y \circ Z)$  e  $\mathfrak{J}(R, v, \mathbf{c}) = 0$  faça
7  |    $(X', Y', Z') \leftarrow \mathfrak{C}_{\mathbf{c}}(R, v)$ 
8  |    $v \leftarrow v \cup (X \mapsto X')$ 
9  |    $v \leftarrow v \cup (Y \mapsto Y')$ 
10 |   $v \leftarrow v \cup (Z \mapsto Z')$ 
11 fim
12 se  $\mathfrak{J}(\varphi_{\mathbb{I}}, v, v_0, \epsilon, \mathbf{c}) = 1$  então
13 |  retorna  $v$ 
14 fim
15 se  $\mathfrak{J}(\varphi_{\mathbb{I}}, v, v_0, \epsilon, \mathbf{c}) = 0$  então
16 |  retorna insatisfatível
17 fim
18  $X \leftarrow$  escolhe $(V(\varphi_{\mathbb{I}}), v, \epsilon)$ 
19  $(v', v'') \leftarrow$  ramifica $(X, v)$ 
20  $\phi \leftarrow$  iSAT $(\varphi_{\mathbb{I}}, v', \epsilon)$ 
21 se  $\phi \neq$  insatisfatível então
22 |  retorna  $\phi$ 
23 fim
24 retorna iSAT $(\varphi_{\mathbb{I}}, v'', \epsilon)$ 

```

Esse algoritmo adiciona ao $\text{SAT}_{\mathbb{I}RP}$ uma verificação de cláusula unitária, como ocorre no DPLL. A definição de cláusula unitária nesse contexto difere da definição usual. Dada

uma cláusula sobre átomos intervalares $\omega^{\mathbb{I}}$ e uma valoração v , a função $unitaria(\omega^{\mathbb{I}}, v, \mathbf{c})$ verifica se existe no máximo um átomo em $\omega^{\mathbb{I}}$ sob a valoração v no qual, aplicada a consistência intervalar, não são gerados intervalos vazios.

$$unitaria(\omega^{\mathbb{I}}, v, \mathbf{c}) = \begin{cases} R & \text{se } \exists R \in A(\omega^{\mathbb{I}}) \mid \forall S \in A(\omega^{\mathbb{I}}) \setminus \{R\} : \mathfrak{C}_c(S, v) = (\emptyset, \emptyset, \emptyset) \\ \diamond & \text{c.c.} \end{cases}$$

A cada passo o algoritmo verifica se existe alguma cláusula unitária $\omega^{\mathbb{I}} \in C(\varphi_{\mathbb{I}})$. Se existir, então ou existe um único átomo R não falso sob a valoração v em $\omega^{\mathbb{I}}$ (o qual deve ser satisfeito, aplicando-se a função \mathfrak{C}_c e atualizando a valoração v) ou todos os átomos são falsos sob v . Nesse último caso, forçamos a aplicação da consistência intervalar em algum átomo falso de $\omega^{\mathbb{I}}$, gerando intervalos vazios. Assim, o condicional da linha 13 verificará que a fórmula é insatisfatível, pois existe uma cláusula cujos átomos são todos falsos. A completude desse corte é garantida pelo fato de que tanto a função \mathfrak{C}_c quanto a ramificação de intervalos gera refinamentos dos intervalos originais e, assim, nunca seria possível encontrar uma nova valoração consistente. É importante observar que aplicar a consistência sobre qualquer restrição R pode tornar outros átomos, mesmo que de outras cláusulas, inconsistentes, sendo necessário nova verificação de cláusulas unitárias.

A estratégia de aplicar a consistência apenas sobre cláusulas unitárias garante excluir alguns átomos durante a execução do algoritmo, enumerando conjunções de restrições candidatas a um modelo. Além disso, se uma cláusula não é unitária, a aplicação da função \mathfrak{C}_c pode excluir soluções, perdendo a completude do algoritmo.

Assim como no DPLL, esse algoritmo é implementado utilizando técnicas de aprendizado e retrocesso não cronológico. Quando um conflito é encontrado, os átomos conflitantes são analisados (utilizando o grafo de implicação) e uma nova cláusula é aprendida, desfazendo as valorações mais recentes até que essa cláusula se torne verdadeira. Essa integração permite utilizar as várias técnicas implementadas em resolvedores SAT, ganhando desempenho em comparação aos resolvedores puramente SMT.

Exemplo 4.2 Seja $(\varphi_{\mathbb{I}}, v_0, \epsilon)$ uma instância SMT tal que $\varphi_{\mathbb{I}} = \omega_1^{\mathbb{I}} \wedge \omega_2^{\mathbb{I}} \wedge \omega_3^{\mathbb{I}} \wedge \omega_4^{\mathbb{I}} \wedge \omega_5^{\mathbb{I}}$:

$$\omega_1^{\mathbb{I}} = (X \asymp H_1 + Z \quad \vee \quad Y \asymp H_2 + Z \quad \vee \quad B \asymp C_1 + Z)$$

$$\omega_2^{\mathbb{I}} = (B \asymp C_2 + Z \quad \vee \quad A_1 \asymp X \wedge C_3)$$

$$\omega_3^{\mathbb{I}} = (B \asymp C_2 + Z \quad \vee \quad A_2 \asymp C_4 \times Y)$$

$$\omega_4^{\mathbb{I}} = (B \asymp C_2 + Z \quad \vee \quad A_3 \asymp A_1 + A_2)$$

$$\omega_5^{\mathbb{I}} = (B \asymp C_2 + Z \quad \vee \quad H_3 \asymp A_3 + Z)$$

O erro ϵ é qualquer e a valoração inicial v_0 é dada por:

$$\begin{array}{lll} v_0(X) & = & [-2, \infty] & v_0(H_2) & = & [-\infty, 0] & v_0(C_2) & = & [0, 0] \\ v_0(Y) & = & [4, \infty] & v_0(H_3) & = & [6.2, \infty] & v_0(C_3) & = & [2, 2] \\ v_0(H_1) & = & (4, \infty] & v_0(C_1) & = & [1, 1] & v_0(C_4) & = & [-2, -2] \\ & & & & & & v_0(Z) & = & [0, 0] \end{array}$$

Pela execução de $\text{iSAT}(\varphi_{\mathbb{I}}, v_0, \epsilon)$, inicialmente, valoramos todas as variáveis sem valor definido: $v_0(B) = v_0(A_1) = v_0(A_2) = v_0(A_3) = [-\infty, \infty]$. Em seguida, definimos uma valoração $v = v_0$ (linha 5) e iniciamos o processo de verificação de cláusula unitária.

Na cláusula $\omega_1^{\mathbb{I}}$ temos que $\text{unitaria}(\omega_1^{\mathbb{I}}, v, \mathbf{c}) = \diamond$, pois existem duas restrições nas quais a aplicação da consistência intervalar não gera intervalos vazios:

$$\begin{aligned} \mathfrak{C}_{\mathbf{c}}(X \asymp H_1 + Z, v) &= \left(v(X) \cap \|v(H_1) + v(Z)\| = [-2, \infty] \cap \|(4, \infty] + [0, 0]\| = (4, \infty], \right. \\ &\quad v(H_1) \cap \|(4, \infty] - v(Z)\| = (4, \infty] \cap \|(4, \infty] - [0, 0]\| = (4, \infty], \\ &\quad \left. v(Z) \cap \|(4, \infty] - (4, \infty]\| = [0, 0] \cap \|(4, \infty] - (4, \infty]\| = [0, 0] \right) \\ \mathfrak{C}_{\mathbf{c}}(Y \asymp H_2 + Z, v) &= \left(v(Y) \cap \|v(H_2) + v(Z)\| = [4, \infty] \cap \|[-\infty, 0] + [0, 0]\| = \emptyset, \right. \\ &\quad v(H_2) \cap \|\emptyset - v(Z)\| = [-\infty, 0] \cap \|\emptyset - [0, 0]\| = \emptyset, \\ &\quad \left. v(Z) \cap \|\emptyset - \emptyset\| = [0, 0] \cap \|\emptyset - \emptyset\| = \emptyset \right) \end{aligned}$$

$$\begin{aligned}
\mathfrak{C}_{\mathbf{c}}(B \asymp C_1 + Z, v) &= \left(v(B) \cap \|v(C_1) + v(Z)\| = [-\infty, \infty] \cap \|[1, 1] + [0, 0]\| = [1, 1], \right. \\
&\quad v(C_1) \cap \|[1, 1 - v(Z)]\| = [1, 1] \cap \|[1, 1] - [0, 0]\| = [1, 1], \\
&\quad \left. v(Z) \cap \|[1, 1] - [1, 1]\| = [0, 0] \cap \|[1, 1] - [1, 1]\| = [0, 0] \right)
\end{aligned}$$

O mesmo ocorre nas demais cláusulas, pois, omitindo os cálculos intermediários, obtemos:

$$\begin{aligned}
\mathfrak{C}_{\mathbf{c}}(B \asymp C_2 + Z, v) &= \left([0, 0], [0, 0], [0, 0] \right) \\
\mathfrak{C}_{\mathbf{c}}(A_1 \asymp X \wedge C_3, v) &= \left([0, \infty], [-2, \infty], [2, 2] \right) \\
\mathfrak{C}_{\mathbf{c}}(A_2 \asymp C_4 \times Y, v) &= \left([-\infty, -8], [-2, -2], [4, \infty] \right) \\
\mathfrak{C}_{\mathbf{c}}(A_3 \asymp A_1 + A_2, v) &= \left([-\infty, \infty], [-\infty, \infty], [-\infty, \infty] \right) \\
\mathfrak{C}_{\mathbf{c}}(H_3 \asymp A_3 + Z, v) &= \left([6.2, \infty], [6.2, \infty], [0, 0] \right)
\end{aligned}$$

Dessa forma, nenhuma cláusula é unitária e a execução de $\text{iSAT}(\varphi_{\mathbb{I}}, v_0, \epsilon)$ segue para a verificação da interpretação da fórmula sob a valoração corrente v . Como todos os átomos da cláusula $\omega_2^{\mathbb{I}}$, por exemplo, não são satisfeitos, pois $v(B) \not\subseteq \|v(C_2) + v(Z)\|$ e $v(A_1) \not\subseteq \|v(X) \wedge v(C_3)\|$ então $\mathfrak{J}(\varphi_{\mathbb{I}}, v, v_0, \epsilon, \mathbf{c}) \neq 1$. Por outro lado, como $\forall X \in V(\varphi_{\mathbb{I}}) : v(X) \neq \emptyset$, então $\mathfrak{J}(\varphi_{\mathbb{I}}, v, v_0, \epsilon, \mathbf{c}) \neq 0$.

Finalmente, suponha que escolhemos a variável X e geramos duas novas valorações $v' = v \cup (X \mapsto [-2, 3])$ e $v'' = v \cup (X \mapsto (3, \infty])$, executando uma nova instância $\text{iSAT}(\varphi_{\mathbb{I}}, v', \epsilon)$.

Nessa nova execução, definimos $v = v'$ e iniciamos novamente o processo de verificação de cláusula unitária.

Agora, a cláusula $\omega_1^{\mathbb{I}}$ é unitária e $\text{unitaria}(\omega_1^{\mathbb{I}}, v, \mathbf{c}) = (B \asymp C_1 + Z)$, pois todas as

outras restrições geram intervalos vazios com a aplicação da consistência intervalar:

$$\begin{aligned}\mathfrak{C}_c(X \asymp H_1 + Z, v) &= (\emptyset, \emptyset, \emptyset) \\ \mathfrak{C}_c(Y \asymp H_2 + Z, v) &= (\emptyset, \emptyset, \emptyset) \\ \mathfrak{C}_c(B \asymp C_1 + Z, v) &= ([1, 1], [1, 1], [0, 0])\end{aligned}$$

Pelo algoritmo, entramos no laço da linha 7 e atualizamos a valoração corrente, obtendo

$$\begin{array}{lll} v(X) &= [-2, 3] & v(H_3) &= [6.2, \infty] & v(A_1) &= [-\infty, \infty] \\ v(Y) &= [4, \infty] & v(C_1) &= [1, 1] & v(A_2) &= [-\infty, \infty] \\ v(B) &= [1, 1] & v(C_2) &= [0, 0] & v(A_3) &= [-\infty, \infty] \\ v(H_1) &= (4, \infty] & v(C_3) &= [2, 2] & v(Z) &= [0, 0] \\ v(H_2) &= [-\infty, 0] & v(C_4) &= [-2, -2]\end{array}$$

Note que a única forma de $\varphi_{\mathbb{I}}$ ser satisfatível é que pelo menos um átomo de cada cláusula seja satisfeito por intervalos não vazios. Como apenas a restrição $B \asymp C_1 + Z$ da cláusula $\omega_1^{\mathbb{I}}$ poderia ser satisfeita por intervalos não vazios, tornamos ela verdadeira, aplicando a consistência intervalar e obtendo uma nova valoração consistente.

Ainda no laço, verificamos que $unitaria(\omega_2^{\mathbb{I}}, v, \mathfrak{c}) = (A_1 \asymp X \wedge C_3)$, pois

$$\begin{aligned}\mathfrak{C}_c(B \asymp C_2 + Z, v) &= (\emptyset, \emptyset, \emptyset) \\ \mathfrak{C}_c(A_1 \asymp X \wedge C_3, v) &= ([0, 9], [-2, 3], [2, 2])\end{aligned}$$

Novamente atualizamos a valoração, de forma que $v(A_1) = [0, 9]$. Como $A_1 \notin V(\omega_1^{\mathbb{I}})$, então as duas primeiras cláusulas continuam satisfeitas por v . Agora a cláusula $\omega_3^{\mathbb{I}}$ é unitária e $unitaria(\omega_3^{\mathbb{I}}, v, \mathfrak{c}) = (A_2 \asymp C_4 \times Y)$, pois

$$\begin{aligned}\mathfrak{C}_c(B \asymp C_2 + Z, v) &= (\emptyset, \emptyset, \emptyset) \\ \mathfrak{C}_c(A_2 \asymp C_4 \times Y, v) &= ([-\infty, -8], [-2, -2], [4, \infty])\end{aligned}$$

Portanto, $v(A_2) = [-\infty, -8]$ e a próxima cláusula também torna-se unitária, pois

$$\text{unitaria}(\omega_4^{\mathbb{I}}, v, \mathbf{c}) = (A_3 \asymp A_1 + A_2).$$

$$\begin{aligned}\mathfrak{C}_{\mathbf{c}}(B \asymp C_2 + Z, v) &= (\emptyset, \emptyset, \emptyset) \\ \mathfrak{C}_{\mathbf{c}}(A_3 \asymp A_1 + A_2, v) &= ([-\infty, 1], [0, 9], [-\infty, -8])\end{aligned}$$

Finalmente, obtemos a seguinte valoraçãõ, sob a qual as cláusulas $\omega_1^{\mathbb{I}}$, $\omega_2^{\mathbb{I}}$, $\omega_3^{\mathbb{I}}$ e $\omega_4^{\mathbb{I}}$ são satisfeitas.

$$\begin{array}{lll} v(X) & = & [-2, 3] & v(H_3) & = & [6.2, \infty] & v(A_1) & = & [0, 9] \\ v(Y) & = & [4, \infty] & v(C_1) & = & [1, 1] & v(A_2) & = & [-\infty, -8] \\ v(B) & = & [1, 1] & v(C_2) & = & [0, 0] & v(A_3) & = & [-\infty, 1] \\ v(H_1) & = & (4, \infty] & v(C_3) & = & [2, 2] & v(Z) & = & [0, 0] \\ v(H_2) & = & [-\infty, 0] & v(C_4) & = & [-2, -2] & & & \end{array}$$

A cláusula $\omega_5^{\mathbb{I}}$ restante também é unitária, pois

$$\begin{aligned}\mathfrak{C}_{\mathbf{c}}(B \asymp C_2 + Z, v) &= (\emptyset, \emptyset, \emptyset) \\ \mathfrak{C}_{\mathbf{c}}(H_3 \asymp A_3 + Z, v) &= (\emptyset, \emptyset, \emptyset)\end{aligned}$$

No entanto, todas as restrições geram intervalos vazios. Se escolhessemos qualquer variável e realizássemos uma ramificação, continuaríamos com esse conflito. Mas como aqui $\omega_5^{\mathbb{I}}$ é unitária então o algoritmo valora, por exemplo, $v(A_3) = \emptyset$, $v(C_4) = \emptyset$ e $v(C_2) = \emptyset$.

Com essa nova valoraçãõ as demais cláusulas tornam-se unitárias e, ao final, todas as variáveis são valoradas vazias. Finalmente, saímos do laço (pois $\forall R \in A(\varphi_{\mathbb{I}}) : \mathfrak{J}(R, v, \mathbf{c}) = 1$) e $\text{iSAT}(\varphi_{\mathbb{I}}, v', \epsilon)$ retorna *insatisfatível*.

Voltando na execução de $\text{iSAT}(\varphi_{\mathbb{I}}, v_0, \epsilon)$ temos que $\phi = \text{iSAT}(\varphi_{\mathbb{I}}, v', \epsilon) = \text{insatisfatível}$ e o algoritmo retorna $\text{iSAT}(\varphi_{\mathbb{I}}, v'', \epsilon)$, cuja valoraçãõ inicial é dada por $v_0 = v''$ tal que

$$\begin{array}{lll}
v_0(X) & = & (3, \infty] \\
v_0(Y) & = & [4, \infty] \\
v_0(B) & = & [-\infty, \infty] \\
v_0(H_1) & = & (4, \infty] \\
v_0(H_2) & = & [-\infty, 0] \\
v_0(H_3) & = & [6.2, \infty] \\
v_0(C_1) & = & [1, 1] \\
v_0(C_2) & = & [0, 0] \\
v_0(C_3) & = & [2, 2] \\
v_0(C_4) & = & [-2, -2] \\
v_0(A_1) & = & [-\infty, \infty] \\
v_0(A_2) & = & [-\infty, \infty] \\
v_0(A_3) & = & [-\infty, \infty] \\
v(Z) & = & [0, 0]
\end{array}$$

Nessa execução, nenhuma cláusula é unitária, pois

$$\begin{aligned}
\mathfrak{C}_c(X \asymp H_1 + Z, v) &= \left((4, \infty], (4, \infty], [0, 0] \right) \\
\mathfrak{C}_c(Y \asymp H_2 + Z, v) &= \left(\emptyset, \emptyset, \emptyset \right) \\
\mathfrak{C}_c(B \asymp C_1 + Z, v) &= \left([1, 1], [1, 1], [0, 0] \right) \\
\mathfrak{C}_c(B \asymp C_2 + Z, v) &= \left([0, 0], [0, 0], [0, 0] \right) \\
\mathfrak{C}_c(A_1 \asymp X \wedge C_3, v) &= \left([0, \infty], (3, \infty], [2, 2] \right) \\
\mathfrak{C}_c(A_2 \asymp C_4 \times Y, v) &= \left([-\infty, -8], [-2, -2], [4, \infty] \right) \\
\mathfrak{C}_c(A_3 \asymp A_1 + A_2, v) &= \left([-\infty, \infty], [-\infty, \infty], [-\infty, \infty] \right) \\
\mathfrak{C}_c(H_3 \asymp A_3 + Z, v) &= \left([6.2, \infty], [6.2, \infty], [0, 0] \right)
\end{aligned}$$

E ainda $\mathfrak{J}(\varphi_{\mathbb{I}}, v, v'', \epsilon, \mathfrak{c}) = \diamond$, realizando mais uma ramificação. Nesse caso, suponha que X seja novamente escolhida e obtemos duas valorações $v' = v_0 \cup (X \mapsto (3, 4])$ e $v'' = v_0 \cup (X \mapsto (4, \infty])$, executando $\text{iSAT}(\varphi_{\mathbb{I}}, v', \epsilon)$.

Finalmente, a primeira cláusula torna-se unitária e valoramos $v(B) = [1, 1]$.

$$\begin{aligned}
\mathfrak{C}_c(X \asymp H_1 + Z, v) &= \left(\emptyset, \emptyset, \emptyset \right) \\
\mathfrak{C}_c(Y \asymp H_2 + Z, v) &= \left(\emptyset, \emptyset, \emptyset \right) \\
\mathfrak{C}_c(B \asymp C_1 + Z, v) &= \left([1, 1], [1, 1], [0, 0] \right)
\end{aligned}$$

Como $\mathfrak{C}_c(B \asymp C_2 + Z) = \left(\emptyset, \emptyset, \emptyset \right)$, as demais cláusulas tornam-se unitárias e atualizamos

a valoração corrente:

$$\begin{array}{lll}
v(A_1) = [0, 16] & \text{pois} & \mathfrak{C}_t(A_1 \asymp X \wedge C_3, v) = ([0, 16], (3, 4], [2, 2]) \\
v(A_2) = [-\infty, -8] & \text{pois} & \mathfrak{C}_t(A_2 \asymp C_4 \times Y, v) = ([-\infty, -8], [-2, -2], [4, \infty]) \\
v(A_3) = [-\infty, 8] & \text{pois} & \mathfrak{C}_t(A_3 \asymp A_1 + A_2, v) = ([-\infty, 8], [0, 16], [-\infty, -8]) \\
v(A_3) = [6.2, 8] & \text{pois} & \mathfrak{C}_t(H_3 \asymp A_3 + Z, v) = ([6.2, 8], [6.2, 8], [0, 0])
\end{array}$$

Embora tenhamos atualizado a valoração para toda cláusula unitária, a cláusula ω_5^{II} inferiu um novo valor à variável A_3 , previamente valorada. Assim, ainda podem existir cláusulas unitárias cuja interpretação é falsa. Observando novamente as cláusulas unitárias, temos mais duas atualizações em v :

$$\begin{array}{ll}
v(A_1) = [14.2, 16] \text{ e } v(A_2) = [-9.8, 8] & \\
\text{pois } \mathfrak{C}_t(A_3 \asymp A_1 + A_2, v) = ([6.2, 8], [0, 16], [-9.8, 8]) & \\
v(Y) = [-4, 4.9] & \text{pois } \mathfrak{C}_t(A_2 \asymp C_4 \times Y, v) = ([-9.8, 8], [-2, -2], [-4, 4.9])
\end{array}$$

Agora, temos uma valoração v sobre a qual todas as cláusulas são unitárias, mas seus átomos satisfeitos. Portanto, saímos do laço e verificamos a interpretação da fórmula φ_{II} sob a valoração corrente dada por

$$\begin{array}{lll}
v(X) = (3, 4] & v(H_3) = [6.2, 8] & v(A_1) = [14.2, 16] \\
v(Y) = [-4, 4.9] & v(C_1) = [1, 1] & v(A_2) = [-9.8, 8] \\
v(B) = [1, 1] & v(C_2) = [0, 0] & v(A_3) = [6.2, 8] \\
v(H_1) = (4, \infty] & v(C_3) = [2, 2] & v(Z) = [0, 0] \\
v(H_2) = [-\infty, 0] & v(C_4) = [-2, -2] &
\end{array}$$

Se o erro dado na entrada é $\epsilon = \infty$, essa valoração é um modelo para a fórmula e o algoritmo retorna v , do contrário, ramificamos alguma variável e prosseguimos com o método.

□

Na prática, esse algoritmo é implementado de maneira não recursiva, utilizando uma pilha para as valorações e um grafo de implicação responsável por podar ramos inconsistentes. Com isso obtém-se uma eficiência superior ao algoritmo recursivo.

Embora aqui apresentamos o iSAT sobre restrições e valorações intervalares, em [17] uma outra abordagem é apresentada, na qual os intervalos são substituídos por variáveis reais limitadas por restrições da fórmula. Nesse caso, valorar uma variável x é equivalente a adicionar uma nova restrição da forma $x \bowtie c$ à fórmula original, onde $x \in \mathbb{R}$, c é uma constante e $\bowtie \in \{<, \leq, =, \geq, >\}$.

O exemplo anterior é apresentado em [17] da seguinte forma: dada uma fórmula $\varphi_{\mathbb{R}} = (x > 4 \vee y \leq 0 \vee x^2 - 2y \geq 6.2)$, queremos encontrar limitantes a_x, b_x, a_y e b_y tais que $b_x - a_x \leq \epsilon$, $b_y - a_y \leq \epsilon$ e $\varphi_{\mathbb{R}}$ seja verdadeira para algum $x, y \in \mathbb{R}$. Da mesma forma, o iSAT utiliza restrições primitivas e, portanto, substitui $\varphi_{\mathbb{R}}$ por uma fórmula $\varphi'_{\mathbb{R}} = \omega_1^{\mathbb{R}} \wedge \omega_2^{\mathbb{R}} \wedge \omega_3^{\mathbb{R}} \wedge \omega_4^{\mathbb{R}} \wedge \omega_5^{\mathbb{R}}$ equivalente, tal que

$$\omega_1^{\mathbb{R}} = (x > 4 \vee y \leq 0 \vee b)$$

$$\omega_2^{\mathbb{R}} = (\neg b \vee h_1 = x^2)$$

$$\omega_3^{\mathbb{R}} = (\neg b \vee h_2 = -2y)$$

$$\omega_4^{\mathbb{R}} = (\neg b \vee h_3 = h_1 + h_2)$$

$$\omega_5^{\mathbb{R}} = (\neg b \vee h_3 \geq 6.2)$$

onde $b \in \{0, 1\}$ é uma variável booleana (que pode ser tratada como real se reescrevermos b por $b \geq 1$ e $\neg b$ por $b \leq 0$).

Conseguimos representar esse mesmo problema em SATII definindo intervalos constantes C_1, C_2, C_3 e C_4 tais que $v_0(C_1) = [1, 1]$, $v_0(C_2) = [0, 0]$, $v_0(C_3) = [2, 2]$ e $v_0(C_4) = [-2, -2]$ e intervalos H_1, H_2 e H_3 que substituem as restrições da forma $x \bowtie c$, de forma que $v_0(H_1) = (4, \infty]$, $v_0(H_2) = [-\infty, 0]$ e $v_0(H_3) = [6.2, \infty]$. Por fim, notemos que ao invés da cláusula $(B \asymp C_2 \vee H_1 \asymp X \wedge C_5)$, adicionamos uma nova variável Z à fórmula original e escrevemos $(B \asymp C_2 + Z \vee H_1 \asymp X \wedge C_5)$, pois definimos restrições

intervalares apenas na forma $X \asymp Y \circ Z$. Como $v_0(Z) = [0, 0]$ é o elemento neutro da adição, ambas as cláusulas são equivalentes.

4.3 Existência de Valoração Degenerada

Como mencionado na seção 2.3, uma restrição intervalar R pode ser satisfeita por uma valoração v e, mesmo assim, suas variáveis não conterem valores que satisfaçam a restrição R no contexto dos reais. Nesse caso, mesmo que uma fórmula $\varphi_{\mathbb{I}}$ seja satisfatível, pode não existir tal conjunto. No entanto, se $v(X)$ é um intervalo degenerado, para todo $X \in V(\varphi_{\mathbb{I}})$, a satisfatibilidade da fórmula $\varphi_{\mathbb{I}}$ é suficiente para garantir a existência de um conjunto de números reais que satisfaz pelo menos uma restrição de cada cláusula de $\varphi_{\mathbb{I}}$.

Definição 4.7 *Uma valoração σ é dita **degenerada** se $\sigma(X) = [x, x]$ para todo $X \in V(\varphi_{\mathbb{I}})$.*

Teorema 4.2 *Dada uma instância SMT $(\varphi_{\mathbb{I}}, v_0, \epsilon, \mathbf{c})$ e uma valoração degenerada $\sigma \subseteq v_0$ sobre $V(\varphi_{\mathbb{I}})$ tal que $\mathfrak{J}(\varphi_{\mathbb{I}}, \sigma, v_0, \epsilon, \mathbf{c}) = 1$, então existe uma função $p : \mathbb{I} \mapsto \mathbb{R}$ que associa a cada intervalo $X \in V(R)$, para todo $R \in A(\varphi_{\mathbb{I}}) \mid \mathfrak{J}(R, \sigma, \mathbf{c}) = 1$, um valor $p(X) \in \sigma(X)$ de forma que*

$$\forall R = (X \asymp Y \circ Z) \in A(\varphi_{\mathbb{I}}) \mid \mathfrak{J}(R, \sigma, \mathbf{c}) = 1 : p(X) = p(Y) \circ p(Z)$$

Prova Seja uma fórmula SMT $\varphi_{\mathbb{I}}$ e uma valoração degenerada σ tal que $\mathfrak{J}(\varphi_{\mathbb{I}}, \sigma, v_0, \epsilon, \mathbf{c}) = 1$. Seja $R = (X \asymp Y \circ Z) \in A(\varphi_{\mathbb{I}}) \mid \mathfrak{J}(R, \sigma, \mathbf{c}) = 1$.

Suponha que $\mathbf{c} = \text{consistência no arco}$. Como $\mathfrak{J}(R, \sigma, \mathbf{c}) = 1$, então $\sigma(X) \subseteq \sigma(Y) \circ \sigma(Z)$ e, pelo teorema 2.1, $\forall x \in \sigma(X), \exists y \in \sigma(Y), \exists z \in \sigma(Z) \mid x = y \circ z$. Como σ é degenerada, então existem únicos $x \in \sigma(X)$, $y \in \sigma(Y)$ e $z \in \sigma(Z)$ tais que $x = y \circ z$, para toda ocorrência de X , Y e Z . Logo, a função p existe.

Por outro lado, se $\mathbf{c} = \text{consistência na envoltória}$, a prova segue análoga ao item anterior, pois como σ é degenerada, $\|\sigma(Y) \circ \sigma(Z)\| = \sigma(Y) \circ \sigma(Z)$, para qualquer $\circ \in \mathfrak{U}$.

□

Pelo teorema 4.2, se existe uma valoração degenerada σ que satisfaz uma dada fórmula $\varphi_{\mathbb{I}}$, então também existe um conjunto de valores reais que satisfaz todas as restrições independente da consistência utilizada. Assim, nosso problema torna-se encontrar (ou verificar a existência de) uma valoração degenerada para a fórmula dada.

Pela relação direta entre intervalos degenerados e números reais, podemos notar que esse problema se distancia do cenário intervalar, no qual queremos encontrar números aproximados, para uma área mais próxima da Programação Não Linear usual.

Os algoritmos 2 e 4 não são capazes de encontrar uma valoração degenerada, pois o método de particionar o intervalo na ramificação pode fazer com que o algoritmo nunca termine. No iSAT, no entanto, como os problemas de interesse apresentam características específicas, é possível definir propriedades que garantem a existência de uma solução degenerada.

A fim de especificar tais propriedades, vamos adequar a definição de uma fórmula *fortemente satisfeita*, apresentada em [17], à nossa representação intervalar:

Definição 4.8 *Uma fórmula $\varphi_{\mathbb{I}}$ é fortemente satisfeita por uma valoração v se existe um conjunto ordenado de átomos $\Omega = (R_1, \dots, R_n)$ sobre o qual vale as seguintes condições:*

1. $\forall \omega^{\mathbb{I}} \in C(\varphi_{\mathbb{I}}) : A(\omega^{\mathbb{I}}) \cap \Omega \neq \emptyset;$
2. $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq Y, X \neq Z$ e $X \notin V(R_j)$, para todo $j < i;$
3. $\forall R_i = (X \asymp Y \circ Z) \in \Omega : v(Y) \circ v(Z) \subseteq v(X) \subseteq \|v(Y) \circ v(Z)\|;$
4. $\forall R \in \Omega : \forall X \in V(R) : v(X) \neq \emptyset.$

Teorema 4.3 *Se uma fórmula $\varphi_{\mathbb{I}}$ é fortemente satisfeita por uma valoração v , então existe um refinamento degenerado σ de v que satisfaz $\varphi_{\mathbb{I}}$.*

Prova Segue em [17].

□

Podemos adicionar, ao algoritmo 4, um módulo que verifica se a valoração corrente v satisfaz fortemente a fórmula $\varphi_{\mathbb{I}}$. Quando todas as cláusulas $\omega^{\mathbb{I}} \in C(\varphi_{\mathbb{I}})$ forem unitárias

e consistentes, definimos um conjunto $\Omega = (R_1, \dots, R_{|C(\varphi_{\mathbb{I}})|})$ composto pelos átomos R_i não falsos de cada cláusula, satisfazendo a condição 1 da definição 4.8. Em seguida verificamos se as condições 2, 3 e 4 também valem e, no caso positivo, definimos uma valoração degenerada σ da seguinte forma:

para $i = 1 \dots |\Omega|$, $R_i = (X_i \asymp Y_i \circ_i Z_i)$:

$$\sigma \leftarrow \begin{cases} \sigma \cup (Z_i \mapsto [z, z]) \mid z \in v(Z_i) & \text{se } \sigma(Z_i) = \diamond \\ \sigma & \text{c.c.} \end{cases}$$

$$\sigma \leftarrow \begin{cases} \sigma \cup (Y_i \mapsto [y, y]) \mid y \in v(Y_i) & \text{se } \sigma(Y_i) = \diamond \\ \sigma & \text{c.c.} \end{cases}$$

$$\sigma \leftarrow \sigma \cup (X_i \mapsto \sigma(Y_i) \circ_i \sigma(Z_i))$$

Essa valoração existe porque, pela condição 2, é garantido que, inicialmente, $\sigma(X_i) = \diamond$. Se tal condição não fosse verdadeira, não poderíamos valorar $\sigma(X_i) = \sigma(Y_i) \circ_i \sigma(Z_i)$, pois $\sigma(X_i)$ já poderia ter um valor que satisfizesse uma outra restrição R_j tal que $X_i \in V(R_j)$ e a valoração $\sigma(X_i) = \sigma(Y_i) \circ_i \sigma(Z_i)$ tornaria R_j não satisfeita sobre σ (como o resultado de uma operação sobre intervalos degenerados é um intervalo degenerado, só existe um valor degenerado para X que satisfaz cada restrição).

Por outro lado, como $\sigma(Z_i) \subseteq v(Z_i)$, $\sigma(Y_i) \subseteq v(Y_i)$ e, pela condição 3, $v(Y_i) \circ_i v(Z_i) \subseteq v(X_i)$, então, pelo teorema 2.1, $\forall \sigma(Z_i), \forall \sigma(Y_i) : \exists \sigma(X_i) \subseteq v(X_i) \mid \sigma(X_i) \asymp \sigma(Y_i) \circ_i \sigma(Z_i)$.

Como existe pelo menos um átomo em cada cláusula satisfeito pela valoração v , que por sua vez é um refinamento da valoração inicial v_0 , então a valoração degenerada σ é um modelo para $\varphi_{\mathbb{I}}$

Enquanto encontrar um conjunto de restrições Ω que satisfaça a condição 1 é particularmente fácil, o mesmo não ocorre com as condições 2 e 3. No primeiro caso, pode ser necessário permutar o conjunto Ω afim de obter um conjunto Ω' ordenado segundo a propriedade 2. Enquanto essas duas condições têm relação com a estrutura da fór-

mula de entrada, as condições 3 e 4 dependem da valoração corrente. Embora a consistência na envoltória garanta que, para toda restrição $R = (X \asymp Y \circ Z)$ consistente, $v(X) \subseteq \|v(Y) \circ v(Z)\|$, não necessariamente $v(X) \supseteq v(Y) \circ v(Z)$, de forma que somente em alguns casos teremos uma valoração que satisfaça fortemente a fórmula de entrada.

Exemplo 4.3 (*Continuação do Ex. 4.2*) Após algumas execuções de $\text{iSAT}(\varphi_{\mathbb{I}}, v_0, \epsilon)$, encontramos uma valoração v sob a qual todas as cláusulas de $\varphi_{\mathbb{I}}$ são unitárias e interpretadas verdadeiras:

$$\begin{array}{lll}
 v(X) & = & (3, 4] & v(H_3) & = & [6.2, 8] & v(A_1) & = & [14.2, 16] \\
 v(Y) & = & [-4, 4.9] & v(C_1) & = & [1, 1] & v(A_2) & = & [-9.8, 8] \\
 v(B) & = & [1, 1] & v(C_2) & = & [0, 0] & v(A_3) & = & [6.2, 8] \\
 v(H_1) & = & (4, \infty] & v(C_3) & = & [2, 2] & v(Z) & = & [0, 0] \\
 v(H_2) & = & [-\infty, 0] & v(C_4) & = & [-2, -2] & & &
 \end{array}$$

Seja Ω o seguinte conjunto de átomos:

$$\Omega = ((B \asymp C_1 + Z), (A_1 \asymp X \wedge C_3), (A_2 \asymp C_4 \times Y), (A_3 \asymp A_1 + A_2), (H_3 \asymp A_3 + Z))$$

Esse conjunto satisfaz as duas primeiras condições e v satisfaz as condições 3 e 4. Aplicando o método de encontrar a valoração degenerada descrito por Fränzle, obtemos:

$$\begin{array}{ll}
 \sigma \leftarrow \emptyset & \\
 \sigma \leftarrow \sigma \cup (Z \mapsto [0, 0]) & \text{pois } \sigma(Z) = \diamond \text{ e } 0 \in v(Z) \\
 \sigma \leftarrow \sigma \cup (C_1 \mapsto [1, 1]) & \text{pois } \sigma(C_1) = \diamond \text{ e } 1 \in v(C_1) \\
 \sigma \leftarrow \sigma \cup (B \mapsto [1, 1]) & \text{pois } \sigma(C_1) + \sigma(Z) = [1, 1] \\
 \sigma \leftarrow \sigma \cup (C_3 \mapsto [2, 2]) & \text{pois } \sigma(C_3) = \diamond \text{ e } 2 \in v(C_3) \\
 \sigma \leftarrow \sigma \cup (X \mapsto [4, 4]) & \text{pois } \sigma(X) = \diamond \text{ e } 4 \in v(X) \\
 \sigma \leftarrow \sigma \cup (A_1 \mapsto [16, 16]) & \text{pois } \sigma(X) \wedge \sigma(C_3) = [16, 16]
 \end{array}$$

$$\begin{aligned}
\sigma \leftarrow \sigma \cup (Y \mapsto [4.5, 4.5]) & \quad \text{pois} \quad \sigma(Y) = \diamond \text{ e } 4.5 \in v(Y) \\
\sigma \leftarrow \sigma \cup (C_4 \mapsto [-2, -2]) & \quad \text{pois} \quad \sigma(C_4) = \diamond \text{ e } -2 \in v(C_4) \\
\sigma \leftarrow \sigma \cup (A_2 \mapsto [-9, -9]) & \quad \text{pois} \quad \sigma(C_4) \times \sigma(Y) = [-9, -9] \\
\sigma \leftarrow \sigma \cup (A_2 \mapsto [-9, -9]) & \quad \text{pois} \quad \sigma(A_2) = [-9, -9] \\
\sigma \leftarrow \sigma \cup (A_1 \mapsto [16, 16]) & \quad \text{pois} \quad \sigma(A_1) = [16, 16] \\
\sigma \leftarrow \sigma \cup (A_3 \mapsto [7, 7]) & \quad \text{pois} \quad \sigma(A_1) + \sigma(A_2) = [7, 7] \\
\sigma \leftarrow \sigma \cup (Z \mapsto [0, 0]) & \quad \text{pois} \quad \sigma(Z) = [0, 0] \\
\sigma \leftarrow \sigma \cup (A_3 \mapsto [7, 7]) & \quad \text{pois} \quad \sigma(A_3) = [7, 7] \\
\sigma \leftarrow \sigma \cup (H_3 \mapsto [7, 7]) & \quad \text{pois} \quad \sigma(A_3) - \sigma(C_2) = [7, 7]
\end{aligned}$$

Para as variáveis restantes, H_1 e H_2 , definimos $\sigma(H_i) = h_i \mid h_i \in v(H_i)$. Para esse exemplo, temos que $\sigma \subseteq v_0$ e, portanto, σ é um modelo degenerado de $\varphi_{\mathbb{I}}$.

□

4.4 Considerações

Neste capítulo mostramos que a consistência intervalar é condição necessária para a existência de um conjunto de valores reais nos intervalos de forma a satisfazer as restrições também no contexto dos números reais. Dessa forma, definimos um método de aplicá-la em uma restrição inconsistente, excluindo valores que garantidamente não pertencem a uma solução real.

Em seguida, enunciamos dois problemas de satisfação de restrições intervalares: SAT \mathbb{I} , no qual a fórmula intervalar a ser satisfeita é constituída por uma conjunção de restrições, e Satisfatibilidade Módulo Álgebra Intervalar, na qual as restrições são estruturadas em uma conjunção de disjunções. Foram apresentadas soluções para ambos os problemas, baseados em um método de ramificação e poda aplicando-se a consistência intervalar. Para o problema SMT, mais especificamente, apresentamos o resolvidor iSAT, que implementa técnicas de resolvidores SAT como o aprendizado de cláusulas e o retrocesso não cronológico, proporcionando um tempo de execução melhor que os demais resolvidores SMT da época.

Ainda sobre o iSAT, apresentamos a definição de uma fórmula fortemente satisfeita, que garante a existência de um modelo degenerado e, conseqüentemente, de uma solução também nos números reais. No entanto, a fórmula de entrada, assim como a valoração consistente, devem apresentar características que nem sempre são obtidas pela consistência intervalar, tornando sua aplicação limitada a algumas classes de instâncias SMT.

CAPÍTULO 5

OTIMIZAÇÃO INTERVALAR

“Nada como procurar quando se quer achar alguma coisa...

Quando se procura geralmente se encontra alguma coisa, sem dúvida.

Mas nem sempre o que estávamos procurando.”

(J.R.R. Tolkien, O Hobbit, Cap. 4)

Este capítulo apresenta um novo problema intervalar, não abordado pelo resolvidor iSAT e pelo algoritmo SAT_{IRP} : a Otimização Intervalar. Vamos definir formalmente esse problema e, em seguida, mostrar como as fórmulas intervalares podem ser representadas por grafos e digrafos, tornando possível uma redefinição de *fórmula fortemente satisfeita*, com uma abordagem em relação aos grafos definidos. Essa nova abordagem permite a descrição de um algoritmo capaz de encontrar a valoração degenerada de fórmulas que apresentem um comportamento acíclico, semelhante ao método apresentado na seção 4.3.

No entanto, como a definição de fórmula fortemente satisfeita exige uma forte propriedade da valoração corrente (condição 3 da definição 4.8), não garantida pela consistência intervalar, nós restringimos ainda mais a estrutura das fórmulas, a fim de que o algoritmo seja capaz de encontrar o modelo degenerado mesmo para valorações consistentes apenas no arco da fórmula de entrada. Com essa modificação, o algoritmo também é capaz de encontrar o modelo ótimo para a fórmula, resolvendo problemas de Otimização Intervalar.

Por fim, estendemos esse algoritmo à outras classes de fórmulas, sem alterar a condição sobre a valoração corrente, obtendo o que vamos chamar de *valoração aproximada*, isto é, na qual algumas variáveis definidas não são refinamentos da valoração inicial. A resposta encontrada, mesmo aproximada, pode ser solução de alguns problemas reais, dependendo do processo de modelagem matemática e da existência de aproximações numéricas.

Vários problemas reais não estão interessados apenas em encontrar uma valoração que satisfaça um conjunto de restrições, mas que esta valoração também seja ótima sob algum critério. No cenário real, esses problemas são, na grande maioria das vezes, representados matematicamente pela Programação Não Linear, na qual queremos encontrar um conjunto de variáveis que, além de satisfazer uma conjunção de restrições, minimiza uma função objetivo dada.

A Satisfatibilidade Intervalar possui uma estrutura muito semelhante à Programação Não Linear, sendo sua principal diferença a ausência de uma função objetivo. A representação intervalar deste problema exige, além de uma fórmula intervalar Φ e uma valoração inicial v_0 , uma função objetivo $f_{\mathbb{I}}$ a ser minimizada.

Embora não definimos a minimização de uma função intervalar $f_{\mathbb{I}}$, sabemos que $f_{\mathbb{I}} : \mathbb{I}^n \mapsto \mathbb{I}$ e, utilizando o conceito de restrições primitivas, podemos representar $f_{\mathbb{I}}$ por um único intervalo F , adicionando à Φ um conjunto de novas restrições intervalares dadas por R_1^f, \dots, R_m^f . Assim, nosso problema torna-se encontrar um modelo que *minimize* F , o que vamos denotar por *min* F .

Definição 5.1 *Uma valoração $v^* \subseteq v_0$ **minimiza** um intervalo $v_0(F)$ se $v^*(F) \preceq v(F)$ para toda valoração $v \subseteq v_0$, onde \preceq é uma relação de ordem tal que*

$$X \preceq Y \text{ se, e somente se, } \underline{x} \leq \underline{y} \text{ e } \bar{x} \leq \bar{y}$$

Repare que a ordenação \preceq não é total. Por exemplo, dados os intervalos $[1, 2]$, $[2, 4]$, $[2, 2]$ e $[0, 5]$, temos que $[1, 2] \preceq [2, 2] \preceq [2, 4]$, mas $[0, 5] \not\preceq X$, para qualquer $X \in \{[1, 2], [2, 4], [2, 2]\}$. Portanto, não podemos afirmar nada sobre o intervalo $[0, 5]$. Em nossos problemas de interesse, no entanto, essa definição é suficiente para caracterizar a minimização de um intervalo. A saber, se temos uma valoração inicial v_0 que deve ser respeitada e $v_0(F) = [\underline{f}, \bar{f}]$, então, pela definição, a valoração degenerada $v^*(F) = [\underline{f}, \underline{f}]$ minimiza F , pois $\forall v \subseteq v_0 : [\underline{f}, \underline{f}] \preceq v(F)$. Importante observar que se $v_0(F)$ for um intervalo aberto ou fechado à direita, não existe uma valoração v^* que minimiza F .

Exemplo 5.1 *Suponha que queremos minimizar a função $f_{\mathbb{I}}(X, Y, Z) = (X + (Y \wedge Z)) \div Z$ sobre um conjunto de restrições Φ .*

Podemos representar $f_{\mathbb{I}}(X, Y, Z)$ pelo intervalo F tal que

$$R_1^f = (F \asymp A_1 \div Z)$$

$$R_2^f = (A_1 \asymp X + A_2)$$

$$R_3^f = (A_2 \asymp Y \wedge Z)$$

E assim, obtemos o seguinte problema:

$$\begin{array}{ll} \min & F \\ \text{s.a} & \Phi \wedge R_1^f \wedge R_2^f \wedge R_3^f \end{array}$$

Pela transformação em restrições primitivas, temos que $f_{\mathbb{I}}(X, Y, Z) = F$, para quaisquer intervalos X, Y e Z .

□

Definição 5.2 *O problema computacional da **Otimização Intervalar** é dado por:*

Otimização Intervalar (POI)

Entrada: fórmula intervalar Φ , valoração inicial v_0 , erro de aproximação ϵ , consistência intervalar \mathbf{c} e intervalo objetivo $F \in V(\Phi)$.

Saída: modelo v de Φ que minimiza F , se tal modelo existir, ou **insatisfável**.

A Otimização Intervalar é estudada desde 1990, quando foi proposta por Ishibuchi e Tanaka [21]. Vários métodos matemáticos envolvendo análise intervalar, adaptações do método Simplex e análise de funções intervalares foram propostos nas últimas décadas [28, 4, 20]. Além desses, algoritmos de ramificação e poda por limitantes também podem ser utilizados para resolver problemas de otimização intervalar.

Embora o iSAT não resolva problemas de otimização, seu método de verificar a existência de uma valoração degenerada (*fórmula fortemente satisfeita*) pode ser estendido

à um algoritmo capaz de encontrar o modelo ótimo de determinadas fórmulas. Nas próximas seções vamos apresentar esse algoritmo e as características que uma determinada fórmula deve apresentar para sua aplicação.

5.1 Representação por Grafos e Digrafos

A definição 4.8, sobre fórmulas fortemente satisfeitas, pode ser estendida de forma natural à fórmulas intervalares. Essas fórmulas, como nas SMT, devem apresentar uma estrutura capaz de satisfazer a condição 2 dessa definição. Assim, afim de uma melhor visualização e manipulação de tais estruturas, vamos representar essas fórmulas por grafos e digrafos.

Definição 5.3 *Dada uma restrição $R = (X \asymp Y \circ Z)$ qualquer, temos que*

- $h(R) = X$ é a **cabeça** de R , isto é, a variável que está isolada em R ;
- $t(R) = \{Y, Z\}$ é a **cauda** de R , isto é, os operandos da função que está sendo relacionada com $h(R)$.

Definição 5.4 *Uma restrição R define um grafo $\mathcal{G}(R)$ e um digrafo $\mathcal{D}(R)$ dados por*

$$\mathcal{G}(R) = (V(R), \{\{h(R), X\} \mid X \in t(R)\})$$

$$\mathcal{D}(R) = (V(R), \{(h(R), X) \mid X \in t(R)\})$$

Assim, o grafo $\mathcal{G}(R)$ e o digrafo $\mathcal{D}(R)$ de uma restrição $R = (X \asymp Y \circ Z)$ são dados por



Note que a operação \circ não é relevante nesta representação. Além disso, a aresta pontilhada que conecta Y e Z não pertence ao grafo propriamente dito, mas podemos desenhá-la para uma melhor visualização da relação entre Y e Z .

Essa representação servirá tanto para facilitar a visualização das restrições como para possibilitar o uso de grafos nos algoritmos que serão apresentados. Assim, trabalhando sobre os grafos somos capazes de obter resultados sobre as restrições propriamente ditas.

Por abuso de notação, definimos os vértices do grafo com o mesmo nome dos intervalos que cada um representa. Alternativamente, poderíamos considerar que existe uma função que mapeia cada intervalo a um vértice correspondente.

Definição 5.5 Uma fórmula Φ define um grafo $\mathcal{G}(\Phi)$ e um digrafo $\mathcal{D}(\Phi)$ dados por

$$\mathcal{G}(\Phi) = \left(\bigcup_{R \in A(\Phi)} P(\mathcal{G}(R)), \bigcup_{R \in A(\Phi)} E(\mathcal{G}(R)) \right)$$

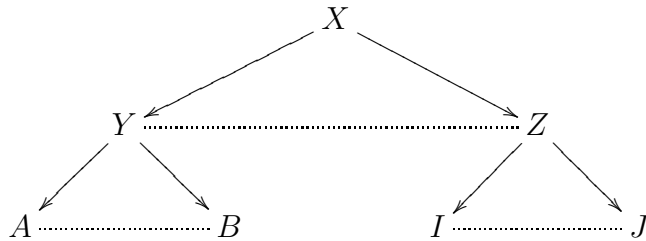
$$\mathcal{D}(\Phi) = \left(\bigcup_{R \in A(\Phi)} P(\mathcal{D}(R)), \bigcup_{R \in A(\Phi)} E(\mathcal{D}(R)) \right)$$

onde $P(\mathcal{G}(R))$ e $E(\mathcal{G}(R))$ são, respectivamente, os conjuntos de vértices (pontos) e arestas de $\mathcal{G}(R)$, isto é, $\mathcal{G}(R) = (P(\mathcal{G}(R)), E(\mathcal{G}(R)))$. O mesmo vale para $\mathcal{D}(R) = (P(\mathcal{D}(R)), E(\mathcal{D}(R)))$.

Neste trabalho, vamos utilizar conceitos sobre os grafos, como **raíz**, **folha**, **conexidade** e **ciclo**, na forma usual [14].

Exemplo 5.2 Seja $\Phi = (X \asymp Y \circ_1 Z) \wedge (Y \asymp A \circ_2 B) \wedge (Z \asymp I \circ_3 J)$ uma fórmula intervalar.

Temos que $\mathcal{D}(\Phi) =$



□

Embora essa representação seja adequada para a maioria das fórmulas intervalares, existem casos nos quais os grafos definidos não são capazes de representar as fórmulas por completo. Se uma fórmula Φ possui uma restrição R tal que $|V(R)| < 3$, então

deveria existir pelo menos uma aresta que conecta um vértice à si mesmo, o que não é representável pelo grafo $\mathcal{G}(R)$. Da mesma forma, se Φ possui dois átomos R e S tais que $E(\mathcal{G}(R)) \cap E(\mathcal{G}(S)) \neq \emptyset$, o grafo $\mathcal{G}(\Phi)$ representará apenas uma das arestas. Enquanto o primeiro problema pode ser resolvido logo na modelagem, manipulando algebricamente a restrição em questão, o mesmo não é possível quando $E(\mathcal{G}(R)) \cap E(\mathcal{G}(S)) \neq \emptyset$ para dois átomos R e S quaisquer.

Além disso, vamos considerar, sem perda de generalização, que toda fórmula Φ define um grafo $\mathcal{G}(\Phi)$ conexo. Se $\mathcal{G}(\Phi)$ for desconexo, então existe pelo menos uma raiz em $\mathcal{G}(\Phi)$ para cada componente conexa. Como uma componente não afeta as demais, podemos simplesmente dividir a fórmula em subfórmulas Φ_1, \dots, Φ_n (tais que $\mathcal{G}(\Phi_i)$ é uma componente conexa maximal de $\mathcal{G}(\Phi)$) e aplicar os algoritmos em cada subfórmula separadamente. No problema POI, por exemplo, encontramos o modelo v_k da subfórmula Φ_k que minimiza $F \in V(\Phi_k)$ e qualquer modelo v_i para as subfórmulas restantes. Como $V(\Phi_i) \cap V(\Phi_j) = \emptyset$, para todo $1 \leq i < j \leq n$, então obtemos o modelo de Φ que minimiza F agrupando os demais modelos v_i .

Definição 5.6 Dizemos que uma fórmula intervalar $\Phi = R_1 \wedge R_2 \wedge \dots \wedge R_n$ é **primitiva** se, e somente se,

- $|V(R_i)| = 3$, para todo $1 \leq i \leq n$;
- $E(\mathcal{G}(R_i)) \cap E(\mathcal{G}(R_j)) = \emptyset$, para todo $j < i$;
- $\mathcal{G}(\Phi)$ é conexo.

Embora fórmulas não primitivas também possam ter solução no contexto de otimização, neste trabalho vamos focar nas fórmulas primitivas, pois o método de encontrar o modelo ótimo que será aqui apresentado não é capaz de garantir solução para as demais fórmulas.

Definição 5.7 Dada uma fórmula intervalar Φ e um conjunto de restrições $\{R_1, \dots, R_n\} \subseteq A(\Phi)$, denotamos por $\Phi' = \Phi \setminus \{R_1, \dots, R_n\}$ a fórmula intervalar Φ' tal que

$$A(\Phi') = A(\Phi) \setminus \{R_1, \dots, R_n\}$$

$$V(\Phi') = \bigcup_{R \in A(\Phi')} V(R)$$

Para o conjunto unitário $\{R\}$, denotamos por $\Phi \setminus R$ a fórmula $\Phi \setminus \{R\}$.

Definição 5.8 Dada uma fórmula intervalar primitiva Φ e um vértice $X \in P(\mathcal{D}(\Phi))$, o grafo não direcionado $\mathcal{A}_\Phi(X)$ é o subgrafo de $\mathcal{G}(\Phi)$ induzido pelos vértices alcançáveis a partir de X em $\mathcal{D}(\Phi)$.

Voltando à definição de uma fórmula fortemente satisfeita, vamos estendê-la às fórmulas intervalares primitivas de forma que possamos visualizar suas propriedades nos grafos por elas definidos.

Definição 5.9 Uma fórmula intervalar primitiva Φ é **fortemente satisfeita** por uma valoração v se:

1. $\mathcal{D}(\Phi)$ é acíclico;
2. $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$;
3. $\forall R = (X \asymp Y \circ Z) \in A(\Phi) : v(Y) \circ v(Z) \subseteq v(X) \subseteq \|v(Y) \circ v(Z)\|$;
4. $\forall X \in V(\Phi) : v(X) \neq \emptyset$;
5. $\forall X \in V(\Phi) : v(X) \in \mathbb{I}$;

Teorema 5.1 Dada uma fórmula intervalar primitiva Φ , temos que $\mathcal{D}(\Phi)$ é acíclico e $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$ se, e somente se, existe uma ordenação Ω dos átomos de Φ tal que $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq Y, X \neq Z$ e $X \notin V(R_j)$, para todo $j < i$.

Prova. Seja Φ uma fórmula intervalar primitiva. Efetuando uma ordenação topológica no grafo $\mathcal{D}(\Phi)$, obtemos uma ordenação de seus vértices de forma que cada vértice X vem antes de todos aqueles que têm arestas que chegam em X , pois $\mathcal{D}(\Phi)$ é acíclico e essa

característica é condição necessária e suficiente para a existência de tal ordenação. Mais formalmente, vamos provar ambos os lados da implicação enunciada:

(\Rightarrow) Suponha que $\mathcal{D}(\Phi)$ é acíclico e $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$. Queremos provar que existe uma ordenação Ω dos átomos de Φ tal que $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq Y, X \neq Z$ e $X \notin V(R_j)$, para todo $j < i$.

Como $\mathcal{D}(\Phi)$ é acíclico, então $\forall (X, Y) \in E(\mathcal{D}(\Phi)) : X \notin \mathcal{A}_\Phi(Y)$. Portanto, $\forall R = (X \asymp Y \circ Z) \in A(\Phi) : X \notin \mathcal{A}_\Phi(Y)$ e $X \notin \mathcal{A}_\Phi(Z)$.

Ordenando as restrições pelos tamanhos dos subgrafos $\mathcal{A}_\Phi(Y)$ e $\mathcal{A}_\Phi(Z)$, obtemos um conjunto $\Omega = (R_1, \dots, R_n)$ tal que

$$\forall R_i \in \Omega : \sum_{Y \in t(R_i)} |\mathcal{A}_\Phi(Y)| \geq \sum_{Z \in t(R_j)} |\mathcal{A}_\Phi(Z)|, \text{ para todo } j < i$$

Dessa forma, temos que $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \notin t(R_j)$, para todo $j < i$. Além disso, como $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$ e $|V(R)| = 3$, pois Φ é primitiva, então

$$\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq Y, X \neq Z \text{ e } X \notin V(R_j), \text{ para todo } j < i$$

(\Leftarrow) Suponha que existe uma ordenação Ω dos átomos de Φ tal que $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq Y, X \neq Z$ e $X \notin V(R_j)$, para todo $j < i$. Queremos provar que $\mathcal{D}(\Phi)$ é acíclico e $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$.

Pela suposição acima, temos que $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq h(R_j)$ e $X \notin t(R_j)$, para todo $j < i$. Logo, $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$.

Reconstruindo o grafo $\mathcal{D}(\Phi)$ a partir do conjunto Ω , observamos que, para a primeira restrição $R_1 \in \Omega$, $\mathcal{D}(R_1)$ é acíclico por definição. Acrescentando a próxima restrição R_2 , como $h(R_2) \notin t(R_1)$, então apenas serão adicionadas arestas no mesmo sentido, não fechando ciclo. Essa mesma propriedade ocorre para todas as inserções de restrições, pois $\forall R_i \in \Omega : h(R_i) \notin t(R_j)$, para todo $j < i$. Portanto, por indução, $\mathcal{D}(\Phi)$ é acíclico. \square

Corolário 5.1 *Sobre fórmulas intervalares primitivas as definições 4.8 e 5.9 são equivalentes.*

Prova. Seja Φ uma fórmula intervalar primitiva e v uma valoração sobre $V(\Phi)$. Como as condições 3 e 4 são iguais nas duas definições e, enquanto a primeira está definida sobre a consistência na envoltória, garantindo que todas as variáveis são valoradas com intervalos em v , a condição 5 da segunda definição garante esta mesma propriedade, basta provar que as condições 1 e 2 são equivalentes entre as definições.

Pela definição 4.8, deve existir um conjunto ordenado de restrições Ω tal que, pela condição 1, $\forall \omega^\perp \in C(\Phi) : A(\omega^\perp) \cap \Omega = \emptyset$. Como Φ é uma fórmula intervalar, então toda cláusula de Φ tem apenas um átomo, e portanto, $\forall R \in A(\Phi) : R \cap \Omega = \emptyset$, isto é, Ω é uma ordenação dos átomos de Φ . Além disso, essa ordenação deve respeitar a condição 2 da mesma definição: $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq Y, X \neq Z$ e $X \notin V(R_j)$, para todo $j < i$.

Por outro lado, pelo teorema 5.1, $\mathcal{D}(\Phi)$ é acíclico e $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$ se, e somente se, existe uma ordenação Ω dos átomos de Φ tal que $\forall R_i = (X \asymp Y \circ Z) \in \Omega : X \neq Y, X \neq Z$ e $X \notin V(R_j)$, para todo $j < i$.

Portanto, as condições 1 e 2 da definição 4.8 também são equivalentes as condições 1 e 2 da definição 5.9.

□

O resolvedor iSAT utiliza a consistência na envoltória como definição da satisfação de restrições. Assim, todas as variáveis e operações resultam em intervalos, não havendo a necessidade da valoração multi-intervalar. Nessa nova abordagem, no entanto, não restringimos a consistência intervalar à consistência na envoltória, sendo necessário acrescentar à definição 5.9 a condição de que $\forall X \in V(\Phi) : v(X) \in \mathbb{I}$. Embora essa condição não seja satisfeita pela consistência no arco, podemos aplicar um método de ramificação em todas as variáveis valoradas por multi-intervalos, obtendo subproblemas cujas variáveis podem ser definidas apenas por intervalos.

Exemplo 5.3 *Seja $\Phi = (X \asymp Y \div Z)$ uma fórmula intervalar primitiva e v_0 uma valoração inicial tal que $v_0(X) = [-2, 4]$, $v_0(Y) = \langle [1, 2], [3, 5] \rangle$ e $v_0(Z) = [-1, 1]$.*

Aplicando a consistência no arco de Φ , obtemos uma valoração consistente v tal que

$$\begin{aligned}
v(X) &= v_0(X) \cap (v_0(Y) \div v_0(Z)) = [-2, 4] \cap (\langle [1, 2], [3, 5] \rangle \div [-1, 1]) \\
&= [-2, 4] \cap \langle [-\infty, -1], [1, \infty], [-\infty, -3], [3, \infty] \rangle = \langle [-2, -1], [1, 4] \rangle \\
v(Y) &= v_0(Y) \cap (v(X) \times v_0(Z)) = \langle [1, 2], [3, 5] \rangle \cap (\langle [-2, -1], [1, 4] \rangle \times [-1, 1]) \\
&= \langle [1, 2], [3, 5] \rangle \cap \langle [-2, 2], [-4, 4] \rangle = \langle [1, 2], [3, 4] \rangle \\
v(Z) &= v_0(Z) \cap (v(X) \otimes v(Y)) = [-1, 1] \cap (\langle [-2, -1], [1, 4] \rangle \otimes \langle [1, 2], [3, 4] \rangle) \\
&= [-1, 1] \cap (\langle [1, 2], [3, 4] \rangle \div \langle [-2, -1], [1, 4] \rangle) = [-1, 1] \cap \langle [-4, -0.5], [0.25, 4] \rangle \\
&= \langle [-1, -0.5], [0.25, 1] \rangle
\end{aligned}$$

No entanto, como $v(X), v(Y), v(Z) \notin \mathbb{I}$, ramificamos todas as variáveis, obtendo as seguintes valorações:

$v_1(X) = [-2, -1]$	$v_1(Y) = [1, 2]$	$v_1(Z) = [-1, -0.5]$
$v_2(X) = [-2, -1]$	$v_2(Y) = [1, 2]$	$v_2(Z) = [0.25, 1]$
$v_3(X) = [-2, -1]$	$v_3(Y) = [3, 4]$	$v_3(Z) = [-1, -0.5]$
$v_4(X) = [-2, -1]$	$v_4(Y) = [3, 4]$	$v_4(Z) = [0.25, 1]$
$v_5(X) = [1, 4]$	$v_5(Y) = [1, 2]$	$v_5(Z) = [-1, -0.5]$
$v_6(X) = [1, 4]$	$v_6(Y) = [1, 2]$	$v_6(Z) = [0.25, 1]$
$v_7(X) = [1, 4]$	$v_7(Y) = [3, 4]$	$v_7(Z) = [-1, -0.5]$
$v_8(X) = [1, 4]$	$v_8(Y) = [3, 4]$	$v_8(Z) = [0.25, 1]$

Essas são todas as combinações possíveis de valorações, mas nem todas são consistentes no arco de Φ , sendo necessário um novo refinamento de intervalos, para cada valoração v_i gerada.

□

Assim como ocorre na separação de fórmulas a fim de se obter subfórmulas conexas, neste método também é possível obter uma solução para a valoração original agrupando as soluções obtidas em cada ramificação. Para o problema POI, por exemplo, basta encontrar o modelo $v_i^* \subseteq v_i$ de toda ramificação v_i , com relação a fórmula de entrada Φ , e compará-los a fim de obter o modelo v_k^* tal que $\forall i : v_k^*(F) \preceq v_i^*(F)$.

Embora, pelo corolário 5.1, as definições 4.8 e 5.9 sejam equivalentes, o método de encontrar a valoração degenerada proposto em [17] utiliza a ordenação de restrições Ω como uma sequência de inferências que devem ser respeitadas no processo de valoração. Dessa forma, iniciamos a valoração pela primeira restrição de Ω , inferindo valorações sobre a segunda e assim sucessivamente.

Na definição 5.9 não temos a ordenação Ω . No entanto, pelo teorema 5.1, verificamos que essa ordenação resulta de uma busca no grafo $\mathcal{D}(\Phi)$ com início nas folhas, alcançando os próximos vértices se suas arestas de saída têm um vértice já alcançado (valorado). Assim, descrevemos no algoritmo 5 um método que, dada uma fórmula primitiva Φ e uma valoração v sobre as variáveis de Φ , devolve um modelo degenerado $\sigma \subseteq v$ de Φ . Esse método é uma reformulação do método de valoração degenerada proposto em [17], que realiza uma busca no grafo $\mathcal{D}(\Phi)$ para valorar todas as variáveis.

Algoritmo 5: Encontra Valoração Degenerada (EVD)

Entrada: (Φ, v)

Saída: modelo degenerado $\sigma \subseteq v$ de Φ ou *insatisfável*

```

1   $\sigma \leftarrow \emptyset$ 
2  para todo  $X \in V(\Phi) \mid \sigma(X) = \diamond$  e  $X \neq h(R)$  para todo  $R \in A(\Phi)$  faça
3  |   Seja  $x \in v(X)$  qualquer
4  |    $\sigma \leftarrow \sigma \cup (X \mapsto [x, x])$ 
5  fim
6  para todo  $R = (X \asymp Y \circ Z) \in A(\Phi) \mid \sigma(X) = \diamond, \sigma(Y) \neq \diamond$  e  $\sigma(Z) \neq \diamond$  faça
7  |    $\sigma \leftarrow \sigma \cup (X \mapsto \sigma(Y) \circ \sigma(Z))$ 
8  fim
9  se  $\forall X \in V(\Phi) : \sigma(X) \neq \diamond$  então
10 |   retorna  $\sigma$ 
11 fim
12 retorna insatisfável

```

Exemplo 5.4 *Seja $(\Phi, v_0, \epsilon, \mathfrak{c})$ uma instância SATII tal que $\Phi = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$ é primitiva:*

$$R_1 = (H_3 \asymp A_3 + Z)$$

$$R_2 = (A_3 \asymp A_1 + A_2)$$

$$R_3 = (A_2 \asymp C_4 \times Y)$$

$$R_4 = (A_1 \asymp X \wedge C_3)$$

$$R_5 = (B \asymp C_1 + Z)$$

O erro ϵ é qualquer, \mathfrak{c} = consistência na envoltória e a valoração v_0 é dada por

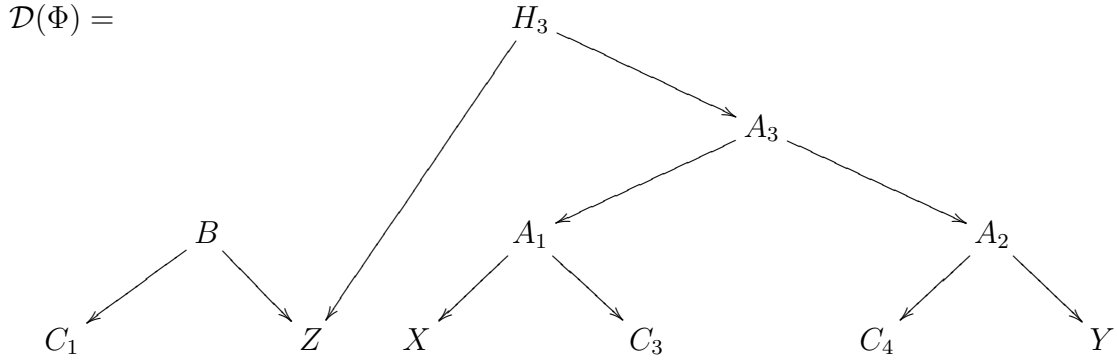
$$\begin{array}{lll} v_0(X) = [-2, \infty] & v_0(C_1) = [1, 1] & v_0(A_1) = [-\infty, \infty] \\ v_0(Y) = [4, \infty] & v_0(Z) = [0, 0] & v_0(A_2) = [-\infty, \infty] \\ v_0(B) = [-\infty, \infty] & v_0(C_3) = [2, 2] & v_0(A_3) = [-\infty, \infty] \\ v_0(H_3) = [6.2, \infty] & v_0(C_4) = [-2, -2] & \end{array}$$

Também seja v uma valoração corrente dada por

$$\begin{array}{lll} v(X) = (3, 4] & v(C_1) = [1, 1] & v(A_1) = [14.2, 16] \\ v(Y) = [-4, 4.9] & v(Z) = [0, 0] & v(A_2) = [-9.8, 8] \\ v(B) = [1, 1] & v(C_3) = [2, 2] & v(A_3) = [6.2, 8] \\ v(H_3) = [6.2, 8] & v(C_4) = [-2, -2] & \end{array}$$

As valoração v e v_0 aqui enunciadas são as mesmas do Ex. 4.3, assim como Φ é um fragmento da fórmula SMT apresentada. Portanto, o conjunto $\Omega = (R_5, R_4, R_3, R_2, R_1)$ satisfaz todas as condições da definição 4.8. Logo, v satisfaz fortemente Φ e podemos encontrar um modelo degenerado $\sigma \subseteq v$ pelo método de [17] (Ex. 4.3).

Por outro lado, $\nexists R, S \in A(\Phi) \mid h(R) = h(S)$ e $\mathcal{D}(\Phi)$ é acíclico:



Executando $\text{EVD}(\Phi, v)$, definimos os valores de $\sigma(C_1)$, $\sigma(Z)$, $\sigma(C_3)$, $\sigma(C_4)$, $\sigma(X)$ e $\sigma(Y)$ com qualquer valor real pertencente, respectivamente, aos intervalos $v(C_1)$, $v(Z)$, $v(C_3)$, $v(C_4)$, $v(X)$ e $v(Y)$ (por exemplo, $\sigma(C_1) = [1, 1]$, $\sigma(Z) = [0, 0]$, $\sigma(C_3) = [2, 2]$, $\sigma(C_4) = [-2, -2]$, $\sigma(X) = [4, 4]$ e $\sigma(Y) = [4.5, 4.5]$), pois essas variáveis não são cabeças de restrições, isto é, seus vértices no grafo $\mathcal{D}(\Phi)$ são folhas. Então, como as restrições $(B \asymp C_1 + Z)$, $(A_1 \asymp X \wedge C_3)$ e $(A_2 \asymp C_4 \times Y)$ têm as variáveis da cauda com valores definidos em σ e as variáveis da cabeça não, obtemos os valores

$$\sigma(B) = \sigma(C_1) + \sigma(Z) = [1, 1]$$

$$\sigma(A_1) = \sigma(X) \wedge \sigma(C_3) = [16, 16]$$

$$\sigma(A_2) = \sigma(C_4) \times \sigma(Y) = [-9, -9]$$

Em seguida, a restrição $(A_3 \asymp A_1 + A_2)$ pode ser valorada, obtendo $\sigma(A_3) = \sigma(A_1) + \sigma(A_2) = [7, 7]$. E, finalmente, $\sigma(H_3) = \sigma(A_3) + \sigma(Z) = [7, 7]$.

□

5.2 Otimização Exata

Como já discutido na seção 4.3, a condição 3 da definição de fórmula fortemente satisfeita não é garantida pela consistência intervalar. No entanto, se essa condição não for verificada, é possível que a valoração degenerada encontrada pelo algoritmo 5 não seja um refinamento da valoração inicial dada. Mesmo assim, vamos relaxar essa condição e

definir um erro associado à valoração degenerada segundo a valoração inicial. Intuitivamente, excluimos a condição 3 da definição 5.9 e garantimos que a valoração corrente v é consistente apenas no arco da fórmula Φ . Assim, após obter uma valoração degenerada σ , verificamos se essa é um refinamento de v . No caso positivo, σ é, de fato, um modelo para a fórmula de entrada. Caso contrário, embora σ satisfaça a fórmula, não há garantias de que a valoração inicial v_0 seja respeitada. Nesse caso, definimos um erro capaz de medir o quanto σ viola v_0 e, dependendo do problema de interesse, aceitamos σ como um modelo ou não.

Definição 5.10 Dada uma variável intervalar X , uma valoração v_0 e uma valoração degenerada σ , o **erro de valoração** $\varepsilon_\sigma^{v_0}(X) \in \overline{\mathbb{R}}$ define a distância entre $\sigma(X) = [x, x]$ e $v_0(X) = [\underline{x}, \bar{x}]$.

$$\varepsilon_\sigma^{v_0}(X) = \begin{cases} \underline{x} - x & \text{se } x < \underline{x} \\ x - \bar{x} & \text{se } x > \bar{x} \\ 0 & \text{c.c.} \end{cases}$$

Exemplo 5.5 Dada uma fórmula $\Phi = (X \simeq Y + Z)$, em $\text{EVD}(\Phi, v)$ definimos os valores $\sigma(Y) \subseteq v(Y)$ e $\sigma(Z) \subseteq v(Z)$ e inferimos o valor $\sigma(X) = \sigma(Y) + \sigma(Z)$:

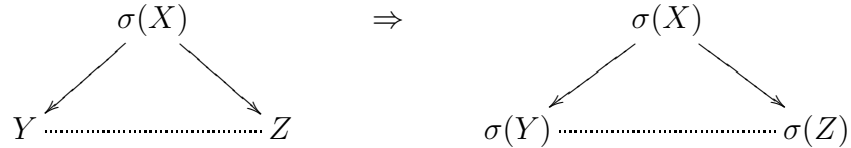


Como garantimos que v é consistente no arco de R , então $v(X) \subseteq v(Y) + v(Z)$. Assim, é possível que $\sigma(X) \not\subseteq v(X)$. Por exemplo, se $v(X) = v_0(X) = [2, 4]$, $v(Y) = v_0(Y) = [1, 3]$ e $v(Z) = v_0(Z) = [1, 2]$, temos que v é consistente no arco de R , mas para $\sigma(Y) = [3, 3]$ e $\sigma(Z) = [2, 2]$, $\sigma(X) = [3, 3] + [2, 2] = [5, 5] \not\subseteq [2, 4]$. Assim, temos que $\varepsilon_\sigma^{v_0}(X) = 1$ e $\varepsilon_\sigma^{v_0}(Y) = \varepsilon_\sigma^{v_0}(Z) = 0$.

□

Definição 5.11 Uma valoração v sobre um conjunto de variáveis $V(\Phi)$ é dita **exata** se, dada uma valoração inicial v_0 sobre $V(\Phi)$, $\forall X \in V(\Phi) : \varepsilon_\sigma^{v_0}(X) = 0$.

No exemplo anterior, se quisermos garantir uma valoração degenerada exata, basta iniciar a valoração pela variável X , pois, pelo teorema 2.1, $\forall x \in v(X) : \exists y \in v(Y), \exists z \in v(Z) \mid x = y + z$, e assim, $\forall \sigma(X) = [x, x] \subseteq v(X) : \exists \sigma(Y) = [y, y] \subseteq v(Y), \exists \sigma(Z) = [z, z] \subseteq v(Z) \mid \sigma(X) \asymp \sigma(Y) + \sigma(Z)$.



Mais do que isso, como v é consistente no arco de R , então

$$v(X) \subseteq v(Y) + v(Z)$$

$$v(Y) \subseteq v(X) - v(Z)$$

$$v(Z) \subseteq v(X) - v(Y)$$

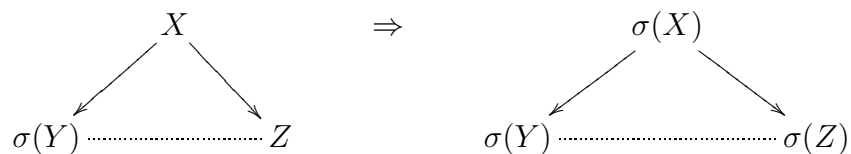
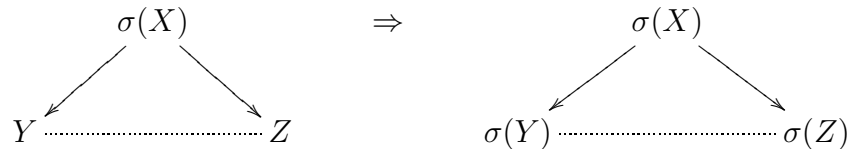
E portanto,

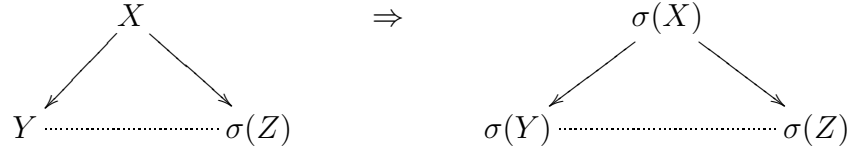
$$\forall \sigma(X) \subseteq v(X) : \exists \sigma(Y) \subseteq v(Y), \exists \sigma(Z) \subseteq v(Z) \mid \sigma(X) \asymp \sigma(Y) + \sigma(Z)$$

$$\forall \sigma(Y) \subseteq v(Y) : \exists \sigma(X) \subseteq v(X), \exists \sigma(Z) \subseteq v(Z) \mid \sigma(Y) \asymp \sigma(X) - \sigma(Z)$$

$$\forall \sigma(Z) \subseteq v(Z) : \exists \sigma(X) \subseteq v(X), \exists \sigma(Y) \subseteq v(Y) \mid \sigma(Z) \asymp \sigma(X) - \sigma(Y)$$

Assim, podemos encontrar uma valoração degenerada exata iniciando em qualquer vértice de $\mathcal{D}(\Phi)$, implicando o valor das variáveis restantes:





Dessa forma, obtemos um método para encontrar a valoração degenerada de uma fórmula Φ diferente do apresentado em [17] e, conseqüentemente, do algoritmo 5. Nesse novo método, descrito pelo algoritmo 6, partimos de uma única variável $W \in V(\Phi)$ e inferimos o valor das demais variáveis.

Algoritmo 6: Encontra Valoração Degenerada Exata (exatoEVD)

Entrada: (Φ, v, W)

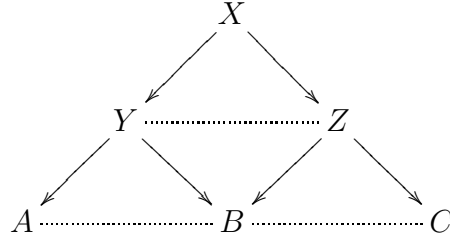
Saída: modelo degenerado $\sigma \subseteq v$ de Φ ou *insatisfável*

- 1 $\sigma \leftarrow \emptyset$
 - 2 Seja $w \in v(W)$ qualquer
 - 3 $\sigma \leftarrow \sigma \cup (W \mapsto [w, w])$
 - 4 **para todo** $R \in A(\Phi) \mid \exists! X \in V(R)$ tal que $\sigma(X) \neq \diamond$ **faça**
 - 5 Seja $(X \asymp Y \circ Z) \equiv R$ tal que $\sigma(X) \neq \diamond$ e $\circ \in \Psi$
 - 6 Seja $y \in (v(Y) \cap (\sigma(X) \bullet v(Z)))$ qualquer
 - 7 $\sigma \leftarrow \sigma \cup (Y \mapsto [y, y])$
 - 8 $\sigma \leftarrow \sigma \cup (Z \mapsto \sigma(X) \odot \sigma(Y))$
 - 9 **fim**
 - 10 **se** $\forall X \in V(\Phi) : \sigma(X) \neq \diamond$ **então**
 - 11 **retorna** σ
 - 12 **fim**
 - 13 **retorna** *insatisfável*
-

Embora este novo método seja capaz de encontrar uma valoração exata, a fórmula deve apresentar uma estrutura ainda mais restrita que a exigida pela condição 2 da definição 5.9: não apenas o digrafo $\mathcal{D}(\Phi)$ deve ser acíclico, mas também o grafo $\mathcal{G}(\Phi)$.

Exemplo 5.6 Seja $\Phi = (X \asymp Y \circ_1 Z) \wedge (Y \asymp A \circ_2 B) \wedge (Z \asymp B \circ_3 C)$ uma fórmula intervalar primitiva e v uma valoração sobre $V(\Phi)$.

$\mathcal{D}(\Phi) =$



Pelo EVD, definimos $\sigma(A) \subseteq v(A)$, $\sigma(B) \subseteq v(B)$ e $\sigma(C) \subseteq v(C)$ quaisquer e inferimos $\sigma(Y) = \sigma(A) \circ_2 \sigma(B)$, $\sigma(Z) = \sigma(B) \circ_3 \sigma(C)$ e $\sigma(X) = \sigma(Y) \circ_1 \sigma(Z)$, encontrando possíveis erros de valoração não nulos $\varepsilon_\sigma^v(X)$, $\varepsilon_\sigma^v(Y)$ e $\varepsilon_\sigma^v(Z)$.

Por outro lado, se iniciarmos a valoração por um único ponto, por exemplo pela raiz X , definimos $\sigma(X) \subseteq v(X)$ qualquer e inferimos

$$\sigma(Y) \subseteq (v(Y) \cap (\sigma(X) \bullet_1 v(Z)))$$

$$\sigma(Z) = \sigma(X) \odot_1 \sigma(Y)$$

Em seguida, inferimos

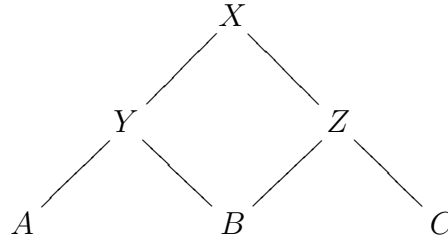
$$\sigma(A) \subseteq (v(A) \cap (\sigma(Y) \bullet_2 v(B)))$$

$$\sigma(B) = \sigma(Y) \odot_2 \sigma(A)$$

Finalmente, temos um estado no qual $\sigma(Z)$ e $\sigma(B)$ estão definidos e $\sigma(C) = \diamond$. Infelizmente, não existem garantias de que $\forall z \in v(Z), \forall b \in v(B) : \exists c \in v(C) \mid z = b \circ_3 c$ e, assim, o método $\text{exatoEVD}(\Phi, v)$ devolve *insatisfável*, pois não encontrou um modelo para Φ com erro nulo.

Mais precisamente, não basta que $\mathcal{D}(\Phi)$ seja acíclico, mas também o grafo $\mathcal{G}(\Phi)$. Nesse exemplo, $\mathcal{G}(\Phi)$ é cíclico:

$\mathcal{G}(\Phi) =$

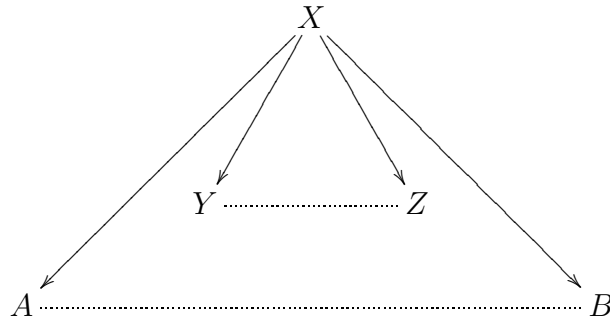


□

Com esse método de valoração também podemos excluir a condição 1 da definição 5.9, pois mesmo em fórmulas com duas ou mais restrições com a mesma variável na cabeça é possível encontrar uma valoração degenerada.

Exemplo 5.7 *Seja $\Phi = (X \asymp Y \circ_1 Z) \wedge (X \asymp A \circ_2 B)$ uma fórmula intervalar primitiva e v uma valoração sobre $V(\Phi)$.*

$\mathcal{G}(\Phi) =$



Pelo método exato $\text{EVD}(\Phi, v)$, primeiramente, definimos $\sigma(X) \subseteq v(X)$ qualquer. Em seguida, obtemos

$$\sigma(Y) \subseteq (v(Y) \cap (\sigma(X) \bullet_1 v(Z)))$$

$$\sigma(Z) = \sigma(X) \odot_1 \sigma(Y)$$

Por fim, obtemos

$$\sigma(A) \subseteq (v(A) \cap (\sigma(X) \bullet_2 v(B)))$$

$$\sigma(B) = \sigma(X) \odot_2 \sigma(A)$$

□

Dessa forma, podemos resolver todas as instâncias POI $(\Phi, v_0, \epsilon, \mathbf{c}, F)$ tais que $\mathcal{G}(\Phi)$ é acíclico. Aplicando a consistência intervalar em v_0 obtemos uma valoração v consistente no arco de Φ tal que $v(F) \in \{[\underline{f}, \bar{f}], [\underline{f}, \bar{f}], (\underline{f}, \bar{f}), (\underline{f}, \bar{f})\}$. Como o mínimo de um intervalo $v(F) \in \{[\underline{f}, \bar{f}], [\underline{f}, \bar{f}]\}$ é o intervalo degenerado $[\underline{f}, \underline{f}]$, basta definirmos $\sigma(F) = [\underline{f}, \underline{f}]$ no algoritmo exatoEVD e inferir o valor degenerado das variáveis restantes.

Observe que mais do que encontrar o modelo que minimiza o intervalo objetivo, esse método devolve uma valoração degenerada, garantindo, pelo teorema 4.2, a existência de um conjunto de valores reais que satisfaz todas as restrições do problema. Dessa forma, podemos utilizar tal método para resolver problemas intervalares que modelam problemas de Programação Não Linear.

Teorema 5.2 *Dada uma fórmula intervalar primitiva Φ tal que $F \in V(\Phi)$, uma valoração inicial v_0 e uma valoração $v \subseteq v_0$ consistente no arco de Φ tal que $v(F) \in \{[\underline{f}, \bar{f}], [\underline{f}, \bar{f}]\}$, $\forall X \in V(\Phi) : v(X) \neq \emptyset$ e $\nexists v' \supset v \mid v' \subseteq v_0$ é consistente no arco de Φ (isto é, v é o refinamento maximal de v_0 consistente no arco de Φ). Se o grafo $\mathcal{G}(\Phi)$ é acíclico, então é possível encontrar uma valoração degenerada $\sigma \subseteq v_0$ que satisfaz Φ e minimiza F .*

Prova. Seja Φ uma fórmula intervalar primitiva tal que $F \in V(\Phi)$, v_0 uma valoração inicial e $v \subseteq v_0$ uma valoração consistente no arco de Φ tal que $v(F) \in \{[\underline{f}, \bar{f}], [\underline{f}, \bar{f}]\}$, $\forall X \in V(\Phi) : v(X) \neq \emptyset$ e $\nexists v' \supset v \mid v' \subseteq v_0$ é consistente no arco de Φ .

Seja $\sigma(F) = [\underline{f}, \underline{f}]$. Como $v \supseteq v'$, para toda valoração $v' \subseteq v_0$ consistente no arco de Φ , então não existe uma expansão do intervalo $\sigma(F)$ que satisfaz Φ e v_0 . Portanto, o menor valor real possível para F consistente é \underline{f} . Logo, $\sigma(F) = [\underline{f}, \underline{f}]$ minimiza F . Dessa forma, basta provar que a partir de $\sigma(F)$ é possível valorar todas as variáveis de $V(\Phi)$.

Suponha que $\mathcal{G}(\Phi)$ é acíclico e conexo e seja $R_1 = (F \asymp X_1 \circ_1 Y_1) \in A(\Phi)$ a restrição que contém F . Como $\sigma(F) = [\underline{f}, \underline{f}]$ e $\underline{f} \in v(F)$, então, pelo teorema 2.1, existem $x_1 \in v(X_1)$ e $y_1 \in v(Y_1)$ tais que $\underline{f} = x_1 \circ_1 y_1$ e, portanto, a valoração degenerada $\sigma(X_1) = [x_1, x_1]$ e $\sigma(Y_1) = [y_1, y_1]$ satisfaz a restrição $F \asymp X_1 \circ_1 Y_1$.

Como $\mathcal{G}(\Phi)$ é conexo, então existe uma ordenação de restrições

$$\Omega = (R_1 = (F \asymp X_1 \circ_1 Y_1), R_2, \dots, R_{|A(\Phi)|})$$

tal que

$$\forall R_i \in \Omega : |V(R_i) \cap V(R_j)| > 0, \text{ para algum } j < i$$

Além disso,

$$\forall R_i \in \Omega : |V(R_i) \cap \bigcup_{j=1}^{i-1} V(R_j)| \leq 1$$

Do contrário, ou $\exists j < i \mid E(\mathcal{G}(R_j)) \cap E(\mathcal{G}(R_i)) \neq \emptyset$, e Φ não é primitiva, ou existem dois pontos de interseção entre os grafos $\mathcal{G}(R_i)$ e $\mathcal{G}(\Phi \setminus \bigcup_{k=i+1}^{|A(\Phi)|} R_k)$ e, portanto, $\mathcal{G}(\Phi)$ é cíclico.

Como já definimos $\sigma(F)$, $\sigma(X_1)$ e $\sigma(Y_1)$, vamos provar, por indução forte, que é possível definir σ para toda variável $X_i \in V(R_i)$, para $i \geq 2$.

- *Caso Base:* Para $R_2 \in \Omega$, como $|V(R_2) \cap V(R_1)| = 1$, existe uma única variável $X_2 \in V(R_2) \mid \sigma(X_2) \neq \diamond$. Seja $(X_2 \asymp Y_2 \circ'_2 Z_2) \equiv R_2$ tal que $\sigma(X_2) \neq \diamond$ e $\circ'_2 \in \mathbb{U}$. Seja também $y_2 \in (v(Y_2) \cap (\sigma(X_2) \bullet'_2 v(Z_2)))$ qualquer. Então, definimos $\sigma(Y_2) = [y_2, y_2]$ e $\sigma(Z_2) = \sigma(X_2) \circ'_2 \sigma(Y_2)$.
- *Hipótese de Indução:* Para todo $1 \leq i \leq k$, as variáveis de R_i têm valor definido em σ .
- *Passo Indutivo:* Para R_{k+1} , temos que $|V(R_{k+1}) \cap V(R_j)| > 0$, para algum $j < k + 1$, e $|V(R_{k+1}) \cap \bigcup_{j=1}^{k+1-1} V(R_j)| \leq 1$. Logo, existe somente uma variável $X_{k+1} \in V(R_{k+1}) \mid X_{k+1} \in V(R_j)$, para $j < k + 1$. Pela H.I., $\sigma(X_{k+1}) \neq \diamond$. Da mesma forma, Seja $(X_{k+1} \asymp Y_{k+1} \circ'_{k+1} Z_{k+1}) \equiv R_{k+1}$ tal que $\sigma(X_{k+1}) \neq \diamond$ e $\circ'_{k+1} \in \mathbb{U}$. Seja também $y_{k+1} \in (v(Y_{k+1}) \cap (\sigma(X_{k+1}) \bullet'_{k+1} v(Z_{k+1})))$ qualquer. Então, definimos $\sigma(Y_{k+1}) = [y_{k+1}, y_{k+1}]$ e $\sigma(Z_{k+1}) = \sigma(X_{k+1}) \circ'_{k+1} \sigma(Y_{k+1})$.

Como Ω contém todas as restrições de $A(\Phi)$, então $\sigma \subseteq v_0$ satisfaz Φ e minimiza F .

□

Note que o algoritmo 6 é consequência direta da prova deste teorema. Se modificarmos as linhas 2 e 3, de forma que o algoritmo sempre escolha o intervalo objetivo F e o valor com o seu mínimo em $v(F) = [\underline{f}, \overline{f}]$ (isto é, $\sigma(F) = [\underline{f}, \underline{f}]$), então podemos utilizá-lo para encontrar o modelo mínimo de fórmulas intervalares primitivas cujo grafo $\mathcal{G}(\Phi)$ é acíclico.

Exemplo 5.8 *Seja uma instância POI $(\Phi, v_0, \epsilon, \mathbf{c}, F)$ tal que $\Phi = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6 \wedge R_7$ é uma fórmula intervalar primitiva dada por*

$$R_1 = (F \asymp X + Y)$$

$$R_2 = (X \asymp A \times B)$$

$$R_3 = (Y \asymp C + D)$$

$$R_4 = (Y \asymp Z \times T)$$

$$R_5 = (W \asymp Z \wedge V)$$

$$R_6 = (T \asymp E \times G)$$

$$R_7 = (U \asymp G - H)$$

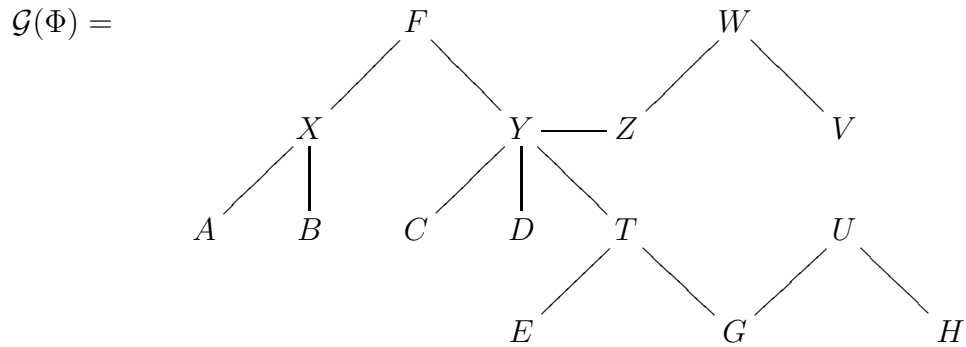
O erro $\epsilon \geq 0$ é qualquer, \mathbf{c} = consistência no arco e a valoração inicial v_0 é dada por

$$\begin{array}{lll} v_0(F) = [-\infty, \infty] & v_0(V) = [2, 2] & v_0(A) = [1, 2] \\ v_0(G) = [-1, 1] & v_0(W) = [-\infty, \infty] & v_0(B) = [0, \infty] \\ v_0(H) = [-2, 7] & v_0(X) = [-\infty, \infty] & v_0(C) = [0, 1] \\ v_0(T) = [-\infty, \infty] & v_0(Y) = [-\infty, \infty] & v_0(D) = [1, 2] \\ v_0(U) = [0, 4] & v_0(Z) = [2, 2] & v_0(E) = [2, 4] \end{array}$$

Aplicando a consistência intervalar em Φ sob a valoração inicial v_0 (algoritmo 3) obtemos uma valoração v tal que, pelo teorema 4.1, $v \subseteq v_0$ é consistente no arco de Φ e $\nexists v' \supset v \mid v' \subseteq v_0$ também é consistente no arco de Φ .

$$\begin{array}{lll}
v(F) = [-8, \infty] & v(V) = [2, 2] & v(A) = [1, 2] \\
v(G) = [-1, 1] & v(W) = [4, 4] & v(B) = [0, \infty] \\
v(H) = [-2, 1] & v(X) = [0, \infty] & v(C) = [-10, 7] \\
v(T) = [-4, 4] & v(Y) = [-8, 8] & v(D) = [1, 2] \\
v(U) = [0, 3] & v(Z) = [2, 2] & v(E) = [2, 4]
\end{array}$$

Dessa forma, $v(F) = [-8, \infty]$ e $\forall X \in V(\Phi) : v(X) \neq \emptyset$. Além disso, $\mathcal{G}(\Phi)$ é conexo e acíclico.



Então, pelo teorema 5.2, é possível encontrar uma valoração degenerada $\sigma \subseteq v_0$ que satisfaz Φ e minimiza F .

Definindo $\sigma(F) = [-8, -8]$, inferimos

$$\sigma(X) = [x, x] \mid x \in (v(X) \cap (\sigma(F) - v(Y))) \Rightarrow \sigma(X) = [0, 0]$$

$$\sigma(Y) = \sigma(F) - \sigma(X) = [-8, -8]$$

Em seguida, inferimos

$$\sigma(B) = [b, b] \mid b \in (v(B) \cap (\sigma(X) \div v(A))) \Rightarrow \sigma(B) = [0, 0]$$

$$\sigma(A) = \sigma(X) \div \sigma(B)$$

Nesse caso, como $\sigma(X) \div \sigma(B) = [-\infty, \infty]$, fazemos $\sigma(A) = [a, a] \mid a \in v(A)$, por exemplo, $\sigma(A) = [1, 1]$.

Pelo outro lado, inferimos

$$\sigma(Z) = [z, z] \mid z \in (v(Z) \cap (\sigma(Y) \div v(T))) \Rightarrow \sigma(Z) = [2, 2]$$

$$\sigma(T) = \sigma(Y) \div \sigma(Z) = [-4, -4]$$

$$\sigma(C) = [c, c] \mid c \in (v(C) \cap (\sigma(Y) - v(D))) \Rightarrow \sigma(C) = [-9, -9]$$

$$\sigma(D) = \sigma(Y) - \sigma(C) = [1, 1]$$

Prosseguindo com o método, inferimos

$$\sigma(W) = [w, w] \mid w \in (v(W) \cap (\sigma(Z) \wedge v(V))) \Rightarrow \sigma(W) = [4, 4]$$

$$\sigma(V) = \sigma(W) \wedge \sigma(V) = [-\infty, \infty] \quad (\sigma(V) = [2, 2])$$

$$\sigma(E) = [e, e] \mid e \in (v(E) \cap (\sigma(T) \div v(G))) \Rightarrow \sigma(E) = [4, 4]$$

$$\sigma(G) = \sigma(T) \div \sigma(E) = [-1, -1]$$

$$\sigma(U) = [u, u] \mid u \in (v(U) \cap (\sigma(G) - v(H))) \Rightarrow \sigma(U) = [0, 0]$$

$$\sigma(H) = \sigma(G) - \sigma(U) = [-1, -1]$$

Por fim, obtemos a seguinte valoração $\sigma \subseteq v_0$ que satisfaz Φ e minimiza F . Como σ é degenerada, ainda garantimos a existência de valores reais que satisfazem todas as restrições reais do problema.

$$\begin{array}{lll} \sigma(F) & = & [-8, -8] & \sigma(V) & = & [2, 2] & \sigma(A) & = & [1, 1] \\ \sigma(G) & = & [-1, -1] & \sigma(W) & = & [4, 4] & \sigma(B) & = & [0, 0] \\ \sigma(H) & = & [-1, -1] & \sigma(X) & = & [0, 0] & \sigma(C) & = & [-9, -9] \\ \sigma(T) & = & [-4, -4] & \sigma(Y) & = & [-8, -8] & \sigma(D) & = & [1, 1] \\ \sigma(U) & = & [0, 0] & \sigma(Z) & = & [2, 2] & \sigma(E) & = & [4, 4] \end{array}$$

□

5.3 Otimização Aproximada

Embora o método proposto seja capaz de encontrar a valoração degenerada que minimiza o intervalo objetivo, as fórmulas de entrada devem apresentar uma estrutura bastante específica. Para que possamos resolver outras classes de fórmulas, vamos observar novamente a definição 5.9 (*fórmula fortemente satisfeita*), na qual substituímos a condição 3 pela consistência no arco. Essa modificação possibilita o método EVD a valorar uma variável com um intervalo degenerado que não é um refinamento da valoração inicial, gerando um erro de valoração não nulo.

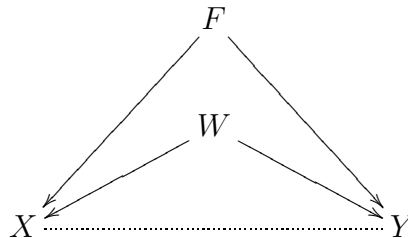
Nesse sentido, vamos definir a *otimização aproximada* como uma valoração que, embora minimize o intervalo objetivo, possa gerar erros de valoração não nulos para algumas variáveis da fórmula de entrada.

A justificativa para tal aproximação é que, como trata-se de uma modelagem intervalar, que permite aproximações e limitantes às variáveis e constantes extraídas de um domínio real, a valoração inicial poderia ser estendida sem perda considerável de informação ou possibilidade de solução real.

Alternativamente, o método de otimização aproximada que será apresentado torna-se um método de otimização exata se garantirmos a condição 3 da definição 5.9, como ocorre no algoritmo iSAT.

Exemplo 5.9 *Seja $\Phi = (F \asymp X + Y) \wedge (W \asymp X - Y)$ uma fórmula intervalar primitiva e uma valoração inicial v_0 consistente no arco de Φ tal que $v_0(F) = [1, 5]$, $v_0(X) = [0, 2]$, $v_0(Y) = [0, 4]$ e $v_0(W) = [-4, 0]$.*

$\mathcal{D}(\Phi) =$



Aplicando $\text{exatoEVD}(\Phi, v)$, definimos $\sigma(F) = [1, 1]$ e inferimos, por exemplo, $\sigma(X) = [1, 1]$ e $\sigma(Y) = [0, 0]$. Como a restrição restante ($W \asymp X - Y$) possui duas variáveis valoradas, não conseguimos inferir o valor da terceira, devolvendo *insatisfável* (o grafo

$\mathcal{G}(\Phi)$ é cíclico).

No entanto, se executarmos, nesse estado, o método EVD, inferimos $\sigma(W) = \sigma(X) - \sigma(Y) = [1, 1]$. Se tivéssemos garantido que $v_0(W) = v_0(X) - v_0(Y)$, então $\sigma(W) \subseteq v_0(W)$. Como apenas garantimos a consistência no arco, então $\sigma(W) \not\subseteq v_0(W)$, pois $[1, 1] \not\subseteq [-4, 0]$. Nesse caso, obtemos um erro de valoração $\varepsilon_\sigma^{v_0}(W) = 1$.

Nessa valoração, aplicamos ambos os métodos. Primeiramente o método exato, necessário para minimizar o intervalo objetivo, e em seguida o método aproximado, valorando as demais variáveis. Assim, vamos propor um novo método, que integra as duas formas de valoração. Da mesma forma, existem restrições quanto às características das fórmulas que este novo método é aplicável, provenientes das limitações dos métodos anteriores.

Em um problema POI $(\Phi, v_0, \epsilon, \mathbf{c}, F)$, se o grafo $\mathcal{A}_\Phi(F)$ é acíclico, podemos valorar todas as variáveis representadas pelos vértices de $\mathcal{A}_\Phi(F)$ com o método exatoEVD, minimizando F . Se o digrafo restante também for acíclico, e não possuir vértices com mais de dois filhos, então podemos valorar as variáveis restantes aplicando o método EVD.

Mais especificamente, o digrafo restante de $\mathcal{A}_\Phi(F)$ é o subgrafo de $\mathcal{D}(\Phi)$ induzido pela união dos vértices que não estão em $\mathcal{A}_\Phi(F)$ com os vértices de $\mathcal{A}_\Phi(F)$ que pertencem a arestas que não estão em $\mathcal{A}_\Phi(F)$:

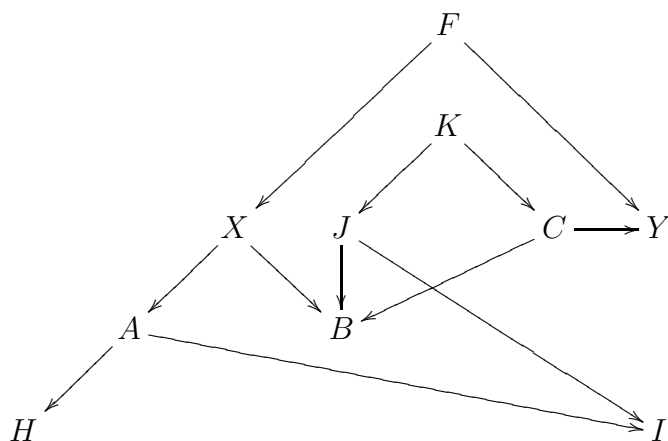
$$(P(\mathcal{D}(\Phi)) \setminus P(\mathcal{A}_\Phi(F))) \cup \{Y \mid (X, Y) \in E(\mathcal{D}(\Phi)), X \notin P(\mathcal{A}_\Phi(F)) \text{ e } Y \in P(\mathcal{A}_\Phi(F))\}$$

Exemplo 5.10 *Seja Φ a seguinte fórmula intervalar primitiva:*

$$\Phi = (F \asymp X \circ_1 Y) \wedge (X \asymp A \circ_2 B) \wedge (C \asymp Y \circ_3 B) \wedge (A \asymp H \circ_4 I) \wedge (J \asymp B \circ_5 I) \wedge (K \asymp C \circ_6 J)$$

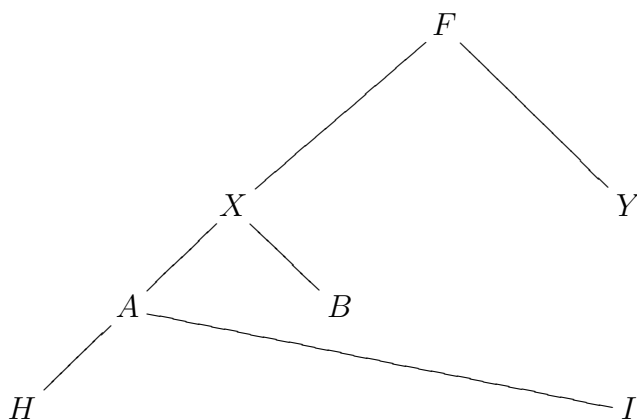
Observando o digrafo $\mathcal{D}(\Phi)$, vemos que os vértices alcançáveis a partir de F são: F , Y , X , B , A , I e H .

$\mathcal{D}(\Phi) =$



Portanto, temos que

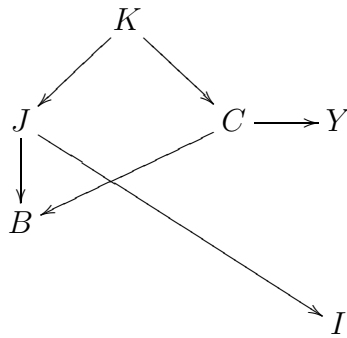
$\mathcal{A}_\Phi(F) =$



Além disso, seja $\mathcal{R}(\Phi)$ o digrafo restante, induzido pelos vértices

$$\begin{aligned}
 & (P(\mathcal{D}(\Phi)) \setminus P(\mathcal{A}_\Phi(F))) \cup \{Y \mid (X, Y) \in E(\mathcal{D}(\Phi)), X \notin P(\mathcal{A}_\Phi(F)) \text{ e } Y \in P(\mathcal{A}_\Phi(F))\} \\
 &= (\{F, K, X, J, C, Y, A, B, H, I\} \setminus \{F, X, Y, A, B, H, I\}) \cup \{B, I, Y\} \\
 &= \{K, J, C\} \cup \{B, I, Y\} \\
 &= \{K, J, C, B, I, Y\}
 \end{aligned}$$

$\mathcal{R}(\Phi) =$



Como o grafo $\mathcal{A}_\Phi(F)$ é acíclico, podemos aplicar a consistência no arco sobre uma dada valoração v_0 , obtendo uma valoração consistente v , e, pelo teorema 5.2, encontrar uma valoração degenerada σ que minimiza F tal que

$$\varepsilon_\sigma^{v_0}(F) = \varepsilon_\sigma^{v_0}(X) = \varepsilon_\sigma^{v_0}(Y) = \varepsilon_\sigma^{v_0}(A) = \varepsilon_\sigma^{v_0}(B) = \varepsilon_\sigma^{v_0}(H) = \varepsilon_\sigma^{v_0}(I) = 0$$

Por outro lado, como $\mathcal{R}(\Phi)$ também é acíclico e não existem vértices com mais de dois filhos, então podemos aplicar o método EVD:

$$\sigma(J) = \sigma(B) \circ_5 \sigma(I)$$

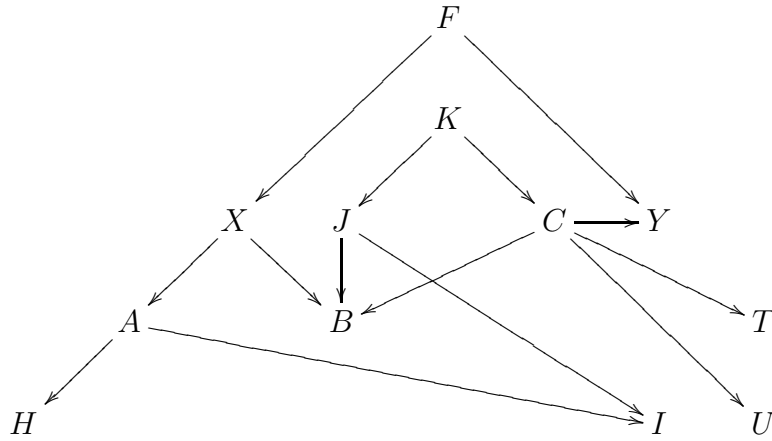
$$\sigma(C) = \sigma(Y) \circ_3 \sigma(B)$$

$$\sigma(K) = \sigma(C) \circ_6 \sigma(J)$$

No entanto, como garantimos apenas a consistência no arco, podemos gerar erros de valoração não nulos $\varepsilon_\sigma^{v_0}(J)$, $\varepsilon_\sigma^{v_0}(C)$ e $\varepsilon_\sigma^{v_0}(K)$.

Agora suponha que adicionamos à fórmula Φ uma nova restrição $(C \asymp T \circ_7 U)$ que não altera a consistência da valoração v .

$\mathcal{D}(\Phi) =$



Como já valoramos todas as variáveis, exceto T e U , podemos inferir seus valores, pelo método exatoEVD. No entanto, como existe um erro $\varepsilon_\sigma^{v_0}(C)$ possivelmente não nulo, as valorações $\sigma(T)$ e $\sigma(U)$ não são necessariamente exatas. Mesmo assim, fomos capazes de encontrar a valoração degenerada para $\mathcal{R}(\Phi)$ com o vértice C com mais de dois filhos.

Generalizando, podemos encontrar a valoração mesmo que todo vértice $X \in (P(\mathcal{G}(\Phi)) \setminus P(\mathcal{A}_\Phi(F)))$ seja raiz de um subgrafo acíclico $\mathcal{A}_\Phi(X)$, pois, como definimos $\sigma(X)$ pelo método EVD, inferimos a valoração dos demais vértices de $\mathcal{A}_\Phi(X)$ através do método exatoEVD.

Definição 5.12 *Dada uma fórmula intervalar primitiva Φ tal que $F \in V(\Phi)$, dizemos que o grafo $\mathcal{G}(\Phi)$ é uma **cascata de árvores segundo F** , se $\mathcal{G}(\Phi)$ é constituído de um conjunto ordenado de subgrafos acíclicos e disjuntos $(\mathcal{A}_\Phi(F), \mathcal{A}_{\Phi \setminus R_1}(X_1), \dots, \mathcal{A}_{\Phi \setminus R_n}(X_n))$, onde, para todo $1 \leq i \leq n$,*

$$h(R_i) = X_i$$

$$t(R_i) \subseteq \left(P(\mathcal{A}_\Phi(F)) \cup \bigcup_{j=1}^{i-1} P(\mathcal{A}_{\Phi \setminus R_j}(X_j)) \right)$$

E, além disso, os subgrafos são conectados por um conjunto de arestas

$$\Upsilon = E(\mathcal{G}(\Phi)) \setminus \left(E(\mathcal{A}_\Phi(F)) \cup \bigcup_{i=1}^n E(\mathcal{A}_{\Phi \setminus R_i}(X_i)) \right)$$

tal que, para todo $1 \leq i \leq n$,

$$|\{X \mid (X, Y) \in \Upsilon \text{ e } X \in V(\mathcal{A}_{\Phi \setminus R_i}(X_i))\}| \leq 1$$

$$|\{(X, Y) \in \Upsilon \mid X \in V(\mathcal{A}_{\Phi \setminus R_i}(X_i))\}| \leq 2$$

Intuitivamente, uma cascata de árvores é um conjunto ordenado de árvores conectadas por arestas de mesma direção, isto é, de forma que todas as arestas com origem em uma árvore $\mathcal{A}_{\Phi \setminus R_i}(X_i)$ têm destino em algum antecessor $\mathcal{A}_{\Phi \setminus R_j}(X_j)$ de $\mathcal{A}_{\Phi \setminus R_i}(X_i)$ ($j < i$) ou no próprio $\mathcal{A}_{\Phi \setminus R_i}(X_i)$. Além disso, no máximo duas arestas têm origem em $\mathcal{A}_{\Phi \setminus R_i}(X_i)$ (e ambas no mesmo vértice X_i) e destino em algum de seus antecessores.

Teorema 5.3 *Dada uma fórmula intervalar primitiva Φ tal que $F \in V(\Phi)$, uma valoração inicial v_0 e uma valoração $v \subseteq v_0$ consistente no arco de Φ tal que $v(F) \in \{[\underline{f}, \bar{f}], [\underline{f}, \bar{f}]\}$, $\forall X \in V(\Phi) : v(X) \neq \emptyset$ e $\nexists v' \supset v \mid v' \subseteq v_0$ é consistente no arco de Φ (isto é, v é o refinamento maximal de v_0 consistente no arco de Φ). Se o grafo $\mathcal{G}(\Phi)$ é uma cascata de árvores segundo F , então é possível encontrar uma valoração degenerada σ que satisfaz Φ e minimiza F .*

Prova. Seja Φ uma fórmula intervalar primitiva e v uma valoração conforme as condições enunciadas no teorema. Queremos provar que é possível encontrar uma valoração degenerada σ que satisfaz Φ e minimiza F .

Como $\mathcal{G}(\Phi)$ é constituído de um conjunto ordenado de subgrafos acíclicos e disjuntos $(\mathcal{A}_{\Phi}(F), \mathcal{A}_{\Phi \setminus R_1}(X_1), \dots, \mathcal{A}_{\Phi \setminus R_n}(X_n))$, conectados por um conjunto de arestas Υ , basta provar que é possível valorar todas as variáveis representadas pelo grafo $\mathcal{A}_{\Phi}(F)$, minimizando F , e também pelos demais grafos $\mathcal{A}_{\Phi \setminus R_i}(X_i)$, para $1 \leq i \leq n$.

Como $\mathcal{A}_{\Phi}(F)$ é um grafo conexo e acíclico e contém F , então, pelo teorema 5.2, é possível obter uma valoração degenerada σ para as variáveis representadas por $P(\mathcal{A}_{\Phi}(F))$ que minimiza F .

Vamos provar, por indução forte, que é possível estender σ aos demais grafos do conjunto.

- *Caso Base:* Para $\mathcal{A}_{\Phi \setminus R_1}(X_1)$, temos que $h(R_1) = X_1$ e $t(R_1) \subseteq P(\mathcal{A}_{\Phi}(F))$. Como σ está definida para as variáveis representadas por $P(\mathcal{A}_{\Phi}(F))$, então também está definida para $t(R_1)$. Seja $R_1 = (X_1 \asymp Y_1 \circ_1 Z_1)$, para algum $\circ_1 \in \mathbb{U}$; definimos $\sigma(X_1) = \sigma(Y_1) \circ_1 \sigma(Z_1)$. Como $\mathcal{A}_{\Phi \setminus R_1}(X_1)$ é disjunto de $\mathcal{A}_{\Phi}(F)$ e, pelas restrições sobre as arestas que conectam os subgrafos, no máximo duas arestas têm origem nos vértices de $\mathcal{A}_{\Phi \setminus R_1}(X_1)$ (e ambas no mesmo vértice), então entre suas variáveis apenas X_1 tem valor definido em σ . Como $\mathcal{A}_{\Phi \setminus R_1}(X_1)$ é acíclico e conexo, podemos inferir σ para as demais variáveis representadas neste grafo, pelo teorema 5.2.
- *Hipótese de Indução:* Para todo $1 \leq i \leq k$, as variáveis representadas pelo conjunto $P(\mathcal{A}_{\Phi \setminus R_i}(X_i))$ têm valor definido em σ .
- *Passo Indutivo:* Para $\mathcal{A}_{\Phi \setminus R_{k+1}}(X_{k+1})$, temos que $h(R_{k+1}) = X_{k+1}$ e

$$t(R_{k+1}) \subseteq \left(P(\mathcal{A}_{\Phi}(F)) \cup \bigcup_{i=1}^{k+1-1} P(\mathcal{A}_{\Phi \setminus R_i}(X_i)) \right)$$

Pela H.I., σ está definida para as variáveis representadas por $P(\mathcal{A}_{\Phi \setminus R_i}(X_i))$, para todo $1 \leq i \leq n$, e portanto, também está definida para $t(R_{k+1})$. Seja $R_{k+1} = (X_{k+1} \asymp Y_{k+1} \circ_{k+1} Z_{k+1})$, para algum $\circ_{k+1} \in \mathbb{U}$; definimos $\sigma(X_{k+1}) = \sigma(Y_{k+1}) \circ_{k+1} \sigma(Z_{k+1})$. Como $\mathcal{A}_{\Phi \setminus R_{k+1}}(X_{k+1})$ é disjunto de $\mathcal{A}_{\Phi}(F)$ e $\mathcal{A}_{\Phi \setminus R_i}(X_i)$ e, pelas restrições sobre as arestas que conectam os subgrafos, no máximo duas arestas têm origem nos vértices de $\mathcal{A}_{\Phi \setminus R_{k+1}}(X_{k+1})$ (e ambas no mesmo vértice), então entre suas variáveis apenas X_{k+1} tem valor definido em σ . Como $\mathcal{A}_{\Phi \setminus R_{k+1}}(X_{k+1})$ é acíclico e conexo, podemos inferir σ para as demais variáveis representadas neste grafo, pelo teorema 5.2.

□

Assim como o método exatoEVD é consequência direta do teorema 5.2, podemos extrair um algoritmo que aplica as valorações como descritas na prova deste teorema, a fim de encontrar o modelo ótimo, embora não necessariamente exato, de uma fórmula intervalar primitiva que contém um intervalo objetivo.

Algoritmo 7: Encontra Valoração Degenerada - Híbrido (hibridEVD)

Entrada: (Φ, v, W)

Saída: modelo degenerado $\sigma \subseteq v$ de Φ que minimiza $F \in V(\Phi)$ ou *insatisfável*

```

1  $\sigma \leftarrow \emptyset$ 
2  $\sigma \leftarrow \sigma \cup (F \mapsto [\underline{f}, \underline{f}])$ 
3 para todo  $R \in A(\Phi) \mid \sigma(X) \neq \diamond$ , para um único  $X \in V(R)$ , ou  $\sigma(h(R)) = \diamond$  faça
4   se  $\exists! X \in V(R) \mid \sigma(X) \neq \diamond$  então
5     Seja  $(X \asymp Y \circ Z) \equiv R$ 
6     Seja  $y \in (v(Y) \cap (\sigma(X) \bullet v(Z)))$  qualquer
7      $\sigma \leftarrow \sigma \cup (Y \mapsto [y, y])$ 
8      $\sigma \leftarrow \sigma \cup (Z \mapsto \sigma(X) \odot \sigma(Y))$ 
9   senão
10    Seja  $R = (X \asymp Y \circ Z)$ 
11     $\sigma \leftarrow \sigma \cup (X \mapsto \sigma(Y) \circ \sigma(Z))$ 
12  fim
13 fim
14 se  $\forall X \in V(\Phi) : \sigma(X) \neq \diamond$  então
15   retorna  $\sigma$ 
16 fim
17 retorna insatisfável

```

Exemplo 5.11 Seja uma instância POI $(\Phi, v_0, \epsilon, \mathbf{c}, F)$ tal que $\epsilon \geq 0$, \mathbf{c} = consistência no arco e $\Phi = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6 \wedge R_7 \wedge R_8 \wedge R_9$ é uma fórmula intervalar primitiva dada por

$$R_1 = (F \succ X \circ_1 A)$$

$$R_2 = (X \succ Y \circ_2 Z)$$

$$R_3 = (B \succ X \circ_3 A)$$

$$R_4 = (B \succ C \circ_4 H)$$

$$R_5 = (C \succ D \circ_5 E)$$

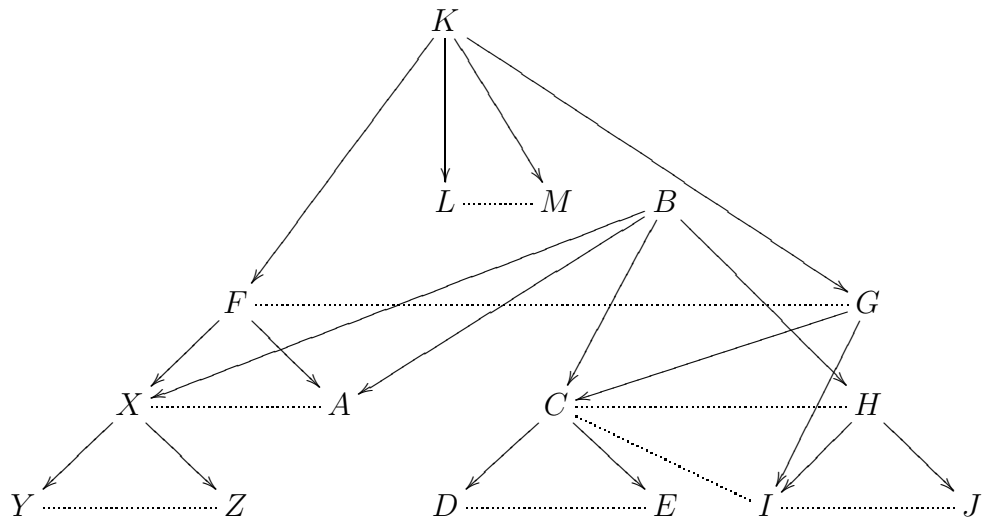
$$R_6 = (H \succ I \circ_6 J)$$

$$R_7 = (G \succ C \circ_7 I)$$

$$R_8 = (K \succ F \circ_8 G)$$

$$R_9 = (K \succ L \circ_9 M)$$

O digrafo $\mathcal{D}(\Phi)$ é dado por:

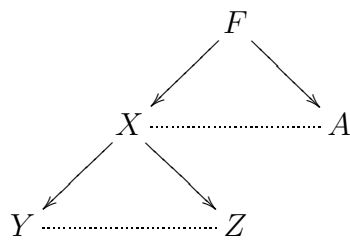


Seja $v \subseteq v_0$ o refinamento maximal de v_0 para Φ , obtido da aplicação da consistência no arco (algoritmo 3), tal que $v(F) = [\underline{f}, \bar{f}]$.

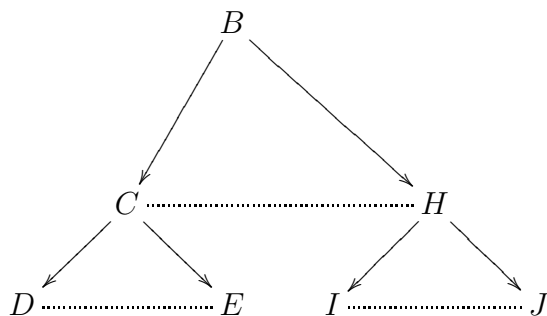
Podemos verificar que o grafo $\mathcal{G}(\Phi)$ é uma cascata de árvores segundo F , pois é constituído de um conjunto de subgrafos acíclicos e disjuntos dado por

$$(\mathcal{A}_\Phi(F), \mathcal{A}_{\Phi \setminus R_3}(B), \mathcal{A}_{\Phi \setminus R_7}(G), \mathcal{A}_{\Phi \setminus R_8}(K))$$

$$\mathcal{A}_\Phi(F) =$$



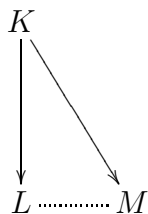
$$\mathcal{A}_{\Phi \setminus R_3}(B) =$$



$$\mathcal{A}_{\Phi \setminus R_7}(G) =$$

G

$$\mathcal{A}_{\Phi \setminus R_8}(K) =$$



Além disso, podemos verificar que os subgrafos são conectados por arestas que respeitam a definição de cascata de árvores:

$$\Upsilon = \{(B, X), (B, A), (G, C), (G, I), (K, F), (K, G)\}$$

$$|\{X \mid (X, Y) \in \Upsilon \text{ e } X \in V(\mathcal{A}_{\Phi \setminus R_3}(B))\}| = |\{B\}| \leq 1$$

$$|\{(X, Y) \in \Upsilon \mid X \in V(\mathcal{A}_{\Phi \setminus R_3}(B))\}| = |\{(B, X), (B, A)\}| \leq 2$$

$$|\{X \mid (X, Y) \in \Upsilon \text{ e } X \in V(\mathcal{A}_{\Phi \setminus R_7}(G))\}| = |\{G\}| \leq 1$$

$$|\{(X, Y) \in \Upsilon \mid X \in V(\mathcal{A}_{\Phi \setminus R_7}(G))\}| = |\{(G, C), (G, I)\}| \leq 2$$

$$|\{X \mid (X, Y) \in \Upsilon \text{ e } X \in V(\mathcal{A}_{\Phi \setminus R_9}(K))\}| = |\{K\}| \leq 1$$

$$|\{(X, Y) \in \Upsilon \mid X \in V(\mathcal{A}_{\Phi \setminus R_8}(K))\}| = |\{(K, F), (K, G)\}| \leq 2$$

Portanto, pelo teorema 5.3, é possível encontrar uma valoração degenerada $\sigma \subseteq v_0$ que satisfaz Φ e minimiza F .

Definindo $\sigma(F) = [\underline{f}, \underline{f}]$, inferimos

$$\sigma(X) = [x, x] \mid x \in (v(X) \cap (\sigma(F) \bullet_1 v(A)))$$

$$\sigma(A) = \sigma(F) \odot_1 \sigma(X)$$

E, em seguida, inferimos

$$\sigma(Y) = [y, y] \mid y \in (v(Y) \cap (\sigma(X) \bullet_2 v(Z)))$$

$$\sigma(Z) = \sigma(X) \odot_2 \sigma(Y)$$

Agora, nenhuma restrição tem apenas uma de suas variáveis sem valor definido em σ , mas a restrição $B \simeq X \circ_3 A$ tem apenas a cabeça ainda não valorada. Portanto, inferimos

$$\sigma(B) = \sigma(X) \circ_3 \sigma(A)$$

Prosseguindo com o método, inferimos

$$\sigma(C) = [c, c] \mid c \in (v(C) \cap (\sigma(B) \bullet_4 v(H)))$$

$$\sigma(H) = \sigma(B) \odot_4 \sigma(C)$$

$$\sigma(D) = [d, d] \mid d \in (v(D) \cap (\sigma(C) \bullet_5 v(E)))$$

$$\sigma(E) = \sigma(C) \odot_5 \sigma(D)$$

$$\sigma(I) = [i, i] \mid i \in (v(I) \cap (\sigma(H) \bullet_6 v(J)))$$

$$\sigma(J) = \sigma(H) \odot_6 \sigma(I)$$

Da mesma forma, a restrição $G \asymp C \circ_7 I$ tem apenas a cabeça ainda não valorada e, inferindo $\sigma(G) = \sigma(C) \circ_7 \sigma(I)$, o mesmo ocorre com a restrição $K \asymp F \circ_8 G$. Portanto, inferimos também $\sigma(K) = \sigma(F) \circ_8 \sigma(G)$

Por fim, inferimos os valores das variáveis restantes:

$$\sigma(L) = [l, l] \mid l \in (v(L) \cap (\sigma(K) \bullet_9 v(M)))$$

$$\sigma(M) = \sigma(K) \odot_9 \sigma(L)$$

□

5.4 Considerações

Neste capítulo, apresentamos o problema POI e uma forma de representação de fórmulas intervalares primitivas utilizando grafos e digrafos. Em seguida, reescrevemos o método de encontrar valoração degenerada para fórmulas acíclicas, apresentado em [17], sob uma abordagem por grafos, restringido à fórmulas cujo grafo que as representa é acíclico. Dessa forma, conseguimos obter a valoração degenerada que minimiza um intervalo objetivo dado.

A fim de estender esse algoritmo a outras classes de fórmulas, relaxamos o problema POI, possibilitando soluções aproximadas. Assim, mesmo uma solução na qual determinados intervalos não sejam refinamentos dos intervalos iniciais, pode ser aceita como solução

do problema real, dependendo da modelagem matemática e do contexto de aplicação. Por exemplo, se os intervalos iniciais estimam a pluviosidade futura de uma determinada região, pode ser admissível um erro de valoração não nulo, visto as aproximações e incertezas de seus limitantes superior e inferior.

É importante observar que as fórmulas intervalares representam relações dadas pela consistência intervalar utilizada: se os intervalos são consistentes, então são capazes de conter aproximações que satisfazem todas as restrições modeladas. Na consistência no arco, mais especificamente, existem três formas equivalentes (teorema 2.4) de representar uma restrição intervalar: $(X \asymp Y \circ Z)$, $(Y \asymp X \bullet Z)$ e $(Z \asymp X \odot Y)$, para $X, Y, Z \in \mathbb{I}$ e $\circ \in \mathcal{M}$ qualquer. Como uma fórmula é uma conjunção de restrições intervalares, somos capazes de definir quando duas fórmulas intervalares representam o mesmo conjunto de restrições, isto é, são *equivalentes*.

Definição 5.13 *Duas fórmulas Φ e Φ' são ditas **equivalentes**, e denotamos esse fato por $\Phi \equiv \Phi'$, se, e somente se,*

1. $\forall R \in A(\Phi) : \exists R' \in A(\Phi') \mid R \equiv R'$;
2. $\forall R' \in A(\Phi') : \exists R \in A(\Phi) \mid R' \equiv R$.

Exemplo 5.12 *As fórmulas $\Phi = (X \asymp Y + Z) \wedge (Y \asymp A \div B) \wedge (B \asymp Y \otimes A)$ e $\Phi' = (Z \asymp X - Y) \wedge (A \asymp Y \times B)$ são equivalentes, pois*

$$(X \asymp Y + Z) \equiv (Z \asymp X - Y)$$

$$(Y \asymp A \div B) \equiv (A \asymp Y \times B)$$

$$(B \asymp Y \otimes A) \equiv (A \asymp Y \times B)$$

□

Se duas fórmulas Φ e Φ' são equivalentes, então possuem o mesmo conjunto de modelos. Dessa forma, se ambas pertencem à instâncias POI $(\Phi, \nu_0, \epsilon, \mathbf{c}, F)$ e $(\Phi', \nu_0, \epsilon, \mathbf{c}, F)$, o modelo ν da fórmula Φ sob valoração ν_0 e consistência \mathbf{c} que minimiza F é igual ao modelo ν' da fórmula Φ' sob as mesmas condições.

Teorema 5.4 *Dada uma fórmula intervalar única Φ tal que $F \in V(\Phi)$, uma valoração inicial v_0 e uma valoração $v \subseteq v_0$ consistente no arco de Φ tal que $v(F) \in \{[\underline{f}, \bar{f}], [\underline{f}, \bar{f}]\}$, $\forall X \in V(\Phi) : v(X) \neq \emptyset$ e $\nexists v' \supset v \mid v' \subseteq v_0$ é consistente no arco de Φ (isto é, v é o refinamento maximal de v_0 consistente no arco de Φ). Se **existe uma fórmula** $\Phi' \equiv \Phi$ tal que $\mathcal{G}(\Phi')$ é uma cascata de árvores segundo F e $\nexists (X \asymp Y \wr W), (X \asymp Y \hat{\wr} W), (X \asymp Y \div Z) \in A(\Phi')$, tal que $v(Z) = [0, 0]$, então é possível encontrar uma valoração degenerada σ que satisfaz Φ e minimiza F .*

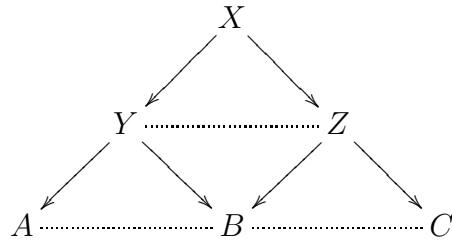
Prova. Seja Φ e v conforme as condições acima e $\Phi' \equiv \Phi$ também conforme as condições enunciadas. Como $\Phi' \equiv \Phi$, então $F \in V(\Phi')$ e $v \subseteq v_0$ é consistente no arco de Φ' , $v(F) \in \{[\underline{f}, \bar{f}], [\underline{f}, \bar{f}]\}$, $\forall X \in V(\Phi') : v(X) \neq \emptyset$ e $\nexists v' \supset v \mid v' \subseteq v_0$ é consistente no arco de Φ' . Logo, pelo teorema 5.3, é possível encontrar uma valoração degenerada σ que satisfaz Φ' e minimiza F . Novamente, como $\Phi' \equiv \Phi$, então σ também satisfaz Φ e minimiza F .

□

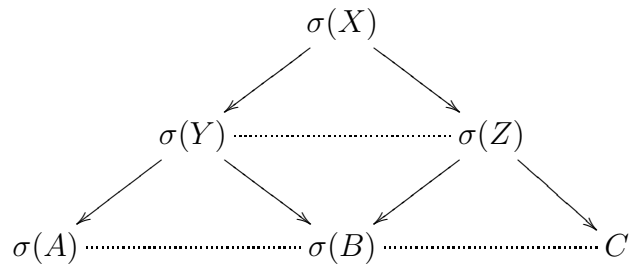
Note que o teorema acima vale porque estamos utilizando a consistência no arco. Se não tivessemos relaxado a definição 4.8, continuando com a exigência de que, $\forall R = (X \asymp Y \circ Z) \in A(\Phi) : v(X) \supseteq v(Y) \circ v(Z)$, uma mesma valoração v não seria consistente com duas restrições R e R' equivalentes. Por exemplo, se $v(X) = [1, 5]$, $v(Y) = [0, 3]$ e $v(Z) = [1, 2]$, temos que $v(X) = v(Y) + v(Z)$, mas $v(Y) \subset v(X) - v(Z)$, pois $[0, 3] \neq [-1, 4]$.

Além disso, como definimos a divisão por zero e o logaritmo de maneira não usual, não há garantias de que os intervalos consistentes contenham números reais que satisfaçam todas as restrições, pois o teorema 2.1 está definido apenas para \oplus e, conseqüentemente, para $\{\oplus, \otimes, \hat{\gamma}, \hat{\lambda}\}$.

Exemplo 5.13 *(Continuação do Ex. 5.5) Seja $\Phi = (X \asymp Y \circ_1 Z) \wedge (Y \asymp A \circ_2 B) \wedge (Z \asymp B \circ_3 C)$ uma fórmula intervalar e v uma valoração consistente no arco de $V(\Phi)$.*

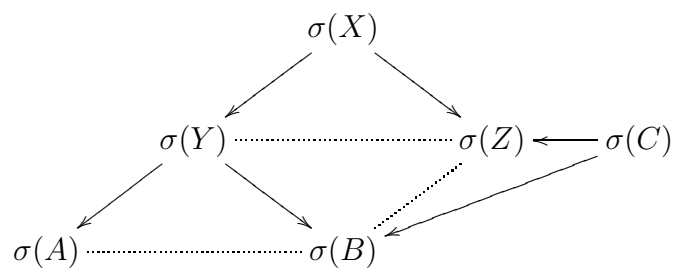
$\mathcal{D}(\Phi) =$


Como discutido no Ex. 5.6, não é possível encontrar uma valoração exata para Φ pelo método exatoEVD. Embora seja possível encontrar uma valoração aproximada, pelo método EVD, se X for o intervalo objetivo e quisermos minimizá-lo, temos que utilizar o método hibridEVD, encontrando um conflito: seja $\sigma(X) \subseteq v(X)$ o intervalo que o minimiza, inferimos os valores $\sigma(Y)$ e $\sigma(Z)$. Em seguida, inferimos $\sigma(A)$ e $\sigma(B)$, de forma que apenas $C \in V(Z \asymp B \circ_3 C)$ não tem valor definido em σ .

 $\mathcal{D}(\Phi) =$


Nesse estado, não podemos inferir $\sigma(C)$ e $\text{hibridEVD}(\Phi, v)$ retorna *insatisfável*. De fato, essa fórmula não contém as propriedades exigidas pelo teorema 5.3, pois o grafo $\mathcal{A}_\Phi(X)$ é cíclico.

Suponha, no entanto, que substituimos a restrição $Z \asymp B \circ_3 C$ pela restrição equivalente $C \asymp Z \odot_3 B$. Essa operação não altera o conjunto solução, pois a fórmula obtida Φ' é equivalente à original e, além disso, seu grafo é uma cascata de árvores segundo X . Agora, executando $\text{hibridEVD}(\Phi v)$, inferimos $\sigma(C) = \sigma(Z) \odot_3 \sigma(B)$ sem gerar conflitos de valoração.



CAPÍTULO 6

CONCLUSÃO

*“E bem, qualquer que seja a solução, uma coisa fica,
e é a soma das sumas, ou o resto dos restos...”*

(Machado de Assis, Dom Casmurro, Cap. 148)

A modelagem intervalar consiste em especificar um conjunto de restrições sobre intervalos que representam valores reais aproximados. Se esse conjunto for uma conjunção de relações algébricas entre as variáveis intervalares e, além disso, tivermos especificações sobre os limitantes e a precisão de cada intervalo, então essa modelagem é, de fato, um problema de Satisfatibilidade Intervalar.

Adicionalmente, se tivermos um intervalo objetivo a ser minimizado, então temos um problema de Otimização Intervalar. Esse problema, por sua vez, é um relaxamento da Programação Não Linear, pois sua solução caracteriza-se por um conjunto de aproximações de números reais que satisfazem as restrições com um erro de precisão preestabelecido, minimizando o intervalo objetivo.

A álgebra intervalar nos oferece uma poderosa ferramenta para solucionar problemas POI: a consistência intervalar. Como esta é condição necessária para a existência de uma solução nos reais, excluimos os valores inconsistentes com os demais intervalos sob as restrições dadas, obtendo refinamentos mais precisos.

Embora a consistência intervalar seja a base dos algoritmos intervalares, ela não garante a existência de um conjunto de valores reais exatos para o problema, tornando impossível, no geral, a redução de problemas de Programação Não Linear usual para problemas POI. No entanto, se a conjunção de restrições intervalares definir um grafo acíclico, mostramos que a consistência no arco é suficiente para garantir a existência de tal conjunto. Além disso, conseguimos extrair a solução ótima do problema de forma

direta e sem a necessidade de retrocesso.

Neste trabalho, também apresentamos uma formalização da álgebra intervalar e de conceitos como restrições intervalares e consistência intervalar por uma abordagem um pouco distinta da qual geralmente se encontra na literatura, deixando explícito o significado de uma modelagem intervalar consistente. Além disso, mostramos que a consistência no arco é condição suficiente para a existência de solução real nas fórmulas acíclicas ou compostas por uma cascata de árvores, se permitirmos a extrapolação de alguns intervalos.

Os algoritmos propostos estendem o trabalho de Fränzle [17], que não aborda problemas de otimização, e tiram proveito de características de fórmulas acíclicas não apenas para garantir a existência de uma valoração degenerada, mas também para encontrar o modelo ótimo.

6.1 Trabalhos Futuros

Neste trabalho, restringimos os algoritmos para POI às fórmulas intervalares primitivas, devido a limitações da representação por grafos. Alternativamente, utilizando multigrafos para a representação das fórmulas intervalares, pode ser possível estender os teoremas e algoritmos para fórmulas não primitivas. Além disso, exigimos que a valoração consistente não valore multi-intervalos, apenas intervalos. Essa restrição facilita a descrição dos algoritmos intervalares, mas exige uma ramificação combinatória. No entanto, como os métodos apresentados são fundamentados nas propriedades apresentadas no teorema 2.1, e essas também valem para multi-intervalos, pelo teorema 2.2, então podemos estender os algoritmos propostos para também resolverem problemas multi-intervalares.

Os algoritmos propostos são corretos e completos se aplicarmos, sobre a valoração inicial dada, a consistência no arco. Embora vários trabalhos já atacaram o problema de aplicar tal consistência [26, 31, 18], sua eficiência ainda é muito dependente da estrutura da fórmula. No entanto, conjecturamos que para os problemas de interesse neste trabalho, a estrutura acíclica da fórmula é suficiente para a aplicação dessa consistência.

Além disso, para a otimização aproximada, existem variáveis que não são garantidamente valoradas por refinamentos da valoração inicial. Essas variáveis, de certa forma,

não tiram proveito da consistência no arco, tornando inútil sua aplicação em tais restrições. Da mesma forma, conjecturamos que não há necessidade de garantir tal consistência para todas as restrições de uma fórmula. Mais do que isso, talvez possamos evitar os erros de valoração não nulos se, para esses átomos, garantirmos alguma outra consistência intervalar.

Por fim, na seção 5.4, mostramos que não necessariamente a fórmula de entrada precisa ser uma cascata de árvores para a aplicação do método `hibridEVD`, mas deve existir uma cascata de árvores equivalente à fórmula original. Dessa forma, também é interessante um método que encontra tal fórmula ou prove sua não existência. Analisando o grafo definido pela fórmula original pode ser possível resolver esse problema, utilizando métodos de busca com retrocesso e poda ou reconhecimento de padrões a fim de encontrar subfórmulas conflitantes.

BIBLIOGRAFIA

- [1] C. Barrett, R. Nieuwenhuis, A. Oliveras, e C. Tinelli. Splitting on demand in SAT modulo theories. Miki Hermann e Andrei Voronkov, editors, *In Proc. LPAR'06, volume 4246 of LNAI*. Springer, 2006.
- [2] F. Benhamou e L. Granvilliers. Continuous and interval constraints. P. F. Rossi, van Beek, e T. Walsh, editors, *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier, New York, NY, USA, p. 569–601. 2006.
- [3] F. Benhamou, D. McAllester, e P. Van Hentenryck. Clp (intervals) revisited. Maurice Bruynooghe, editor, *ILPS*. MIT Press, 1994.
- [4] A. K. Bhurjee e G. Panda. Efficient solution of interval optimization problem. *Mathematical Methods of Operations Research*, 76(3):273–288, Agosto de 2012.
- [5] A. Biere, M. Heule, H. van Maaren, e T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
- [6] M. Bofill, R. Nieuwenhuis, A. Oliveras, E. Rodríguez-Carbonell, e A. Rubio. The barcelogic smt solver. A. Gupta e S. Malik, editors, *Proceedings of the 20th international conference on Computer Aided Verification*, volume 5123, Berlin, Heidelberg, 2008. Springer.
- [7] D. W. Cantrell. “affinely extended real numbers.” from mathworld. Disponível em: <http://mathworld.wolfram.com/AffinelyExtendedRealNumbers.html>, Acesso em: 20/01/2014.
- [8] S. A. Cook. The complexity of theorem-proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, volume 5, Bulgaria, 1971. Hikari Ltd.
- [9] I. M. COPI. *Introdução à lógica*. Mestre Jou, São Paulo, 1978.

- [10] Ernest Davis. Constraint propagation with interval labels. *Artif. Intell.*, 32(3):281–331, julho de 1987.
- [11] M. Davis, G. Logemann, e D. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, julho de 1962.
- [12] L. De Moura e N. Bjørner. Z3: an efficient smt solver. *Proceedings of the Theory and practice of software, 14th international conference on Tools and algorithms for the construction and analysis of systems, TACAS’08/ETAPS’08*, Berlin, Heidelberg, 2008. Springer-Verlag.
- [13] Rina Dechter. *Constraint Processing*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.
- [14] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- [15] B. Dutertre e L. De Moura. Integrating simplex with dpll(t). CSL, SRI INTERNATIONAL, 2006. Relatório Técnico.
- [16] Boi Faltings. Arc consistency for continuous variables. *Artificial Intelligence*, 65: 363–376, 1998.
- [17] M. Franzle, C. Herde, T. Teige, S. Ratschan, e T. Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation*, 1:209–236, 2007.
- [18] C. Gilles, T. Gilles, e N. Bertrand. New light on arc consistency over continuous domains. COPRIN - INRIA Sophia Antipolis, novembro de 2004. Relatório Técnico.
- [19] A. Griggio, F. Bruno, K. Irst, e V. Sommarive. A practical approach to satisfiability modulo linear integer arithmetic. *Journal on Satisfiability, Boolean Modeling and Computation*, 1–27, 2012.

- [20] M. Hladík. Interval linear programming: A survey. Z. A. Mann, editor, *Linear Programming - New Frontiers in Theory and Applications*. Nova Science Publishers, New York, p. 85–120. 2012.
- [21] H. Ishibuchi e H. Tanaka. Multiobjective programming in optimization of the interval objective function. *Eur J Oper Res*, 48(2), 1990.
- [22] N. Jussien e O. Lhomme. Dynamic domain splitting for numeric csps. *ECAI*, 1998.
- [23] J. P. Marques-Silva e K. A. Sakallah. Grasp: a new search algorithm for satisfiability. *In Proceedings of International Conference on Computer-Aided Design*, 1996.
- [24] R. E. Moore, R. B. Kearfott, e M. J. Cloud. *Introduction to interval analysis*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.
- [25] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, e S. Malik. Chaff: engineering an efficient sat solver. *In Proceedings of the 38th Design Automation Conference*, 2001.
- [26] Thang Nguyen, Yves Deville, Universite Catholique De Louvain, e Place Ste Barbe. A distributed arc-consistency algorithm. *Science of Computer Programming*, 1995.
- [27] B. Selman, H. Kautz, e B. Cohen. Local search strategies for satisfiability testing. *Dimacs Series in Discrete Mathematics and Theoretical Computer Science*, 521–532, 1995.
- [28] H. Suprajitno e I. bin Mohd. Linear programming with interval arithmetic. *International Journal of Contemporary Mathematical Sciences*, 5:323–332. 2010.
- [29] G. S. Tseitin. On the complexity of derivation in propositional calculus. *Leningrad Seminar on Mathematical Logic*, 1966.
- [30] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1936.

- [31] P. Van Hentenryck, Y. Deville, e C. Teng. A generic arc-consistency algorithm and its specializations. *Artificial Intelligence*, 57:291–321, 1992.
- [32] P. van Hentenryck, D. McAllester, e D. Kapur. Solving polynomial systems using a branch and prune approach. *In SIAM Journal on Numerical Analysis*, 34(2), 1997.
- [33] E. W. Weisstein. “set.” from mathworld—a wolfram web resource. Disponível em: <<http://mathworld.wolfram.com/Set.html>>, Acesso em: 20/01/2014.

APÊNDICE A

Teorema 2.1 *Sejam três intervalos não vazios X , Y e Z tais que $X \subseteq Y \circ Z$, para algum $\circ \in \mathfrak{M}$, então $\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y \circ z$.*

Prova. Em [24] temos que, para $\circ \in \{+, -, \times, \div\}$, $Y \circ Z = \{y \circ z \mid y \in Y \text{ e } z \in Z\}$. Portanto, $X \subseteq Y \circ Z$ implica que $\forall x \in X : x = y \circ z$ para algum $y \in Y$ e $z \in Z$. Isto é

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y \circ z$$

Portanto, o teorema vale para $\circ \in \{+, -, \times, \div\}$. Basta provar que ele também vale para $\circ \in \{\wedge, \Upsilon\}$.

Se $\circ = \wedge$, temos 5 casos:

1. $z > 0$ é ímpar:

Como $Y \wedge Z = [\underline{y}^z, \overline{y}^z]$ e $X \subseteq Y \wedge Z$, então, para todo $x \in X$,

$$\underline{y}^z \leq x \leq \overline{y}^z$$

Por outro lado, $\forall y \in Y : \underline{y} \leq y \leq \overline{y}$. Como $z > 0$ é ímpar, então temos que

$$\underline{y}^z \leq y^z \leq \overline{y}^z$$

Como $Z = \{z\}$, então

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y^z$$

2. $z > 0$ é par e $\underline{y} \geq 0$:

Nesse caso, $\underline{y}^z \leq y \leq \overline{y}^z$ também é verdadeiro, para todo $y \in Y$. Como $Y \wedge Z = [\underline{y}^z, \overline{y}^z]$ e $X \subseteq Y \wedge Z$, então a prova segue do item anterior.

3. $z > 0$ é par e $\bar{y} < 0$:

Como $Y \wedge Z = [\bar{y}^z, \underline{y}^z]$ e $X \subseteq Y \wedge Z$, então, para todo $x \in X$,

$$\bar{y}^z \leq x \leq \underline{y}^z$$

Por outro lado, $\forall y \in Y : \underline{y} \leq y \leq \bar{y}$. Como $\underline{y} \leq \bar{y} < 0$, então $|\bar{y}| \leq |\underline{y}|$ e $|\bar{y}|^z \leq |\underline{y}|^z$.

Mas como $z > 0$ é par, então $|\bar{y}|^z = \bar{y}^z$ e $|\underline{y}|^z = \underline{y}^z$.

Portanto

$$\bar{y}^z \leq y^z \leq \underline{y}^z$$

Como $Z = \{z\}$, então

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y^z$$

4. $z > 0$ é par, $\underline{y} < 0$ e $\bar{y} \geq 0$:

Como $Y \wedge Z = [0, \max\{\underline{y}^z, \bar{y}^z\}]$ e $X \subseteq Y \wedge Z$, então, para todo $x \in X$,

$$0 \leq x \leq \max\{\underline{y}^z, \bar{y}^z\}$$

Por outro lado, $\forall y \in Y : \underline{y} \leq y \leq \bar{y}$. Pelo mesmo argumento do item anterior, temos que $\forall y \leq 0 \in Y : 0 \leq y^z \leq \underline{y}^z$. E, pelo mesmo argumento do item 1, $\forall y \geq 0 \in Y : 0 \leq y^z \leq \bar{y}^z$.

Portanto

$$0 \leq y^z \leq \max\{\underline{y}^z, \bar{y}^z\}$$

Como $Z = \{z\}$, então

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y^z$$

5. $z < 0$:

Pela definição, $Y \wedge Z = [1, 1] \div (Y \wedge -Z)$. Como $-z > 0$, então, pelos itens anteriores, o teorema vale para $(Y \wedge -Z)$.

Isto é

$$\forall h \in (Y \wedge -Z) : \exists y \in Y, \exists z \in (-Z) \mid h = y^z$$

Como o teorema também vale para a divisão, então é direto que

$$\forall x \in X : \exists y \in Y, \exists z \in (-Z) \mid x = \frac{1}{y^z} = y^{-z}$$

E portanto,

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = y^z$$

Se $\circ = \Upsilon$, temos 5 casos:

1. $z > 0$ é ímpar:

Como $Y \Upsilon Z = [\sqrt[z]{\underline{y}}, \sqrt[z]{\bar{y}}]$ e $X \subseteq Y \wedge Z$, então, para todo $x \in X$,

$$\sqrt[z]{\underline{y}} \leq x \leq \sqrt[z]{\bar{y}}$$

Por outro lado, $\forall y \in Y : \underline{y} \leq y \leq \bar{y}$. Como $z > 0$ é ímpar, então temos que

$$\sqrt[z]{\underline{y}} \leq \sqrt[z]{y} \leq \sqrt[z]{\bar{y}}$$

Como $Z = \{z\}$, então

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = \sqrt[z]{y}$$

2. $z > 0$ é par e $\underline{y} \geq 0$:

Como $Y \Upsilon Z = \langle [-\sqrt[z]{\bar{y}}, -\sqrt[z]{\underline{y}}], [\sqrt[z]{\underline{y}}, \sqrt[z]{\bar{y}}] \rangle$ e $X \subseteq Y \wedge Z$, então, para todo $x \in X$,

$$-\sqrt[z]{\bar{y}} \leq x \leq -\sqrt[z]{\underline{y}} \text{ ou } \sqrt[z]{\underline{y}} \leq x \leq \sqrt[z]{\bar{y}}$$

Por outro lado, $\forall y \in Y : \underline{y} \leq y \leq \bar{y}$. Como $0 \leq \underline{y} \leq \bar{y}$, então $\sqrt[z]{\underline{y}} \leq \sqrt[z]{\bar{y}}$ e $-\sqrt[z]{\bar{y}} \leq -\sqrt[z]{\underline{y}}$.

Portanto

$$-\sqrt[z]{\bar{y}} \leq \sqrt[z]{y} \leq -\sqrt[z]{\underline{y}} \text{ ou } \sqrt[z]{\underline{y}} \leq \sqrt[z]{y} \leq \sqrt[z]{\bar{y}}$$

Como $Z = \{z\}$, então

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = \sqrt[z]{y}$$

3. $z > 0$ é par e $\bar{y} < 0$:

Como $Y \cap Z = \emptyset$ e $X \subseteq Y \cup Z$, então $X = \emptyset$.

Por outro lado, como $\forall y \in Y : y < 0$ e $\sqrt[z]{y} \notin \mathbb{R}$, então

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = \sqrt[z]{y}$$

4. $z > 0$ é par e $\underline{y} < 0$ e $\bar{y} \geq 0$:

Pela definição, $Y \cap Z = [0, \bar{y}] \cap Z$. Como $X \subseteq Y \cap Z$ então $X \subseteq [0, \bar{y}] \cap Z$ e, pelo item 2,

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = \sqrt[z]{y}$$

5. $z < 0$:

Pela definição, $Y \cap Z = [1, 1] \div (Y \cap -Z)$. Como $-z > 0$, então, pelos itens anteriores, o teorema vale para $(Y \cap -Z)$.

Isto é

$$\forall h \in (Y \cap -Z) : \exists y \in Y, \exists z \in (-Z) \mid h = \sqrt[z]{y}$$

Como o teorema também vale para a divisão, então é direto que

$$\forall x \in X : \exists y \in Y, \exists z \in (-Z) \mid x = \frac{1}{\sqrt[z]{y}} = \sqrt[-z]{y}$$

E pertanto,

$$\forall x \in X : \exists y \in Y, \exists z \in Z \mid x = \sqrt[z]{y}$$

□

GUILHERME ALEX DERENIEVICZ

**OTIMIZAÇÃO DE SISTEMAS INTERVALARES NÃO
LINEARES ACÍCLICOS**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Fabiano Silva

CURITIBA

2014