

**Abdução Clássica
e
Abdução Probabilística:
a Busca pela Explicação
de Dados Reais**

Alexandre Matos Arruda

TESE APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
DOUTOR EM CIÊNCIA DA COMPUTAÇÃO

Programa: Doutorado em Ciência da Computação

Orientador: Prof. Dr. Marcelo Finger

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

São Paulo, abril de 2014

Abdução Clássica e Abdução Probabilística: a Busca pela Explicação de Dados Reais

Esta versão da tese contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 16/04/2014. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Marcelo Finger (orientador) - IME-USP
- Profa. Dra. Renata Wassermann - IME-USP
- Prof. Dr. Fábio Gagliardi Cozman - EP-USP
- Prof. Dr. Mario Roberto Folhadela Benevides - UFRJ
- Prof. Dr. Paulo Eduardo Santos - FEI

Agradecimentos

Agradeço a Deus por ter me guiado e mostrado o caminho certo, mesmo que por linhas tortas.

Agradeço a minha família pelo apoio moral dado.

Agradeço a meu orientador Prof. Marcelo Finger, pela sua orientação, apoio e compreensão.

Agradeço a todos meus amigos do Laboratório de Inteligência Artificial e Métodos Formais (LIAMF) da USP.

Resumo

A busca por explicações de fatos ou fenômenos é algo que sempre permeou o raciocínio humano. Desde a antiguidade, o ser humano costuma observar fatos e, de acordo com eles e o conhecimento presente, criar hipóteses que possam explicá-los. Um exemplo clássico é quando temos consulta médica e o médico, após verificar todos os sintomas, descobre qual é a doença e os meios de tratá-la. Essa construção de explicações, dado um conjunto de evidências que o indiquem, chamamos de *abdução*. A abdução tradicional para a lógica clássica estabelece que o dado meta não é derivado da base de conhecimento, ou seja, dada uma base de conhecimento Γ e um dado meta A temos $\Gamma \not\vdash A$. Métodos clássicos de abdução buscam gerar um novo dado H que, juntamente com uma base de conhecimento Γ , possamos inferir A ($\Gamma \cup H \vdash A$). Alguns métodos tradicionais utilizam o tableaux (como em (MP93)) para a geração da fórmula H . Aqui, além de lidarmos com a abdução baseada em corte, através do KE-tableaux, que não necessita assumir que o dado meta não seja derivado da base de conhecimento, lidaremos também com a lógica probabilística, redescoberta por Nilsson, em (Nil86), onde temos a atribuição de probabilidades a fórmulas. Dizemos que uma instância em lógica probabilística é consistente se existe uma distribuição probabilística consistente sobre as valorações. Determinar essa distribuição probabilística é que o chamamos de problema PSAT. O objetivo de nosso trabalho é definir e estabelecer o que é uma abdução em Lógica Probabilística (abdução em PSAT) e, além disso, fornecer métodos de abdução para PSAT: dada uma instância PSAT $\langle \Gamma, \Psi \rangle$ na forma normal atômica (Bon11) e uma fórmula A tal que existe uma distribuição probabilística π que satisfaz $\langle \Gamma, \Psi \rangle$ e $\pi(A) = 0$, cada método é capaz de gerar uma fórmula H tal que $\langle \Gamma \cup H, \Psi \rangle \approx A$ onde $\pi(A) > 0$ para toda distribuição π que satisfaça $\langle \Gamma \cup H, \Psi \rangle$. Iremos também demonstrar que alguns dos métodos apresentados são corretos e completos na geração de fórmulas H que satisfaçam as condições de abdução.

Palavras-chave: lógica probabilística, abdução, abdução probabilística, abdução para PSAT.

Abstract

The search for explanations of facts or phenomena is something that has always permeated human reasoning. Since antiquity, the human being usually observes facts and, according to them and his knowledge, create hypotheses that can explain them. A classic example is when we have medical consultation and the doctor, after checking all the symptoms, discovers what is the disease and the ways to treat it. This construction of explanations, given a set of evidence, we call *abduction*. In traditional abduction methods it is assumed that the goal data has not yet been explained, that is, given a background knowledge base Γ and a goal data A we have $\Gamma \not\vdash A$. Classical methods want to generate a new datum H in such way that with the background knowledge base Γ , we can infer A ($\Gamma \cup H \vdash A$). Some traditional methods use the analytical tableaux (see (MP93)) for the generation of H . Here we deal with a cut-based abduction, with the KE-tableaux, which do not need to assume that the goal data is not derived from the knowledge base, and, moreover, with probabilistic logic (PSAT), rediscovered in (Nil86), where we have probabilistic assignments to logical formulas. A PSAT instance is consistent if there is a probabilistic distribution over the assignments. The aim of our work is to define and establish what is an abduction in Probabilistic Logic (abduction for PSAT) and, moreover, provide methods for PSAT abduction: given a PSAT instance $\langle \Gamma, \Psi \rangle$ in atomic normal form (Bon11) and a formula A such that there is a probabilistic distribution π that satisfies $\langle \Gamma, \Psi \rangle$ and $\pi(A) = 0$, each method is able to generate a formula H such that $\langle \Gamma \cup H, \Psi \rangle \approx A$ where $\pi(A) > 0$ for all distribution π that satisfies $\langle \Gamma \cup H, \Psi \rangle$. We demonstrated that some of the our methods, shown in this work, are correct and complete for the generation of H .

Keywords: probabilistic logic, abduction, probabilistic abduction, abduction for PSAT.

Sumário

Lista de Abreviaturas	ix
Lista de Símbolos	xi
1 Introdução	1
1.1 O que é abdução?	1
1.2 A Lógica Probabilística	7
1.3 Abdução Probabilística?	8
1.4 Objetivos	9
1.5 Contribuições	9
1.6 Organização do Trabalho	9
2 Tableaux Clássicos	11
2.1 Introdução	11
2.2 Conceitos e Definições Preliminares	11
2.2.1 Tableaux Analítico para a Lógica Proposicional	14
2.2.2 KE-Tableau para a Lógica Proposicional	18
2.3 A Linguagem de Primeira-Ordem	20
2.3.1 Tableaux Analítico e KE-Tableaux para a Lógica de Primeira-Ordem	22
3 Abdução Clássica via Tableaux Clássicos	25
3.1 Introdução	25
3.2 Conceitos e Definições Preliminares	26
3.3 Abdução Baseada em Tableaux Analíticos	27
3.4 Abdução Baseada em Corte com KE-Tableaux	29
3.4.1 Completude para a Abdução Baseada em Corte Proposicional . . .	33
3.4.2 Completude para Abdução Minimal Baseada em Corte de Primeira- Ordem	34
4 O Problema PSAT	37
4.1 Introdução	37
4.2 Conceitos e Definições Preliminares	38

4.3	O Algoritmo Simplex	39
4.4	PSAT via Simplex com Geração de Colunas	41
4.4.1	O Método de Geração de Colunas	42
4.5	A Forma Normal Atômica	43
4.6	PSAT via Simplex com Geração de Colunas Utilizando SAT- <i>solver</i>	44
4.7	Troca da Função Objetivo para a Segunda Fase do Simplex	47
5	Abdução Probabilística	55
5.1	Introdução	55
5.2	Conceitos e Definições Preliminares	56
5.3	Geração de Fórmulas com Probabilidades Positivas para Instâncias PSAT	56
5.4	Abdução de Certezas	60
5.4.1	Utilizando o Algoritmo Simplex	60
5.4.2	Utilizando Geração de Fórmulas com Probabilidade Positiva	65
5.5	Abdução de informações probabilísticas	70
5.5.1	Utilizando abdução clássica para gerar informações probabilísticas	70
6	Conclusões	73
6.1	Contribuições e Considerações Finais	73
6.2	Pesquisas Futuras	73
	Referências Bibliográficas	75

Lista de Abreviaturas

SAT	Problema da Satisfabilidade Clássica
PSAT	Problema da Satisfabilidade Probabilística
FOL	Lógica de Primeira-Ordem (First-Order Logic)
PB	Princípio da Bivalência
sse	se, e somente se

Lista de Símbolos

\wedge	Conjunção da lógica clássica
\vee	Disjunção da lógica clássica
\rightarrow	Implicação da lógica clássica
\leftrightarrow	Equivalência da lógica clássica
\neg	Negação da lógica clássica
Γ	Conjunto de fórmulas da lógica clássica proposicional
Ψ	Conjunto de atribuição de probabilidades
$Abd(\Gamma, A)$	Abdução clássica para o conjunto de fórmulas Θ e uma fórmula A
π	Vetor de distribuição probabilística
A^{-1}	Matriz inversa de A
A^T	Matriz transposta de A
c^T	Vetor transposto de c
c^i	Vetor complemento de c

Capítulo 1

Introdução

1.1 O que é abdução?

A busca por explicações de fatos ou fenômenos é algo que sempre permeou o raciocínio humano. Desde a antiguidade, o ser humano costuma observar fatos e, de acordo com o conhecimento presente, criar hipóteses que possam explicá-los. Um exemplo clássico é quando temos uma consulta médica e o médico, após verificar todos os sintomas, sugere qual é a doença e os meios de tratá-la. Essa construção de explicações, dado um conjunto de evidências que o indiquem, chamamos de *abdução*. O termo abdução (também chamado de retrodução) foi adotado por Charles Sanders Peirce (Pei58) como uma terceira forma de raciocínio além da dedução e da indução. Peirce a descreve como sendo a inferência lógica da descoberta, isto é, da criação de hipóteses (ou conjecturas) para explicar fatos e pensamentos. Em (Pei58), ele a exemplifica em forma de silogismo:¹

Um fato surpreendente C é observado;
Mas se A é verdadeiro, C precisa ser verdadeiro também;
Portanto, há alguma razão de *suspeitar* que A é verdadeiro.

Do ponto de vista da lógica dedutiva, o último argumento é inválido, mas do ponto de vista abduutivo está correto já que C representa algo que pode ser uma novidade, uma informação nova, algo inesperado, e deve, portanto, ser explicado. São exemplos de abdução:

1. O astrônomo alemão Johannes Kepler formulou, em 1609, a teoria de que os planetas orbitam em trajetórias elípticas por constatar que, em determinadas épocas, ora eles estavam mais distantes ora mais próximos do sol (detalhes em (HB09)).

¹Em inglês:

The surprising fact, C , is observed.
But if A were true, C would be a matter of course.
Hence, there is reason to suspect that A is true. (Pei58)

2. A sonda Curiosity, lançada ao espaço em novembro de 2011, chegou, em agosto de 2012, à Marte e constatou a presença de rochas sedimentares em lugares que, com outros fatores, nos dá fortes indícios de que o planeta vermelho pode ter abrigado um lago (Gt14).
3. O Bóson de Higgs, conhecido popularmente como “a partícula de Deus”, foi conjecturado em 1964 pelo físico britânico Peter Higgs como o responsável por conferir propriedades inerciais (ou seja, “massa”) a partículas próximas, em acordo com as previsões teóricas do modelo padrão das partículas elementares (Hig64).

Além de observações científicas, temos que o cotidiano também pode apresentar, mesmo de forma inconsciente, processos de abdução:

4. Alguém vai a um consultório médico diagnosticar uma doença e o médico, ao verificar os sintomas, identifica a doença e os meios de tratá-la.
5. Após acordar, uma pessoa constata que seu celular não liga e suspeita de que a bateria deva estar descarregada.
6. Após uma noite de chuva, ao amanhecer, uma árvore caída na pista interrompe o fluxo do trânsito. Um motorista imagina que a chuva deve tê-la derrubado.

A abdução, portanto, é o tipo de pensamento característico nas diferentes situações onde lidamos com uma informação incompleta, onde precisamos de uma hipótese plausível. Por tratar de hipóteses plausíveis, elas podem vir a ser falhas em algumas situações *a posteriori*. Nos casos acima (1-6), por exemplo, temos que a hipótese do primeiro caso já foi confirmada há tempos, a segunda e a terceira ainda estão, no presente momento, em processo de confirmação dentro das teorias vigentes. As restantes podem ser, em algum momento, inválidas: um diagnóstico errado de alguma doença, um defeito no aparelho, um carro subiu no canteiro e derrubou a árvore.

Essa forma de raciocínio, em certo aspecto, reflete a concepção do método científico. Um exemplo é o caso quando um cientista infere a existência de um novo tipo de partícula atômica, ele está inferindo uma explicação a partir dos diversos dados disponíveis que indiquem sua presença, mas um experimento posterior pode muito bem contestar sua validade. Alguns filósofos da ciência afirmam que a abdução é o grande pilar do método científico, como podemos ver em (Lip04), (Psi99) e, em (McM92), onde Ernan McMullin intitula a abdução como “A inferência que faz ciência”.

Dedução, Indução e Abdução

Em (Pei58), Peirce descreve suas três formas de raciocínio lógico: a dedução, indução e a abdução. A dedução é caracterizada como sendo a forma mais simples: através de

implicações válidas chegamos a uma conclusão diretamente das hipóteses, ou seja, nada é acrescentado à teoria, partimos de um conjunto de hipóteses, e obtemos uma conclusão. A indução é vista como sendo a formulação de uma regra dado que temos o caso e o resultado, portanto, é uma generalização feita para algo maior a partir da observação de algo menor. A abdução é definida como o processo de formação de uma hipótese, onde temos uma regra e um resultado e, a partir deles, criamos uma hipótese. Peirce as diferencia em termos gerais:

“A dedução prova algo que **deve ser**; a indução mostra que algo **realmente é operativo**²; a abdução meramente sugere algo que **pode ser**.³”

(C. S. Peirce, em (Pei58))

Em (Pei58) um exemplo de cada uma é fornecido:

DEDUÇÃO

Regra	Todos os feijões desta sacola são brancos.
Caso	Estes feijões são desta sacola.
Resultado	Estes feijões são brancos.

INDUÇÃO

Caso	Estes feijões são desta sacola.
Resultado	Estes feijões são brancos.
Regra	Todos os feijões desta sacola são brancos.

ABDUÇÃO (OU HIPÓTESE)

Regra	Todos os feijões desta sacola são brancos.
Resultado	Estes feijões são brancos.
Caso	Estes feijões são desta sacola.

Na literatura moderna, entretanto, há muitas convenções e, às vezes, confusões, principalmente entre a distinção entre abdução e indução. Em geral, a maioria dos autores não fazem a distinção e consideram o pensamento dedutivo no sentido clássico e tudo que não for dedutivo como indutivo, por exemplo, na filosofia computacional da ciência, a indução é entendida como “qualquer tipo de inferência que expande o conhecimento em face da incerteza” (Tha88). O termo mais frequentemente utilizado na literatura nesse caso é *raciocínio indutivo*. Em (Ali96), Aliseda Atocha propõe uma nova abordagem sem

²No sentido de que é uma regra.

³Em inglês:

Deduction proves that something *must be*; Induction shows that something *actually* is operative; Abduction merely suggests that something *may be*. (Pei58)

dizer explicitamente onde temos abdução ou indução através do *raciocínio explicatório*, onde há dois tipos de abordagens, a novidade e a anomalia, que podem levar, cada uma a três graus de saídas diferentes: fatos, regras ou teorias.

Em contrapartida, alguns autores buscam fazer a distinção entre elas. Em Harman (Har65), por exemplo, a abdução é considerada como uma forma de inferência não-dedutiva que inclui a indução enumerativa⁴ como caso especial. Em termos gerais, Harman argumenta que se alguém entende que “todos os *A*'s são *B*'s” é a melhor explicação, no sentido de ser a mais simples e plausível para o fato de que “todos os *A*'s observados são *B*'s”, então a indução enumerativa não explica nada mais além do que a *inferência para a melhor explicação* é capaz de explicar. Em (JJ94), Josephson reafirma a hipótese de Harman onde a generalização indutiva é uma forma de abdução visto que ela é ampliativa:

Todos *A*'s observados são *B*'s
Portanto, Todos *A*'s são *B*'s (JJ94, p.19)

onde temos que a teoria é ampliada com a inclusão da última afirmação. Contudo, Josephson discorda do ponto que Harman afirma que a inferência de “Todos *A*'s são *B*'s” de fato explica uma instância particular “Este *A* é *B*”, sob a alegação de que apesar da hipótese nos trazer algo novo ela não explica a correlação entre *A* e *B*.

Abdução como Inferência para a Melhor Explicação

Em 1897, o físico britânico Joseph John Thomson conduziu um experimento que o levou a postular que as partículas constituintes dos raios catódicos são cargas carregadas negativamente, segue sua formulação:

“Como os raios catódicos carregam uma carga de eletricidade negativa, são defletidos por uma força eletrostática como se estivessem eletrizados negativamente, e são afetados por uma força magnética do mesmo modo como essa força atua em corpos eletrizados negativamente em movimento ao longo do trajeto desses raios, eu não vejo alternativa senão concluir que eles são cargas de eletricidade negativa carregadas por partículas de matéria.”⁵ (traduzido da citação em (Ach01))

Thomson, portanto, postulou o que pra ele é a **melhor explicação plausível** para

⁴Trata do caso em que a partir da observação de alguma propriedade de uma amostra, podemos inferir essa propriedade para o todo.

⁵Em inglês:

“As the cathode rays carry a charge of negative electricity, are deflected by an electrostatic force as if they were negatively electrified, and are acted on by a magnetic force in just the way in which this force would act on a negatively electrified body moving along the path of these rays, I can see no escape from the conclusion that they are charges of negative electricity carried by particles of matter.”

a deflexão dos raios catódicos no experimento. Na literatura, a abdução é entendida por muitos filósofos como sendo *a inferência para a melhor explicação (ou hipótese)*. Essa visão teve origem em (Har65), onde Harman propôs que a “melhor explicação” é aquela que explica um fato tão bem que podemos rejeitar qualquer outra hipótese. Em (JJ94), Josephson nos apresenta essa ideia em forma de silogismo:⁶

D é uma coleção de informações (fatos, observações, dados)
 H explica D (Se for verdade, explica D)
 Nenhuma outra hipótese pode explicar D tão bem como H explica.
 Portanto, H é provavelmente verdade ((JJ94), 5).

Essa forma de abdução difere da abdução tradicional proposta por C. S. Peirce em dois aspectos: 1- considera apenas a **melhor** hipótese ao invés de **qualquer** hipótese; 2- temos agora uma exigência explicativa da hipótese e não apenas como adjunta para inferir um fato. Um exemplo, mostrado em (Har65), é quando um detetive coloca todas as evidências juntas e decide que o culpado é o mordomo depois de concluir que nenhuma outra explicação que leve em conta todos os fatos, é plausível ou simples o bastante para ser aceita.

A forma acima não explicita como todas as hipóteses são obtidas e, mais ainda, não é mostrado como a melhor hipótese é escolhida entre elas. Uma abordagem clássica é gerar todas as hipóteses possíveis (*geração de hipóteses*) e descartar entre elas todas que são irrelevantes, implausíveis o que não possuem importância suficiente. Das que restarem buscamos selecionar aquela que melhor representa a relação dos dados observados com o fato a ser explicado, avaliando os custos e benefícios de adotá-la como melhor explicação (*seleção de hipóteses*). Em (JJ94), Josephson propõe 6 regras para avaliar uma hipótese abdutiva H :⁷

1. Quão decisivamente H supera as alternativas.
2. Quão bom H é por si mesmo.
3. Julgamentos sobre confiabilidade dos dados.
4. Quanta confiança há em todas as hipóteses plausíveis consideradas.
5. Considerações pragmáticas incluindo os custos dela estar errada e os benefícios de estar certa.
6. Quão forte é a necessidade dela levar a a uma conclusão.

⁶Em inglês:

D is a collection of data (facts, observations, givens).
 H explains D (would if true, explain D).
 No other hypothesis can explain D as well as H does.
 Therefore, H is probably true (Josephson, 5).

⁷Em inglês:

1. how decisively does H surpasses the alternatives.
2. how good H is all by itself.
3. judgments of the reliability of the data.
4. how much confidence there is hat all plausible explanations have been considered.
5. pragmatic considerations including the costs of being wrong and the benefits of being right.
6. how strong the need is to come to a conclusion at all (JJ94)

Essas duas fases (geração e seleção de hipóteses) estão fortemente correlacionadas visto que se temos uma geração de hipóteses falha, por conseguinte, corremos o risco de ter péssimas opções para a seleção. E, por outro lado, podemos ter uma excelente fase de geração de hipóteses, mas nada disso adiantará se escolhermos uma péssima hipótese dentre elas.

Dentre outros trabalhos que defendem a abdução como “inferência para a melhor hipótese” podemos citar (Lip04), (Psi99) e (DK94), este último contém uma visão geral de como a inferência para a melhor hipótese é usada em diferentes áreas da filosofia.

Abdução em Lógica Clássica

A abdução em lógica clássica estabelece que dada teoria Γ e uma fórmula A , temos $\Gamma \not\models A$, ou seja, A representa uma informação nova e ainda não explicada por Γ . A solução de um problema de abdução tradicional busca gerar uma fórmula H que juntamente com Γ “explique” A . A ideia é que H represente um fato que associe a base de conhecimento ao dado meta, ou seja, $\Gamma, H \models A$. Para que H seja uma explicação relevante, a literatura impõe algumas restrições de admissibilidade para gerar H , as mais comuns encontradas na literatura, conforme Aliseta em (Ali97), são:

Dada um teoria Γ (conjunto de fórmulas) e um dado meta A (uma fórmula), H é uma explicação se

1. $\Gamma \cup H \models A$;
2. H é consistente com Γ ;
3. H é “minimal”;
4. H possui algumas restrições sintáticas (por exemplo, H é uma fórmula atômica ou uma conjunto delas, negadas ou não).

Essas são algumas condições tradicionalmente impostas à geração da fórmula H . Note que todas partem da precondição $\Gamma \not\models A$. Temos, porém, que a condição $\Gamma \not\models A$ é indecidível para a lógica de primeira-ordem (Tur36) e, trabalhos recentes, como visto em (MDG08), contornam esse problema ao fornecerem além de uma visão explicatória (caso $\Gamma \not\models A$), uma visão geradora de lemas que facilitem a prova (caso $\Gamma \models A$).

Neste trabalho focaremos em métodos de abdução que utilizam o tableaux analítico (Smu68) e o KE-tableaux (DM94) como sistemas subjacentes (como veremos nos capítulos 2 e 3).

1.2 A Lógica Probabilística

O estudo da probabilidade é de grande interesse na Ciência da Computação. Atualmente, vivemos em uma era em que temos que lidar com grandes quantidades de dados e, por diversas vezes, esses dados podem não ser exatos. Por exemplo, na área de Processamento de Linguagem Natural (PLN), temos diversos problemas, onde muitas vezes fazemos o uso de dados completamente probabilísticos para classificação textual, outro exemplo é o diagnóstico médico automatizado, onde buscamos todas as informações a respeito de uma doença genética e uma aplicação é capaz de calcular qual a probabilidade dela ocorrer sob vários fatores probabilísticos. São casos onde temos que os dados não são conhecidos com exatidão, onde temos incertezas, e a abordagem clássica pode ter custo computacional elevado ou simplesmente ser inviável como modelo. Para esse intuito, no presente trabalho, utilizaremos a lógica probabilística.

A lógica probabilística foi introduzida por Boole (em (Boo54)) como uma forma de expressar as incertezas do raciocínio humano. Durante um grande período seu trabalho foi redescoberto por diversas vezes (HJ00). Em 1965, Hailperin revisitou o problema e o trouxe para a programação linear (Hai65). Em 1974, vemos sua influência no trabalho de Finetti em sua teoria subjetiva de probabilidade (Fin74). Em 1986, Nilsson (Nil86) formalizou o problema da satisfabilidade probabilística (PSAT) para a Inteligência Artificial (IA): dadas fórmulas lógicas e probabilidades atribuídas à elas, queremos saber se essa atribuição é satisfatível.

O Problema PSAT

Em complexidade computacional, temos que o problema da satisfabilidade probabilística (PSAT) é um problema de decisão, onde se pergunta se um conjunto de probabilidades atribuídas a sentenças proposicionais é consistente. Além disso, ele é NP-completo, ou seja, de acordo com o teorema de Cook-Levin (Coo71) temos a existência de uma redução polinomial do PSAT para o SAT. Tal redução à primeira vista é atraente pelo fato de que algoritmos que resolvem SAT estão em contínua pesquisa e, em geral, são eficientes, porém o problema PSAT tem se mostrado mais difícil e métodos que utilizam o simplex apresentam melhores resultados do que as reduções para SAT apresentadas até o presente momento como podemos ver em (Bon11).

A primeira formulação do problema utilizando o algoritmo simplex foi apresentada em (GGP88), onde temos uma estratégia de geração de colunas para a elaboração da matriz solução. Em (Bon11) é proposto uma forma normal para o problema (a forma normal atômica) e ela é utilizada de modo que a cada iteração uma coluna é gerada usando o problema SAT como oráculo na busca. Resultados como transição de fase e comparações com uma redução para SAT também são apresentados em (Bon11).

Ambos trabalhos utilizam o simplex de duas fases, porém restringem-se somente à

primeira fase, onde buscam uma solução que satisfaça o problema. Uma segunda fase visa a minimização e maximização da probabilidade de uma fórmula em relação ao problema em questão e é algo que será tratado durante este trabalho.

1.3 Abdução Probabilística?

Os métodos de abdução que utilizam a lógica clássica buscam criar sentenças que, juntamente com uma base de conhecimento, auxiliam na explicação de um fato observado. Todo esse processo é feito sobre uma base clássica e tais métodos, portanto, não se adequariam bem ao lidar com incertezas, onde os dados não são precisos. Quando temos, por exemplo, que uma doença genética acontece com taxa elevada de frequência e temos uma base de conhecimento sobre condições para doenças genéticas, uma das questões é saber quais são as condições que nos levaram a ter essa frequência. A *abdução probabilística* é, portanto, a criação de hipóteses que, juntamente com uma *base de conhecimento probabilística*, permite que um fato que acontece com determinada frequência seja explicado. A importância dela reside no crescente uso de sistemas que possuem dados probabilísticos, onde a geração de hipóteses pode ser vista como uma forma de explicação de um fato que ocorre com determinada probabilidade. Outro uso possível também é a adequação do sistema, onde temos que a hipótese gerada é adicionada à base de conhecimento assim como é feito tradicionalmente na abdução clássica, de forma que o sistema agora passe a incluir uma explicação do dado meta. Quanto à geração de hipótese, temos dois tipos:

1. Uma fórmula com probabilidade 1, ou seja, uma sentença que, à rigor, age como uma sentença verdadeira na lógica clássica. A importância da geração desse tipo de fórmula reside na apresentação de fatos concretos que implicam a probabilidade da sentença meta. Por exemplo, a probabilidade de termos falta de energia na semana é positiva e geramos uma hipótese de que a subestação responsável teve paralização no fornecimento naquela semana.
2. Uma fórmula com probabilidade entre 0 e 1. Esse é o caso quando temos que a adição de alguma incerteza ao banco de conhecimento explica a probabilidade da sentença meta. Por exemplo, a possibilidade de termos trânsito ruim em uma grande metrópole dado que hipotetizamos uma probabilidade alta de chuva.

Temos, portanto, que a abdução probabilística permite a criação de hipóteses, tanto precisas como imprecisas, para a explicação de um fato ou um fenômeno observado. Uma aplicação prática em que podemos ter ambos casos de hipóteses é o problema de gerar *tags part-of-speech* para textos em NLP. Alguns algoritmos utilizam metologias baseadas em regras, sem probabilidade, como é o caso do algoritmo TBL⁸ (Bri), outras metologias

⁸Transformation-Based Learning

clássicas utilizam algoritmos completamente estocásticos, baseados em HMM⁹ ou MEM¹⁰ (DJ09). Neste trabalho procuramos fornecer um ambiente que permite a utilização da lógica probabilística em situações que podemos ter ambos os casos: regras bem definidas ou associadas a probabilidades.

1.4 Objetivos

Nosso objetivo primordial neste trabalho é definir o que é a abdução probabilística utilizando uma linguagem proposicional e apresentar alguns métodos de abdução probabilística capazes de gerar hipóteses precisas e imprecisas. Para cumprir esse objetivo, iremos apresentar a visão clássica de abdução que utiliza tanto o tableaux semântico como o KE-tableaux. Quanto ao último, fornecemos uma prova de completude que foi publicada em (AF13). A ideia de apresentar essas abordagens clássicas consiste no seu uso em métodos de abdução probabilística, conforme veremos. Além de expormos essas novas definições, também trazemos à luz novos resultados no uso da lógica probabilística, dentre eles temos o processo de psat-normalização que permite a utilização da segunda fase do simplex.

1.5 Contribuições

As principais contribuições deste trabalho são as seguintes:

- Uma prova da completude da abdução baseada em corte (capítulo 3);
- Apresentamos um método que permite o uso da segunda fase do simplex para estabelecer a probabilidade de uma fórmula A nas distribuições que satisfazem uma dada instância PSAT (capítulo 4);
- Definimos o problema da abdução probabilística, assim como apresentamos alguns métodos corretos e completos que o resolvem (capítulo 5).

1.6 Organização do Trabalho

Quanto à divisão deste trabalho, temos que no capítulo 2 iremos apresentar a lógica proposicional, que será utilizada pelo restante deste trabalho, e a lógica de primeira-ordem assim como os métodos tableaux e KE-tableaux para elas. No capítulo 3 abordaremos os métodos clássicos de abdução que utilizam o tableaux analítico e o KE-tableaux. O problema da satisfabilidade probabilística (PSAT) será apresentado no capítulo 4. No

⁹Hidden Markov Model

¹⁰Maximum Entropy Model

capítulo 5, iremos apresentar nossa definição de abdução probabilística e os métodos propostos. Por fim, temos nossas conclusões e trabalhos futuros descritos no capítulo no capítulo 6.

Capítulo 2

Tableaux Clássicos

2.1 Introdução

Neste capítulo introduziremos as noções básicas da lógica proposicional e da lógica de primeira-ordem para o entendimento dos capítulos seguintes. Apresentaremos a linguagem adotada e, além disso, os métodos de prova que utilizam o tableaux analítico e o KE-tableaux. Ambos são métodos de inferência por refutação, porém possuem paradigmas diferentes que remetem ao *cálculo de sequentes*, proposto por Gentzen em (Gen35), conforme iremos apresentar.

Quanto à estruturação deste capítulo, apresentaremos, na seção 2.2, a linguagem proposicional, que será utilizada pelos capítulos restantes, seus principais conceitos presentes na literatura e, nas subseções 2.2.1 e 2.2.2, iremos apresentar o tableaux analítico e o KE-tableaux respectivamente. Na seção 2.3, abordaremos a linguagem de primeira-ordem e suas versões do tableaux analítico e do KE-tableaux (subseção 2.3.1) que possuem quantificação universal e existencial.

2.2 Conceitos e Definições Preliminares

A linguagem proposicional clássica é constituída de *variáveis booleanas* (ou átomos) combinadas com os *operadores lógicos*. Temos, portanto, o alfabeto da linguagem:

1. Um conjunto infinito enumerável de símbolos $X = \{p_1, p_2, \dots\}$ correspondente às *variáveis proposicionais atômicas*;
2. As constantes lógicas \top e \perp ;
3. Os *conectivos binários*: \vee , \wedge e \rightarrow que representam respectivamente a disjunção, conjunção e implicação;
4. O *conectivo unário* \neg que representa a negação;

5. Parênteses como símbolos auxiliares: ‘(’ e ‘)’.

Dado o alfabeto da linguagem proposicional, iremos agora definir como as fórmulas proposicionais são geradas. Por *fórmula bem formada* (que chamaremos simplesmente de fórmula ou sentença) em lógica proposicional, utilizaremos a definição tradicional:

1. Toda variável proposicional atômica ou constante lógica é uma fórmula;
2. Se A e B são fórmulas, então $(A \vee B)$, $(A \wedge B)$ e $(A \rightarrow B)$ são fórmulas;
3. Se A é uma fórmula, então $\neg A$ é uma fórmula.

Definimos o conjunto \mathcal{L}_{LP} como sendo o menor conjunto contendo todas as fórmulas formadas pelas regras 1, 2 e 3.

Os parênteses nas fórmulas podem ser omitidos para simplificar a notação. Utilizaremos a seguinte ordem de precedência: \neg , \wedge , \vee , \rightarrow . A equivalência entre duas fórmulas A e B é expressa como $(A \leftrightarrow B)$ e será uma abreviação para $(A \rightarrow B) \wedge (B \rightarrow A)$. Por questões de praticidade, as letras maiúsculas A, B, C, \dots serão utilizadas como representantes das fórmulas proposicionais, assim como usaremos os símbolos p, q e r como proposicionais atômicas. Por *literais* entenderemos como sendo proposições atômicas que podem estar negadas ou não.

Definimos também o conceito de *subfórmula* utilizando o predicado Sub , como abaixo:

1. Para toda variável proposicional p , $Sub(p) = \{p\}$;
2. Se $(A \square B)$, então $Sub((A \square B)) = \{(A \square B) \cup Sub(A) \cup Sub(B)\}$,
com $\square \in \{\vee, \wedge, \rightarrow\}$;
3. Se $\neg A$, então $Sub(\neg A) = \{\neg A \cup Sub(A)\}$.

Dizemos, portanto, que uma fórmula A é subfórmula de uma fórmula B sse¹ $A \in Sub(B)$.

Cada variável proposicional pode ser associada a um valor verdade, verdadeiro ou falso, que iremos representar, respectivamente, por 1 e 0. Uma valoração é uma função $v : X \rightarrow \{0, 1\}$ que associa cada variável proposicional a um valor verdade. Em particular, temos associado às constantes \top e \perp o valor $v(\top) = 1$ e $v(\perp) = 0$ para toda valoração v . Podemos estender uma valoração v a todo o conjunto \mathcal{L}_{LP} associando cada fórmula a um valor verdade obedecendo aos conectivos:

- $v(A \wedge B) = 1$ sse $v(A) = 1$ e $v(B) = 1$;
- $v(A \vee B) = 1$ sse $v(A) = 1$ ou $v(B) = 1$;
- $v(\neg A) = 1$ sse $v(A) = 0$;

¹Abreviação para “se, e somente se”

- $v(A \rightarrow B) = 1$ sse $v(A) = 0$ ou $v(B) = 1$;
- $v(A \leftrightarrow B)$ sse $v(A) = v(B)$.

Definimos agora quando uma fórmula proposicional A ou um conjunto de fórmulas proposicionais Γ são satisfatíveis:

Definição 2.2.1 (Satisfabilidade Proposicional). *Uma fórmula proposicional A é satisfatível sse existe uma valoração v que a satisfaz (ou seja, $v(A) = 1$) e um conjunto Γ de fórmulas proposicionais é dito satisfatível sse existe uma valoração v que satisfaz todos seus elementos. Dizemos que a valoração v satisfaz Γ .*

Agora que temos a definição de satisfabilidade para uma fórmula proposicional e para um conjunto de fórmulas proposicionais, vamos definir quando uma fórmula possui uma relação de consequência a partir de um conjunto de fórmulas:

Definição 2.2.2 (Consequência Lógica). *Dizemos que uma fórmula A é consequência lógica de um conjunto Γ , ou que Γ implica logicamente A , se toda valoração que satisfaz Γ também satisfaz A . Essa relação é representada como $\Gamma \models A$.*

Exemplo 2.2.3. Seja o conjunto $\Gamma = \{p, (\neg p \vee q)\}$ e a fórmula q . É fácil verificar que toda valoração v que satisfaz Γ , ou seja, $v(p) = 1$ e $v((\neg p \vee q)) = 1$ também faz com que $v(q) = 1$, logo $\Gamma \models q$.

Vamos agora ao caso quando uma fórmula sempre é verdadeira para quaisquer valorações:

Definição 2.2.4 (Tautologia). *Uma fórmula proposicional A é uma tautologia se, e somente se, todas as valorações a satisfazem, ou seja, para qualquer valoração v , temos $v(A) = 1$.*

Exemplo 2.2.5. Considere uma das relações De Morgan da literatura $(\neg p \wedge \neg q) \leftrightarrow \neg(p \vee q)$. Verificamos que ela é uma tautologia: $v(\neg p \wedge \neg q) = 1$ sse $v(\neg p) = 1$ e $v(\neg q) = 1$ sse $v(p) = 0$ e $v(q) = 0$ sse $v(p \vee q) = 0$ sse $v(\neg(p \vee q)) = 1$. Usando a relação \models , temos $\models \neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$.

Por fim, temos o caso em que A não é satisfeito por nenhuma valoração:

Definição 2.2.6. *Quando uma dada fórmula A é tal que não há nenhuma valoração em que tenhamos $v(A) = 1$, dizemos que A é uma fórmula insatisfatível.*

Exemplo 2.2.7. Um exemplo clássico é a fórmula $(A \wedge \neg A)$ na qual nenhuma valoração a satisfaz.

Para entender os paradigmas que envolvem tanto o tableaux analítico como o KE-tableaux, é necessário mencionar o trabalho de Gentzen (em (Gen35)), onde foi proposto o *cálculo de sequentes*. Um sequente possui o seguinte formato:

$$\Gamma \Rightarrow \Delta$$

onde Γ é um conjunto de afirmações ou proposições em linguagem lógica e Δ é o conjunto que representa suas implicações ou o resultados que obtemos dada a validade de Γ . O cálculo de sequentes possui regras na forma:

$$\frac{\Gamma_1 \Rightarrow \Delta_1 \quad \dots \quad \Gamma_n \Rightarrow \Delta_n}{\Gamma' \Rightarrow \Delta'} R$$

onde cada sequente $\Gamma_i \Rightarrow \Delta_i$ é uma *premissa* da regra R e o sequente $\Gamma' \Rightarrow \Delta'$ é sua *conclusão*. Dizemos que uma regra R como apresentada acima é *analítica* se cada fórmula ou subfórmula presente em Γ ou Δ ocorre em algum Γ_i ou Δ_i . Todas as regras propostas no cálculo de sequentes são *analíticas*, com exceção da regra do corte:

$$\frac{\Gamma \Rightarrow H, \Delta \quad \Gamma', H \Rightarrow \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'} \text{Corte}$$

onde temos que H é uma fórmula que está nas premissas de uma regra do corte, porém, pode não ocorrer na conclusão.

O cálculo de sequentes foi proposto por Gentzen através do qual foi possível demonstrar que a *regra do corte* em dedução natural é eliminável, ou seja, dispensável, de forma que temos cálculos *livres de corte* (Gen35). Em um sistema livre de corte, todas as regras são analíticas e uma consequência imediata que temos é o *princípio da subfórmula*, que nos diz que todas as fórmulas que ocorrem em uma prova usando um sistema livre de corte são subfórmulas da conclusão ou das hipóteses em questão.

O método do tableaux analítico é um descendente direto do cálculo de sequentes livres de cortes de Gentzen, enquanto o KE-tableaux permite a utilização de uma regra do corte sob a forma do *princípio da bivalência*, como veremos nas subseções seguintes.

2.2.1 Tableaux Analítico para a Lógica Proposicional

Agora que definimos a linguagem proposicional e sua semântica, a ideia é automatizar a verificação da relação $\Gamma \models A$ através da manipulação de A e das fórmulas presentes em Γ . Para esse fim, iremos apresentar uma introdução ao tableaux analítico, proposto por Raymond Smullyan em (Smu68). O método do tableaux analítico e o cálculo de sequentes de Gentzen (Gen35) são essencialmente o mesmo sistema visto por pontos de vista diferentes. No primeiro as provas são feitas por refutação e o segundo é uma dedução

direta a partir de um conjunto de hipóteses. O tableaux, apresentado em (Smu68), é analítico no sentido de que as regras são baseadas em sistemas livres de corte e, portanto, sempre geram subfórmulas das hipóteses.

Um tableaux analítico é construído sobre uma árvore de nós, conforme (Smu68), que possui as seguintes características:

1. Um conjunto S de elementos chamados nós;
2. Uma função n que assinala para cada nó x um inteiro positivo $n(x)$ chamado de nível de x ;
3. Uma relação xRy definida em S , que pode ser lida como “ x é um predecessor de y ” ou “ y é um sucessor de x ”. E essa relação obedece às seguintes condições:
 - a. Existe um único nó de nível 1. Esse nó chamamos de *nó raiz* da árvore;
 - b. Todo nó, com exceção da raiz, possui um único predecessor;
 - c. Para quaisquer nós x, y se y é um sucessor de x , então $n(y) = n(x) + 1$.

Chamaremos de *folha* qualquer nó que não possuir sucessores e de *nó simples* se possui um único sucessor. Quando um nó possuir mais de um sucessor o chamaremos de *nó composto*. Por *caminho* chamaremos qualquer sequência finita ou enumerável de nós começando da raiz da árvore tal que cada elemento da sequência é o predecessor do próximo. Chamaremos de *ramo* qualquer caminho cujo o último nó é uma folha ou é um caminho infinito. O caso em que especificarmos um *caminho partindo de um nó x* significa que em vez do caminho começar pela raiz, irá começar pelo nó x em questão. Adotaremos a mesma convenção para a definição de *ramo partindo de um nó*.

Com base na noção de árvore, definimos agora as regras do tableaux analítico nas Tabelas 2.1 e 2.2 onde cada nó é rotulado por uma fórmula proposicional marcada por T

Tabela 2.1: Regras do tipo α

$$\frac{TA \wedge B}{\frac{TA}{TB}} (T\wedge) \quad \frac{FA \vee B}{\frac{FA}{FB}} (F\vee) \quad \frac{FA \rightarrow B}{\frac{TA}{FB}} (F \rightarrow)$$

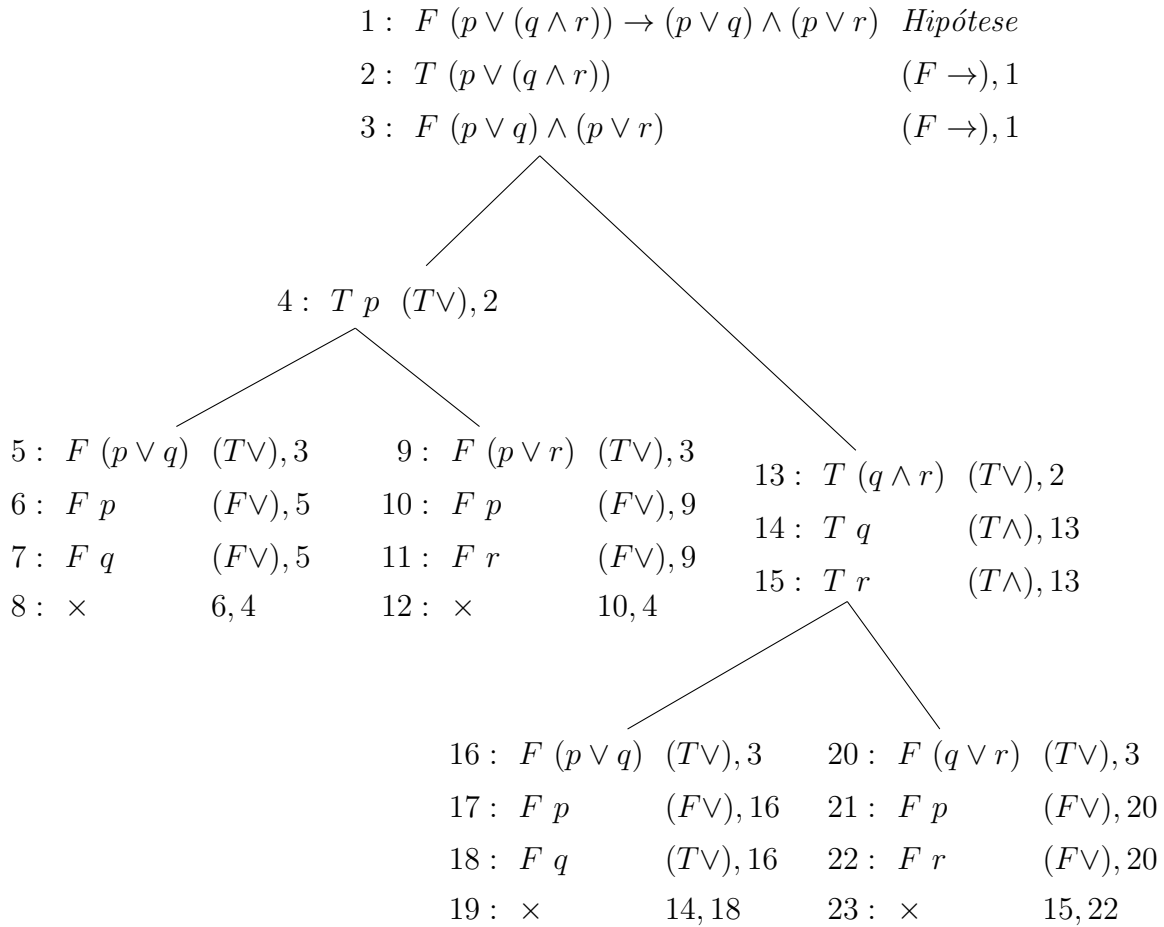
$$\frac{T\neg A}{FA} (T\neg) \quad \frac{F\neg A}{TA} (F\neg)$$

Tabela 2.2: Regras do tipo β

$$\begin{array}{c} FA \wedge B \quad (F\wedge) \\ \swarrow \quad \searrow \\ FA \quad FB \end{array} \quad \begin{array}{c} TA \vee B \quad (T\vee) \\ \swarrow \quad \searrow \\ TA \quad TB \end{array} \quad \begin{array}{c} TA \rightarrow B \quad (T \rightarrow) \\ \swarrow \quad \searrow \\ FA \quad TB \end{array}$$

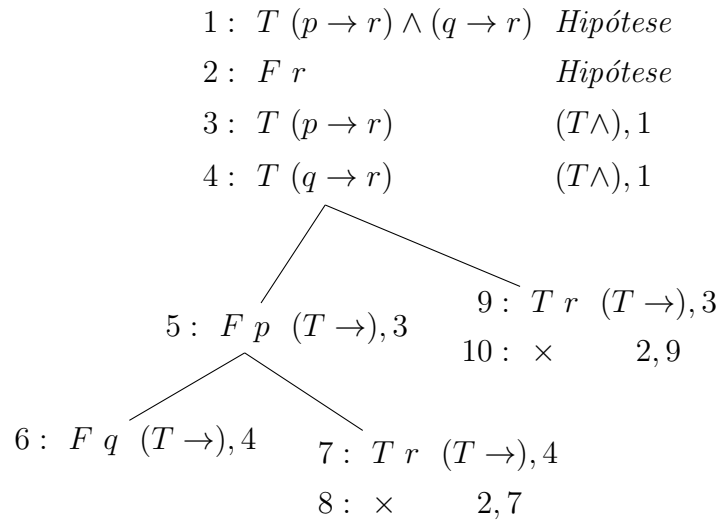
ou F . A marcação com T ou F ao lado de uma fórmula é vista como uma associação de

valor “verdadeiro” (T) ou “falso” (F) à respectiva fórmula. As fórmulas $TA \wedge B$, $FA \vee B$, $FA \rightarrow B$, $T\neg A$, $F\neg A$ são as premissas das regras do tipo α e $FA \wedge B$, $TA \vee B$, $TA \rightarrow B$ são as premissas das regras do tipo β , as demais são as conclusões de cada regra. Note que as regras do tipo α (chamaremos regras- α) são aplicadas linearmente (cada premissa é um nó simples), enquanto que as regras do tipo β (regras- β) dividem um ramo em dois (cada premissa é um nó composto). Quando uma regra é aplicada a uma fórmula na árvore dizemos que aquela fórmula foi *expandida*. Se um ramo possui uma fórmula TA e uma fórmula FA em sua composição, dizemos que ele é um *ramo fechado*. Ramos fechados possuem um nó rotulado com \times como folhas. Como exemplo do uso do tableaux analítico, vamos provar a fórmula $\models (p \vee (q \wedge r)) \rightarrow (p \vee q) \wedge (p \vee r)$:



Temos que cada ramo do tableaux acima é fechado, basta notar a presença de fórmulas da forma TA e FA em cada ramo. O tableau é iniciado com a fórmula $F(p \vee (q \wedge r)) \rightarrow (p \vee q) \wedge (p \vee r)$ como hipótese. A ideia é que se todos os ramos fecham então afirmar que a fórmula $(p \vee (q \wedge r)) \rightarrow (p \vee q) \wedge (p \vee r)$ é falsa (usando a marcação F) é um absurdo. Quando todos os ramos estão fechados, como o caso acima, temos então um *tableaux fechado* e, portanto, a sequência de hipóteses iniciais está *provada*. Um ramo está *saturado* se todas as fórmulas não-atômicas já foram expandidas. Quando temos um tableaux onde todos seus ramos estão ou fechados ou saturados dizemos que ele é um *tableaux completo* (ou

saturado). Como exemplo de um tableau completo, temos:

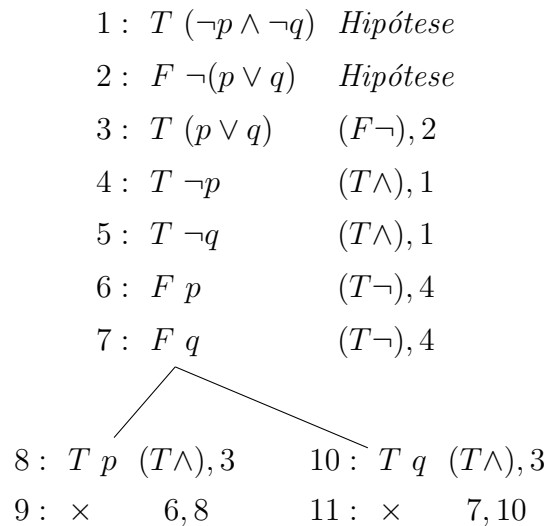


O ramo aberto nos fornece um contramodelo, basta utilizar as proposições atômicas que ocorrem nele. No caso acima temos que se fizermos $v(p) = 1$, $v(q) = 1$ e $v(r) = 0$, obteremos uma valoração que falsifica $(p \rightarrow r) \wedge (q \rightarrow r)$.

O símbolo \vdash será utilizado para indicar quando obtemos uma prova de A a partir de um conjunto Γ (dizemos que Γ prova A) através do uso do tableaux.

Definição 2.2.8. $A_1, \dots, A_n \vdash B_1, \dots, B_m$ se, e somente se, existe um Tableau fechado para $TA, \dots, TA_n, FB_1, \dots, FB_m$.

Como mais um exemplo, considere a prova da lei De Morgan $(\neg p \wedge \neg q) \vdash \neg(p \vee q)$:



Nos exemplos anteriores provamos, portanto, $\vdash (p \vee (q \wedge r)) \rightarrow (p \vee q) \wedge (p \vee r)$ e $(p \rightarrow r) \wedge (q \rightarrow r) \not\vdash r$.

Os dois teoremas a seguir, provados em (Smu68), atestam que o tableaux é um sistema correto e completo para a lógica proposicional:

Teorema 2.2.9 (Correção). *Dado que temos um conjunto de fórmulas proposicionais Γ e uma fórmula A tal que $\Gamma \vdash A$, então $\Gamma \models A$.*

Demonstração. Em (Smu68). □

Teorema 2.2.10 (Completude). *Dado que temos um conjunto de fórmulas proposicionais Γ e uma fórmula A tal que $\Gamma \models A$, então $\Gamma \vdash A$.*

Demonstração. Em (Smu68). □

Vamos, agora, apresentar um método de tableaux não-analítico que possui uma regra com características equivalente à regra do corte.

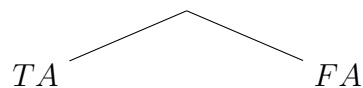
2.2.2 KE-Tableau para a Lógica Proposicional

O KE-tableaux foi proposto por D'Agostino e Mondadori, em (DM94), como um modo de incorporar a regra do corte em provas por tableaux. Uma consequência imediata que temos é que suas regras nem sempre produzem subfórmulas da fórmula expandida. É demonstrado que, em comparação ao tableaux analítico de Smullyan, ele possui melhores propriedades computacionais e é capaz de preservar todas as propriedades desejáveis que os sistemas livres de corte. Da mesma forma que o tableaux analítico, o KE-tableaux é feito utilizando a noção de árvore de prova com fórmulas marcadas com T ou F . Todas as regras, com exceção da regra do corte (*Princípio da Bivalência*) geram expansões lineares, como pode ser visto na Tabela 2.3. Observe que não há ramificação na aplicação dessas

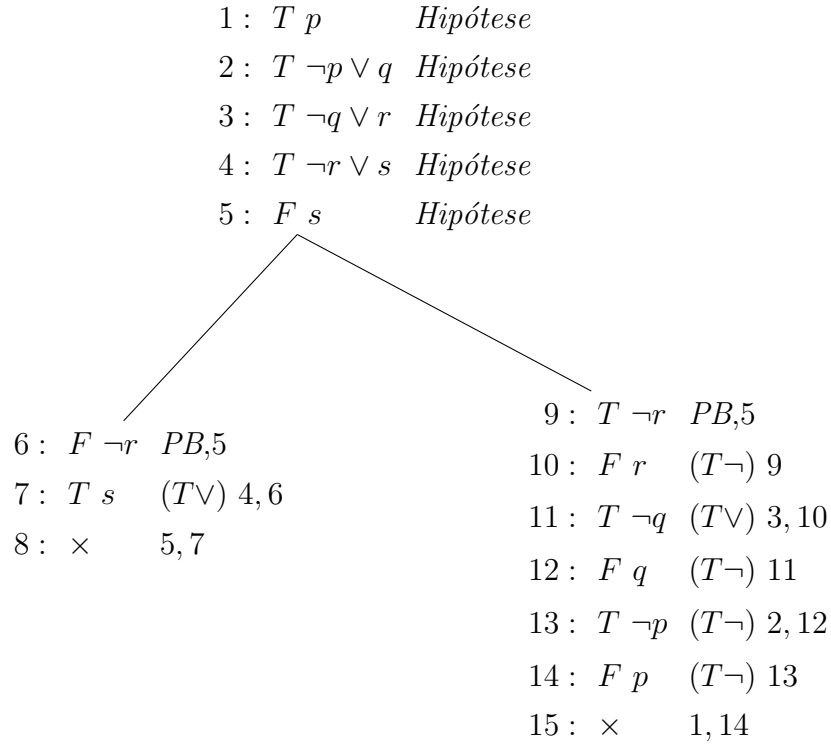
Tabela 2.3: Regras- α do KE-tableaux

$$\begin{array}{ccc}
 \frac{TA \wedge B}{\frac{TA}{TB}} (T\wedge) & \frac{FA \wedge B}{\frac{TA}{FB}} (F\wedge_1) & \frac{FA \wedge B}{\frac{TB}{FA}} (F\wedge_2) \\
 \\
 \frac{FA \vee B}{\frac{FA}{FB}} (T\vee) & \frac{TA \vee B}{\frac{FA}{TB}} (T\vee_1) & \frac{TA \vee B}{\frac{FB}{TA}} (T\vee_2) \\
 \\
 \frac{FA \rightarrow B}{\frac{TA}{FB}} (T \rightarrow) & \frac{TA \rightarrow B}{\frac{TA}{TB}} (T \rightarrow_1) & \frac{TA \rightarrow B}{\frac{FB}{FA}} (T \rightarrow_2) \\
 \\
 \frac{T\neg A}{FA} (T\neg) & \frac{F\neg A}{TA} (F\neg) &
 \end{array}$$

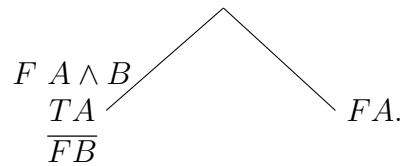
regras. O corte é expressado pelo *Princípio da Bivalência* (PB) como única regra que ramifica:



Isso significa que a qualquer momento em um KE-tableau podemos expandi-lo, para qualquer fórmula A , com uma bifurcação que contém TA e FA . As definições vistas de ramo fechado, ramo saturado, tableaux fechado e tableaux completo também são utilizadas da mesma forma aqui. Como exemplo, considere a prova, em KE-tableaux, de $p, \neg p \vee q, \neg q \vee r, \neg r \vee s \vdash s$:



Apesar da regra PB poder introduzir fórmulas arbitrárias, violando o princípio da subfórmula, D'Agostino e Mondadori mostraram que não há perda de completude em restringir as aplicações da regra PB de forma a preservar o princípio da subfórmula. Por exemplo, no tableaux analítico temos, pela Tabela 2.2, que a regra $F A \wedge B$ bifurca em dois ramos contendo $F A$ e $F B$ como folhas. Obtemos um resultado equivalente com as regras PB e \wedge_1 do KE-tableaux :



A relação de dedutibilidade \vdash_{KE} é definida como:

Definição 2.2.11. $A_1, \dots, A_n \vdash_{KE} B_1, \dots, B_m$ se, e somente se, existe um KE-Tableaux fechado para $TA, \dots, TA_n, FB_1, \dots, FB_m$.

Assim como o tableaux analítico, temos que as provas que utilizam o KE-Tableaux são corretas e completas:

Teorema 2.2.12 (Correção). *Dado que temos um conjunto de fórmulas proposicionais Γ e uma fórmula A tal que $\Gamma \vdash_{KE} A$, então $\Gamma \models A$.*

Demonstração. Em (DM94). □

Teorema 2.2.13 (Completude). *Dado que temos um conjunto de fórmulas proposicionais Γ e uma fórmula A tal que $\Gamma \models A$, então $\Gamma \vdash_{KE} A$.*

Demonstração. Em (DM94). □

Dado que apresentamos o tableaux analítico e o KE-tableaux para a lógica proposicional, vamos agora apresentar a linguagem de primeira-ordem, que permite a elaboração de sentenças que possuem quantificação universal e existencial, e, em seguida, apresentaremos as respectivas versões do tableaux analítico e do KE-tableaux.

2.3 A Linguagem de Primeira-Ordem

Na lógica de primeira-ordem, além de possuímos os conectivos proposicionais apresentados, temos a quantificação universal e existencial sobre um domínio de indivíduos, fato que a faz ter maior poder de expressão do que a lógica proposicional. O alfabeto para a linguagem de primeira-ordem que iremos adotar é o apresentado em (Smu68), que possui os seguintes símbolos:

1. $\neg, \wedge, \vee, \rightarrow$ (conectivos);
2. \forall, \exists (quantificadores universal e existencial, respectivamente);
3. Uma quantidade contável de símbolos chamados de *variáveis individuais*;
4. Uma quantidade contável de símbolos chamados de *parâmetros individuais*;
5. Para cada inteiro positivo n , uma quantidade contável de símbolos chamados de predicados n -ários (ou predicados de grau n).

Assim como no caso proposicional, os parênteses nas fórmulas podem ser omitidos para simplificar a notação. Utilizaremos a seguinte ordem de precedência: $\neg, \wedge, \vee, \rightarrow$. A equivalência entre duas fórmulas A e B é expressa como $(A \leftrightarrow B)$ e será uma abreviação para $(A \rightarrow B) \wedge (B \rightarrow A)$. Iremos utilizar as letras “ x ”, “ y ”, “ z ”, sobrescritas ou não, para denotar variáveis individuais arbitrárias e as letras “ a ”, “ b ” e “ c ”, sobrescritas ou não, para denotar parâmetros individuais. As letras capitais “ P ”, “ Q ”, “ R ”, com ou sem sobrescrita, denotarão predicados e seus graus ficarão claros no contexto. Usaremos o termo *símbolos individuais* para expressar tanto variáveis individuais como parâmetros individuais.

Fórmulas Atômicas Uma fórmula atômica é uma tupla- $(n + 1)$ Pc_1, \dots, c_n onde P é um predicado qualquer de grau n e c_1, \dots, c_n são quaisquer símbolos individuais.

Agora vamos a definição da linguagem L_{FOL} da Lógica de primeira-ordem:

1. Se A é uma fórmula atômica, então também é uma fórmula em L_{FOL} ;
2. Se A e B são fórmulas em L_{FOL} , então $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ também são fórmulas em L_{FOL} ;
3. Se A é uma fórmula em L_{FOL} , então $\neg A$ é uma fórmula em L_{FOL} ;
4. Se A é uma fórmula e x é uma variável então $\forall xA$ e $\exists xA$ estão em L_{FOL} ;
5. L_{FOL} é o menor conjunto contendo as regras acima.

Substituição Para toda fórmula A , variável x e parâmetro a , definimos a fórmula A_x^a pela seguinte regra de indução:

1. Se A é uma fórmula atômica, então A_x^a é o resultado de substituir qualquer ocorrência de x em A por a ;
2. $[A \square B]_x^a = A_x^a \square B_x^a$, para $\square \in \{\wedge, \vee, \rightarrow\}$;
3. $[\neg A]_x^a = \neg[A]_x^a$;
4. $[(\Delta x A)]_x^a = (\Delta x)A$ para $\Delta \in \{\forall, \exists\}$.

Por exemplo, dada a fórmula $\forall yPx, y \rightarrow \exists xQx$ temos que $[\forall yPx, y \rightarrow \exists xQx]_x^a = \forall yPa, y \rightarrow \exists xQx$. Quando uma variável x está ligada a um quantificador, como no exemplo, dizemos que x é uma *variável quantificada*, caso contrário, é uma *variável livre*. Referimos A_x^a como a fórmula resultante de substituir toda ocorrência livre de x por a .

Não iremos nos ater à semântica da lógica de primeira-ordem porque, neste capítulo e até mesmo no resto deste trabalho, ela não será diretamente utilizada. Definições semânticas da lógica de primeira-ordem podem ser encontradas em livros conhecidos da literatura, como em (Smu68) e em (End72), os quais a definem utilizando a teoria dos conjuntos. A noção de consequência lógica entre um conjunto Γ e uma fórmula A também é definida utilizando o símbolo \models de forma que A é consequência lógica de Γ se, e somente se, $\Gamma \models A$. Os métodos de tableaux da seção a seguir são, *mutatis mutandis*, corretos e completos quanto a essas semânticas mencionadas, porém temos um agravante em relação aos tableaux proposicionais: a lógica de primeira-ordem é indecidível (Tur36), ou seja, não há algoritmos que decidam $\Gamma \models A$ em primeira-ordem.

2.3.1 Tableaux Analítico e KE-Tableaux para a Lógica de Primeira-Ordem

No caso da lógica de primeira-ordem, as regras proposicionais permanecem as mesmas que as apresentadas tanto para o tableaux analítico quanto para o KE-tableaux. Para obtermos o tableaux analítico e o KE-tableaux de primeira-ordem adicionamos as seguintes regras de quantificação para cada:

$$\frac{\gamma}{\gamma_1(t)} \quad \text{para qualquer termo } t \qquad \frac{\delta}{\delta_1(a)} \quad \text{parâmetro novo } a$$

γ	$\gamma_1(t)$	δ	$\delta_1(t)$
$T\forall xP(x)$	$TP(t)$	$T\exists xP(x)$	$TP(a)$
$F\exists xP(x)$	$FP(t)$	$F\forall xP(x)$	$FP(a)$

Como exemplo, considere a prova de $\vdash \forall xPx \vee \forall xQx \rightarrow \forall x(Px \vee Qx)$ utilizando o tableaux analítico:

	1 : $F \forall xPx \vee \forall xQx \rightarrow \forall x(Px \vee Qx)$ <i>Hipótese</i>	
	2 : $T \forall xPx \vee \forall xQx$	($F \rightarrow$), 1
	3 : $F \forall x(Px \vee Qx)$	($F \rightarrow$), 1
	4 : $F (Pa \vee Qa)$	($T\forall$), 3
	5 : $F Pa$	($F\vee$), 4
	6 : $F Qa$	($F\vee$), 4
7 : $T \forall xPx$ ($T\forall$), 2	10 : $T \forall xQx$ ($T\forall$), 2	
8 : $T Pa$ ($T\forall$), 7	11 : $T Qa$ ($T\forall$), 10	
9 : \times 5, 8	12 : \times 6, 12	

A diferença das regras do tipo γ e δ em relação a todas as outras que foram apresentadas é que há casos em que necessariamente temos que expandir uma fórmula utilizando elas mais de uma vez. Isso faz jus ao fato de que verificar se $\Gamma \models A$ é válido em primeira-ordem é indecidível (Tur36). Por esse motivo, há casos em que não poderemos decidir se um tableaux para $\Gamma \vdash A$ fecha, já que não é possível determinar quantas expansões γ e δ são necessárias para fechá-lo. Considere, por exemplo, uma das formas de expandir um tableau iniciado por $F \exists xPx \rightarrow \forall xPx$:

1 :	$F \exists xPx \rightarrow \forall xPx$	<i>Hipótese</i>
2 :	$T \exists xPx$	$(T \rightarrow), 1$
3 :	$F \forall xPx$	$(T \rightarrow), 1$
4 :	$T Pa$	$(T \exists), 2$
5 :	$F Pb$	$(T \forall), 3$
6 :	$T Pc$	$(T \forall), 2$
7 :	$T Pd$	$(T \forall), 2$
	\vdots	

Vemos que há um ramo infinito e, de fato, não podemos fechar o tableaux pois se obtivéssemos $\vdash \exists xPx \rightarrow \forall xPx$ teríamos consequentemente $\models \exists xPx \rightarrow \forall xPx$, que claramente não é válido. Tanto $T \exists xPx$ como $F \forall xPx$ podem ser instanciados infinitas vezes, sempre gerando um novo parâmetro diferente dos anteriores. É o caso que temos $\not\vdash \exists xPx \rightarrow \forall xPx$.

O motivo principal de termos vistos todos os tableaux apresentados neste capítulo são seus usos na geração de fórmulas H que resolvem o problema da abdução clássica, conforme veremos no próximo capítulo.

Capítulo 3

Abdução Clássica via Tableaux Clássicos

3.1 Introdução

O *problema da abdução tradicional* estabelece que dada uma teoria Γ e uma fórmula A , temos $\Gamma \not\models A$. A ideia central é que Γ representa uma base de conhecimento e A um fato novo e ainda não explicado por Γ . A *solução de um problema de abdução* busca gerar uma fórmula H que, juntamente com Γ , “explique” A , ou seja, $\Gamma \cup H \models A$. Assim, para resolvermos o problema da abdução devemos buscar um H que represente um fato que associa a base de conhecimento ao dado meta. Para que H seja uma explicação relevante, a literatura impõe algumas restrições de admissibilidade para sua geração, as mais comuns, conforme Aliseta (Ali97), são:

Dada um teoria Γ (conjunto de fórmulas) e um dado meta A (uma fórmula), H é uma explicação se

1. $\Gamma \cup H \models A$;
2. H é consistente com Γ ;
3. H é “minimal”;
4. H possui algumas restrições sintáticas (por exemplo, H é uma fórmula atômica ou uma conjunto delas, negadas ou não).

Essas são algumas condições tradicionalmente impostas à geração da fórmula H . Note que todas partem da precondição $\Gamma \not\models A$. Temos, porém, que a condição $\Gamma \not\models A$ é indecidível para a lógica de primeira-ordem (Tur36) e, trabalhos recentes, como visto em (MDG08), contornam esse problema ao fornecerem além de uma visão *explicatória* (caso $\Gamma \not\models A$), uma visão *geradora de lemas* que facilitem a prova (caso $\Gamma \models A$).

Uma estratégia para a geração de fórmulas que resolvam o problema da abdução é o uso de tableaux, como foi feito pioneiramente em (MP93). Como vimos no capítulo anterior, dado $\Gamma \vdash^? A$, onde $\Gamma = \{H_1, \dots, H_n\}$ e a interrogação significa que não sabemos se a consequência lógica é válida ou não, temos que $\Gamma \vdash A$ sse o (KE-)tableaux fecha para $T H_1, \dots, T H_n, FA$. A ideia do uso do tableaux para resolver o problema da abdução consiste em gerar uma fórmula H que ao ser inserida como hipótese em um tableaux para $T H_1, \dots, T H_n, FA$, ela de fato o fecha, ou seja, temos que $T H_1, \dots, T H_n, T H, FA$ se expande para um tableaux fechado, conforme apresentado em (MP93).

Neste trabalho focaremos em métodos de abdução que utilizam o tableaux analítico (Smu68) e o KE-tableaux (DM94) como sistemas subjacentes. Quanto ao primeiro, temos que o método segue a visão clássica e, portanto, é regido pela precondição da abdução tradicional onde temos $\Gamma \not\vdash G$ e uma hipótese H é gerada de forma a obtermos $\Gamma, H \vdash G$ (geração de hipóteses). No segundo método, a precondição $\Gamma \not\vdash G$ não é obrigatória e uma fórmula H pode ser produzida tanto para explicação, caso em que temos $\Gamma \not\vdash G$, como para facilitar a prova de G a partir de Γ , caso em que já temos $\Gamma \vdash G$ (geração de lemas).

Quanto à organização deste capítulo, iremos, na seção 3.2, apresentar os conceitos básicos sobre a abdução tradicional e a abdução com geração de lemas. Na seção seguinte, a 3.3, iremos apresentar o método de abdução, visto na literatura, utilizando o tableaux analítico. Iremos abordar, na seção 3.4, o método de abdução proposto em (MDG08), que utiliza o KE-tableaux como sistema de inferência.

Como contribuição nossa neste capítulo, e publicada em (AF13), apresentaremos, nas subseções 3.4.1 e 3.4.2, a prova da completude para a abdução baseada em corte tanto para o caso proposicional quanto para o caso de primeira-ordem.

3.2 Conceitos e Definições Preliminares

Um *problema de abdução* é um par, escrito como $Abd(\Gamma, G)$, onde Γ é a base de conhecimento e G é o dado meta. A *hipótese de abdução tradicional* estabelece que $\Gamma \not\vdash G$. A solução para um problema de abdução é uma *fórmula abduzida* H tal que

$$\Gamma, H \vdash G$$

que é uma condição que será mantida neste trabalho. Tradicionalmente, abdução como “inferência da melhor explicação” impõe algumas condições extras sobre as fórmulas abduzidas tais como admissibilidade e condições minimais. Condições de admissibilidade comuns na literatura são:

- $H \not\vdash G$: a fórmula abduzida não implica o dado meta sem a base de conhecimento. Essa condição inclui o caso trivial onde $H = G$.

- $\Gamma, H \vdash \perp$ se e somente se $\Gamma \vdash \perp$: a fórmula abduzida leva em consideração o dado meta e não apenas contradiz a teoria. O caso onde a teoria não é levada em consideração é bloqueada pela condição $\Gamma \not\vdash G$.

Essas duas condições são, de modo geral, indecidíveis para a lógica de primeira-ordem. O método proposto em (MDG08) nos fornece meios de satisfazer essas condições por construção, sem submetê-las a um procedimento de decisão. Em geral, dado $\Gamma \vdash^? G$, onde a interrogação significa que não sabemos *a priori* a validade do sequente, produzimos um H tal que $\Gamma, H \vdash G$ que nos leva a dois possíveis casos:

1. Se $\Gamma \not\vdash G$, então o problema se reduz a abdução tradicional a qual é chamada de *geração de hipóteses*. Uma explicação H é produzida tal que $\Gamma, H \vdash G$. Algumas condições de minimalidade ou preferência são envolvidas para escolher a melhor explicação.
2. Se $\Gamma \vdash G$ é verdadeiro, então a questão não é explicar G a partir de Γ , mas facilitar sua prova por meio da *geração de lema*, onde produzimos um sequente $\Gamma, H \vdash G$ que possui uma prova mais simples que $\Gamma \vdash G$.

Iremos mostrar, na seção a seguir, o método que utiliza o tableaux analítico.

3.3 Abdução Baseada em Tableaux Analíticos

Descreveremos nesta seção a técnica de abdução que utiliza tableaux analíticos, apresentada em (MP93), onde temos a precondição $\Gamma \not\vdash A$ e a geração de hipóteses são restritas às C -fórmulas (conjunção de literais).

A ideia básica é que a solução para o problema da abdução $Abd(\Gamma, A)$ pode ser achada através de fórmulas que formam o fechamento de um tableau para $\Gamma \cup \{\neg A\}$. Condições de admissibilidade aqui são exigidas, por exemplo, hipóteses falsas ou equivalentes à A não são permitidas. O método é realizado em tableaux completos, onde todos os ramos são ou fechados ou saturados.

Definição 3.3.1. *Seja β um ramo saturado de um tableaux. Seja $\rho(\beta)$ a função que indica o conjunto de nós literais de β . O conjunto de fechamento para β é $\tau(\beta) = \{\neg\lambda \mid \lambda \in \rho(\beta)\}$, onde $\neg\lambda$ é a negação de λ .*

Seja \mathcal{T} um tableaux completo, o conjunto $\mathcal{S}(\mathcal{T})$ dos conjuntos minimais de fechamento de ramos abertos de \mathcal{T} é $\{\tau(\beta) \mid \beta \text{ é um ramo aberto em } \mathcal{T} \text{ e não há nenhum ramo } \beta' \text{ em } \mathcal{T} \text{ tal que } \tau(\beta') \subset \tau(\beta)\}$.

$\mathcal{S}(\mathcal{T})$ será chamado de conjunto de conjuntos de fechamentos de \mathcal{T} . Nota-se que ele abrange o conjunto de subconjuntos-minimais de conjuntos de fechamentos de ramos abertos de \mathcal{T} .

Definição 3.3.2 (Fechamento). *Dado um tableaux completo \mathcal{T} para um problema de abdução $Abd(\Gamma, A)$ e $\mathcal{S}(\mathcal{T}) = \{\tau_1, \dots, \tau_n\}$ o conjunto de conjuntos de fechamentos de \mathcal{T} . Seja g qualquer escolha dos elementos dos conjuntos de elementos de $\mathcal{S}(\mathcal{T})$, isto é, $g(\tau_i) \in \tau_i$.*

1. *Se $\mathcal{S}(\mathcal{T}) = \emptyset$, então \mathcal{T} é um tableaux fechado (Esse é o caso onde $\Gamma \models A$);*
2. *Se $\mathcal{S}(\mathcal{T}) \neq \emptyset$ e para qualquer função de escolha g dos elementos dos elementos de $\mathcal{S}(\mathcal{T})$ o conjunto $\{g(\tau_1) \wedge \dots \wedge g(\tau_n)\}$ contém o par de literais complementares, então o único fechamento para \mathcal{T} é uma fórmula \perp (Esse é o caso onde $\Gamma \models \neg A$);*
3. *Caso contrário, $\alpha = g(\tau_1) \wedge \dots \wedge g(\tau_n)$ é um fechamento para \mathcal{T} se, e somente se, α não contém um par de literais complementares.*

Na definição acima, é entendido que α é a conjunção de literais no conjunto $\{g(\tau_1) \wedge \dots \wedge g(\tau_n)\}$, isto é, literais não são repetidos.

Se \mathcal{T} é qualquer tableau completo, denotamos por $\mathcal{E}(\mathcal{T})$ o conjunto de todos os fechamentos de \mathcal{T} . Claramente, diferentes árvores de tableaux para o mesmo problema de abdução $Abd(\Gamma, A)$ possuem o mesmo fechamento. Temos, portanto:

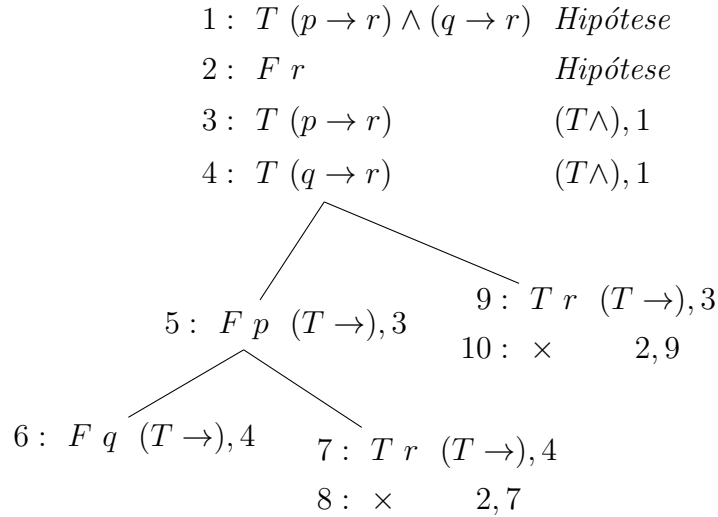
$$\mathcal{E}(\Gamma, A) = \mathcal{E}(\mathcal{T}) \text{ para qualquer árvore } \mathcal{T} \text{ para } Abd(\Gamma, A).$$

O seguinte teorema estabelece completude e correção para o método:

Teorema 3.3.3. *Seja $Abd(\Gamma, A)$ um problema de abdução. Então todo elemento de $\mathcal{E}(\Gamma, A)$ é uma explicação não contraditória para $Abd(\Gamma, A)$ (correção) e qualquer explicação minimal e não contraditória para $Abd(\Gamma, A)$ é um elemento de $\mathcal{E}(\Gamma, A)$ (completude).*

Demonstração. Em (MP93). □

Como exemplo de uma fórmula abduzida no caso proposicional, considere novamente o sequente $(p \rightarrow r) \wedge (q \rightarrow r) \not\vdash r$. Pelo método apresentado anteriormente, obtemos o seguinte tableaux \mathcal{T} :



Como candidatos ao processo de abdução nos temos os literais p e q sobre o único ramo aberto e saturado. Note que a fórmula $p \vee q$ não é uma candidata para fórmula abduzida. Como podemos ver, a abdução utilizando o tableaux analítico apresenta dois problemas: o primeiro é a necessidade de gerar um tableaux completo e o segundo é a restrição à C -fórmulas (conjunção de literais).

A geração de um tableaux completo pode gerar um alto custo computacional, já que expandí-lo, no pior caso, pode ser pior do que gerar uma tabela verdade para o problema (D'A92). A restrição a C -fórmulas também nos impõe um limitante na capacidade de fórmulas que podemos gerar, já que fórmulas “mais fracas” que as C -fórmulas são ignoradas. Como forma de contornar ambos os problemas, a abdução baseada em corte foi proposta em (MDG08).

3.4 Abdução Baseada em Corte com KE-Tableaux

Proposta em (MDG08), a abdução baseada em corte não necessita da pré-condição $\Gamma \not\vdash G$, que é indecidível no caso da lógica de primeira-ordem. Como vimos, dado $\Gamma \vdash^? G$, onde a interrogação significa que não sabemos *a priori* a validade do sequente, produzimos um H tal que $\Gamma, H \vdash G$ que nos leva a dois possíveis casos:

1. Se $\Gamma \not\vdash G$, então o problema se reduz a abdução tradicional (geração de hipóteses), onde uma explicação H é produzida tal que $\Gamma, H \vdash G$ e algumas condições de minimalidade ou preferência são envolvidas para escolher a melhor explicação.
2. Se $\Gamma \vdash G$ é verdadeiro, então geramos um H que facilita sua prova (geração de lema), e produzimos um sequente $\Gamma, H \vdash G$ que possui uma prova mais simples que $\Gamma \vdash G$. Em (MDG08) há o compromisso entre encontrar uma prova menor e a busca por uma prova com baixo custo computacional.

O último caso é pertinente pois encontramos em muitos textos matemáticos a situação onde um lema é adicionado a uma prova para fazê-la mais compreensível. Portanto, temos, além de uma visão “explicativa” da hipótese H , uma visão “simplificativa”. Em último caso, dado uma prova eficiente de $\Gamma, H \vdash G$, podemos colocar essas provas juntas e aplicar a regra do corte:

$$\frac{\frac{\Gamma \vdash H \quad \Gamma, H \vdash G}{\Gamma, \Gamma \vdash G}}{\Gamma \vdash G}$$

e, portanto, obtemos uma prova eficiente de $\Gamma \vdash G$ (MDG08).

Contrário ao procedimento adotado em (MP93), aqui os ramos não necessitam ser saturados. Esse fato nos permite um ganho computacional pelo fato de não precisarmos expandir todo ramo aberto, porém impõe um problema extra à abdução baseada em corte que é decidir quando parar a expansão dos ramos do tableau.

Geramos uma hipótese abduzida como segue: dado um ramo aberto de um tableau e o subconjunto de suas fórmulas não-expandidas $\Phi = \{TA_1, \dots, TA_n, FB_1, \dots, FB_m\}$, computamos $H(\Phi)$:

$$H(\Phi) = \begin{cases} \neg A_1 & , \text{ se } n = 1, m = 0 \\ B_1 & , \text{ se } n = 0, m = 1 \\ \neg(A_1 \wedge \dots \wedge A_n) & , \text{ se } n > 1, m = 0 \\ B_1 \vee \dots \vee B_m & , \text{ se } n = 0, m > 1 \\ (A_1 \wedge \dots \wedge A_n) \rightarrow (B_1 \vee \dots \vee B_m) & , \text{ se } n > 0, m > 0. \end{cases}$$

É provado em (MDG08) que dado um ramo de um KE-tableau que contenha fórmulas Ψ e um conjunto $\Phi \subseteq \Psi$, se adicionarmos $TH(\Phi)$ ao ramo como nova hipótese, esse ramo pode ser expandido em uma subárvore fechada. Se o tableau inicial tem n ramos β_i abertos então fazemos a conjunção $H(\Phi_{\beta_1}) \wedge \dots \wedge H(\Phi_{\beta_n})$ para garantir que o KE-tableau feche.

Como exemplo, obtido de (MDG08), considere o seguinte $p \vee q, p \rightarrow r \not\vdash r$ e um de seus KE-tableaux:

$$\begin{array}{ll} 1 : T (p \vee q) & \text{Hipótese} \\ 2 : T (p \rightarrow r) & \text{Hipótese} \\ 3 : F r & \text{Hipótese} \\ 4 : F p & (T \rightarrow_2)2, 3 \\ 5 : T q & (T \vee_1)1, 4 \end{array}$$

Observe que o KE-tableau acima não foi completamente expandido, o que representa um custo computacional menor do que o método analítico. Produzimos o conjunto $\Psi = \{Fp, Tq, Fr\}$ e geramos os seguintes subconjuntos e candidatos:

<i>Candidato</i>	Φ	H
1	$\{Fp\}$	p
2	$\{Tq\}$	$\neg q$
3	$\{Fr\}$	r
4	$\{Fp, Tq\}$	$q \rightarrow p$
5	$\{Fp, Tr\}$	$p \vee r$
6	$\{Tq, Fr\}$	$q \rightarrow r$
7	$\{Tq, Fp, Fr\}$	$q \rightarrow (p \vee r)$

O candidato 3 foi eliminado por provar o dado meta sozinho, violando uma condição de admissibilidade. As fórmulas 1 e 2 são C -fórmulas que podem ser geradas pelo método de tableau analítico. Todas as outras fórmulas não são C -fórmulas mas explicações admissíveis para o dado meta. Note que todas as fórmulas implicam a última que é chamada *hipótese menos comprometedora*.

O algoritmo para a abdução baseada em corte proposicional é dado como a seguir:

Algoritmo 1 *AbducaoPROP*(Γ, G, \mathcal{T})

Sejam β_1, \dots, β_k ramos abertos em \mathcal{T} .

for $i = 1$ para k **do**

 Seja $\Psi_i = \{XA \in \beta_i \mid \text{não expandido em } \beta_i\}$

 Escolha um $\Phi_i \subseteq \Psi_i$ admissível

 Seja $H_i = H(\Phi)$

end for

return $H_1 \wedge \dots \wedge H_k$

Temos que o algoritmo apresentado é correto pelo teorema:

Teorema 3.4.1 (Correção). *O algoritmo AbducaoPROP de abdução baseada em corte proposicional é correto, isto é, dada uma entrada $\Gamma \vdash^? G$, uma fórmula H é gerada tal que $\Gamma, H \vdash G$.*

Demonstração. Em (MDG08). □

Para a abdução proposicional o método acima é o suficiente, mas não para a abdução em primeira-ordem. Considere o seguinte exemplo, obtido de (MDG08),

$$\forall x(\text{homem}(x) \rightarrow \text{mortal}(x)) \not\vdash \forall y \text{ mortal}(y),$$

e seu tableau parcialmente expandido:

$$\frac{\frac{\frac{T \forall x(\text{homem}(x) \rightarrow \text{mortal}(x))}{F \forall y \text{ mortal}(y)}}{F \text{ mortal}(a)} \quad a \text{ é novo}}{T \text{ homem}(a) \rightarrow \text{mortal}(a)}}{F \text{ homem}(a)}$$

O conjunto de todas as hipóteses candidatas para o único ramo é $\Phi = \{Fhomem(a)\}$, pelo último procedimento abduativo obtemos $H = homem(a)$. Verificamos que $\forall x(homem(x) \rightarrow mortal(x))$, $H \not\vdash \forall y mortal(y)$ e, portanto, o último método é incorreto quando aplicado à lógica de primeira-ordem.

A razão dessa falha é porque o parâmetro usado não tem significado fora do tableau. Para evitar isso aplicamos o *procedimento de Skolemização reversa* que consiste de mudar os parâmetros para variáveis e quantificá-los. Revertendo a expansão δ de $F homem(a)$ obtemos $F \forall z homem(z)$. Definindo $H = \forall z homem(z)$, obtemos o seguinte:

$$\forall x(homem(x) \rightarrow mortal(x)), \forall z homem(z) \vdash \forall y mortal(y)$$

que é claramente válido.

Aplicamos o processo de Skolemização recursivamente para as fórmulas marcadas. Como uma fórmula pode ter mais de um parâmetro, então a Skolemização reversa é aplicada em uma lista de parâmetros π obtida na ordem na qual foram introduzidas ao tableau. Formalmente o processo de Skolemização reversa é definido como:

$$\begin{aligned} Skolem^{-1}(SA, \emptyset) &= SA \\ Skolem^{-1}(SA(a), a \cup \pi) &= SQzB(z) \\ &\text{onde } SB(a) = Skolem^{-1}(SA(a), \pi); \\ &Q = \begin{cases} \forall & , \text{se } S = F \\ \exists & , \text{se } S = T \end{cases} \\ &z \text{ é uma variável que não ocorre em } B(a). \end{aligned}$$

Por exemplo, considere o sequente $\forall x \exists y m(y, x) \vdash^? \exists y \forall x m(y, x)$ e o tableau infinito para ele:

1. $T \forall x \exists y m(y, x)$ hipótese
2. $F \exists y \forall x m(y, x)$ hipótese
3. $T \exists y m(y, t)$ γ in 1
4. $F \forall x m(u, x)$ γ in 2
5. $T m(a, t)$ δ in 3
6. $F m(u, b)$ δ in 4
- \vdots

Aplicando o método proposicional obtemos $H(\Phi) = Fm(a, t) \rightarrow m(u, b)$ e aplicando a Skolemização reversa sobre os parâmetros $\{a, b\}$ do ramo temos uma fórmula $Skolem^{-1}(H(\Phi), \{a, b\}) = F \forall zw(m(w, t) \rightarrow m(u, z))$, que é uma solução correta e podemos ver facilmente a validade do sequente

$$\forall x \exists y m(y, x), \forall zw(m(w, t) \rightarrow m(u, z)) \vdash \exists y \forall x m(y, x).$$

Em (MDG08) é provado que dado um ramo aberto β em uma expansão parcial de um KE-tableau, o conjunto Ψ_β de fórmulas desse ramo, e o subconjunto $\Phi \subset \Psi_\beta$ tal que $SA \in \Phi$ não foi expandido, então adicionando $Skolem^{-1}(H(\Phi))$ às hipóteses, o ramo pode ser expandido em uma subárvore fechada.

Apresentamos agora o algoritmo para a abdução baseada em corte em primeira-ordem:

Algoritmo 2 *AbducaoFOL*(Γ, G, \mathcal{T})

Sejam β_1, \dots, β_k ramos abertos em \mathcal{T} , seja π os parâmetros usados.

for $i = 1$ para k **do**

 Seja $\Psi_i = \{XA \in \beta_i \mid \text{não expandido em } \beta_i\}$

 Escolha um $\Phi_i \subseteq \Psi_i$ admissível

 Seja $H_i = H(\Phi_i)$

end for

return $Skolem^{-1}(H_1 \wedge \dots \wedge H_k, \pi)$

Teorema 3.4.2 (Correção). *O algoritmo *AbducaoFOL* de abdução baseada em corte em primeira-ordem é correto, isto é, dada uma entrada $\Gamma \vdash^? G$, uma fórmula H é gerada tal que $\Gamma, H \vdash G$.*

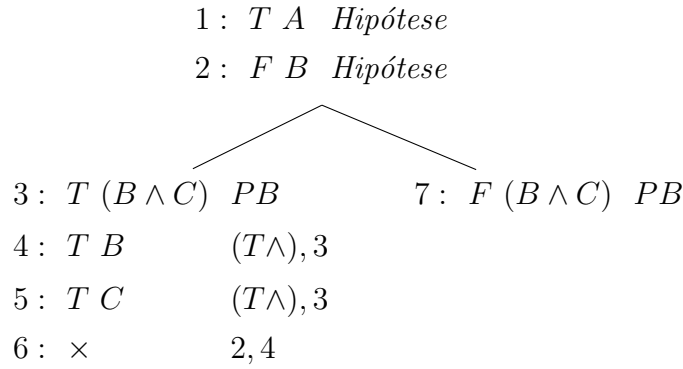
Demonstração. Em (MDG08). □

A seguir, iremos mostrar a completude tanto para a abdução baseada em corte proposicional como para a abdução baseada em corte de primeira-ordem. Iremos mostrar que o método de primeira-ordem tem uma completude sobre fórmulas “minimais”. Isso significa que o método pode sempre computar uma fórmula menos comprometedora em lógica de primeira-ordem como um todo. A ideia para ambas as provas são similares e consistem basicamente em buscar as fórmulas em cada ramo que poderiam levar a um fechamento na presença de H , se elas fossem hipóteses.

Ambas as próximas subseções são resultados novos, publicados em nosso artigo (AF13).

3.4.1 Completude para a Abdução Baseada em Corte Proposicional

Considere que o sequente $A \vdash B$ não é válido. Existem diversas fórmulas H possíveis tais que $A, H \vdash B$. Entretanto, um método analítico nunca computará, por exemplo, $A \rightarrow (B \wedge C)$, pois como C não ocorre no sequente inicial, pelas regras de expansão nunca iremos encontrá-lo em um ramo aberto e saturado e, portanto, nunca será parte de uma fórmula abduzida. Com o uso da regra não analítica PB, essa restrição simplesmente não aplica. Considere o seguinte KE-tableau não analítico para $A \vdash B$:



onde $H = A \rightarrow (B \wedge C)$ pode agora ser abduzida do ramo direito aberto com o procedimento de abdução baseada em corte.

Mantendo esse exemplo em mente, definimos a *completude de um sistema de abdução* \mathcal{S} como dado $\Gamma \vdash^? G$ e um H admissível tal que $\Gamma, H \vdash G$, então o sistema \mathcal{S} computa H' que é logicamente equivalente à H .

Para provar a completude do sistema de abdução baseado em corte, temos que considerar, para um dado $\Gamma \vdash^? G$ e um H admissível tal que $\Gamma, H \vdash G$, que existe um KE-tableau (não-analítico) que é uma expansão de $\Gamma \vdash^? G$, onde um H' , tal que $H' \equiv H$, é obtível de uma abdução baseada em corte. Esse expansão do tableau pode ser saturada ou, normalmente, apenas uma expansão parcial que deixa algum ramo aberto.

Teorema 3.4.3 (Completude). *Considere o conjunto de fórmulas $\Gamma = \{\alpha_1, \dots, \alpha_n\}$ e uma fórmula G onde $\Gamma \vdash^? G$. Seja H uma fórmula admissível tal que $\Gamma, H \vdash G$. Então existe uma expansão do tableau \mathcal{T} e uma fórmula H' gerada pela abdução baseada em corte, tal que H e H' são logicamente equivalentes.*

Demonstração. Considere uma expansão parcial de um KE-tableau \mathcal{T} de $\Gamma \vdash^? G$, cujo primeiro passo é um ramo PB sobre H . Esse passo de ramificação é possivelmente não-analítico. Suponha que TH é inserido no ramo da esquerda. Como $\Gamma, H \vdash G$, então existe uma expansão do tableau que fecha o ramo da esquerda.

Se aplicarmos a abdução baseada em corte no ramo da direita, uma dessas possíveis fórmulas abduzidas é simplesmente H , que, por hipótese, é uma fórmula admissível. Portanto, sempre podemos computar H' idêntico à H e não apenas logicamente equivalente.

Note que H pode não ser a hipótese menos comprometedora. É apenas uma possível hipótese abduzida. Mais expansões no ramo da direita podem nos levar a computar outras fórmulas abduzidas. \square

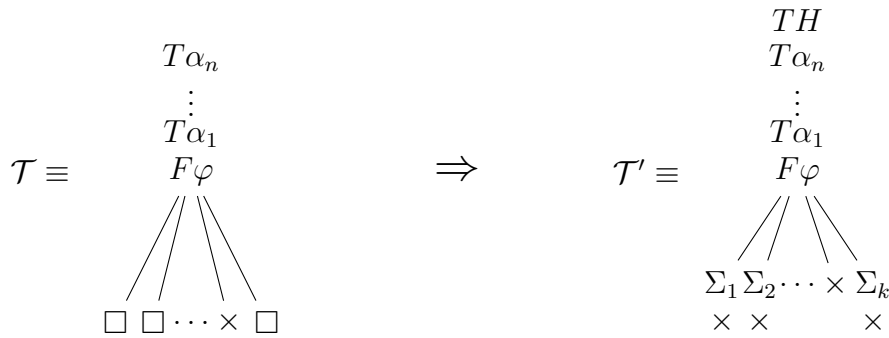
3.4.2 Completude para Abdução Minimal Baseada em Corte de Primeira-Ordem

A prova da seção anterior é aplicável ao tableau sempre que o tableau é proposicional ou de primeira-ordem. No caso da primeira-ordem, examinamos o caso do método ser

capaz de computar uma fórmula abduzida “minimal”, onde minimal pode ser entendido com respeito a propriedade de derivação, isto é, H' é menor do que H se $H \vdash H'$.

Lema 3.4.4. *Dado $\Gamma, H \vdash G$ tal que $\Gamma \not\vdash G$ e $\Gamma = \{\alpha_1, \dots, \alpha_n\}$, então a partir de um tableau saturado \mathcal{T} inicializado por $T\alpha_1, \dots, T\alpha_n, FG$ podemos construir um tableau \mathcal{T}' adicionando TH como hipótese e para cada ramo aberto σ_i de \mathcal{T}' podemos expandir uma subárvore Σ_i com expansões de TH tal que o fecha.*

Demonstração. A ideia é, partindo de um tableau saturado \mathcal{T} , construir um tableau \mathcal{T}' que tem TH como hipótese tal que cada ramo aberto π_i em \mathcal{T}' é expandido e fechado com uma subárvore Σ_i . Iremos mostrar que é possível obter Σ_i usando somente expansões de TH . A estratégia da prova é representada como a seguir:



Assumimos que \mathcal{T} foi construído do seguinte modo: considere um tableau saturado para $\Gamma, H \vdash G$, apague dele todas as fórmulas que são expansões de TH e, recursivamente, todas as fórmulas que são resultantes da expansão de uma fórmula apagada. O resultado é um tableau aberto \mathcal{T} . Portanto, Σ_i não tem nenhuma fórmula que pode ser expandida com as regras α, β, γ e δ a partir das fórmulas de \mathcal{T} porque essas expansões poderiam claramente já estarem em \mathcal{T} . Portanto, Σ_i não tem nenhuma expansão de fórmulas a partir de \mathcal{T} . \square

Teorema 3.4.5. *Considere o conjunto de fórmulas de primeira-ordem $\Gamma = \{\alpha_1, \dots, \alpha_n\}$ e a fórmula de primeira-ordem G onde $\Gamma \not\vdash G$. Seja H uma fórmula tal que $\Gamma, H \vdash G$, então existe uma fórmula de primeira-ordem H' gerada pelo algoritmo tal que $\Gamma, H' \vdash G$ e $H \vdash H'$.*

Demonstração. Nos temos, de acordo com o lema anterior, um tableau saturado \mathcal{T} , iniciado por $T\alpha_1, \dots, T\alpha_n, FG$, com os ramos abertos $\sigma_1, \dots, \sigma_k$ tal que podemos construir \mathcal{T}' a partir de \mathcal{T} adicionando TH como hipótese de \mathcal{T} e fechando cada ramo aberto com expansões $\Sigma_1, \dots, \Sigma_k$ de TH . Basicamente, para cada ramo aberto σ_i em \mathcal{T} podemos construir o conjunto de fórmulas Ψ_i que são fórmulas fechadas para cada subárvore Σ_i se tivéssemos TH como hipótese. Para cada Ψ_i geramos fórmulas $H(\Psi_i)$ usando o método. Aplicando Skolemização reversa em cada $H(\Psi_i)$ geramos Ψ'_i e, finalmente, podemos definir $H' = \Psi'_1 \wedge \dots \wedge \Psi'_k$.

É fácil verificar $H \vdash H'$. De fato, considere um tableau \mathcal{T}'' inicializado com TH, FH' . Expandimos FH' até obter k ramos com fórmulas finais da forma $F\Psi'_i$ (para $1 \leq i \leq k$). Expandimos cada $F\Psi'_i$ de forma a obter k ramos com fórmulas finais da forma $FH(\Psi_i)$ (para $1 \leq i \leq k$). Expandimos cada $FH(\Psi_i)$ e, em seguida, expandimos TH para cada ramo aberto π_i com uma subárvore Σ_i para obter o resultado. \square

O estudo da abdução clássica é fundamental para o capítulo que trata sobre a abdução probabilística, visto que, alguns métodos de abdução probabilística, para serem completos dependem que a parte da abdução proposicional que eles possuem também seja.

Capítulo 4

O Problema PSAT

4.1 Introdução

O estudo da lógica probabilística tem origens no século XIX, onde Boole, em (Boo54), já estudava a atribuição de probabilidades a sentenças lógicas. Durante um longo período o problema foi redescoberto diversas vezes (HJ00), até que, em 1965, Hailperin trouxe o problema para a programação linear (Hai86) e, finalmente em 1986, Nilsson formalizou o problema da satisfabilidade probabilística para a inteligência artificial (Nil86).

Em termos gerais, o problema da satisfabilidade probabilística (PSAT) é definido como: dadas fórmulas lógicas e probabilidades atribuídas a elas, desejamos saber se essa atribuição de probabilidades é consistente. Como lidamos constantemente com probabilidade e lógicas clássicas em inteligência artificial, a resolução do problema tem despertado interesse em áreas que trabalham com premissas probabilísticas dentro de ambientes que também possuem um formalismo clássico em sua parte.

Quanto a sua complexidade computacional, o problema da satisfabilidade probabilística (chamaremos PSAT) é NP-completo e, portanto, o teorema de Cook-Levin (Coo71) garante a existência de uma redução polinomial do PSAT para o SAT. Apesar de possuir redução à SAT, que possui bons algoritmos que o resolvem e estão em pesquisa contínua, o problema tem se mostrado mais difícil. Algoritmos que utilizam o simplex mostraram-se mais eficientes como podemos ver em (GGP88) e, atualmente, usando a combinação de ambos simplex e SAT como é visto em (Bon11), sendo esse último a principal referência deste capítulo.

Quanto à organização deste capítulo, temos que na seção 4.2 apresentaremos os conceitos e definições preliminares para o entendimento do resto do capítulo. Em seguida, na seção 4.3, o algoritmo simplex, que será utilizado com frequência neste trabalho, será apresentado. A solução do problema PSAT via simplex será exposta na seção 4.4. A forma normal atômica do problema PSAT será abordada na seção 4.5 para, na seção seguinte, a 4.6, apresentarmos o método, proposto em (Bon11), que utiliza tanto o simplex como o

SAT sua resolução. Como resultado novo, iremos apresentar, na seção 4.7, a utilização da segunda fase do simplex para estabelecer a probabilidade de uma fórmula em relação às distribuições que satisfazem uma dada instância PSAT.

4.2 Conceitos e Definições Preliminares

Adotaremos a linguagem proposicional clássica, constituída de variáveis booleanas (ou átomos) combinadas com conectivos lógicos, como apresentadas no capítulo 2 e, além disso, por convenção da área, fórmulas atômicas também poderão ser representadas pelas variáveis x e y subscritas e/ou superescritas. Apresentamos agora a definição formal do que é uma instância em PSAT:

Definição 4.2.1. *Seja $\mathcal{S} = \{S_1, \dots, S_k\}$ um conjunto de k sentenças sobre um conjunto de n proposições booleanas atômicas $X = \{x_1, \dots, x_n\}$ com os conectivos usuais da lógica proposicional clássica. Dado um conjunto de probabilidades $P = \{p_i \mid 0 \leq p_i \leq 1, 1 \leq i \leq k\}$, uma instância PSAT é formada pela atribuição das probabilidades de P às sentenças de \mathcal{S} : $\{P(S_i) = p_i \mid 1 \leq i \leq k\}$.*

Definimos agora quando uma instância PSAT é consistente:

Definição 4.2.2. *Seja $V = \{v_1, \dots, v_{2^n}\}$ o conjunto das valorações possíveis sobre um conjunto de proposições booleanas atômicas $X = \{x_1, \dots, x_n\}$, e π , uma distribuição de probabilidade sobre V . A probabilidade de uma fórmula A de acordo com π é dada por $P_\pi(A) = \sum\{\pi(v_j) \mid v_j(A) = 1\}$. Dado um conjunto $\mathcal{S} = \{S_1, \dots, S_k\}$, as probabilidades p_i atribuídas às fórmulas de \mathcal{S} serão consistentes se, e somente se, houver uma distribuição de probabilidade π sobre V que faça $P_\pi(S_i) = p_i; 1 \leq i \leq k$.*

Dizemos que uma instância PSAT é satisfatível se, e somente se, tal atribuição de probabilidades é consistente para seu conjunto de fórmulas. O exemplo a seguir mostra uma formalização de um problema em PSAT:

Exemplo 4.2.3. Três amigos afirmam que todo dia pelo menos dois deles vão ao bar, no final do expediente, com uma probabilidade de 60%. Deseja-se saber se a afirmação é plausível.

Cada amigo pode ser representado por uma variável x_1, x_2, x_3 . A hipótese de que pelo menos dois deles vão todo dia pode ser expressa pela fórmula probabilística: $P((x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_3)) = 1$ e a probabilidade de que pelo menos dois vão ao bar com probabilidade 60% pode ser expressa como $P(x_1 \wedge x_2) = 0.6$, $P(x_1 \wedge x_3) = 0.6$ e $P(x_2 \wedge x_3) = 0.6$.

Os melhores algoritmos para determinar se uma instância PSAT é satisfatível utilizam o algoritmo simplex como base. Apresentaremos a seguir uma breve descrição do algoritmo.

4.3 O Algoritmo Simplex

O problema PSAT pode ser formulado como um problema de programação linear: seja Δ uma instância em PSAT formada pela atribuição das probabilidades de um conjunto P de probabilidade às k fórmulas do conjunto S , ou seja, $\Delta = \{P(S_i) = p_i \mid 1 \leq i \leq k\}$. Dado que as sentenças de S são sobre um conjunto $X = \{x_1, \dots, x_n\}$ de proposições atômicas, definimos uma matriz $A_{k \times 2^n} = [a_{ij}]$ tal que $a_{ij} = v_j(S_i)$ e o vetor $p_{k \times 1} = [p_i]$. Temos que Δ é satisfatível se, e somente se, existe um vetor π que satisfaz as restrições:

$$A\pi = p \quad (4.1)$$

$$\pi \geq 0 \quad (4.2)$$

$$\sum \pi = 1 \quad (4.3)$$

Se uma solução π existir, dizemos que π satisfaz Δ , caso contrário dizemos que Δ é probabilisticamente insatisfatível. As restrições 4.2 e 4.3 garantem que π seja uma distribuição de probabilidades. A equação 4.3 pode ser suprimida se adicionarmos à A uma nova linha composta só de 1's tal que $A_{k+1,j} = 1$, para $1 \leq j \leq 2^n$, e também adicionarmos a posição $p_{k+1} = 1$ no vetor p .

Em (Nil86), Nilsson previu a dificuldade técnica em resolver o sistema acima devido a quantidade exponencial de colunas, já que cada coluna é uma das 2^n valorações. Em (GGP88), Kavvadias e Papadimitriou propuseram a utilização do método de geração de colunas no algoritmo simplex para resolver o problema PSAT sem explicitar tal número de colunas. Como estratégia para evitar a quantidade exponencial de colunas, eles utilizaram o teorema de Carathéodory que nos garante que se o problema de programação linear (4.1-4.3) tem solução, então existe uma solução com no máximo $k + 1$ elementos de π diferentes de zero.

O algoritmo simplex viabiliza a solução de muitos problemas de programação linear. A estratégia é caminhar por soluções viáveis minimizando (maximizando) uma função objetivo até chegar em um ponto ótimo. Dado m equações e n incógnitas de uma problema de programação linear, podemos representá-lo no seguinte formato padrão:

$$\begin{aligned} \min z &= cx \\ \text{sujeito a:} & \\ Ax &= b \\ x &\geq 0. \end{aligned} \quad (4.4)$$

A função objetivo (ou custo) é z e c é o vetor custo. Sem perda de generalidade supomos que $b \geq 0$. Também supomos que a matriz A possui m colunas linearmente independentes.

Chamamos de *Base* da matriz A um conjunto com m colunas suas linearmente independentes, $\beta = \{A_{j_1}, \dots, A_{j_m}\}$, e chamaremos por B a matriz formada por tais colunas. As variáveis correspondentes às que estão na base são chamadas de *variáveis básicas* e as demais *variáveis não básicas*. Uma base de A implica uma Solução Básica para o problema 4.4, já que basta fazer $x_j = 0$ para toda coluna $A_j \notin \beta$ e $x_B = B^{-1}b$, onde x_B é o vetor com as variáveis básicas.

Dizemos que uma solução básica é uma *Solução Básica Viável* (SBV) quando ela satisfaz todas as restrições do sistema 4.4. Cada SBV possui um custo correspondente, calculado pela aplicação da função objetivo a suas variáveis básicas, $z = c_B x_B$, onde c_B é formado a partir de c , descartando os coeficientes correspondentes às variáveis não-básicas, que são nulas e não colaboram com o custo. Tal custo é chamado de *Custo da Base*.

Através do *Pivoteamento* sempre podemos inserir uma nova coluna A_j de A em B e, retirando a coluna certa, chegarmos a outra SBV. Dada uma base não singular $\beta = \{A_{B_1}, \dots, A_{B_m}\}$ e $x_0 = \{x_{10}, \dots, x_{m0}\}$ a SBV correspondente com $\sum_{i=1}^m x_{i0} A_{B_i} = b$, temos que com a combinação linear das colunas da base, dada uma coluna A_j de A , obtemos $A_j = \sum_{i=1}^m x_{ij} B_i$. Sendo l o valor de i que minimiza a expressão $\frac{x_{i0}}{x_{ij}}$, para $1 \leq i \leq m$ e $x_{ij} > 0$, podemos substituir A_l por A_j na base, obtendo uma nova solução também viável.

A ideia do simplex é iniciar o algoritmo com uma SBV, e alterar a base B de modo iterativo, substituindo uma coluna da base a cada iteração (através do pivoteamento), obtendo novas SBVs e minimizando (ou maximizando) a função objetivo até que não haja substituição na base que diminua (ou aumente) tal função. Teremos encontrado uma base que minimiza (ou maximiza) a função objetivo quando não for possível mais diminuir (aumentar) o custo.

Agora precisamos definir um método para encontrar tais colunas, que quando entram na base, diminuem (ou aumentam) o custo da função objetivo. Dada uma base β com uma SBV x_0 , ao inserirmos uma nova coluna A_j na base, retirando A_l pelo processo de pivoteamento, essa coluna será responsável por uma parcela do custo da base $c_j x_{j1}$, onde x_{j1} é a componente correspondente à coluna A_j da nova SBV x_1 . Porém, cada coluna A_i que estava na base contribuirão diferentemente com o custo total, descontando $x_{j1} x_{ij} c_{B(i)}$ de sua parcela de custo. Então, para cada unidade de x_{j1} , a entrada de A_j na base deve colaborar no custo total com $\bar{c}_j = c_j - c'_B B^{-1} A_j$, o que denotamos por *custo reduzido*. Os coeficientes x_{ij} podem ser obtidos pelo produto $x_i = B^{-1} A_j$, então podemos reescrever o custo reduzido como $\bar{c}_j = c_j - c'_B B^{-1} A_j$. Ao escolher uma coluna com custo reduzido negativo para entrar na base, garantimos que o custo desta descreça, a menos de casos degenerados (para lidar com casos degenerados podemos usar a regra de Bland, como em (PS98)). Quando não encontrarmos mais colunas com custo reduzido negativo, teremos encontrado uma solução ótima para 4.4. Como, ao longo dos próximos capítulos, iremos lidar com minimização, buscaremos sempre colunas de custo reduzido negativo.

Embora o número de iterações do Simplex seja exponencial no pior caso com relação ao

tamanho da entrada, o caso médio é polinomial, e na prática ele acaba sendo largamente usado para resolver problemas de programação linear.

4.4 PSAT via Simplex com Geração de Colunas

Como vimos, na formulação original do problema PSAT como nas Equações (4.1) - (4.3) da seção 4.2 temos uma matriz A que possui k linhas e 2^n colunas, onde n é o número de proposições atômicas presentes na instância. Vemos de imediato que para resolver PSAT nos falta uma função objetivo, que pode ser acrescentada artificialmente como $z = 0\pi$, tanto para minimizar como para maximizar, já que qualquer solução viável nos serve. A condição 4.3 é inserida ao sistema adicionando uma linha de 1's à matriz A e fazendo $p_{k+1} = 1$. Temos agora um problema de programação linear no formato padrão:

$$\min \quad z = 0\pi \quad (4.5)$$

sujeito a:

$$A\pi = p \quad (4.6)$$

$$\pi \geq 0. \quad (4.7)$$

Encontrar uma primeira SBV para começar a solução do Simplex pode não ser trivial. Utilizaremos o *Método Simplex de Duas Fases*, onde na primeira fase encontramos uma SBV, para então otimizar a função objetivo na segunda fase. Como, nesta parte do trabalho, nos interessa apenas a viabilidade, executaremos somente a primeira fase em busca de uma SBV e, se não houver solução viável, a instância PSAT correspondente ao problema de programação linear será probabilisticamente insatisfável.

Para encontrar uma solução viável, se esta existir, introduzimos $k + 1$ variáveis artificiais $\pi_1^a, \dots, \pi_{k+1}^a$ a serem adicionadas em cada equação representada pelas linhas de A : $\sum_{j=1}^{k+1} a_{i,j}\pi_j + \pi_i^a = p_i$, $\pi_i^a \geq 0$, para $1 \leq i \leq k + 1$. Para cada variável artificial π_i^a concatenamos a A uma coluna composta de 0's, a menos do i -ésimo elemento, igual a 1, formando uma matriz $[A|I]$. Assim, as colunas referentes às variáveis artificiais π^a formam a matriz identidade, possibilitando uma base que fornece uma SBV, já que $p \geq 0$. Ao fim, queremos uma SBV com variáveis artificiais tendo probabilidade nula, então temos uma nova função objetivo artificial $\vec{1}\pi^a$ e a SBV só será solução para a instância PSAT em questão quando a função objetivo se anular. Considerando π^a dentro do vetor π , temos:

$$\min \quad z = \vec{1}\pi^a \quad (4.8)$$

sujeito a:

$$[A|I]\pi = p \quad (4.9)$$

$$\pi \geq 0. \quad (4.10)$$

Agora podemos aplicar o algoritmo Simplex, incluindo na base inicialmente as $k+1$ colunas correspondentes às variáveis incluídas artificialmente. Como mencionamos, a matriz B dessa base é a própria matriz identidade, logo as colunas são linearmente independentes e a solução é viável, com $\pi_i^a = p_i$, para $1 \leq i \leq k+1$. Armazenamos então apenas a base B , sua inversa B^{-1} , o valor das variáveis básicas π^a e o vetor p . Para procurar uma coluna não básica a ser inserida na base sem precisar explicitá-las, aplicamos o método de geração de colunas.

4.4.1 O Método de Geração de Colunas

No Simplex tradicional aplicado ao PSAT, a cada iteração calcularíamos o custo reduzido de cada coluna não básica para escolher, utilizando dos critérios mencionados, aquela que encontraria na base. Como o número de colunas é exponencial, isso é praticamente inviável para grandes instâncias PSAT. Podemos então adotar uma estratégia gulosa, gerando as colunas não básicas até encontrarmos uma com custo reduzido negativo. Porém, essas podem ser escassas, ou ainda não existirem, o que leva novamente à visitação de um número exponencial de colunas.

O custo reduzido de uma coluna A_j pode ser calculado como $\bar{c}_j = c_j - c'_B B^{-1} A_j$, e, como as colunas de A fora da base inicial tem custo c_j nulo no PSAT, podemos reescrevê-lo como $\bar{c}_j = -c'_B B^{-1} A_j$. Queremos encontrar colunas de A com custo reduzido negativo, o que significa achar colunas A_j tal que $c'_B B^{-1} A_j > 0$. O *Método de Geração de Colunas* consiste em encontrar tal coluna sem ter de listar explicitamente as colunas de A , o que é possível quando estas colunas possuem uma estrutura especial. No caso do PSAT, cada elemento a_{ij} da matriz A corresponde ao valor da fórmula S_i na valoração v_j , ou $a_{ij} = v_j(S_i)$. Logo, podemos encontrar tal coluna A_j que satisfaz $c'_B B^{-1} A_j > 0$ procurando no espaço de valorações, sem explicitar as colunas propriamente. Esse problema é chamado de *Problema Auxiliar*, que deve ser resolvido a cada iteração do Simplex. Kavvadias e Papadimitriou mostraram que o problema auxiliar é NP-completo, no caso do PSAT, quando as fórmulas são dadas no formato clausal (GGP88). Se quisermos encontrar a coluna com menor custo reduzido, temos um problema de otimização NP-difícil.

O algoritmo 3 mostra como o Simplex com geração de colunas pode ser usado para resolver uma instância PSAT. A função *Pivoteamento*(B, A_j, π) insere a coluna A_j na ma-

triz B eliminando outra coluna de maneira que mantenha a solução básica viável. O vetor custo da base c_B possui 1 para as variáveis artificiais na base e 0 para as demais, sendo este atualizado pela função *Pivoteamento* a cada iteração. A função *ProblemaAuxiliar*(S, w) recebe um conjunto de sentenças S , e um vetor w com pesos para essas sentenças e devolve a coluna que representa a valoração que satisfaz o problema auxiliar correspondente ou \emptyset se este for insatisfável. Enquanto o custo $c_B \pi_B$ for positivo, não teremos de encontrar uma solução; caso contrário, se este se anular, teremos uma solução e o algoritmo irá parar. Consideramos ainda que a última sentença, S_{k+1} , é uma tautologia e que $p_{k+1} = 1$ para representar a restrição 4.10.

Algoritmo 3 PSAT via Simplex com Geração de Colunas

Entrada: Uma instância PSAT com $k + 1$ sentenças $S = \{S_1, \dots, S_{k+1}\}$ e um vetor p com probabilidades atribuídas a elas, $P(S_i) = p_i$.

Saída: Uma matriz B e um vetor π_B tal que $B\pi_B = p$ se a instância PSAT for probabilisticamente satisfável; “UNSAT” caso contrário.

```

1:  $B \leftarrow I_{k+1}$ 
2:  $c_B \leftarrow [1 \dots 1]_{k+1,1}$ 
3:  $\pi_B \leftarrow p$ 
4: while  $c'_B \pi_B > 0$  do
5:    $A_j \leftarrow \text{ProblemaAuxiliar}(S, c'_B B^{-1})$ 
6:   if  $A_j = \emptyset$  then
7:     return UNSAT
8:   end if
9:   Pivoteamento( $B, A_j, \pi_B$ )
10:   $\pi_B \leftarrow B^{-1}p$ 
11: end while
12: return  $B, \pi_B$ 

```

4.5 A Forma Normal Atômica

Assim como na lógica proposicional, em que temos que qualquer sentença possui uma equivalente na forma normal conjuntiva (ou disjuntiva), temos também uma forma normal para o problema PSAT, introduzida em (Bon11), no sentido de que, dada uma instância PSAT, ela é probabilisticamente satisfável se, e somente se, a equivalente na forma normal também é. A vantagem da introdução de formas normais é facilitar a elaboração de métodos de solução e também a investigação de propriedades do problema em questão.

Definição 4.5.1. *Seja $S = \{S_1, \dots, S_l\}$ um conjunto de sentenças sobre um conjunto de variáveis Booleanas $X = \{x_1, \dots, x_n\}$. Dizemos que uma instância PSAT, $\Delta = \{P(S_i) = p_i \mid 1 \leq i \leq l\}$, $0 \leq p_i \leq 1$, está na Forma Normal Atômica se, e somente se, puder ser particionada em dois conjuntos, $\langle \Gamma, \Psi \rangle$, onde $\Gamma = \{P(s_i) = 1 \mid 1 \leq i \leq m\}$, e $\Psi = \{P(y_i) = p_i \mid y_i \text{ é um átomo, e } 1 \leq i \leq k\}$, com $0 \leq p_i \leq 1$, onde $l = m + k$.*

Temos que a parte Γ , da definição acima, é a parte SAT da forma normal atômica e Ψ é a parte probabilística. É demonstrado, em (Bon11), que toda instância em PSAT pode ser transformada em outra instância PSAT equivalente na forma normal atômica.

Teorema 4.5.2 (Forma Normal Atômica). *Seja $\Delta = \{P(S_i) = p_i \mid 1 \leq i \leq k\}$ uma instância do problema PSAT, então podemos construir, em tempo polinomial, uma instância PSAT $\langle \Gamma, \Psi \rangle$ na forma normal atômica, tal que Δ é satisfatível se, e somente se, $\langle \Gamma, \Psi \rangle$ também o é.*

Demonstração. Em (Bon11). □

Exemplo 4.5.3. Como exemplo da criação de uma instância PSAT na forma normal a partir de outra que não está na forma normal, considere $\Delta = \{P(x_1 \vee \neg x_2) = 0.2; P(x_2 \vee x_3) = 0.7; P(\neg x_1 \wedge x_3) = 0.5\}$. Basta adicionarmos 3 novas variáveis y_1, y_2 e y_3 e para toda valoração com probabilidade positiva fazemos com que cada nova variável tenha o mesmo valor verdade das fórmulas em Δ , ou seja, $\Gamma = \{y_1 \leftrightarrow (x_1 \vee \neg x_2), y_2 \leftrightarrow (x_2 \vee x_3), y_3 \leftrightarrow (\neg x_1 \wedge x_3)\}$ e para Ψ , definimos $\Psi = \{P(y_1) = 0.2; P(y_2) = 0.7; P(y_3) = 0.5\}$. Temos então uma instância PSAT $\langle \Gamma, \Psi \rangle$ na forma normal atômica que é satisfatível sse Δ também for.

Uma das vantagens do uso de formas normais atômicas é que o tamanho da matriz solução dela é ditada somente pela parte probabilística, como segue no teorema abaixo:

Teorema 4.5.4. *Seja $\Psi = \{P(y_i) = p_i \mid 1 \leq i \leq k\}$. Uma instância PSAT na forma normal atômica $\langle \Gamma, \Psi \rangle$ é satisfatível se, e somente se, existe uma matriz A_Ψ , com k linhas e até $k + 1$ colunas que, juntamente com um vetor π , atende às restrições 4.1 - 4.3, tal que cada coluna de A_Ψ corresponde a uma valoração consistente com Γ .*

Demonstração. Em (Bon11). □

4.6 PSAT via Simplex com Geração de Colunas Utilizando SAT-solver

Iremos apresentar aqui a ideia utilizada em (Bon11), que consiste em usar instâncias na forma normal atômica, apresentada na seção anterior, combinada com o uso de um SAT-solver para resolver o problema auxiliar no simplex para PSAT.

A ideia consiste em adaptar o algoritmo 3 para entradas na forma normal atômica. Como vimos, para toda instância em PSAT existe uma outra instância equivalente na forma normal atômica. Através do teorema 4.5.4 chegamos a um problema auxiliar diferente no qual podemos utilizar um SAT-solver como oráculo para buscar colunas que diminuam o custo. Partimos então de uma solução básica viável (SBV) para a seguinte inserindo colunas, pelo pivoteamento, que não aumentem nossa função objetivo.

Retomamos aqui a apresentação de uma instância PSAT no formato padrão do simplex:

$$\min z = 0\pi \quad (4.11)$$

sujeito a:

$$A\pi = p \quad (4.12)$$

$$\pi \geq 0, \quad (4.13)$$

onde a primeira linha de A representa a condição $\sum \pi = 1$. Dada uma instância $\langle \Gamma, \Psi \rangle$ na forma normal atômica, com m cláusulas na parte SAT Γ e k elementos na parte $\Psi = \{P(y_i) = p_i \mid 1 \leq i \leq k\}$, se aplicássemos diretamente a primeira fase do algoritmo simplex, apresentado na seção 4.3, teríamos uma matriz B como base com $m + k + 1$ linhas e colunas, pois cada cláusula C em Γ corresponderia a uma atribuição de probabilidade $P(C) = 1$. Em (Bon11) uma matriz $A_{(k+1) \times (k+1)}$ não singular é utilizada, chegando ao seguinte resultado:

Teorema 4.6.1. *Seja $\Psi = \{P(y_i) = p_i \mid 1 \leq i \leq k\}$. Uma instância PSAT na forma normal atômica $\langle \Gamma, \Psi \rangle$ é satisfatível se, e somente se, existe uma matriz binária não singular A_Ψ , com $k + 1$ linhas e colunas e um vetor $\pi_{(k+1) \times 1}$ tais que $A_\Psi \pi = [1 | p_1 \dots p_k]^T$, e que cada coluna de A_Ψ com probabilidade não nula corresponda a uma valoração sobre $\{y_1 \dots y_k\}$ consistente com Γ , a menos da primeira linha composta por 1's.*

Isso significa que, para uma problema PSAT $\Delta = \langle \Gamma, \Psi \rangle$ na forma normal, se utilizássemos o algoritmo da seção 4.3, teríamos matrizes bases com $m + k + 1$ linhas e colunas. Utilizando o teorema 4.6.1 essa quantidade cai para $k + 1$ colunas, onde k é o número de probabilidades atribuídas em Ψ .

A cada iteração adicionamos à base apenas colunas que representam valorações consistentes com Γ (chamaremos de Γ -consistentes, assim como é feito em (Bon11)), obtendo uma matriz base $B_{k+1, k+1}$ e um vetor π tais que $B\pi = p$ e $\sum \pi = 1$. Durante o processo uma coluna Γ -inconsistentes pode ter eventualmente probabilidade não nula. Se Γ for vazio, todas as valorações são consistentes com Γ , por vacuidade, caso esse que uma matriz $B_{k+1, k+1}$ e o vetor π já nos dão a solução diretamente com $B\pi = p$. A ideia do algoritmo PSAT utilizando SAT-solver para geração de colunas é resolver, a cada iteração, uma instância PSAT relaxada, ou seja, uma instância $\langle \Gamma, \Psi \rangle$ com $\Gamma = \emptyset$.

Definição 4.6.2. *Para toda instância PSAT $\langle \Gamma, \Psi \rangle$, definimos a instância PSAT relaxada como $\langle \emptyset, \Psi \rangle$, ignorando a instância SAT Γ .*

Se anularmos a soma das probabilidades das colunas Γ -inconsistentes ou obtermos uma matriz B composta somente com colunas Γ -consistentes então teremos encontrado uma solução para a instância original. Além disso é demonstrado que:

Lema 4.6.3. *Toda instância PSAT relaxada é satisfatível.*

Demonstração. Em (Bon11) □

Na prova acima é utilizada uma matriz $I^* = [a_{i,j}]$, onde:

$$a_{i,j} = \begin{cases} 1, & i \leq j, \\ 0, & i > j. \end{cases}$$

É demonstrada uma clara vantagem em utilizar esse tipo de matriz ao invés da matriz identidade: as colunas podem representar valorações, assim a função objetivo a ser minimizada será a soma das probabilidades referentes às colunas de I^* não consistentes com Γ .

O algoritmo 4 é o resultado da aplicação do Simplex com o método de geração de colunas utilizando um SAT-*solver*. A função *ProblemaAuxiliar* deve retornar colunas com custo reduzido negativo e que sejam Γ -consistentes. Um exemplo, encontrado em (Bon11) é:

Algoritmo 4 PSAT via Simplex com Geração de Colunas para Forma Normal Atômica

Entrada: Uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ na forma normal atômica, onde $\Psi = \{P(S_i) = p_i \mid 1 \leq i \leq k\}$ e Γ é uma instância SAT sobre as variáveis $y_1, \dots, y_k, x_1, \dots, x_n$.

Saída: Uma matriz B e um vetor π_B tal que $B\pi_B = [1|p]'$ se Δ for probabilisticamente satisfatível; “UNSAT” caso contrário.

```

1:  $B \leftarrow I_{k+1}^*$ 
2:  $c_B \leftarrow [1 \dots 1]_{k+1,1}$ 
3:  $\pi_B \leftarrow p$ 
4: while  $c'_B \pi_B > 0$  do
5:    $A \leftarrow ProblemaAuxiliar(S, c'_B B^{-1}, \Gamma)$ 
6:   if  $A_j = \emptyset$  then
7:     return UNSAT
8:   end if
9:   Pivoteamento( $B, A_j, \pi_B$ )
10:   $\pi_B \leftarrow B^{-1}p$ 
11: end while
12: return  $B, \pi_B$ 

```

Exemplo 4.6.4. Seja $\langle \Gamma, \Psi \rangle$ uma instância PSAT na forma normal atômica, tal que $\Gamma = \{\neg y_1 \vee y_2 \vee y_3\}$ e $\Psi = \{p(y_1) = 0.7, p(y_2) = 0.6, p(y_3) = 0.2\}$.

Como temos $k = 3$ probabilidades atribuídas em Ψ , ao submetermos $\langle \Gamma, \Psi \rangle$ ao algoritmo PSAT obtemos $B = I_{4 \times 4}^*$ como base inicial:

$$B = \begin{matrix} 1 \\ y_1 \\ y_2 \\ y_3 \end{matrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

além disso, temos o vetor

$$p = \begin{bmatrix} 1 \\ 0.7 \\ 0.6 \\ 0.2 \end{bmatrix},$$

e inicialmente $c_B = [1 \ 1 \ 1 \ 1]$. Na execução encontrada em (Bon11), após algumas iterações, obtemos a mesma matriz B e $c_B = [0 \ 1 \ 0 \ 0]$, onde a segunda coluna é Γ -inconsistente. Uma coluna de custo reduzido A_j tal que $-c'_B B^{-1} A_j < 0$ é calculada de forma que obtemos $A_j = [1 \ 1 \ 0 \ 1]$. O problema é resolvido ao substituir a segunda coluna de B por A_j :

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix},$$

atualizando $c_B = [0 \ 0 \ 0 \ 0]$. O vetor solução π era obtido ao fazer

$$\pi_B = B^{-1}p = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0.7 \\ 0.6 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.1 \\ 0.4 \\ 0.2 \end{bmatrix}.$$

Os métodos que vimos para resolver o problema PSAT utilizam o simplex de duas fases e, mais precisamente, só a primeira fase do método, onde é determinada uma solução básica viável. Iremos apresentar, na seção a seguir, como resultado novo, um procedimento que permite o uso da segunda fase do simplex de forma que podemos aumentar ou diminuir o valor da probabilidade de uma fórmula dada uma instância PSAT.

4.7 Troca da Função Objetivo para a Segunda Fase do Simplex

Nas seções até agora apresentamos métodos que, no intuito de encontrar uma solução básica viável para uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$, utilizavam somente a primeira fase do simplex de duas fases. No presente trabalho, após estabelecer uma solução básica viável, fornecemos um método que permite a mudança da função objetivo de forma que podemos diminuí-la (ou aumentá-la) dada como solução básica inicial aquela obtida ao fim da primeira fase. A nova função objetivo representa a probabilidade de uma determinada sentença proposicional A em uma distribuição probabilística que satisfaça a instância PSAT Δ em questão.

Para que possamos diminuir (ou aumentar) o custo da função objetivo na segunda fase do simplex, precisamos preparar a base para que a inserção de uma nova coluna Γ -consistente não acarrete que uma outra coluna Γ -inconsistente passe a ter probabilidade maior que zero. Um exemplo em que isso pode acontecer:

Exemplo 4.7.1. Suponha que dada uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$, onde $\Gamma = \{(\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4) \vee (x_1 \wedge x_2 \wedge x_3 \wedge \neg x_4) \vee (x_1 \wedge x_2 \wedge x_3 \wedge x_4) \vee (x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4)\}$, com $P(x_1) = 0.8$, $P(x_2) = 0.4$, $P(x_3) = 0.4$, $P(x_4) = 0.2$. Temos uma das soluções dada pela primeira fase do simplex:

$$B\pi = \begin{matrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.2 \\ 0.4 \\ 0.0 \\ 0.2 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.8 \\ 0.4 \\ 0.4 \\ 0.2 \end{bmatrix}$$

onde $c = [0 \ 0 \ 1 \ 0 \ 0]$.

Note que a única coluna consistente fora da base é $v = [1 \ 1 \ 0 \ 1 \ 0]^T$ e para ser inserida, considerando o algoritmo de pivoteamento, temos:

$$\alpha = B^{-1}v = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix},$$

vemos que a quarta coluna é a candidata a ser substituída, pois a menor relação π_j/α_j é $\pi_4/\alpha_4 = 0.2$. Após a substituição temos um novo sistema:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.8 \\ 0.4 \\ 0.4 \\ 0.2 \end{bmatrix}$$

com solução $\pi = [0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.2]$ e vemos que a terceira coluna, que é Γ -inconsistente, passou a ter valor $\pi_3 = 0.2 > 0$.

O motivo é que as colunas que são Γ -inconsistentes podem estar na combinação linear das colunas que entram na base. Isso faz com que a inserção de algumas colunas Γ -consistentes possam aumentar a probabilidade de colunas Γ -inconsistentes. A ideia para contornar esse problema é obter uma base viável que minimiza a quantidade de colunas Γ -inconsistentes (lembrando que uma coluna i é Γ -inconsistente se $c_i = 1$), ou seja,

que tenhamos um cc^T mínimo. Dessa forma nenhuma outra coluna Γ -consistente poderia substituir qualquer coluna Γ -inconsistente pois senão teríamos uma base não-invertível (se pudessemos substituir não teríamos então uma quantidade mínima). O seguinte teorema elucida o que queremos:

Teorema 4.7.2 (psat-normalização). *Dada uma instância PSAT $\langle \Gamma, \Psi \rangle$ tal que ela seja resolvida pelo PSAT-solver retornando uma matriz B , um vetor c e a solução π . Existe uma solução com uma matriz B' , com c' e π' , obtida a partir de B , c e π , tal que B' é invertível, $\pi' = \pi$ e $c'c'^T$ é mínimo.*

Demonstração. Vamos por casos:

1. Se, na solução inicial, já temos $cc^T = 0$, então já estamos no caso.
2. Se $cc^T = n$ então criamos n vetores w_i , com $1 \leq i \leq n$, onde cada w_i é composto por zeros exceto por uma única posição que é igual a 1 e, além disso, temos $c = w_1 + \dots + w_n$. Dado inicialmente $B' = B$, para cada vetor w_i de $i = 1$ até $i = n$ buscamos uma coluna Γ -consistente C tal que $w_i B'^{-1} C \neq 0$, se tal coluna existir, substituímos a coluna B'_i de B' por C . Temos que:
 - A cada substituição a solução π continuará intacta pois para $c_i = 1$ temos $\pi_i = 0$ e portanto a nova coluna inserida nessa posição terá probabilidade zero.
 - Após uma substituição B' continuará invertível: suponha o caso em que a substituição de uma coluna B'_j por C tornou B' não mais invertível, logo, C é uma combinação linear das outras colunas $B'_k (k \neq j)$ colunas. Seja então $C = a_1 B'_1 + \dots + a_{j-1} B'_{j-1} + a_{j+1} B'_{j+1} + \dots + a_m B'_m$ e, antes de entrar na base que era invertível, tínhamos $C = b_1 B'_1 + \dots + b_{j-1} B'_{j-1} + b_j B'_j + b_{j+1} B'_{j+1} + \dots + b_m B'_m$ com $b_j \neq 0$ já que, pela hipótese, temos $w_j B'^{-1} C \neq 0$. É fácil ver que B'_j é uma combinação linear das colunas $B'_k (k \neq j)$, basta fazer $B'_j = \frac{(a_1 - b_1)}{b_j} B'_1 + \dots + \frac{(a_{j-1} - b_{j-1})}{b_j} B'_{j-1} + \frac{(a_{j+1} - b_{j+1})}{b_j} B'_{j+1} + \dots + \frac{(a_m - b_m)}{b_j} B'_m$ e, portanto, antes da substituição B' já não era invertível, um absurdo.
 - Se para um determinado w_i não existe uma coluna Γ -consistente C tal que $w_i B'^{-1} C \neq 0$, significa que a substituição de B'_i por qualquer coluna C que é Γ -consistente torna a matriz B' não-invertível (com $B'^{-1} C$ notamos que C é a combinação das outras $B'_j (j \neq i)$ colunas Γ -consistentes). Não substituímos nada e pulamos para o próximo passo com o vetor w_{i+1} (se $i < n$). Outro fato é que a coluna B'_i não pode ser substituída posteriormente após a substituição de uma coluna B'_j (com $i < j$), pois toda coluna que entra possui coeficiente zero para B'_i em sua combinação linear com as colunas de B' e, conseqüentemente, a coluna que sai também terá coeficiente zero para B'_i .

Logo, percorrendo esses passos de $i = 1$ até $i = n$, obteremos uma matriz B' com $c'c'^T$ mínimo e $\pi' = \pi$. \square

Chamaremos uma matriz solução B de uma instância $\Delta = \langle \Gamma, \Psi \rangle$, como acima mostrada, de *matriz psat-normalizada para Δ* e de *psat-normalização* o processo descrito acima para gerar uma matriz psat-normalizada para Δ a partir de outra que não é psat-normalizada. Note que o vetor c em matrizes psat-normalizadas só possui valor $c_i = 1$ se e somente se a respectiva coluna B_i é Γ -inconsistente e linearmente independente de qualquer outra coluna Γ -consistente fora da base. No algoritmo 5 vemos uma versão explícita do método. O próximo teorema nós fornece a principal vantagem de obtermos matrizes

Algoritmo 5 Geração de Matriz Psat-normalizada

Entrada: Uma matriz B , um vetor c e um vetor π que resolvem uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ na forma normal atômica.

Saída: Uma matriz B' e um vetor c' que resolvem a instância Δ e $c'c'^T$ é mínimo, e o vetor π' .

```

1:  $B' := B$ 
2:  $c' := c$ 
3:  $\pi' := \pi$ 
4: if  $c'c'^T = 0$  then
5:   return  $B', c', \pi'$ 
6: end if
7: Defina  $w_i$  como um vetor unitário tal que  $w_1 + w_2 + \dots + w_n = c$ 
8: for  $i = 1$  to  $|cc^T|$  do
9:   Encontre uma coluna  $C$ , que é  $\Gamma$ -consistente, tal que  $w_i B'^{-1} C \neq 0$ 
10:  if  $C$  existe then
11:     $B' := B'[B_i/C]$ 
12:     $c'_i := 0$ 
13:  end if
14: end for
15: return  $B', c', \pi'$ 

```

psat-normalizadas para uma dada instância PSAT:

Teorema 4.7.3. *Dada uma instância $\langle \Gamma, \Psi \rangle$ e sua solução com uma matriz psat-normalizada B , um vetor c e π , então temos que a inserção, através do pivoteamento do PSAT-solver, de qualquer coluna não-básica C que é Γ -consistente, gerando B' , c' e π' , é tal que $c'\pi = 0$.*

Demonstração. Se a coluna não-básica C substituísse B_i em B , onde $c_i = 1$, teríamos então uma matriz não-invertível já que B é psat-normalizada. Portanto, toda coluna B_i de B com $c_i = 1$ possui coeficiente zero na combinação linear de C na base B . Logo, a inserção de C não afetará a probabilidade dessas colunas. \square

Exemplo 4.7.4. Dada a matriz B , o vetores $c = [0 \ 0 \ 1 \ 0 \ 0]^T$ e $v = [1 \ 1 \ 0 \ 1 \ 0]^T$ do exemplo 4.7.1, temos que o único vetor w_i que podemos formar é $w_1 = [0 \ 0 \ 1 \ 0 \ 0]^T$. Vemos

que $w_1 B^{-1}v = -1$ e, portanto, aplicando o algoritmo 5, temos que a terceira coluna será substituída por v e obtemos B' tal que:

$$B'\pi = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.2 \\ 0.4 \\ 0.0 \\ 0.2 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.8 \\ 0.4 \\ 0.4 \\ 0.2 \end{bmatrix}.$$

Note que π_3 continua a ter valor 0, já que a coluna inserida é Γ -consistente. Observe que a nova matriz B' é invertível e preserva a solução original.

No exemplo anterior vimos que a solução com a matriz psat-normalizada possui as mesmas colunas Γ -consistentes com valor positivo (maior que zero) que a instância original. Isso nos dá margem a criar a seguinte definição:

Definição 4.7.5 (Psat-equivalência). *Dada uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ e sua solução com a matriz B e vetores c e π , dizemos que uma outra solução, com a matriz B' e vetores c' e π' , é psat-equivalente à primeira se toda coluna B_i que é Γ -consistente que ocorre com probabilidade positiva ($\pi_i > 0$) também ocorre em B' com a mesma probabilidade π_i .*

Agora podemos estabelecer o seguinte teorema:

Teorema 4.7.6. *Dada duas soluções para uma instância $\Delta = \langle \Gamma, \Psi \rangle$, uma com a matriz B , um vetor c e π e outra com a matriz B' , vetor c' e π' , podemos obter uma solução psat-equivalente à solução com B' , c' e π' , partindo da solução básica com B , c e π , inserindo apenas colunas Γ -consistentes, de tal forma que, a cada iteração, as colunas que são Γ -consistentes ainda formam uma solução para Δ .*

Demonstração. Para cada coluna B'_i de B' que é Γ -consistente e possui $\pi_i > 0$ criamos uma fórmula proposicional $F_{B'_i}$ correspondente à valoração que a satisfaz. Seja F a disjunção de cada fórmula $F_{B'_i}$. A ideia é resolver a instância $\Delta' = \langle \Gamma \cup F, \Psi \rangle$ dando como base inicial uma matriz psat-normalizada, obtida a partir de B , e iniciar o algoritmo PSAT-*solver* com $c' = [1 \ 1 \ \dots \ 1]$. A cada iteração buscamos adicionar colunas que são $\Gamma \cup F$ -consistentes e por fim obteremos uma matriz B^* tal que $c'_i = 0$ somente para colunas $\Gamma \cup F$ -consistentes. Cada coluna B^*_i de B^* que possui probabilidade positiva também está em B' com a mesma probabilidade, caso contrário, teríamos duas soluções para Δ com a mesma base B' , absurdo.

Como tanto para psat-normalizar como para, posteriormente, inserir colunas sempre buscamos colunas Γ -consistentes (são inseridas sempre sobre uma matriz já psat-normalizada), temos que cada iteração é uma solução para Δ . \square

E com base na prova do teorema anterior, agora podemos dizer como, a partir de uma solução de uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$, obtemos uma solução de uma outra instância $\Delta' = \langle \Gamma \cup H, \Psi \rangle$ pelo próximo teorema:

Teorema 4.7.7. *Dada duas instâncias $\Delta = \langle \Gamma, \Psi \rangle$ e $\Delta' = \langle \Gamma \cup H, \Psi \rangle$, ambas probabilisticamente satisfatíveis, podemos obter iterativamente a partir de qualquer solução com B, c e π da instância Δ uma solução com B', c', π' da instância Δ' , tal que B' é psat-normalizada para Δ' e, além disso, cada iteração é uma solução de Δ .*

Demonstração. Suponha que resolvamos a instância Δ e, aplicando o algoritmo de psat-normalização, tenhamos como solução com B, c, π onde B é psat-normalizada para Δ . Inicializamos o PSAT-solver dando B como matriz inicial e um vetor $c' = [1 \ 1 \ \dots \ 1]$. Ao final obteremos uma solução de Δ' com B', c', π' e essa solução é obtida buscando somente colunas $\Gamma \cup H$ -consistentes a cada iteração, pelo teorema 4.7.3, temos que cada iteração é uma solução para Δ . Por fim, basta psat-normalizarmos para Δ' a solução com B', c', π' , onde cada passo ainda preserva uma solução para Δ , e obteremos o resultado. \square

A prova do teorema anterior nos fornece um método de encontrar distribuições de probabilidades que satisfaçam $\Delta = \langle \Gamma, \Psi \rangle$ e que diminuam (ou aumentem) a probabilidade de uma fórmula A . Primeiramente, mudamos a função objetivo de forma a obter o valor $\pi(A)$ como custo: dada uma solução inicial com B, c e π , com B psat-normalizada, para $\Delta = \langle \Gamma, \Psi \rangle$, criamos um novo vetor de custo w tal que atualizamos w fazendo $w_i = 1$ se a coluna B_i é $\Gamma \cup A$ -consistente ou $w_i = 0$, caso contrário. Podemos, agora, buscar colunas $\Gamma \cup A$ -consistentes que diminuam (ou aumentem) o custo de $\pi(A)$. Uma versão explícita pode ser vista no algoritmo 6.

Algoritmo 6 ReduzCusto

Entrada: Uma matriz B psat-normalizada, os vetores c, π que resolvem uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ e uma fórmula A com probabilidade $\pi(A) = p_1 > 0$.

Saída: Uma matriz B' , um vetor w e um vetor π' que resolvem a instância Δ e tal que $w\pi' = \pi'(A) < \pi(A)$. NULO se não for encontrado.

- 1: $w_i := 1$ se e somente se B_i é $\Gamma \cup A$ -consistente, $w_i := 0$ caso contrário.
 - 2: $C := ProblemaAuxiliar(\Gamma \cup \neg A, B', w, \pi)$
 - 3: **if** C existe **then**
 - 4: *Pivoteamento*(B, w, C)
 - 5: **else**
 - 6: **return** NULL
 - 7: **end if**
 - 8: $\pi' := B^{-1}p$
 - 9: **return** B', w, π'
-

Os teoremas vistos aqui são importantes porque nos dizem que dada uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ com duas soluções s_1 e s_2 , podemos, a partir de s_1 , obter uma outra

solução que é psat-equivalente à s_2 e, além disso, a cada iteração as colunas que são Γ -consistentes formam uma solução para Δ . Isto nos diz que, dada uma solução para uma instância Δ , todo o seu conjunto de soluções é “alcançável” a partir dela.

Outro ponto importante desta seção é a possibilidade de buscar distribuições que diminuam (ou aumentem) o valor da probabilidade de A para uma dada instância PSAT Δ de forma que a cada iteração ainda tenhamos uma solução de Δ . Isto é importante quando A representar algum fato ou propriedade desejável em um sistema ou modelo computacional representado por Δ .

Capítulo 5

Abdução Probabilística

5.1 Introdução

No capítulo 3 vimos a definição da abdução tradicional para a lógica proposicional, que consistia em dada uma base de conhecimento Γ e uma fórmula A não derivada dessa base, gerávamos uma hipótese H , que juntamente com a base de conhecimento explicava A , ou seja, $\Gamma, H \vdash A$. Vamos agora para o caso da lógica probabilística em que temos fórmulas proposicionais associadas a probabilidades. O *problema da abdução probabilística* é formulado aqui como: dada uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ na forma normal atômica e uma fórmula A , existe uma distribuição de probabilidades π tal que π satisfaz Δ e temos $P_\pi(A) = 0$. A *solução de um problema de abdução probabilística* é a geração de uma fórmula H que juntamente com Δ faz com que A tenha probabilidade positiva. Dada uma instância PSAT $\langle \Gamma, \Psi \rangle$, temos duas formas de fazer isso:

1. Adicionar uma nova fórmula H à Γ , ou seja, $P(H) = 1$ (Abdução de certezas);
2. Adicionar uma nova fórmula $(H \leftrightarrow a)$, onde a é uma proposição atômica nova, à Γ e a à Ψ tal que $P(a) = p$, onde $0 < p < 1$ (Abdução de informações probabilísticas).

A primeira forma representa a geração de uma explicação dentro de uma visão clássica de lógica, onde um fato ou evento realmente aconteceu e, portanto, ele explica porque A acontece com probabilidade positiva. A segunda forma é quando uma informação, apesar de ser imprecisa, faz com que A possua uma probabilidade positiva, ou seja, a probabilidade do fato ou um evento H , juntamente com Δ , possui uma correlação com A . Uma condição de admissibilidade para ambos os casos é que a hipótese H não explique sozinha a probabilidade de A , ou seja, que exista uma distribuição probabilística π , tal que $\pi(H) > 0$ e $\pi(A) = 0$.

Apresentaremos, neste capítulo, alguns métodos de abdução para a Lógica Probabilística (PSAT). Dada uma base de conhecimento $\Delta = \langle \Gamma, \Psi \rangle$ e uma fórmula A tal que exista uma distribuição de probabilidades π que satisfaz Δ e $\pi(A) = 0$, a ideia central, neste

capítulo, é fornecer métodos que utilizam aquele visto na seção 4.7 do capítulo 4 para a geração de uma hipótese H que faça com que a fórmula A tenha probabilidade positiva quando H é adicionada à Γ . Também apresentaremos um método que permite a geração de uma hipótese H associada a uma probabilidade, ou seja, $P(H) = a$, onde $0 < a < 1$, de forma que adicionamos informações novas tanto à Γ quanto à Ψ .

Quanto à organização deste capítulo, iremos apresentar, na seção 5.2, os conceitos preliminares para o entendimento das seções seguintes. Na seção seguinte, em 5.3, apresentaremos um método que gera fórmulas com probabilidades positivas dada uma instância PSAT qualquer e que servirá de base para a elaboração dos outros métodos. Na seção 5.4, forneceremos dois métodos que utilizam os métodos clássicos apresentados no capítulo 3 em conjunto com o método de minimização da seção 4.7 do capítulo 4. Em seguida, na seção 5.5, temos um método que utiliza a hipótese H gerada por um método clássico de abdução, juntamente com o método da seção 4.7 do capítulo 4, para gerar uma fórmula abduzida H com probabilidade entre zero e um.

5.2 Conceitos e Definições Preliminares

Vamos utilizar um símbolo que nos fale quando uma instância Δ , como descrita acima, é satisfeita por uma distribuição π tal que $\pi(A) = 0$ para uma dada fórmula A . Utilizaremos o símbolo $|\approx$, como em (FdB10):

Definição 5.2.1. $\langle \Gamma, \Psi \rangle |\approx A$, se e somente se, para toda distribuição de probabilidade π tal que π satisfaz $\langle \Gamma, \Psi \rangle$ então $P_\pi(A) > 0$.

A motivação por trás dessa definição é que se $\langle \Gamma, \Psi \rangle \not|\approx A$ então temos que $\langle \Gamma, \Psi \rangle$ não força a probabilidade de A ser positiva, ou seja, existe uma distribuição de probabilidades π tal que π satisfaz $\langle \Gamma, \Psi \rangle$ e $\pi(A) = 0$. Não devemos confundir o símbolo $|\approx$ (inferência probabilística) com \vdash (inferência clássica), visto que, o último também será bastante utilizado daqui por diante.

5.3 Geração de Fórmulas com Probabilidades Positivas para Instâncias PSAT

Antes de apresentar nossos métodos de como gerar uma hipótese H em PSAT, iremos apresentar um método que nos permite inferir fórmulas que ocorrem com probabilidade positiva dada uma instância PSAT $\langle \Gamma, \Psi \rangle$ qualquer. A ideia é similar ao operador $Th(\Gamma)$ que encontramos em lógica proposicional na literatura (em (End72), por exemplo), onde temos que se $\Gamma \vdash A$ então $A \in Th(\Gamma)$. Aqui estabelecemos um método que permite que dada uma instância $\langle \Gamma, \Psi \rangle$ na forma normal, podemos gerar qualquer fórmula A tal que $\langle \Gamma, \Psi \rangle \not|\approx A$. Para esse fim, temos o seguinte teorema, que é similar ao teorema da dedução:

Teorema 5.3.1. *Se uma fórmula H é tal que $\langle \Gamma \cup H, \Psi \rangle$ é probabilisticamente insatisfável, então $\langle \Gamma, \Psi \rangle \approx \neg H$*

Demonstração. A prova é imediata. Temos que se $\langle \Gamma \cup H, \Psi \rangle$ é probabilisticamente insatisfável significa que associar $\neg H$ com uma probabilidade $P(\neg H) = 0$ também é insatisfável. Logo $\langle \Gamma, \Psi \rangle \approx \neg H$. □

Vamos, agora, apresentar um método que gera fórmulas H tais que $\langle \Gamma, \Psi \rangle \approx H$ utilizando o teorema anterior. Suponha que apliquemos o algoritmo PSAT-*solver* em $\langle \Gamma, \Psi \rangle$, resolvendo o sistema:

$$\begin{aligned} \min \quad & c\pi \\ \text{sujeito a} \quad & \\ & A\pi = p \\ & \pi \geq 0. \end{aligned}$$

Temos que o método irá parar quando $c\pi = 0$. A ideia aqui é fazer o método, apresentado em (Bon11) e exposto no capítulo 4, funcionar reduzindo o custo $c\pi$, mas sem zerá-lo.

Assim como no algoritmo PSAT-*solver*, o vetor c é composto inicialmente com 1's e uma matriz inicial I^* representando uma solução básica é fornecida para um vetor de probabilidades p . A cada iteração uma coluna Γ -consistente com custo reduzido é buscada de forma que o custo $c\pi$ seja reduzido. Como vimos, é garantido que se não houver uma coluna Γ -consistente com custo reduzido, então a instância em questão é probabilisticamente insatisfável. Para gerar um H' tal que $\langle \Gamma \cup H', \Psi \rangle$ é probabilisticamente insatisfável, utilizando esse método, basta impedir que o algoritmo encontre colunas Γ -consistentes com custo reduzido em algum ponto da execução. Lembramos que para encontrar uma coluna com custo reduzido, o método PSAT-*solver*, busca, a cada iteração, encontrar uma coluna Γ -consistente A_j tal que $-cB^{-1}A_j < 0$. Suponha que, em determinado ponto, as colunas que reduzem o custo sejam c_1, c_2, \dots, c_n e as fórmulas que as representam as conjunções de suas proposições atômicas como sendo C_1, C_2, \dots, C_n , basta fazermos $H' = \neg C_1 \wedge \neg C_2 \wedge \dots \wedge \neg C_n$ e então teremos $\langle \Gamma \cup H', \Psi \rangle$ insatisfável, pois não há colunas que reduzam o custo $c\pi$ à zero e, portanto, $\langle \Gamma, \Psi \rangle \approx \neg H$. A qualquer momento onde tenhamos $0 < c\pi < 1$, podemos parar e gerar H' .

Se as únicas fórmulas que pudéssemos gerar fossem do modo acima, estaríamos bem restritos. Uma quantidade bem maior de fórmulas H , tais que $\langle \Gamma, \Psi \rangle \approx H$, pode ser geradas observando o seguinte fato: dado $H' = \neg C_1 \wedge \neg C_2 \wedge \dots \wedge \neg C_n$, como visto anteriormente, podemos obter, através dos métodos vistos de abdução clássica, fórmulas H'' tais que $\Gamma, H'' \vdash H'$. É fácil ver que uma fórmula H'' obtida dessa forma faz com que $\langle \Gamma \cup H'', \Psi \rangle$ também seja probabilisticamente insatisfável. Podemos fazer $H = \neg H'$ ou $H = \neg H''$ ou $H = \neg(H'' \vee H')$ que obteremos $\langle \Gamma, \Psi \rangle \approx H$. Expressando o algoritmo de

forma explícita, temos:

Algoritmo 7 GeraçãoPositivadeFormulas

Entrada: Uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ satisfatível.

Saída: Uma fórmula H tal que $\Delta \approx H$.

```

1:  $B \leftarrow I_{k+1}$ 
2:  $c_B \leftarrow [1 \dots 1]_{k+1,1}$ 
3:  $\pi_B \leftarrow p$ 
4:  $IniciarAbducao \leftarrow F$ 
5: while  $c_B \pi_B > 0$  e  $IniciarAbducao = F$  do
6:    $C \leftarrow ProblemaAuxiliar(\Gamma, c_B, B^{-1})$ 
7:    $Pivoteamento(B, C, \pi_B)$ 
8:    $\pi_B \leftarrow B^{-1}p$ 
9:   if  $c_B \pi_B = 0$  then
10:     $c_B \leftarrow [1 \dots 1]_{k+1,1}$ 
11:   end if
12:    $IniciarAbducao \leftarrow EscolhaValorPositivo(c_B, \pi_B)$ 
13: end while
14:  $H' \leftarrow \neg C_1 \wedge \dots \wedge \neg C_n$ , onde  $C_i$  é obtido da coluna  $c_i$  que reduz o custo.
15:  $H \in \{H', Abd(\Gamma, H'), H' \vee Abd(\Gamma, H')\}$ 
16: return  $\neg H$ 

```

Podemos ver que a geração de fórmulas dessa forma é o suficiente pelo próximo teorema:

Teorema 5.3.2 (Completeness). *Dado $\langle \Gamma, \Psi \rangle$ probabilisticamente satisfatível, temos que toda fórmula A tal que $\langle \Gamma, \Psi \rangle \approx A$ pode ser gerada pelo método acima.*

Demonstração. Dado $\langle \Gamma, \Psi \rangle \approx A$, onde $\langle \Gamma, \Psi \rangle$ é probabilisticamente satisfatível, temos que $\langle \Gamma \cup \neg A, \Psi \rangle$ é probabilisticamente insatisfatível. Ao iniciarmos *PSAT-solver* com a instância $\langle \Gamma \cup \neg A, \Psi \rangle$, ele irá parar em um ponto onde não encontra mais colunas $\Gamma \cup \neg A$ -consistentes que reduzam o custo. Nesse ponto, onde não temos mais colunas $\Gamma \cup \neg A$ -consistentes que reduzam o custo, ao obter o vetor c e a matriz B , determinamos as colunas Γ -consistentes que reduzem o custo $c\pi$ para a instância $\langle \Gamma, \Psi \rangle$ (lembramos que uma coluna Γ -consistente A_j só reduz o custo $c\pi$ se $-cB^{-1}A_j < 0$). Supondo essas colunas como sendo c_1, c_2, \dots, c_n e as fórmulas que a representam como C_1, C_2, \dots, C_n , definimos $T = \neg C_1 \wedge \neg C_2 \wedge \dots \wedge \neg C_n$. Note que $\Gamma, \neg A \vdash T$, pois, caso contrário, alguma coluna c_i seria $\Gamma \cup \neg A$ -consistente e o método não teria parado ao buscar colunas $\Gamma \cup \neg A$ -consistentes para reduzir o custo. Como $\neg A$ pode ser obtido por abdução clássica a partir de Γ e T e, além disso, todas essas iterações são possíveis para uma dada instância $\langle \Gamma, \Psi \rangle$ utilizando o método, temos, portanto, que dado $\langle \Gamma, \Psi \rangle \approx A$ o método é capaz de gerar A . \square

Apresentamos agora uma aplicação do método em um exemplo obtido a partir de (Bon11):

Exemplo 5.3.3. Seja $\langle \Gamma, \Psi \rangle$ uma instância PSAT na forma normal atômica, tal que $\Gamma = \{\neg y_1 \vee y_2 \vee y_3\}$ e $\Psi = \{p(y_1) = 0.7, p(y_2) = 0.6, p(y_3) = 0.2\}$.

Como temos $k = 3$ probabilidades atribuídas em Ψ , ao submetermos $\langle \Gamma, \Psi \rangle$ ao algoritmo PSAT obtemos $B = I_{4 \times 4}^*$ como base inicial:

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Além disso, temos o vetor

$$p = \begin{bmatrix} 1 \\ 0.7 \\ 0.6 \\ 0.2 \end{bmatrix}$$

e inicialmente $c_B = [1 \ 1 \ 1 \ 1]$. Na execução encontrada em (Bon11), após algumas iterações, obtemos a mesma matriz B e $c_B = [0 \ 1 \ 0 \ 0]$, onde a segunda coluna é Γ -inconsistente. Uma coluna de custo reduzido A_j tal que $-c'_B B^{-1} A_j < 0$ é calculada de forma que obtemos $A_j = [1 \ 1 \ 0 \ 1]$. O problema era resolvido ao substituir a segunda coluna de B por A_j :

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix},$$

atualizando $c_B = [0 \ 0 \ 0 \ 0]$. O vetor solução π era obtido ao fazer

$$\pi_B = B^{-1} p = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0.7 \\ 0.6 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.1 \\ 0.4 \\ 0.2 \end{bmatrix}.$$

Nosso algoritmo, como vimos, não concluiria essa substituição: podemos parar e gerar uma fórmula a partir das colunas que reduzem o custo. Note que com $-c_B B^{-1} = [0 \ -1 \ 1 \ 0]$ a única coluna Γ -consistente com custo reduzido que reduz o custo $c_B \pi$ é $A_j = [1 \ 1 \ 0 \ 1]$. Logo criamos uma fórmula $T = \neg(y_1 \wedge \neg y_2 \wedge y_3)$. Temos que para toda fórmula H' tal que $\Gamma, H' \vdash T$ obtemos $\langle \Gamma, \Psi \rangle \mid \approx \neg H'$. Como exemplo, podemos fazer: $H' = (y_1 \wedge y_2 \wedge \neg y_3) \vee (y_1 \wedge y_2 \wedge y_3)$, onde vemos claramente que $\Gamma, H' \vdash T$, e, portanto, $\langle \Gamma, \Psi \rangle \mid \approx \neg H'$.

Outra forma de execução do algoritmo seria: iniciar com $c'_B = [1 \ 1 \ 1 \ 1]$ e na primeira iteração obter a coluna $B_1 = [1 \ 0 \ 0 \ 0]$ atualizando $c'_B = [0 \ 1 \ 1 \ 1]$; na segunda iteração, obter a coluna $B_2 = [1 \ 1 \ 0 \ 1]$ e atualizar $c'_B = [0 \ 0 \ 1 \ 1]$; na iteração seguinte, obter a coluna $B_4 = [1 \ 1 \ 1 \ 1]$ e atualizar $c'_B = [0 \ 0 \ 1 \ 0]$. Nesse ponto podemos parar o simplex e gerar uma fórmula H bastando notar que $-c'_B B^{-1} = [0 \ -1 \ 0 \ 1]$ possui somente uma única coluna Γ -consistente que reduz o custo que é $A_j = [1 \ 1 \ 1 \ 0]$. Definimos $T = \neg(y_1 \wedge y_2 \wedge \neg y_3)$ e obtemos que para toda fórmula H' tal que $\Gamma, H' \vdash T$, temos $\langle \Gamma, \Psi \rangle \mid \approx \neg H'$. Como exemplo, podemos ter $\neg H' = T = \neg(y_1 \wedge y_2 \wedge \neg y_3)$.

5.4 Abdução de Certezas

A abdução de certezas representa a geração de uma explicação dentro de uma visão clássica de lógica, onde um fato ou evento realmente aconteceu e, portanto, ele explica porque A acontece com probabilidade positiva. Dada uma instância $\langle \Gamma, \Psi \rangle$, e uma fórmula A tal que $\langle \Gamma, \Psi \rangle \not\approx A$, a estratégia é:

1. Adicionar uma nova fórmula H à Γ de forma que tenhamos $\langle \Gamma \cup H, \Psi \rangle \mid \approx A$.

Definindo formalmente, temos:

Definição 5.4.1 (Abdução de Certeza). *Dado $\langle \Gamma, \Psi \rangle \not\approx A$ dizemos que uma fórmula H é abduzida em Γ , se e somente se, para toda distribuição de probabilidade π temos $\langle \Gamma \cup H, \Psi \rangle \mid \approx A$ e, além disso, $\langle \Gamma \cup H, \Psi \rangle$ é probabilisticamente satisfável e $H \not\approx A$.*

A ideia por trás dessa definição, assim como no caso da abdução clássica, é que a fórmula H não aumente sozinha a probabilidade da fórmula A e nem torne $\langle \Gamma \cup H, \Psi \rangle$ insatisfável. Vemos que a fórmula adicionada H é puramente proposicional clássica, ou seja, a probabilidade associada à ela é 1. Adicionando H à base de conhecimento temos que a probabilidade de A passa a ser positiva.

Iremos apresentar nas próximas duas subseções dois métodos de geração da fórmula H como descrito acima. O primeiro método utiliza somente o simplex através de uma matriz psat-normalizada enquanto que o segundo, além do simplex, usa também o método de geração de fórmulas com probabilidades positivas, visto na seção 5.3 deste capítulo. Ambos os métodos são corretos sendo que o último, além de correto, é completo.

5.4.1 Utilizando o Algoritmo Simplex

Dada uma instância PSAT $\langle \Gamma, \Psi \rangle$ e uma fórmula A tal que $\langle \Gamma, \Psi \rangle \not\approx A$, o método é realizado utilizando o simplex de duas fases: na primeira fase buscamos uma distribuição que satisfaça a instância $\langle \Gamma \cup \neg A, \Psi \rangle$; na fase seguinte percorremos as soluções de forma a escolher uma distribuição π tal que $\pi(A) > 0$.

Na primeira fase temos o problema simplex na forma:

$$\begin{aligned} & \min \quad c\pi \\ & \text{sujeito a} \\ & \quad A\pi = p \\ & \quad \pi \geq 0. \end{aligned}$$

Para essa fase, resolvemos a instância $\langle \Gamma \cup \neg A, \Psi \rangle$. A matriz que representa a SBV (Solução Básica Viável) inicial para a primeira fase do simplex é dada então por:

$$\begin{array}{l} 1 \\ \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{array} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \vdots \\ \pi_n \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}.$$

Resolvemos o problema auxiliar, aplicando o algoritmo *PSAT-solver*, para obtermos, ao final, uma matriz B , um vetor c e uma distribuição π tais que $B\pi = p$, com $\pi \geq 0$ e $c\pi = 0$. Aplicamos o processo de psat-normalização (seção 4.7) em B obtendo uma nova matriz psat-normalizada B' , a mesma solução π e um vetor c atualizado tal que c agora indica que colunas de B' são Γ -inconsistentes e linearmente independentes das colunas fora da base (possuem valor 1 em c) e Γ -consistentes (possuem valor 0 em c). Encerramos assim a primeira fase do simplex obtendo uma solução para a instância $\langle \Gamma \cup \neg A, \Psi \rangle$.

Na segunda fase do simplex percorremos o conjunto de soluções de $\langle \Gamma, \Psi \rangle$ de forma a gerar a hipótese H para alguma solução com uma distribuição probabilística π que possua $\pi(A) > 0$. Para esse fim, mudamos a função objetivo do sistema anterior e criamos um vetor w . Temos que w fará parte da nova função objetivo e indicará a inserção de colunas que são $\Gamma \cup A$ -consistentes do seguinte modo: se uma coluna B'_i em B' é $\Gamma \cup A$ -consistente então $w_i = 0$ e, caso contrário, se B'_i for $\Gamma \cup A$ -inconsistente então $w_i = 1$.

Temos então a nova versão do simplex:

$$\begin{aligned} & \min \quad w\pi \\ & \text{sujeito a} \\ & \quad B'\pi = p. \end{aligned}$$

Note que no fim da primeira fase, todas as colunas que possuem probabilidade positivas são $\Gamma \cup \neg A$ -consistentes, temos que o valor de $w\pi$ inicialmente é 1. A ideia é buscarmos colunas que reduzam o custo de $w\pi$ e, conseqüentemente, aumente a probabilidade da fórmula A . Buscamos colunas $\Gamma \cup A$ -consistentes C tal que $-wB^{-1}C < 0$.

O método pode parar a qualquer momento que tivermos $w\pi < 1$ ou irá parar, no máximo, quando não podemos mais encontrar colunas que reduzam $w\pi$. Pelo fato de B ser

uma matriz psat-normalizada, e de que sempre inserimos colunas Γ -consistentes, temos que a cada iteração é uma solução da instância $\langle \Gamma, \Psi \rangle$.

Ao parar o simplex, geramos uma fórmula abduzida H da seguinte forma:

1. Buscamos as colunas de B que são Γ -consistentes;
2. Para cada coluna obtida no passo 1, se a coluna for B_i e tivermos $w_i = 0$ (sendo $\Gamma \cup \{A\}$ -consistente) então elaboramos uma fórmula a partir de uma valoração que a satisfaz (e que satisfaz A também) fazendo uma conjunção das proposições atômicas da valoração. Se for o caso, podemos elaborar mais de uma fórmula que seja $\Gamma \cup \{A\}$ -consistente para cada coluna.
3. Para cada uma das colunas restantes dos passos 1 e 2, onde temos B_i e $w_i = 1$, buscamos colunas que são $\Gamma \cup \neg A$ -consistentes, elaboramos uma fórmula sendo a conjunção das proposições atômicas de uma valoração que a satisfaz. Se for o caso, podemos elaborar mais de uma fórmula que seja $\Gamma \cup \neg A$ -consistente para cada coluna.
4. Criamos H' fazendo a disjunção de todas as fórmulas obtidas.

O algoritmo 8 descreve de forma explícita o método. Temos que $\langle \Gamma \cup H, \Psi \rangle \approx A$ e, além disso, com o $w^T \pi$ final temos que para toda distribuição probabilística π que satisfaz $\langle \Gamma \cup H, \Psi \rangle$ ocorre $\pi(A) \geq 1 - w^T \pi$.

Exemplo 5.4.2. Em uma cidade ao norte da amazônia estatísticas revelam que chove em 60% dos dias do mês e que em 20% dos dias houve chuvas com queda de árvores. Um levantamento, feito pelo estado, na região onde a cidade se encontra revelou que em todas as cidades todos os dias que tiveram queda de árvores ou incidência de raios também tiveram falta de energia e toda falta de energia foi ocasionada por esses dois fatores. A prefeitura não teve acesso ao levantamento de dados sobre os efeitos de raios na cidade, mas um levantamento posterior concluiu que houve dias em que faltou energia e não houve quedas de árvores. Buscam-se hipóteses prováveis para esse fato.

Podemos formular o problema da seguinte forma:

- A letra c representará o percentual de chuvas em um mês;
- A letra q representará a incidência de quedas de árvores;
- A letra e representará o percentual de energia elétrica em um mês;
- A letra r o de incidência de raios.

Temos, portanto, $P(c) = 60\%$, $P(q) = 20\%$ e, conforme o enunciado, $P(\neg e \wedge \neg q) > 0$. Além disso nos é informado que $P((q \vee r) \leftrightarrow \neg e) = 1$.

Algoritmo 8 AbduçãoSimplex

Entrada: Uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ satisfatível e uma fórmula A tal que $\langle \Gamma, \Psi \rangle \not\approx A$.

Saída: Uma fórmula H tal que $\langle \Gamma \cup H, \Psi \rangle \approx A$

```

1:  $(B, c, \pi) = \text{PSAT-solver}(\Delta \cup \neg A, \Psi)$ 
2:  $w \leftarrow [1 \dots 1]_{k+1,1}$ 
3:  $B' \leftarrow B$ 
4:  $\pi_B \leftarrow \pi$ 
5:  $\text{IniciarAbducao} \leftarrow F$ 
6: while  $w\pi_B > 0$  e  $\text{IniciarAbducao} = F$  do
7:    $C \leftarrow \text{ProblemaAuxiliar}(\Gamma \cup A, w, B'^{-1})$ 
8:    $\text{Pivoteamento}(B', C, \pi_B)$ 
9:    $\pi_B \leftarrow B'^{-1}p$ 
10:  if  $w\pi_B = 0$  then
11:     $w \leftarrow [1 \dots 1]_{k+1,1}$ 
12:  end if
13:   $\text{IniciarAbducao} \leftarrow \text{EscolhaValorPositivo}(w, \pi_B)$ 
14: end while
15:  $H := \emptyset$ 
16: for cada  $B'_i$   $\Gamma$ -consistente em  $B'$  do
17:  if se  $w_i = 0$  then
18:    Crie  $C_i^v$  como sendo a conjunção das proposições atômicas de uma valoração  $v$ 
    que satisfaz  $\Gamma$ ,  $B'_i$  e  $A$ 
19:     $H := H \vee C_i^v$ 
20:  else
21:    Crie  $C_i^v$  como sendo a conjunção das proposições atômicas de uma valoração  $v$ 
    que satisfaz  $\Gamma$ ,  $B'_i$  e  $\neg A$ 
22:     $H := H \vee C_i^v$ 
23:  end if
24: end for
25: return  $H$ 

```

Utilizando o método simplex, obtemos inicialmente a matriz:

$$\begin{array}{c} 1 \\ \psi_1 \\ \psi_2 \end{array} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \pi'_1 \\ \pi'_2 \\ \pi'_3 \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ p_2 \end{bmatrix}$$

e o vetor de custos c como $c = [1 \ 1 \ 1]$. Ao buscar colunas consistentes com Γ , suponha que na primeira iteração o algoritmo nos retornou a coluna $Aj' = [1 \ 1 \ 1]$, e atualizamos a coluna $c = [1 \ 1 \ 0]$; na segunda iteração obtemos $Aj' = [1 \ 1 \ 0]$ com a atualização do vetor $c = [1 \ 0 \ 0]$; na terceira iteração obtemos $Aj' = [1 \ 0 \ 0]$ com $c = [0 \ 0 \ 0]$ e, por fim, o algoritmo encerra por chegar ao valor mínimo. Temos que $\pi' = [0.4 \ 0.4 \ 0.2]$.

Agora iremos à segunda parte do simplex, começamos verificando que colunas são consistentes com $(\neg e \wedge \neg q)$ de forma que se a coluna for consistente, então sua posição em w será 0, logo temos: $w = [0 \ 0 \ 1]$. Podemos encerrar o algoritmo nesse momento e buscarmos valorações que satisfaçam o problema. Temos duas colunas que satisfazem Γ . Geramos uma fórmula H como mencionado antes, ou seja, buscamos valorações para cada coluna que for Γ -consistente. Para a primeira coluna com $\psi_1 = 0$ e $\psi_2 = 0$ uma valoração que satisfaz é representada pela fórmula $(\neg c \wedge \neg e \wedge \neg q \wedge r \wedge \neg \psi_1 \wedge \neg \psi_2)$ e a segunda coluna com $\psi_1 = 1$ e $\psi_2 = 0$ temos uma valoração que a satisfaz representada por $(c \wedge \neg e \wedge \neg q \wedge r \wedge \psi_1 \wedge \neg \psi_2)$. Agora que geramos fórmulas com todas as colunas que são Γ -consistentes com $w_i = 0$, buscamos as colunas que são Γ -consistentes com $w_i = 1$, então para a terceira coluna com $\psi_1 = 1$ e $\psi_2 = 1$ temos uma valoração representada por $(c \wedge \neg e \wedge q \wedge r \wedge \psi_1 \wedge \psi_2)$. Fazemos então

$$H = (\neg c \wedge \neg e \wedge \neg q \wedge r \wedge \neg \psi_1 \wedge \neg \psi_2) \vee (c \wedge \neg e \wedge \neg q \wedge r \wedge \psi_1 \wedge \neg \psi_2) \vee (c \wedge \neg e \wedge q \wedge r \wedge \psi_1 \wedge \psi_2).$$

A primeira vista parece uma fórmula gigantesca, mas ela possui equivalente em tamanho menor:

$$H = (\neg c \wedge \neg q \wedge r) \vee (c \wedge \neg q \wedge r) \vee (c \wedge q \wedge r),$$

que é uma das fórmulas que podem ser abduzidas pelo método. Temos que, fazendo $\langle \Gamma \cup H, \Psi \rangle \approx A$, temos $P(A) = 0.8$. É fácil ver que $H \not\approx A$ pois existe uma distribuição probabilística que associa valor 1 à coluna que é Γ -consistente e que não é $A \cup \Gamma$ -consistente.

Temos que o método acima só permite distribuições π que satisfazem A com um valor $\pi(A) = p$, ou seja, qualquer outra possível distribuição na qual A ocorra com uma probabilidade diferente dessa não será permitida, o que de certa forma impõe um limite

ao método. O próximo método nos fornece um conjunto maior de possibilidades visto que nos permite gerar uma hipótese H na qual várias distribuições a satisfazem com valores diferentes e positivos.

5.4.2 Utilizando Geração de Fórmulas com Probabilidade Positiva

Apresentaremos agora um método que mescla parte do método da seção 5.3 com parte do método da seção 4.7. Utilizaremos uma versão relaxada do simplex de duas fases. Dada uma instância $\langle \Gamma, \Psi \rangle$ satisfatível, na primeira fase buscamos uma distribuição que a satisfaça. Novamente temos uma instância PSAT $\langle \Gamma, \Psi \rangle$ tal que $\langle \Gamma, \Psi \rangle \not\approx A$. A ideia é, ao invés de gerar um H , utilizando a segunda fase clássica do simplex, tal que $\langle \Gamma \cup H, \Psi \rangle \approx A$, iremos usar uma versão relaxada da segunda fase, onde não estamos necessariamente interessados em maximizar ou minimizar a função objetivo, mas sim buscar distribuições com valores positivos para A . Após essa segunda fase utilizamos também o método de geração de fórmulas com probabilidade positiva.

Na primeira fase utilizamos o simplex na forma:

$$\begin{aligned} \min \quad & c\pi \\ \text{sujeito a} \quad & \\ & A\pi = p \\ & \pi \geq 0. \end{aligned}$$

Resolvemos a instância $\langle \Gamma, \Psi \rangle$, obtendo uma matriz solução B e um vetor c e a solução π , conforme já visto e, em seguida, psat-normalizamos a matriz B , obtendo B' , atualizamos o vetor c e mantemos a mesma solução π anterior.

Dado que obtivemos a matriz B' e π na fase anterior, trocamos a função objetivo na segunda fase relaxada do simplex. Criamos um vetor w que corresponde às colunas de B' que são $\Gamma \cup A$ -consistentes tal que se uma coluna B'_i em B' for $\Gamma \cup \neg A$ -consistente então o respectivo valor w_i dela em w será 0, caso contrário será 1. Criamos também vetor w^i que é o complementar do vetor w e é sempre atualizado quando w também for, ou seja, se em determinada posição k temos $w_k = 1$ então $w_k^i = 0$ e se $w_k = 0$ então $w_k^i = 1$. A ideia é utilizar w na função objetivo quando quisermos inserir colunas $\Gamma \cup \neg A$ -consistentes e utilizar w^i quando quisermos inserir colunas $\Gamma \cup A$ -consistentes. Só poderemos gerar uma fórmula H se a nova função objetivo que utiliza w for maior que zero, ou seja, se $w\pi > 0$.

Temos então a nova versão do simplex:

$$\begin{aligned} \text{custo: } & v\pi \\ \text{sujeito a} \quad & \\ & B\pi = p, \end{aligned}$$

onde v acima pode ser w ou w^i . A cada iteração atualizamos B , w , w^i e π . O método pode encerrar a segunda fase a qualquer momento em que $w\pi > 0$ (ou seja, $P(A) > 0$ na distribuição de probabilidades). Como inicialmente temos B' como uma matriz psat-normalizada, qualquer inserção de colunas Γ -consistentes utilizando o pivoteamento faz com que cada iteração seja solução da instância $\langle \Gamma, \Psi \rangle$.

Ao buscarmos alguma coluna $\Gamma \cup \neg A$ -consistente para inserir em B' em uma posição k , temos que seu respectivo valor w_k será igual a 0 ($w_k = 0$) e o valor w^i como igual a 1 ($w_k^i = 1$). Da mesma forma, se quisermos inserir alguma coluna $\Gamma \cup A$ -consistente fora da base, temos que seu respectivo valor w_k^i será igual a 0 ($w_k^i = 0$) e o valor w como igual a 1 ($w_k = 1$). Em determinado ponto onde $w\pi > 0$ podemos parar o simplex e gerar uma fórmula abduzida da seguinte forma:

- 1 Buscamos as colunas de B que são Γ -consistentes;
- 2 Para cada coluna obtida no passo 1, se a coluna for B_i e tivermos $w_i = 0$ (sendo $\Gamma \cup \{A\}$ -consistente) então elaboramos uma fórmula a partir de uma valoração que a satisfaz (e que satisfaz A também) fazendo uma conjunção de suas proposições atômicas. Se for o caso, podemos elaborar mais de uma fórmula que seja $\Gamma \cup \{A\}$ -consistente para cada coluna.
- 3 Para cada uma das colunas restantes dos passos 1 e 2, onde temos B_i e $w_i = 1$, buscamos colunas que são $\Gamma \cup \neg A$ -consistentes, elaboramos uma fórmula sendo a conjunção das proposições atômicas de uma valoração que as satisfazem. Se for o caso, podemos elaborar mais de uma fórmula que seja $\Gamma \cup \neg A$ -consistente para cada coluna.
- 4 Criamos H' fazendo a disjunção de todas as fórmulas obtidas.

A geração de H' aqui é similar à forma apresentada na seção 5.3 e temos necessariamente que $\langle \Gamma \cup H', \Psi \rangle \models A$. Porém criamos, além de H' , a fórmula T' como descrita abaixo:

- 5 Definimos a fórmula $T = \neg C_1 \wedge \neg C_2 \wedge \dots \wedge \neg C_n$, onde cada C_i é a conjunção das proposições atômicas de uma coluna com custo reduzido utilizando o vetor w ($-wB^{-1}C_i < 0$);
- 6 A partir de T , definimos uma nova fórmula $T' = (T \vee A)$, que é uma fórmula que só é falsa em valorações que satisfaçam algum C_i e $\neg A$ ao mesmo tempo. A ideia é reproduzir o momento em que não temos mais colunas $\Gamma \cup \neg A$ -consistentes que reduzam o custo $w\pi$;

é fácil notar, no item 5, que $\langle \Gamma \cup T', \Psi \rangle$ é probabilisticamente satisfável, basta ver que a matriz B obtida no ponto em que paramos o simplex serve como solução de $\langle \Gamma \cup T', \Psi \rangle$.

Outro fato é que temos $\langle \Gamma \cup T', \Psi \rangle \approx A$ pois o PSAT-*solver* irá falhar ao tentar achar uma solução para a instância $\langle \Gamma \cup T' \cup \neg A, \Psi \rangle$ devido à presença de T' , já que esse impede que em algum ponto, com $w'\pi > 0$, que encontremos colunas $\Gamma \cup \neg A$ -consistentes que reduzam o custo.

Como H' é construído a partir de uma matriz B obtida da instância acima, temos que $\langle \Gamma' \cup T' \cup H', \Psi' \rangle$ também é probabilisticamente satisfável. Note que T' restringe valorações que satisfaçam algum C_i e $\neg A$ ao mesmo tempo, logo, pela construção de H' é fácil ver que $\Gamma \cup H' \vdash T'$, senão teríamos uma valoração que não satisfaz T' (satisfaz algum C_i e $\neg A$) mas satisfaz $\Gamma \cup H'$.

O método leva à uma sétima etapa que nos fornece bem mais possibilidades de fórmulas que o método anterior:

7 Podemos adicionar mais fórmulas à disjunção de valorações H' da seguinte forma:

Se uma fórmula C é tal que $\Gamma, C \vdash T'$, com $\Gamma \cup C$ satisfável, então podemos adicioná-la à fórmula H' com $H' := (H' \vee C)$.

Não é difícil ver que, dado $\langle \Gamma' \cup H', \Psi' \rangle \approx A$ pelo método acima, então $\langle \Gamma' \cup (H' \vee C), \Psi' \rangle \approx A$, já que $\Gamma, (H' \vee C) \vdash T'$.

8 Faça $H := H'$.

Como já mencionamos, a fórmula $T = \neg C_1 \wedge \neg C_2 \wedge \dots \wedge \neg C_n$ não precisa ser diretamente escrita podemos criar uma fórmula lógica para $-cB^1C < 0$ (em (Bon11) esse procedimento é descrito com detalhes). Em algoritmo 9 vemos uma forma explícita do método.

O método como descrito acima, além de correto, é completo:

Teorema 5.4.3 (Completeness). *Dado $\langle \Gamma, \Psi \rangle \not\approx A$ e uma fórmula H tal que temos $\langle \Gamma \cup H, \Psi \rangle \approx A$, o método acima é capaz de gerar a fórmula H .*

Demonstração. A prova é baseada no seguinte: dado que temos uma solução usando o simplex para uma instância $\langle \Gamma \cup H, \Psi \rangle$, sendo ela composta por uma matriz psat-normalizada B^H para $\langle \Gamma \cup H, \Psi \rangle$, um vetor c^H e um vetor π^H tal que $c^H \pi^H = 0$, temos que se mudarmos a função objetivo usando um vetor w^H tal que $w_i^H = 0$ se B_i^H for consistente com $\Gamma \cup H \cup \neg A$ e, caso contrário, $w_i^H = 1$, vemos que $w^H \pi^H$ nós dará $P(A) > 0$ para a distribuição dada. Se procurarmos reduzir o valor de w^H buscando colunas $\Gamma \cup H \cup \neg A$ -consistentes, temos que em algum momento não poderemos mais achar essas colunas pois $\langle \Gamma \cup H \cup \neg A, \Psi \rangle$ é probabilisticamente insatisfável, ou seja, não podemos reduzir $w^H \pi^H$ à zero. A ideia da completude é simular a computação acima para uma solução da instância $\langle \Gamma, \Psi \rangle$ como se tivéssemos $\Gamma \cup H$ o tempo todo. No ponto em que não pudermos mais buscar colunas $\Gamma \cup H \cup \neg A$ consistentes que reduzam o custo $w\pi$, teremos que somente colunas $\Gamma \cup \neg A$ -consistentes podem reduzi-lo e, então, geramos a fórmula H' com essas colunas utilizando a segunda parte do método.

Algoritmo 9 AbduçãoSimplexcomGeraçãodeFormula

Entrada: Uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ satisfatível e uma fórmula A tal que $\langle \Gamma, \Psi \rangle \not\approx A$.

Saída: Uma fórmula H tal que $\langle \Gamma \cup H, \Psi \rangle \approx A$

```

1:  $(B, c, \pi) = \text{PSAT-solver}(\Delta \cup \neg A, \Psi)$ 
2:  $w \leftarrow [1 \dots 1]_{k+1,1}$ 
3:  $B' \leftarrow B$ 
4:  $\pi_B \leftarrow \pi$ 
5:  $\text{IniciarAbducao} \leftarrow F$ 
6: while  $w\pi_B > 0$  e  $\text{IniciarAbducao} = F$  do
7:    $C \leftarrow \text{ProblemaAuxiliar}(\Gamma \cup A, w, B'^{-1})$ 
8:    $\text{Pivoteamento}(B', C, \pi_B)$ 
9:    $\pi_B \leftarrow B'^{-1}p$ 
10:  if  $w\pi_B = 0$  then
11:     $w \leftarrow [1 \dots 1]_{k+1,1}$ 
12:  end if
13:   $\text{IniciarAbducao} \leftarrow \text{EscolhaValorPositivo}(w, \pi_B)$ 
14: end while
15: Defina  $T := C_1 \wedge \dots \wedge C_n$ , onde cada  $C_i$  é a conjunção das proposições atômicas de
    uma coluna que reduz o custo  $w\pi$ 
16: Defina  $T' := (T \vee A)$ 
17:  $H' := \emptyset$ 
18: for cada  $B'_i$   $\Gamma$ -consistente em  $B'$  do
19:   if se  $w_i = 0$  then
20:     Crie  $C_i^v$  como sendo a conjunção das proposições atômicas de uma valoração  $v$ 
      que satisfaz  $\Gamma$ ,  $B'_i$  e  $A$ 
21:      $H' := H' \vee C_i^v$ 
22:   else
23:     Crie  $C_i^v$  como sendo a conjunção das proposições atômicas de uma valoração  $v$ 
      que satisfaz  $\Gamma$ ,  $B'_i$  e  $\neg A$ 
24:      $H' := H' \vee C_i^v$ 
25:   end if
26: end for
27:  $H \in \{H', H' \vee \text{Abd}(\Gamma, T')\}$ 
28: return  $H$ 

```

Através do teorema 4.7.7, temos que a partir de qualquer solução $\langle \Gamma, \Psi \rangle$, com B, c e π , podemos obter uma solução para $\langle \Gamma \cup H, \Psi \rangle$, onde a cada iteração as colunas de B que são Γ -consistentes formam uma solução para $\langle \Gamma, \Psi \rangle$ e, no final, B é psat-normalizada e solução para $\langle \Gamma \cup H, \Psi \rangle$. Temos então que terminada a primeira fase do método, onde obtemos uma solução B, c, π para a instância $\langle \Gamma, \Psi \rangle$, mudamos a função objetivo para w , e então, através do teorema 4.7.7, podemos obter através de várias iterações uma matriz B' psat-normalizada para $\langle \Gamma \cup H, \Psi \rangle$, e um vetor π' tal que as colunas $\Gamma \cup H$ -consistentes de B' formam uma solução para $\langle \Gamma \cup H, \Psi \rangle$ e em toda iteração as colunas Γ -consistentes formam uma solução para $\langle \Gamma, \Psi \rangle$. A cada iteração atualizamos o vetor w onde, como vimos, $w_i = 0$ se a coluna inserida for $\Gamma \cup H \cup \neg A$ -consistente, e $w_i = 1$, caso contrário. Obtemos então B', w, π' onde B' é uma matriz psat-normalizada para $\langle \Gamma \cup H, \Psi \rangle$, e $w\pi'$ representa a probabilidade de A (onde, necessariamente, $P(A) > 0$).

Como vimos, iremos reduzir o custo $w\pi'$ até não haver colunas $\Gamma \cup H \cup \neg A$ -consistentes que o reduzam e, quando não houver mais, buscamos as colunas $\Gamma \cup \neg A$ -consistentes que o reduzem. Supondo que as fórmulas que representam essas colunas $\Gamma \cup \neg A$ -consistentes são C_1, \dots, C_n , criamos a fórmula $T = \neg C_1 \wedge \dots \wedge \neg C_n$ e também definimos $T' = (T \vee A)$. Temos que $\Gamma, H \vdash T'$ pois não há uma valoração v que satisfaça Γ e H e não satisfaça T' , basta ver que a única forma de falsificar T' é que uma coluna c_i seja $\Gamma \cup H \cup \neg A$ satisfatível.

Geramos uma fórmula H' a partir da matriz B utilizando os passos 1-3. Dada uma fórmula C tal que $H' \vee C \leftrightarrow H$, podemos gerar C por abdução clássica, já que $\Gamma, C \vdash T'$. Fazendo $H' := H' \vee C$ e como todos os passos acima podem ser uma das formas de execução do método para uma instância $\langle \Gamma, \Psi \rangle$ onde $\langle \Gamma, \Psi \rangle \not\approx A$, obtemos o resultado. \square

Exemplo 5.4.4. Suponha que resolvemos a primeira fase do exemplo 5.4.2, onde obtemos B, c e π para o problema anterior. Agora criamos o vetor w no qual cada posição w_i possui valor 0 se B_i é $\Gamma \cup \neg A$ -consistente e 1 caso contrário. Obtemos o vetor $w = [0 \ 0 \ 1]$, com $P(A) = 0.8$ que é o valor máximo que conseguimos obter. Criamos um vetor w^i que é formado a partir de w trocando os valores de 0's por 1's e de 1's por 0's. Obtemos $w^i = [1 \ 1 \ 0]$ e reduzimos o custo procurando uma coluna A_j tal que $-w^i B^{-1} A_j < 0$. Encontramos $A_j = [1 \ 0 \ 0]$ e atualizamos $w^i = [0 \ 1 \ 0]$, que faz com que tenhamos $P(A) = 0.4$. Paramos o simplex aqui e calculamos $-w B^{-1} = [0 \ -1 \ 1]$. Vemos que a única coluna $\Gamma \cup \neg A$ -consistente que reduz o custo é $B_j = [1 \ 1 \ 0]$, tal que a coluna B_j é representada por $C = (\psi_1 \wedge \neg \psi_2)$ e então criamos $T = (\neg C \vee A)$. Buscamos as colunas Γ -consistentes em B . Pelo algoritmo temos que gerar as fórmulas com as colunas que são Γ -consistentes, temos a fórmula $(c \wedge \neg e \wedge q \wedge \neg r \wedge \psi_1 \wedge \psi_2)$ para a primeira coluna de B , $(\neg c \wedge \neg e \wedge \neg q \wedge r \wedge \neg \psi_1 \wedge \neg \psi_2)$ para a segunda coluna e $(\neg c \wedge e \wedge \neg q \wedge \neg r \wedge \psi_1 \wedge \neg \psi_2)$ para a terceira. Fazendo equivalências lógicas, obtemos a fórmula abreviada:

$$H = (\neg c \wedge \neg q \wedge r) \vee (c \wedge \neg q \wedge r) \vee (c \wedge q \wedge \neg r),$$

que é uma fórmula que força a probabilidade de A ser positiva.

Podemos adicionar mais fórmulas a H como vimos. É fácil ver que $\Gamma, r \vdash T$ já que a única valoração v que satisfaz $\Gamma \cup r$ e não satisfaz T é $v(c) = 1, v(e) = 1, v(q) = 0, v(r) = 0$, logo, temos que $(H \vee r)$ é uma fórmula que satisfaz o problema. Note que $\Gamma \vdash (H \vee r) \leftrightarrow r$ e, portanto, temos $\langle \Gamma \cup r, \Psi \rangle \approx A$.

Na próxima seção apresentaremos o caso em que a fórmula H produzida é imprecisa e possui uma probabilidade associada entre 0 e 1.

5.5 Abdução de informações probabilísticas

A abdução de informação probabilística é a geração de uma hipótese que, apesar de ser imprecisa, faz com que A possua uma probabilidade positiva, ou seja, a probabilidade do fato ou um evento H , juntamente com Δ , possui uma correlação com A .

A abdução de informações probabilísticas é formulada como a seguir: dada uma instância $\langle \Gamma, \Psi \rangle$, e uma fórmula A tal que $\langle \Gamma \cup H, \Psi \rangle \not\approx A$, a estratégia é:

1. Adicionar uma nova fórmula $(H \leftrightarrow a)$, onde a é uma proposição atômica nova, à Γ e a à Ψ tal que $P(a) = p$, onde $0 < p < 1$.

Definindo formalmente, temos:

Definição 5.5.1 (Abdução de informações probabilísticas). *Dado $\langle \Gamma, \Psi \rangle \not\approx A$ dizemos que uma fórmula $(H \leftrightarrow a)$, onde a é novo, é abduzida em Ψ , se e somente se, para toda distribuição de probabilidade π temos $\langle \Gamma \cup (H \leftrightarrow a), \Psi \cup \{a\} \rangle \approx A$ e, além disso, $\langle \Gamma \cup (H \leftrightarrow a), \Psi \cup \{a\} \rangle$ é probabilisticamente satisfável e $\langle (H \leftrightarrow a), a \rangle \not\approx A$.*

Diferentemente do método da seção anterior, temos que H representa uma informação imprecisa. O interesse na abdução de informações probabilísticas é descobrir que fatores ocorrem, mesmo que de forma imprecisa ou incerta, acarretam, juntamente com a base de conhecimento, que uma probabilidade positiva de A aconteça.

Apresentaremos na próxima subseção um método que utiliza os métodos clássicos de abdução para gerar informações probabilísticas. Lembramos que \vdash significa a inferência clássica.

5.5.1 Utilizando abdução clássica para gerar informações probabilísticas

Dada uma instância PSAT $\langle \Gamma, \Psi \rangle$ e uma fórmula A tal que $\langle \Gamma, \Psi \rangle \not\approx A$ não é válido, podemos gerar, utilizando métodos clássicos de abdução até então vistos (capítulo 3), uma fórmula H tal que $\Gamma, H \vdash A$.

Na maioria dos casos, temos que ao obter um H dessa forma teremos $\langle \Gamma \cup H, \Psi \rangle$ probabilisticamente insatisfável. Logo uma forma de resolver é associar H a uma probabilidade positiva de forma que $P(A) > 0$.

Teorema 5.5.2. *Dado $\langle \Gamma, \Psi \rangle \not\approx A$ e uma fórmula H obtida por abdução clássica tal que $\Gamma, H \vdash A$, temos que se $\langle \Gamma \cup \{H \leftrightarrow a\}, \Psi \cup \{P(a) = b\} \rangle$ com $b > 0$ for probabilisticamente satisfável então $P(A) > 0$, ou seja, para toda distribuição de probabilidade π que satisfaz $\langle \Gamma \cup \{H \leftrightarrow a\}, \Psi \cup \{P(a) = b\} \rangle$ temos $\pi(A) > 0$.*

Demonstração. suponha que exista uma distribuição π que satisfaça $\langle \Gamma \cup \{H \leftrightarrow a\}, \Psi \cup \{P(a) = b\} \rangle$ com $b > 0$. Como a possui probabilidade positiva, significa que alguma valoração v , na qual Γ e a são verdadeiros, possui probabilidade positiva. Como $H \leftrightarrow a$, temos que H também é verdadeiro nessa valoração. No enunciado temos $\Gamma, H \vdash A$, logo a valoração v satisfaz A e portanto, A possui probabilidade positiva, logo $\pi(A) > 0$. \square

Temos que mostrar também que a fórmula probabilisticamente abduzida obtida é tal que $\langle \{H \leftrightarrow a\}, \{P(a) = b\} \rangle \not\approx A$, em outras palavras, ela sozinha não garante a probabilidade positiva de A .

Teorema 5.5.3. *Dado $\langle \Gamma, \Psi \rangle \not\approx A$ e uma fórmula H obtida por abdução clássica tal que $\Gamma, H \vdash A$, considere $\langle \Gamma \cup \{H \leftrightarrow a\}, \Psi \cup \{P(a) = b\} \rangle \approx A$ com $b > 0$, temos então que $\langle \{H \leftrightarrow a\}, \{P(a) = b\} \rangle \not\approx A$.*

Demonstração. Basta notar que, pela abdução clássica, que temos $H \not\vdash A$ e também $H \not\vdash \neg A$. Criamos uma matriz que representa uma solução via programação linear:

$$\begin{array}{l} H \leftrightarrow a \\ a \end{array} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 - b \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ b \end{bmatrix}.$$

Como $H \not\vdash \neg A$, podemos obter uma valoração consistente com $H \leftrightarrow a$ que não satisfaz $\neg A$ e atribuir probabilidade $1 - b$. Da mesma forma, como temos $H \not\vdash A$, podemos obter uma valoração consistente com $H \leftrightarrow a$ que não satisfaz A e atribuir probabilidade b . Dessa forma temos uma solução do sistema tal que $P(A) = 0$, logo $\langle \{H \leftrightarrow a\}, \{P(a) = b\} \rangle \not\approx A$. \square

A ideia do método é, dado um problema de abdução probabilística com $\langle \Gamma, \Psi \rangle \not\approx A$, geramos um H pela abdução clássica tal que $\Gamma, H \vdash A$. Como em muitos casos não podemos adicionar H à Γ devido a problemas de inconsistência, associamos H a uma probabilidade entre 0 e 1 usando o símbolo proposicional a , ou seja, adicionamos $(H \leftrightarrow a)$ à Γ e a à Ψ com uma probabilidade associada entre 0 e 1. Para determinar a probabilidade de a de forma que $\langle \Gamma \cup (H \leftrightarrow a), P(a) = b \cup \Psi \rangle$ seja probabilisticamente satisfável, utilizamos o simplex de duas fases, onde na primeira fase obtemos uma solução de $\langle \Gamma, \Psi \rangle$ e

Algoritmo 10 AbduçãoInformaçãoProbabilística

Entrada: Uma instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$ satisfatível e uma fórmula A tal que $\langle \Gamma, \Psi \rangle \not\approx A$.

Saída: Uma fórmula proposicional $(H \leftrightarrow a)$ com $P(a) = b > 0$ tal que $\langle \Gamma \cup (H \leftrightarrow a), P(a) = b \cup \Psi \rangle \approx A$

- 1: $H := Abd(\Gamma, A)$
- 2: $(B, c, \pi) = \text{PSAT-solver}(\Gamma, \Psi)$
- 3: Gere B' psat-normalizado para Δ a partir de B
- 4: Defina w tal que $w_i = 1$ se B'_i é $\Gamma \cup H$ -consistente, $w_i = 0$ se caso contrário
- 5: $\pi' \leftarrow \pi$
- 6: $\text{pararMetodo} \leftarrow F$
- 7: **while** $w\pi' > 0$ e $\text{pararMetodo} = F$ **do**
- 8: $C \leftarrow \text{ProblemaAuxiliar}(\Gamma \cup H, w, B'^{-1})$
- 9: $\text{Pivoteamento}(B', C, \pi')$
- 10: $\pi' \leftarrow B'^{-1}p$
- 11: $\text{pararMetodo} \leftarrow \text{EscolhaValorPositivo}(w, \pi')$
- 12: **end while**
- 13: **return** $(H \leftrightarrow a), P(a) = w\pi'$

na segunda fase procuramos uma distribuição de probabilidades que nos forneça um valor $p = \pi(H) > 0$. Fazendo $P(a) = p$ obtemos uma instância $\langle \Gamma \cup (H \leftrightarrow a), P(a) = b \cup \Psi \rangle \approx A$. O algoritmo 10 é uma versão explícita do método.

O método a rigor é incompleto: todas as colunas $\Gamma \cup H$ -consistentes da base de uma solução que possuem probabilidades positivas são $\Gamma \cup A$ -consistentes, já que $\Gamma, H \vdash A$, ou seja, as valorações que satisfazem a fórmula $(H \leftrightarrow a)$, com $P(a) = b > 0$, na instância Δ , satisfazem A com probabilidade positiva. Fórmulas que podem ser imprecisas e que possuem valorações com probabilidades positivas que não satisfazem A não são levadas em conta pela abordagem acima. Temos, porém, um ganho de propriedades da lógica proposicional, como o princípio da subfórmula, onde a fórmula H gerada pode conter subfórmulas presentes tanto em A como em Γ , e, além disso, os métodos apresentados de abdução clássica além de completos apresentam um ganho computacional, como o caso do KE-tableaux, que podem ser utilizados na abordagem proposta.

Capítulo 6

Conclusões

6.1 Contribuições e Considerações Finais

Dentre as contribuições deste trabalho, apresentamos uma prova de completude para a abdução baseada em corte (DM94), que utiliza o KE-tableaux, tanto para o caso proposicional como para o caso de primeira-ordem. Esse resultado foi publicado no artigo (AF13).

Um procedimento de normalização (psat-normalização) foi apresentado no capítulo 4, esse procedimento nos possibilita a mudança de função objetivo no algoritmo simplex de forma que, a cada iteração na segunda fase do simplex, tenhamos ainda uma solução para a instância PSAT em questão. Por possibilitar a segunda fase do simplex para buscar distribuições que diminuam (ou aumentam) a probabilidade de uma fórmula proposicional A em relação a uma dada instância Δ , esse método é utilizado no capítulo de abdução probabilística. Além das aplicações em abdução probabilística, a utilização da segunda fase do simplex para diminuir (ou aumentar) a probabilidade de uma fórmula A pode apresentar diversas outras aplicações, visto que A pode representar uma característica ou informação desejável (ou indesejável) presente em um programa descrito por uma instância PSAT.

Definimos a noção de abdução probabilística para PSAT, vista no capítulo 5, que segue a ideia tradicional de geração de hipóteses da abdução clássica, só que dentro de um contexto probabilístico. Apresentamos um método que é capaz de gerar todas as fórmulas que possuem probabilidade positiva dada a ocorrência de uma instância PSAT e além disso, fornecemos métodos que resolvem o problema, sendo um deles completo.

6.2 Pesquisas Futuras

Como próximo passo, pretendemos implementar todos os métodos propostos na seção 4.7 e no capítulo 5. Também há margem para a criação de um método que seja completo

para a abdução de informações probabilísticas proposta na seção 5.5 do capítulo 5. Uma base para a criação do método pode ser: dado um problema de abdução $\langle \Gamma, \Psi \rangle \not\approx A$, ao gerarmos uma fórmula tal que $\langle \Gamma \cup H, \Psi \rangle \approx A$, resolvemos essa instância com uma matriz B , um vetor c e π , buscamos as colunas na matriz B que satisfazem $\Gamma \cup A$, geramos uma fórmula H_A , correspondente às valorações que as satisfazem, e associamos ela à probabilidade p_A obtida para A fazendo $H_A \leftrightarrow c$, com c novo, e $P(c) = p_A$. Teremos então $\langle \Gamma \cup H \cup (H_A \leftrightarrow c), \Psi \cup \{P(c) = p_A\} \rangle \approx A$. A ideia é criar um método que gere, para todos os casos, utilizando abdução clássica, uma fórmula proposicional C tal que $\langle \Gamma \cup ((H_A \vee C) \leftrightarrow p_A), \Psi \cup p_A \rangle \approx A$, ou seja, não necessitamos de H para obter uma probabilidade positiva de A .

Outro trabalho futuro é a possibilidade de gerar uma fórmula abduzida H quando é dado que uma fórmula A ocorreu com uma probabilidade p em uma base de conhecimento representada pela instância PSAT $\Delta = \langle \Gamma, \Psi \rangle$, que possui as condições de abdução probabilística, de forma que para $\langle \Gamma \cup H, \Psi \rangle$ tenhamos que toda distribuição probabilística π que o satisfaz é tal que $\pi(A) \geq p$. O primeiro método apresentado na seção 5.4 do capítulo 5 é capaz de resolver esse problema, porém todas as soluções ficam restritas ao caso onde temos $\pi(A) = p$. Um método que permita de forma genérica que tenhamos $\pi(A) \geq p$ pode ser produzido desde que consigamos ajustar a matriz B e os vetores w e π (com $w\pi = p$) que formam uma solução de $\langle \Gamma, \Psi \rangle$.

Uma possibilidade também é o emprego da abdução probabilística para efetuar ajustes da probabilidade de uma fórmula A em uma dada instância PSAT Δ : dado que toda distribuição de probabilidades π que satisfaz Δ é tal que $0 < \pi(A) < 1$ (não temos a condição $\Delta \not\approx A$) a ideia é criar um método que gere fórmulas abduzidas H_1 e H_2 tais que, adicionadas à Δ , façam com que toda distribuição π seja tal que $p_1 < \pi < p_2$, ou seja, a adição das fórmulas abduzidas H_1 e H_2 à Δ gerou novos limitantes à probabilidade de A .

Referências Bibliográficas

- [Ach01] P. Achinstein. *The Book of Evidence*. Oxford University Press, 2001. 4
- [AF13] Alexandre M. Arruda e Marcelo Finger. Completeness for cut-based abduction. *Logic Journal of the IGPL*, 16:537–560, 2013. 9, 26, 33, 73
- [Ali96] A. Aliseda. A unified framework for abductive and inductive reasoning in philosophy and ai. *ECAI'96 Workshop on Abductive and Inductive Reasoning*, 1996. 3
- [Ali97] A. Aliseda. *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence*. Doctoral Dissertation, Stanford University, 1997. 6, 25
- [Bon11] G. De Bona. Satisfatibilidade probabilística. Dissertação de Mestrado, Instituto de Matemática e Estatística (IME-USP), 2011. iii, v, 7, 37, 43, 44, 45, 46, 47, 57, 58, 59, 67
- [Boo54] G. Boole. *An Investigation of the Laws of Thought: on which are Founded the Mathematical Theories of Logic and Probabilities*. Walton and Marberly, 1854. 7, 37
- [Bri] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, páginas 543–565. 8
- [Coo71] S. A. Cook. The complexity of theorem-proving procedures. *Proceedings of the third annual ACM symposium on Theory of computing*, 1971. 7, 37
- [D'A92] M. D'Agostino. Are tableaux an improvement on truth-tables? cut-free proofs and bivalence. *Journal of Logic, Language and Information*, 1:235–252, 1992. 29
- [DJ09] J. H. Martin D. Jurafsky. *An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, 2009. 9
- [DK94] T. Day e H. Kincaid. Putting inference to the best explanation in its place. *Synthese*, 98:271–295, 1994. 6
- [DM94] Marcello D'Agostino e Marco Mondadori. The taming of the cut. classical refutations with analytic cut. *Journal of Logic and Computation*, 4:285–319, 1994. 6, 18, 20, 26, 73

- [End72] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 1972. 21, 56
- [FdB10] M. Finger e G. de Bona. A refuted conjecture on probabilistic satisfiability. *Advances in Artificial Intelligence SBIA 2010: 20th Brazilian Symposium on Artificial Intelligence, Proceedings, Springer-Verlag*, 2010. 56
- [Fin74] B. De Finetti. *Theory of probability*. John Wiley and Sons, 1974. 7
- [Gen35] G. Gentzen. Untersuchungen über das logische schliessen. *Mathematische Zeitschrift, tradução em (Sza69)*, 39:176–219, 1935. 11, 14
- [GGP88] D. Kavvadias G. Georgakopoulos e C. H. Papadimitriou. Probabilistic satisfiability. *Journal of Complexity*, páginas 1–11, 1988. 7, 37, 39, 42
- [Gt14] J. P. Grotzinger e *et al.* A habitable fluvio-lacustrine environment at yellowknife bay, gale crater, mars. *Science*, 343(6169), 2014. 2
- [Hai65] T. Hailperin. Best possible inequalities for the probability of a logical function of events. *American Mathematical Monthly*, 72:343–359, 1965. 7
- [Hai86] T. Hailperin. *Boole's Logic and Probability*. North-Holland Pub. Co., 1986. 37
- [Har65] G. H. Harman. The inference to the best explanation. *Philosophical Review*, 74:88–95, 1965. 4, 5
- [HB09] G. J. Holton e S. G. Brush. *Physics, the Human Adventure: From Copernicus to Einstein and Beyond*. Rutgers University Press, 2009. 1
- [Hig64] P. W. Higgs. Broken symmetries and the masses of gauge bosons. *Physical Review Letters*, 13:508–509, 1964. 2
- [HJ00] P. Hansen e B. Jaumard. *Probabilistic satisfiability*. Handbook of Defeasible Reasoning and Uncertainty Management Systems: Algorithms for uncertainty and defeasible reasoning, 2000. 7, 37
- [JJ94] J. R. Josephson e S. G. Josephson. *Abductive Inference*. Cambridge University Press, 1994. 4, 5
- [Lip04] P. Lipton. *Inference to the Best Explanation*. Routledge, 2nd edição, 2004. 2, 6
- [McM92] E. McMullin. *The Inference that Makes Science*. Marquette University Press, 1992. 2
- [MDG08] M. Finger M. D'Agostino e D. Gabbay. Cut-based abduction. *Logic Journal of the IGPL*, 2008. 6, 25, 26, 27, 29, 30, 31, 33
- [MP93] M. C. Mayer e F. Pirri. First-order abduction via tableau and sequent calculi. *Journal of the IGPL*, páginas 99–117, 1993. iii, v, 26, 27, 28, 30
- [Nil86] Nils J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–87, 1986. iii, v, 7, 37, 39

- [Pei58] C. S. Peirce. *Collected Papers of Charles Sanders Peirce*, volume 1-6, editado por C. Hartshorne, P. Weiss. Harvard University Press, 1958. 1, 2, 3
- [PS98] C.H. Papadimitriou e K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Pubns, 1998. 40
- [Psi99] S. Psillos. *Scientific Realism: How Science Tracks Truth*. Routledge, 1999. 2, 6
- [Smu68] Raymond M. Smullyan. *First-Order Logic*. Dover, 1968. 6, 14, 15, 17, 18, 20, 21, 26
- [Sza69] M. E. Szabo. *The Collected Papers of Gerhard Gentzen*. North-Holland, 1969. 76
- [Tha88] P. R. Thagard. *Computational Philosophy of Science*. Cambridge, MIT Press. Bradford Books, 1988. 3
- [Tur36] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 42:230–265, 36. 6, 21, 22, 25