



Bernardo Pinto de Alkmim

**A Lógica sobre Leis iALC: Implementação de
Provas de Correção e Completude e Proposta
de Formalização da Legislação Brasileira**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio.

Orientador : Prof. Edward Hermann Haeusler

Co-Orientador: Prof. Alexandre Rademaker

Rio de Janeiro
Março de 2019



Bernardo Pinto de Alkmim

**A Lógica sobre Leis iALC: Implementação de
Provas de Correção e Completude e Proposta
de Formalização da Legislação Brasileira**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

Prof. Edward Hermann Haeusler

Orientador

Departamento de Informática — PUC-Rio

Prof. Alexandre Rademaker

Co-Orientador

Fundação Getulio Vargas — Matriz

Prof. Luiz Carlos Pinheiro Dias Pereira

Departamento de Filosofia — PUC-Rio

Prof. Mario Roberto Folhadela Benevides

UFRJ

Rio de Janeiro, 11 de Março de 2019

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Bernardo Pinto de Alkmim

Bacharel em Ciência da Computação pela PUC-Rio em 2016, com Domínio Adicional em Matemática. Participou de workshops e congressos nas áreas de Teoria da Computação e Lógica, em especial XVIII EBL.

Ficha Catalográfica

Alkmim, Bernardo

A Lógica sobre Leis iALC: Implementação de Provas de Correção e Completude e Proposta de Formalização da Legislação Brasileira / Bernardo Pinto de Alkmim; orientador: Edward Hermann Haeusler; co-orientador: Alexandre Rademaker. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2019.

v., 83 f: il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. – Tese. 2. iALC. 3. Legislação Brasileira. 4. Lógica de Descrição. 5. Lógica Intuicionista. 6. Formalização de Textos Jurídicos. 7. Raciocínio Jurídico. I. Haeusler, Edward. II. Rademaker, Alexandre. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Agradecimentos

Queria começar agradecendo ao Hermann, pela orientação tanto na minha graduação quanto agora, e a toda a equipe do TecMF e o pessoal da FGV que me auxiliaram no decorrer deste projeto, em especial ao Alexandre, que topou o papel de coorientador. Ambos foram cruciais para a motivação e desenvolvimento deste trabalho, além do artigo que conseguimos publicar.

Agradeço aos funcionários e ao corpo docente do Departamento de Informática, em especial ao Hermann e ao Roberto, pelas disciplinas lecionadas e todo o conhecimento passado, que aumentam minha maturidade como pesquisador. Também agradeço aos colegas do corpo discente junto dos quais aprendi, ensinei, e cresci.

Sou grato à minha família, pelo apoio constante e incentivo em ir atrás daquilo que me satisfaz profissionalmente.

Gostaria de agradecer também aos amigos de toda a vida, em especial ao pessoal do Coral da PUC-Rio. Muitas vezes os pontos altos das minhas semanas são os ensaios e o tempo que passamos juntos.

Também agradeço à PUC-Rio pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Alkmim, Bernardo; Haeusler, Edward; Rademaker, Alexandre. **A Lógica sobre Leis iALC: Implementação de Provas de Correção e Completude e Proposta de Formalização da Legislação Brasileira.** Rio de Janeiro, 2019. 83p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A lógica iALC é uma lógica de descrição de caráter intuicionista, criada para lidar com textos jurídicos como alternativa à mais comumente utilizada lógica deôntica, por conseguir contornar problemas que se encontra ao utilizar esta última. Nesta dissertação, introduzimos os principais conceitos que formam iALC, argumentamos sobre sua utilização em vez de demais lógicas para formalização de leis, implementamos suas provas de correção e completude no assistente de provas $L\exists\forall N$, e apresentamos uma proposta de formalização de leis brasileiras em iALC. Além disso, mostramos um exemplo de aplicação desta formalização para resolução de questões de múltipla escolha da primeira fase do exame da OAB, que tem por objetivo avaliar a aptidão dos candidatos para a prática da advocacia no Brasil. São vistos 3 exemplos de questões, cujas características são discutidas e comparadas umas às outras.

Palavras-chave

iALC; Legislação Brasileira; Lógica de Descrição; Lógica Intuicionista; Formalização de Textos Jurídicos; Raciocínio Jurídico.

Abstract

Alkmim, Bernardo; Haeusler, Edward (Advisor); Rademaker, Alexandre (Co-advisor). **The Logic on Laws iALC: Implementation of Soundness and Completeness Proofs and a Proposal for Formalization of Brazilian Law.** Rio de Janeiro, 2019. 83p. MsC Thesis — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The logic iALC is a description logic with an intuitionistic aspect to it, created to deal with legal texts as an alternative to the more common deontic logic, by being able to avoid problems found when utilizing the latter. In this dissertation, we introduce the core concepts which form iALC, debate on its utilization instead of other logics for legal formalization, implement the soundness and completeness proofs for it in the proof assistant $L\exists\forall N$, and present a proposal for formalization of Brazilian law in iALC. Furthermore, we show an example of application of this formalization in order to reason on multiple choice questions of the first part of the OAB Exam (the Brazilian national Bar exam), which aims to test candidates for their aptitude to practice the law in Brazil. We will show 3 examples, whose characteristics will be discussed and then compared to the others.

Keywords

iALC; Brazilian Law; Description Logic; Intuitionistic Logic; Legal Text Formalization; Legal Reasoning.

Sumário

1	Introdução	9
1.1	Estrutura	10
2	Estado da Arte	12
2.1	Lógicas para Leis	13
2.2	Formalização de Leis	13
3	iALC	16
3.1	Lógicas de Descrição	16
3.2	Lógica Intuicionista	20
3.3	Uma Lógica de Descrição Intuicionista	23
3.4	Comparando com Lógica Deôntica	28
4	Provas de Correção e Completude	30
4.1	LEAN	31
4.2	Um cálculo de seqüentes para iALC	32
4.3	Prova de Completude	36
4.4	Prova de Correção	39
5	Leis Brasileiras	43
5.1	Organização e Estrutura	43
5.2	Lei 8906	46
6	Exemplo de Uso: Exame da OAB	52
6.1	Estrutura do Exame	52
6.2	Exemplo 1: Marcas	53
6.3	Exemplo 2: Deise	55
6.4	Exemplo 3: Luana, Leonardo, e Bruno	58
6.5	Discussão dos Exemplos	59
7	Considerações Finais	62
7.1	Estudo e Aprendizado	62
7.2	Trabalho Futuro	63
A	Apêndice - Lei 8906	69
A.1	Capítulo I - Da Atividade da Advocacia	69
A.2	Capítulo II - Dos Direitos do Advogado	72
A.3	Capítulo III - Da Inscrição	79

The enemy of art is the absence of limitations.
Orson Welles, *The Movie Business Book*.

1

Introdução

Realizar uma análise mais precisa e automatizada de sistemas legais não é uma tarefa simples, uma vez que leis são criadas e alteradas continuamente no tempo, de acordo com a sociedade na qual estão inseridas. Há, claro, métodos de formação e padronização do processo de criação de leis, mas eles ocorrem de modo não centralizado, podendo, inclusive, variar em função do tempo. Da promulgação de uma lei à de outra, podem haver variações linguísticas significativas que requeiram uma mudança na construção legal, por exemplo. Podem haver diversos outros tipos de mudanças além da linguística, sem contar com leis que podem alterar textos de outras leis previamente escritas, dificultando o processo de análise, pois é necessário considerar os *corpora* legais em determinados instantes de tempo (i. e. lidar com *instâncias* das legislações).

Com isso, os textos que representam diferentes sistemas legais, além de poderem apresentar formatos diferentes entre si, apresentam muitas distinções entre várias de suas partes, o que dificulta ainda mais tentativas de formalização via análise textual, puramente.

Mesmo assim, há certo grau de ordem e estrutura. Seguindo a Teoria do Ordenamento Jurídico de Bobbio [BS95], temos melhor noção de como isso ocorre. Na legislação brasileira, por exemplo, cada lei é dividida em artigos, que por sua vez são divididos em parágrafos, que são divididos em incisos etc. Há leis com maior precedência que outras por virem diretamente da constituição, há leis que alteram outras leis, entre outras, e busca-se obter uma ordenação total das leis para resolução de antinomias legais de modo mais preciso e claro. Aqui foi dito *busca-se* pois isso não é sempre possível, havendo casos em que cabe ao juiz interpretar e aplicar a lei corretamente. Discorreremos mais sobre antinomias e a estrutura das leis brasileiras no capítulo 5.

Isso nos leva a buscar soluções que não se limitam apenas a uma análise textual lei por lei, uma vez que é necessário saber como elas se relacionam e como podem ser aplicadas a casos reais, i. e., como se realiza o processo de jurisdição, especialmente em casos onde ocorrem as ditas *antinomias reais*. Discutiremos mais sobre antinomias reais e outros problemas encontrados em textos jurídicos mais à frente, nos capítulos 2 e 3. Uma solução encontrada que permite organização da

informação adquirida com o adicional de permitir realizar raciocínio e argumentação sobre as leis é via o uso de lógica.

Propostas para essa formalização de modo geral envolvem alguma lógica específica para lidar com leis. Há muitos trabalhos que utilizam lógica deôntica (ou alguma variação dela), na qual as leis são atribuídos valores-verdade. Tais lógicas apresentam problemas com certas interações específicas entre leis, que serão discutidas mais a fundo nos capítulos 2 e 3. Outras propostas envolvem lógicas de descrição, que já lidam de outro modo com os relacionamentos entre leis, como, por exemplo, a lógica de descrição intuicionista iALC, que será o foco deste trabalho. Essa lógica foi criada por Valeria de Paiva, Edward Hermann Haeusler e Alexandre Rademaker e teve seus conceitos e componentes introduzidos em [HPR10], e percorridos em [HR16], [Del+17] e [AHR18], com alguns exemplos de uso. As argumentações presentes nesses artigos serão debatidas ao longo da dissertação.

Este documento apresenta como dissertação de mestrado do Departamento de Informática da PUC do Rio de Janeiro uma pesquisa sobre diferentes tentativas de formalizar leis (especialmente sob o ponto de vista da lógica), focando enfim na lógica iALC, motivando para seu uso. Além disso, apresenta uma implementação das provas de correção e completude de iALC no provador de teoremas *LEAN*¹ a fim de explicitar a possibilidade e viabilidade de realizar cálculos com essa lógica, e, por fim, uma proposta de formalização da legislação brasileira utilizando iALC, com uma possível aplicação na resolução da fase de múltipla escolha do Exame da OAB.

Pretendemos utilizar os resultados aqui encontrados para um uso futuro em um projeto de resolvidor automatizado de provas de múltipla escolha da Ordem dos Advogados do Brasil (OAB). O papel deste trabalho no projeto desse resolvidor será na etapa de busca de leis no corpus legal e de raciocínio dedutivo a fim de buscar justificativas para indicar se uma alternativa de resposta a uma questão está correta ou não, ocorrendo após o processamento de linguagem natural da questão em si.

1.1

Estrutura

Este documento está organizado nos seguintes capítulos:

O capítulo 2 contém as principais ideias publicadas sobre o tema em artigos técnicos específicos da área. Os assuntos fundamentais a serem resumidos vêm em duas partes principais: lógicas sobre leis e os problemas por elas encontrados, e teorias legais e da argumentação em conjunto com formalização em lógica de textos

¹<https://leanprover.github.io/>

legais. A presença dos artigos da primeira parte têm como propósito contribuir para melhor definição e implementação de iALC, e os da segunda para dar maior fundamentação para formalizar as leis brasileiras.

O capítulo 3 apresenta a fundamentação teórica de iALC, bem como suas características, propriedades e conceitos, focando nas diferenças a outras lógicas sobre leis e como os problemas por elas encontrados são solucionados. Mencionaremos além disso os conceitos de teoria do direito que inspiraram sua concepção.

No capítulo 4 estão explicadas as provas de completude e correção, bem como o detalhamento de suas implementações e a motivação para essas implementações.

O capítulo 5 aborda como realizamos a formalização dos códigos legais brasileiros utilizando iALC. São listadas características das leis brasileiras e como elas são representadas (ou como podem ser representadas) por iALC, além de como são tratados possíveis conflitos encontrados. Também ficam descritas as heurísticas encontradas para formalização em iALC. Esse capítulo faz variadas referências ao Apêndice A, onde estão completamente formalizados os 3 primeiros capítulos da Lei 8906, contendo 14 artigos.

O capítulo 6 discorre sobre o exame de ingresso na OAB como possível aplicação para a formalização realizada em iALC, em especial na etapa da prova de múltipla escolha. Há 3 questões de exemplo, cujas formalizações e cujos métodos de resolução são explicados e comparados.

O capítulo 7 fecha a pesquisa como um todo. Destaca as principais contribuições realizadas e tenta confrontá-las com os objetivos inicialmente estabelecidos. Os pontos ainda em aberto são apresentados e uma relação de possíveis trabalhos futuros enumerada. Focaremos em especial no projeto de *solver* de questões de múltipla escolha da prova da OAB.

2

Estado da Arte

Lógica e leis tiveram estudos realizados com o passar dos anos, seja na representação de raciocínio jurídico via silogismos ou na formalização de textos legais em lógica, muito mais na área de lógica do que na do direito em si, o que tende a render pesquisas mais teóricas. O uso prático desses resultados acaba ficando em segundo plano, e isso cria um distanciamento entre o resultado dessas pesquisas e a área que se utilizaria do conhecimento gerado efetivamente.

O raciocínio jurídico é, por muitas vezes, aprendido apenas na prática e não são apresentadas ou difundidas metodologias mais concretas para seu uso, apesar de ser muito baseado em silogismos lógicos e teoria da argumentação. De fato, Walker, em [Wal07], afirma que trabalhos que busquem estabelecer metodologias são pontuais, e não amplamente difundidos dentro da área do direito, mantendo-se em áreas que utilizam o direito como *domínio*. Nesse mesmo trabalho, Walker também propõe uma estruturação do raciocínio legal baseando-se em regras i.e. deduções, explicitando sua busca por um uso prático de uma lógica que possa representar esse tipo de raciocínio.

Outro ponto interessante no trabalho de Walker é a estruturação da resolução de um caso jurídico. A argumentação para um caso apresenta duas partes: uma de *evidências* e outra de *regras legais*. Para as evidências, ele utiliza o que chama de *esquemas de plausibilidade instanciados*, ou seja, instâncias que podem ou não estar de acordo com a lei (que acabam agindo como as hipóteses de um raciocínio dedutivo). Já para as regras legais, ele utiliza *árvores de implicação*, que são basicamente árvores de dedução restritas à implicação clássica sobre alguma formalização do texto legal em lógica deôntica (pois os textos se tornam proposições). Por utilizar apenas implicação, e ficar no domínio de lógica deôntica, há muitas restrições naquilo que pode ser feito, mas tal estruturação se assemelha de alguma forma à realizada em iALC para resolver as questões do exame da OAB no capítulo 6.

É necessário ter uma representação lógica para poder então aplicar a dedução em si, caso contrário, está apenas sendo realizado raciocínio jurídico, que, conforme dito anteriormente, não apresenta bases realmente formais para uso consistente em lógica, o que leva à impossibilidade de implementação de sistemas dedutivos

automatizados.

A seguir apresentamos alguns trabalhos cujos temas foram úteis para o desenvolvimento desta dissertação. Os trabalhos que se relacionam com esse ocorrem em dois frentes: há os que exploram diferentes lógicas que podem ser utilizadas para representar leis, como lógica de descrição e lógica deôntica e suas variações, e há os que buscam formalização de estrutura entre leis utilizando lógica.

2.1

Lógicas para Leis

Hariri et al [Har+14] discorrem sobre Bases de Conhecimento e Ações de lógicas de descrição (KAB, do inglês *Knowledge Action Bases*), que são um mecanismo com uma semântica rica para representar informação do domínio de interesse em termos de uma base de conhecimento de lógica de descrição e ações para modificá-la com o passar do tempo, podendo introduzir novos objetos. Esse caráter mutável é útil para avaliar como construir uma base de conhecimento de leis onde elas podem alterar umas às outras, podendo inclusive eliminar outras leis da base.

Apesar de não utilizarmos KAB, realizamos um processo parecido no capítulo 5, onde formalizamos artigos da Lei 8906 da legislação brasileira. Nela, há artigos que alteram os anteriores devido a mudanças posteriores à criação da Lei 8906 em si.

Hansen, Pigozzi, e Torre [HPT07] apresentam dez problemas filosóficos encontrados em lógica deôntica, como, por exemplo, o Dilema de Jørgensen [Jør38], que faz menção à possibilidade de noção de validade em normas (no nosso caso, leis). Esse artigo entra como parte da motivação para o uso de soluções alternativas ao lidar com leis, como iALC.

Os problemas descritos nesse artigo serão discutidos no capítulo 3, onde introduziremos iALC e a compararemos à lógica deôntica.

2.2

Formalização de Leis

Há também as propostas de formalização aplicadas a diferentes contextos legais, de modo geral utilizando uma base deôntica ou de descrição, mas não limitado a elas.

Pertierra et al. [Per+17] realizam um projeto que visa facilitar representação e checagem de leis antes da passagem para linguagem formal (focando em problemas encontrados em estatutos e regulamentos de impostos), representando as relações semânticas entre as leis na semântica da lógica por eles utilizada, chamada de lógica *default* (criada por Reiter em [Rei80]), na qual se supõe que algo é verdadeiro até

que o contrário seja provado. Um ponto interessante é que essa lógica é não-monotônica¹. De fato, são apresentadas algumas formalizações de leis britânicas utilizando diferentes variações dessa lógica default, que se traduzem justamente em leis que cancelam ou atualizam outras.

Em iALC, por ser uma lógica de descrição, temos que uma futura implementação de um raciocinador gerará construtivamente uma base de conhecimento de leis, podendo remover fórmulas já existentes da base, de acordo com as novas leis adicionadas, apresentando um análogo ao caráter não monotônico apresentado nesse artigo.

Além disso, eles implementam um sistema de auxílio a advogados, e apresentam uma *pipeline* para o processo realizado, que pode nos ser útil para uma futura implementação de um raciocinador sobre leis brasileiras.

Verheij [Ver17] busca formalizar argumentos legais, tendo em vista suas principais bases: casos e regras, e os aplica à lei holandesa. O artigo busca desenvolver as conexões entre esses conceitos para aplicar em situações reais na lei holandesa. Seus métodos envolvem uso de inteligência artificial, mas nosso foco aqui está na modelagem por ele realizada.

O autor utiliza três modos de formalização: para casos, para regras, e para argumentos. Para o primeiro, busca-se encontrar relações entre diferentes situações. Para o segundo, são apresentados modos de unir diferentes regras legais de forma coerente. Já para o terceiro, o foco muda para o raciocínio jurídico sobre os dois primeiros. O primeiro e segundo modos serviram como guias da formalização realizada no capítulo [5], e o terceiro, dos raciocínios feitos no capítulo [6].

Letia e Groza [AG09] utilizam lógica de descrição para lidar com a modelagem de argumentos imprecisos, via ontologias e lógica fuzzy. Lógica fuzzy é usada para criar medidas de precisão ou veracidade dos argumentos apresentados. Como esse artigo lida diretamente com argumentos e não com leis, a veracidade é um ponto realmente vital, mas o é um pouco menos para nós, posto que leis são normas. Isso pode ser interessante a nível de alternativa a uma implementação futura para um raciocinador sobre a legislação brasileira. As formalizações utilizando lógica de descrição apresentadas traçam um paralelo quase direto com o que fizemos no capítulo [5].

Khan et al [Kha+16] convertem texto legal para um conjunto de regras lógicas no contexto de leis dos EUA que envolvem a área de medicina, por meio de ferramentas de modelagem relacional e um sistema de apoio à decisão especializado

¹Numa lógica monotônica, como a clássica ou a intuicionista, por exemplo, quando se tem um conjunto Γ de premissas levando a uma conclusão α ($\Gamma \vdash \alpha$), não importa qual nova premissa γ adicionarmos, sempre será possível concluir α desse novo conjunto de premissas ($\Gamma, \gamma \vdash \alpha$), diferentemente de lógicas não monotônicas, nas quais pode ser que o acréscimo de γ impeça que α seja conclusão.

para a área médica.

Desse artigo tiramos de relevante a metodologia para geração de regras sobre o domínio de leis, posto que uma modelagem relacional é uma abordagem razoavelmente diferente da que estamos propondo.

3

iALC

Há diversos tipos diferentes de lógicas, com diversas características e propriedades, o que acaba gerando variações em suas bases teóricas, expressividade e aplicabilidade. A base de iALC, ALC, é utilizada amplamente para a construção de ontologias sobre a *web*, por exemplo. Lógica proposicional clássica é utilizada para a geração de circuitos lógicos, e a lógica de primeira ordem (FOL, do inglês *first-order logic*) permeia toda a matemática, e fragmentos decidíveis de FOL são utilizados em bancos de dados relacionais. Há também lógicas de maior ordem, lógicas modais, de descrição, entre outras.

Em se tratando de leis, a mais comumente utilizada é a lógica deôntica. A versão padrão desta lógica é a lógica modal **KD**, indicando que \Box (o operador de lógicas modais, que no caso da deôntica indica obrigatoriedade, necessidade) é um operador para o qual valem os axiomas **K** e **D** de lógica modal, mas há extensões que consideram também que vale o axioma **5**. Essa lógica lida com sentenças imperativas de modo geral, focando em obrigações e permissões, portanto a passagem para o mundo das leis é algo aparentemente natural. Ao final deste capítulo mostraremos por que isso não necessariamente é o caso.

Antes de apresentar a lógica iALC, apresentamos os principais conceitos e operadores das lógicas de descrição e da lógica intuicionista. Depois mostramos como iALC as funde, assim como as bases teóricas que motivam seu uso com leis. Ao fim deste capítulo, a comparamos com lógica deôntica, para motivar sua utilização no lugar desta última.

3.1

Lógicas de Descrição

O capítulo 3 de [HLP08], de autoria de Franz Baader, Ian Horrocks, e Ulrike Sattler fornece uma grande e detalhada base teórica para o estudo de lógicas de descrição (DL, do inglês *description logic*). Aqui utilizamos a notação de acordo.

Criadas nos anos 80, as DLs buscam realizar representação de conhecimento (KR, do inglês *Knowledge Representation*) sob o ponto de vista da lógica, que permite maior precisão na definição e o uso via uma representação de ontologias que permite implementação, e, por conseguinte, que possamos utilizar deduções

automatizadas sobre elas.

Lógicas de descrição apresentam equivalência a fragmentos, geralmente decidíveis, de FOL. A nível de terminologia, há três principais componentes, cada um com análogo em FOL:

- Indivíduos

Análogos às constantes de FOL, representam a instanciação de alguma entidade, à qual atribuem-se diferentes papéis e diferentes conceitos.

- Conceitos

Análogos aos predicados unários de FOL, representam propriedades que podem se referir a diferentes indivíduos, e assim agrupá-los.

- Papéis

Análogos aos predicados binários de FOL, representam relações que podem existir entre diferentes indivíduos.

Utilizamos estes componentes para preencher bases de conhecimento da seguinte maneira: seja $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ uma base de conhecimento. \mathcal{A} dizemos ser o *ABox* (*Axiomatic Box*) de \mathcal{K} , no qual encontramos sentenças referindo-se a indivíduos instanciados, e \mathcal{T} , o *TBox* (*Terminologic Box*), que é um conjunto de definições de terminologia, ou seja, conceitos e papéis de \mathcal{K} .

Para poder popular a ABox e a TBox, é necessário saber como se formam seus elementos. O alfabeto de uma DL consiste em:

- Conjuntos de nomes de indivíduos, conceitos e papéis

Os nomes dados a indivíduos, conceitos e papéis para fins de identificação também fazem parte da linguagem da lógica de descrição. Por convenção, indivíduos e papéis começam por minúscula, ¹ e conceitos por maiúscula.

- \perp

Em DLs, \perp é uma constante que representa um conceito sem indivíduos, ou seja, um conceito que não vale para indivíduo algum.

- \top

É o complemento de \perp , ou seja, um conceito que vale para qualquer indivíduo.

- \sqcap

Conjunção de conceitos. $C \sqcap D$ é um novo conceito, representando a interseção de C e D .

¹Na literatura, quando se define um papel genérico, é comum utilizar a letra R maiúscula, mas sempre que um papel é instanciado utiliza-se um nome iniciado por letra minúscula. Neste texto manteremos essa convenção.

– \sqcup

Disjunção de conceitos. Funciona de modo análogo a \sqcap .

– \neg

Complemento de conceitos, que gera um novo conceito.

– \forall

Restrição universal de conceitos utilizando papéis. $\forall R.C$ é um novo conceito representando uma restrição universal do conceito C sob o papel R .

– \exists

Restrição existencial de conceitos utilizando papéis. Funciona de modo análogo a \forall .

– $:$

Gera asserções do tipo $a : C$ (chamada asserção de conceitos) e $(a, b) : R$ (asserção de papéis), para C conceito, R papel, e a e b indivíduos, e indicam que os indivíduos em destaque populam o conceito ou papel a que faz referência. Com essas sentenças formamos o ABox.

– \sqsubseteq

Representando a inclusão de conceitos, este operador é utilizado da seguinte maneira: $C \sqsubseteq D$ indica que o conceito C está incluso no conceito D . Funciona do mesmo modo com papéis.

– \equiv

Indica equivalência de conceitos e papéis, pode ser derivado do anterior, pois $C \sqsubseteq D \wedge D \sqsubseteq C \Leftrightarrow C \equiv D$. Sentenças com este operador ou com \sqsubseteq como o operador principal formam a TBox, e recebem o nome de inclusão geral de conceitos (GCI, do inglês *general concept inclusion*).

Agora que temos uma introdução geral à sintaxe de DLs, vejamos uma das DLs canônicas chamada ALC, que serve de base para iALC. O \mathcal{AL} em ALC significa linguagem de atribuição de conceitos (do inglês *attributive language*), o que indica que ela apresenta negação atômica de conceitos, interseção de conceitos, restrições universais, e quantificação existencial limitada. Já o \mathcal{C} indica que ela apresenta a negação de conceitos complexos, ou seja, conceitos não atômicos como o conceito formado pela interseção $C \sqcap D$, para C e D conceitos simples, pode ser negado, resultando no conceito $\neg(C \sqcap D)$. Assim, ALC significa Linguagem de Atribuição de conceitos com Complementos.

A semântica de ALC pode ser definida tomando conceitos como conjuntos de indivíduos e papéis como conjuntos de pares de indivíduos (que assumimos

pertencer a um determinado domínio). Portanto, seja $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ uma interpretação para ALC, onde $\Delta^{\mathcal{I}}$ é um conjunto não vazio que representa o domínio, e $\cdot^{\mathcal{I}}$, uma função de interpretação que mapeia nomes de indivíduos a elementos de $\Delta^{\mathcal{I}}$, conceitos a subconjuntos de $\Delta^{\mathcal{I}}$, e papéis a subconjuntos de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, tais que, para R papel e C e D conceitos:

$$\begin{aligned}\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\ \perp^{\mathcal{I}} &= \emptyset \\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\ (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\ (\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y, (x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\} \\ (\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y, (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}\end{aligned}$$

A interpretação para estruturas mais complexas funciona de modo recursivo na sintaxe de ALC. Agora, para formação efetiva da ABox e da TBox iremos aos conceitos de validade. Definimos $\mathcal{I} \models$ (indicando que a interpretação \mathcal{I} valida o que vem ao lado direito) como:

$$\begin{aligned}\mathcal{I} \models a : C &\Leftrightarrow a^{\mathcal{I}} \in C^{\mathcal{I}} \\ \mathcal{I} \models (a, b) : R &\Leftrightarrow (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \\ \mathcal{I} \models \mathcal{A} &\Leftrightarrow \forall \phi \in \mathcal{A}, \mathcal{I} \models \phi\end{aligned}$$

Essas três sentenças formam a interpretação da ABox \mathcal{A} em ALC. A da TBox \mathcal{T} é como segue:

$$\begin{aligned}\mathcal{I} \models C \sqsubseteq D &\Leftrightarrow C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \\ \mathcal{I} \models \mathcal{T} &\Leftrightarrow \forall \phi \in \mathcal{T}, \mathcal{I} \models \phi\end{aligned}$$

Vale lembrar que \equiv pode ser derivado de \sqsubseteq . Com isso, falta apenas a base de conhecimento \mathcal{K} em si.

$$\mathcal{I} \models \mathcal{K} \Leftrightarrow \mathcal{I} \models \mathcal{T} \wedge \mathcal{I} \models \mathcal{A}$$

Para termos um exemplo de uma base de conhecimento populada, sejam

os indivíduos *caio*, *rita* e *marta*, conceitos *Humano*, *Homem*, *Pai*, *Mulher*, e *Mae* e papéis *temFilha*, e *conjugueDe*. Um exemplo de base de conhecimento $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ sobre estes termos seria:

$$\begin{aligned}\mathcal{A} &= \{caio : Homem, rita : Mulher, marta : Mulher, \\ &\quad (caio, marta) : temFilha, (caio, rita) : conjugueDe\} \\ \mathcal{T} &= \{Homem \sqsubseteq Humano, Mulher \sqsubseteq Humano, Pai \equiv \\ &\quad Homem \sqcap \exists temFilha.Humano, Mae \equiv Mulher \sqcap \exists temFilha.Humano\}\end{aligned}$$

Note que definimos os conceitos *Pai* e *Mae* a partir da restrição existencial do conceito *Humano* via o papel *temFilha*. Ao observar a ABox, podemos concluir que, apesar de termos *caio : Pai*, não temos *rita : Mae*, pois não há a asserção $(rita, a) : temFilha$ para qualquer indivíduo *a*. Uma ABox diferente poderia fazer *rita : Mae* valer, mesmo com a mesma TBox.

Abriremos aqui um pequeno parênteses do mundo de DLs para explicar intuicionismo e lógica intuicionista, a fim de preparar a explicação dos fundamentos de iALC.

3.2 Lógica Intuicionista

O intuicionismo tem fortes relações com a matemática construtiva e lida com o conceito de *verdade* de um modo diferente da lógica clássica. Utilizaremos como base aqui os textos de Heyting, em [Hey56], e Priest, no capítulo 6 de [Pri08].

No intuicionismo, para alguma fórmula (digamos, proposicional) ϕ ser verdadeira, é necessário que haja uma construção de ϕ , uma *prova*. Isso se contrapõe à lógica clássica, que tem uma definição mais geral de o que é verdade, não havendo, muitas vezes, a necessidade de uma construção concreta de uma resposta. Há um exemplo canônico disso, com o seguinte questionamento: existem dois números irracionais, *a* e *b* tais que a^b seja racional? A resposta² envolve o número irracional $\sqrt{2}$. Se $\sqrt{2}^{\sqrt{2}}$ é racional, então a resposta foi encontrada, e $a = b = \sqrt{2}$. Caso contrário, é irracional, e $\sqrt{2}^{\sqrt{2}^{\sqrt{2}}}$ é racional pois é igual a 2, e $a = \sqrt{2}^{\sqrt{2}}$ e $b = \sqrt{2}$. Aparentemente a questão foi resolvida, mas fica o questionamento: *qual* das respostas é a correta? De fato, por usar o princípio do terceiro excluído, $\phi \vee \neg\phi$, esta prova constitui uma prova em lógica clássica, mas não em lógica intuicionista. Além do princípio do terceiro excluído, há várias equivalências clássicas que não

²Há, de fato, uma resposta construtiva a esta pergunta, envolvendo o Teorema de Gelfond-Schneider.

são intuicionistas, em especial a da dupla negação: $\neg\neg\phi \leftrightarrow \phi$. Em [Bel71], Belding explora mais as relações da negação intuicionista com estas fórmulas.

Essa equivalência não é válida em lógica intuicionista justamente porque ele não necessariamente envolve a construção de uma resposta específica, podendo apenas esgotar as possibilidades de resposta exaustivamente. Por lidar sempre com o conceito de provas e construções, a semântica do termo verdade está sempre atrelada a uma prova. Por exemplo, uma prova de $\phi \wedge \psi$ é formada a partir de um par de provas, uma de ϕ e outra de ψ , e a partir daí podemos dizer que a conjunção é verdadeira.

Um operador que se torna realmente diferente sob a ótica intuicionista é o da implicação, \rightarrow . Em lógica clássica, dizer $\phi \rightarrow \psi$ é o mesmo que dizer $\neg\phi \vee \psi$, ao passo que na intuicionista é dizer que ter uma prova de ϕ nos leva a uma prova de ψ . Há, de fato, menos mistura no uso dos operadores na lógica intuicionista do que na clássica, o que acaba afetando, por exemplo, a negação, \neg . Dizer que $\neg\phi$ é verdadeiro é dizer que há uma prova de que não há prova de ϕ , ou seja, de que ϕ leva a alguma contradição. De fato, esse operador pode ser definido simplesmente por $\phi \rightarrow \perp$.

Uma característica interessante da lógica intuicionista decorrente desta mudança na implicação é que não há necessariamente equivalência de uma implicação com sua contrapositiva, ou seja, $A \rightarrow B$ e $\neg B \rightarrow \neg A$ não são equivalentes para quaisquer A e B . De fato, temos que $(A \rightarrow B) \vdash (\neg B \rightarrow \neg A)$ (também conhecido como *modus tollens*), mas o contrário, $(\neg B \rightarrow \neg A) \vdash (A \rightarrow B)$, não ocorre. Mostraremos uma prova do primeiro, e argumentaremos sobre a impossibilidade do segundo. Segue a prova em dedução natural de acordo com [Pra65] (aqui utilizamos o teorema da dedução³ para a conclusão):

$$\frac{\frac{\frac{[A]^3 \quad [A \rightarrow B]^1}{B} \quad [\neg B]^2}{\frac{\perp}{\neg A} \quad 3} \quad 2}{(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A) \quad 1}$$

A volta não ocorre devido à semântica da implicação intuicionista, explicada acima. Podemos supor que $\neg B \rightarrow \neg A$, e que nos é dada uma prova de A . Queremos, a partir disso, ter uma prova de B . Suponhamos que tenhamos uma prova de $\neg B$, para transformar em uma prova de $\neg A$ via eliminação da implicação/*modus ponens*, e então obter uma prova de $A \wedge \neg A$, nos levando a um absurdo. Podemos, então, descartar nossa hipótese de $\neg B$ e concluir

³No caso, via o teorema da dedução, temos que $(A \rightarrow B) \vdash (\neg B \rightarrow \neg A)$ implica em $\vdash (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$.

$\neg\neg B$ via introdução da negação. Vale notar que não podemos realizar um *reductio ad absurdum* para gerar B , pois ele só é possível em lógica clássica, e assim não podemos remover essa negação. Temos, portanto, que $(\neg B \rightarrow \neg A) \vdash (A \rightarrow \neg\neg B)$. Em lógica clássica, como $\neg\neg B \rightarrow B$, teríamos a equivalência da contrapositiva, mas em lógica intuicionista isso é impossível.

Outro modo de ver isso é supondo que B é uma fórmula para a qual vale a contrapositiva entre ela e qualquer outra fórmula ϕ . Temos, portanto, que $(\neg B \rightarrow \neg\phi) \rightarrow (\phi \rightarrow B)$. Suponhamos que ϕ seja $\neg\perp$. Assim, $\neg\phi$ é $\neg\neg\perp$ e, portanto, é \perp , pois mesmo intuicionisticamente vale a dupla negação para \perp , dado que $\neg\perp$ é $\perp \rightarrow \perp$, uma tautologia, tornando $\neg\neg\perp$ um absurdo, \perp . Daí, temos $(\neg\neg B) \rightarrow (\neg\perp \rightarrow B)$, que é equivalente a $(\neg\neg B) \rightarrow (B)$. Assim, se B satisfaz a contrapositiva com qualquer ϕ , então $\neg\neg B \rightarrow B$, que sabemos não valer em lógica intuicionista. Assim, não podemos assumir intuicionisticamente que para alguma fórmula vai valer a contrapositiva globalmente, mostrando que ela é mais restrita que a clássica.

Mas apesar de ser uma lógica mais restrita, a intuicionista apresenta grande aplicabilidade e relação com conceitos diferentes, especialmente quando se trata de metamatemática e informática. De fato, pelo Isomorfismo de Curry-Howard, há uma equivalência de funções recursivas primitivas para fórmulas em lógica intuicionista e de algoritmos para funções recursivas primitivas para provas em lógica intuicionista. Ao lidarmos diretamente com lógica intuicionista, nos colocamos mais próximos das limitações de implementações computacionais dos problemas com os quais lidamos.

Lógica intuicionista também apresenta uma rica semântica de mundos possíveis, introduzida por Kripke em [Kri65]. Consideraremos lógica intuicionista proposicional, com os operadores \vee , \wedge , \neg , e \rightarrow . Uma interpretação para essa linguagem é uma tripla $\langle W, R, v \rangle$, onde W é um conjunto de mundos, R uma relação binária entre mundos reflexiva e transitiva, e v uma função de valoração, que recebe um mundo e uma fórmula e retorna se a fórmula é verdadeira, 1, ou falsa, 0, naquele mundo. É preciso que v atenda à condição de hereditariedade (sendo p uma variável proposicional da lógica):

$$\forall w \in W, v(w, p) = 1 \wedge wRw' \Rightarrow v(w', p) = 1$$

A valoração por v se dá recursivamente da seguinte forma (para ϕ e ψ fórmulas intuicionistas, e w um mundo):

$$v(w, \phi \wedge \psi) = 1 \text{ se } v(w, \phi) = 1 \wedge v(w, \psi) = 1. \text{ Se não, é } 0.$$

$v(w, \phi \vee \psi) = 1$ se $v(w, \phi) = 1 \vee v(w, \psi) = 1$. Se não, 0.

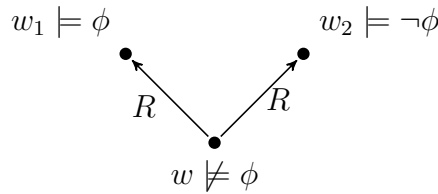
$v(w, \neg\phi) = 1$ se $\forall w'(wRw' \Rightarrow v(w', \phi) = 0)$. Se não, 0.

$v(w, \phi \rightarrow \psi) = 1$ se $\forall w'(wRw' \Rightarrow v(w', \phi) = 0 \vee v(w', \psi) = 1)$. Se não, 0.

Podemos ver que $\neg\phi$ se mantém como um caso especial da implicação, $\phi \rightarrow \perp$. \perp é uma constante que não é válida em mundo algum.

Pode-se pensar um mundo em semântica de mundos possíveis como sendo um grupo de informações em determinado momento, ou seja, aquilo que nele vale é aquilo que pode ser provado naquele momento (dizemos que na interpretação \mathcal{I} o mundo w valida a fórmula ϕ via $w \models_{\mathcal{I}} \phi$), e mundos que a esse são relacionados como extensões dele, podendo conter novas provas ou manter as provas já existentes. Pela condição de hereditariedade, uma vez provado algo não pode ser falsificado. Um fato interessante sobre a relação entre lógica intuicionista e clássica é que, se colapsarmos todos os mundos de um modelo de Kripke de lógica intuicionista em um mundo só, temos uma lógica clássica, pois a noção de construção de provas se perde, importando apenas os valores-verdade diretamente das fórmulas. Este fato será importante para explicar as mudanças entre as semânticas de iALC e de ALC mais à frente.

Com isso, a característica *paracompleta*⁴ da negação intuicionista fica mais aparente. Podemos ter um mundo w que não valida ϕ apenas, e ele pode estar relacionado a dois mundos, w_1 e w_2 , tais que $w_1 \models_{\mathcal{I}} \phi$ e $w_2 \models_{\mathcal{I}} \neg\phi$.



Agora que temos noção de o que é uma lógica de descrição e como se realiza um modelo de Kripke de uma lógica intuicionista, podemos introduzir iALC e apresentar como ela mescla esses conceitos e como ela os relaciona à teoria de leis que utiliza como referência.

3.3

Uma Lógica de Descrição Intuicionista

iALC é uma lógica que foi criada para lidar com leis e evitar os chamados paradoxos *contrary-to-duty* (CTD), encontrados ao se utilizar lógica deôntica. Em [HR16] há uma discussão sobre tais paradoxos. Essencialmente, eles ocorrem em

⁴Diz-se que algo é paracompleto quando não se aceita o princípio do terceiro excluído.

lógica deôntica por poder haver normas representando uma implicação considerando o não cumprimento de uma outra norma. Como as normas são proposições em lógica deôntica, chega-se a um paradoxo envolvendo a norma cujo cumprimento foi desconsiderado. Na seção 3.4 explicamos melhor como isso ocorre.

A base teórica no direito que inspirou o uso de iALC com leis é a jurisprudência de Kelsen, apresentada em muitos de seus trabalhos (uma referência mais atual que contém bastante da sua teoria é [Kel99]), cujos conceitos básicos são introduzidos em [HR16] para explicitar sua relação com iALC. O ponto central da teoria de Kelsen é a definição de *o que é lei*. Para Kelsen, *lei* pode ser vista de dois modos: como *Lei Natural*, ou sob o ponto de vista do *Positivismo Legal*. Para o uso em informática, o positivismo legal se aproxima mais daquilo que buscamos. Do ponto de vista prático, o positivismo legal é mais simples de lidar pois nele *lei* é apenas um conjunto de normas que mira direcionar determinados comportamentos em sociedade, ao passo que a visão de *Lei Natural* mescla o conceito de legalidade com o de moralidade, o que é bem mais complexo, e leva o foco do assunto para as áreas que envolvem Ética e o impacto na Sociedade, tirando-o das normas em si e suas relações umas com as outras.

Pelo Positivismo Legal, Kelsen cunha o conceito de *groundnorm*, que representa o grupo de normas básicas que dão amparo a todas as outras normas de um sistema legal. O amparo se dá via uma relação de precedência legal, uma relação que é reflexiva e transitiva. Fora isso, há a consideração de que normas são os indivíduos de primeira classe de uma ontologia legal. Por fim, há relações entre normas análogas entre diferentes sistemas legais, ou entre normas diferentes dentro de um mesmo sistema.

O fato de normas serem nossos indivíduos em conjunto com a precedência entre leis ser reflexiva e transitiva é um grande motivador para utilizar uma lógica intuicionista, pois um modelo de Kripke para lógica intuicionista surge quase naturalmente tomando os indivíduos como mundos e a precedência de leis como a relação entre mundos. Chamemos, então, essa relação de \preceq . Ainda assim se torna necessário manter em um mundo de lógicas de descrição para evitar os problemas de lógica deôntica.

Outro motivador para o uso de lógica intuicionista em vez da clássica é o operador da negação. Pela semântica de lógica intuicionista, um indivíduo i estaria em um conceito $\neg A$, $i : \neg A$, apenas se $\forall j$ tal que $i \preceq j$ não é o caso que $j : A$. Isso está de acordo com Kelsen, pois indica que i não precede nenhuma lei de A . Se utilizarmos lógica clássica, $\neg A$ se torna um conjunto representando o complemento de A , significando as leis em que não vale A ou que não são sentenças legais válidas (VLS, do inglês *valid legal statement*), mas todas as sentenças que consideramos são VLS. A semântica clássica acaba não refletindo fielmente a teoria

jurídica. Num ponto de vista mais prático, podemos considerar o conceito *Brasil* contendo as VLSs válidas em território brasileiro, e *pedroMaiorDeIdade* um VLS indicando que Pedro, de 17 anos é maior de idade. Como a maioridade no Brasil é de 18 anos, claramente não é o caso que $\text{pedroMaiorDeIdade} : \text{Brasil}$. Se pensarmos classicamente, pelo princípio do terceiro excluído, nos é forçado que $\text{pedroMaiorDeIdade} : \neg \text{Brasil}$, ou seja, que Pedro será maior de idade em todo sistema legal que não apresenta precedência com o brasileiro, o que sabemos não ser verdade, pois há países cuja maioridade é aos 18 anos assim como no Brasil. Do ponto de vista intuicionista, podemos simplesmente ter $\text{pedroMaiorDeIdade} \not\models \text{Brasil}$ sem ter necessariamente $\text{pedroMaiorDeIdade} \models \neg \text{Brasil}$.

Um último fator é que o intuicionismo permite a coexistência de normas aparentemente contraditórias, mas em sistemas legais diferentes: podemos ter dois conceitos, *Texas* e *Brasil*, representando as normas do estado norteamericano do Texas, e do Brasil, respectivamente, onde $\text{Texas} \sqsubseteq \neg \text{Brasil}$ e vice-versa, pois são sistemas legais diferentes baseados em groundnorms distintas, e não há precedência das normas de um para as do outro. Podemos ter então, as sentenças *há pena de morte* e *não há pena de morte* coexistindo, mas valendo em sistemas legais distintos, $\text{temPenaDeMorte} : \text{Texas}$ e $\text{naoTemPenaDeMorte} : \text{Brasil}$, sem problema algum.

Com as bases teóricas cobertas, introduziremos agora a sintaxe de iALC. Por ser baseada em ALC, seu alfabeto é o mesmo. Há indivíduos, conceitos e papéis, além dos mesmos operadores. A diferença está no fato de \sqsubseteq (e \equiv por consequência) ser um operador formador de conceitos. Ou seja, dados C e D conceitos, $C \sqsubseteq D$ é um novo conceito. A negação, por ser intuicionista, poderia ser definida por meio de subjuncção e \perp , $\neg C = C \sqsubseteq \perp$, mas é conveniente mantê-la. O motivo para \sqsubseteq ser formador de conceitos é que $\neg C$ deve ser um conceito, como é em ALC, mas justamente por ser definido pela subjuncção, é necessário propagar essa condição para o outro operador.

A semântica de iALC apresenta diferenças à de ALC na negação e na restrição universal, e na adição da subjuncção, além do operador $:$, com os outros operadores se mantendo os mesmos. O principal fator de mudança é a introdução da relação entre indivíduos \preceq , representando a precedência entre leis. Em ALC, por ser clássica, a semântica não leva em conta diferentes mundos, já que eles estão todos colapsados em um só. Como lidamos agora intuicionisticamente, precisamos separá-los e relacioná-los. As mudanças com os operadores foram as seguintes, tomando em conta uma interpretação \mathcal{I} :⁵

⁵Vale notar que, apesar de ser uma relação envolvendo indivíduos, R (em $\forall R.C$) não representa nossa relação de precedência \preceq , e sim um papel (do inglês *role*). A letra R é utilizada para manter consistência com nomenclatura da literatura.

$$\begin{aligned}
(\neg C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y, x \preceq y \Rightarrow y \notin C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y(x \preceq y \Rightarrow \forall z((y, z) \in R^{\mathcal{I}} \Rightarrow z \in C^{\mathcal{I}}))\} \\
(C \sqsubseteq D)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y, (x \preceq y \wedge y \in C^{\mathcal{I}}) \Rightarrow y \in D^{\mathcal{I}}\}
\end{aligned}$$

Os indivíduos y e z referenciados acima são todos pertencentes a $\Delta^{\mathcal{I}}$. Isso foi omitido para buscar limpar a notação.

A ABox é populada por asserções de indivíduos em conceitos e em papéis como em ALC, com a adição da possibilidade de termos asserções encadeadas em conceitos i.e. $a : (b : C)$. Neste caso, dizemos que a é o nominal ou indivíduo externo (esta definição será útil para compreender as provas do capítulo 4, em especial a de correção). Seguem as alterações na interpretação de fórmulas na ABox que permitem isso (vale notar que agora a interpretação deve levar um mundo como referência, que no caso chamaremos de w e que há a necessidade de preservar a hereditariedade da semântica intuicionista da lógica):

$$\begin{aligned}
w \models_{\mathcal{I}} a : C &\Leftrightarrow \forall x(a \preceq x \Rightarrow x \models_{\mathcal{I}} C) \\
w \models_{\mathcal{I}} (a, b) : R &\Leftrightarrow \forall x \forall y(a \preceq x \wedge b \preceq y \Rightarrow (x^{\mathcal{I}}, y^{\mathcal{I}}) \in R^{\mathcal{I}})
\end{aligned}$$

A interpretação de \mathcal{I} da ABox \mathcal{A} em si se mantém. Agora a TBox \mathcal{T} sofre alterações devido à nova semântica da subjunção.

$$w \models_{\mathcal{I}} C \sqsubseteq D \Leftrightarrow \forall x(w \preceq x \wedge x \in C^{\mathcal{I}} \Rightarrow x \in D^{\mathcal{I}})$$

De modo análogo à ABox, a interpretação da TBox em si se mantém.

3.3.1

Um Exemplo de Uso

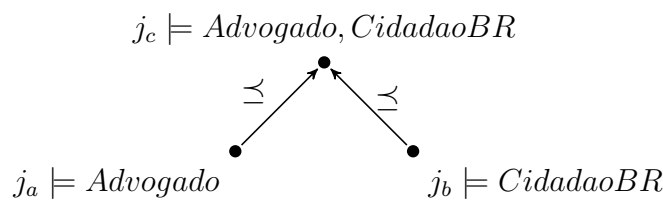
Em iALC, VLS não são proposições, mas indivíduos em uma ontologia legal, i.e., uma lei não pode ser verdadeira ou falsa, ela apenas existe ou não em um conceito. As proposições que consideramos são alguns dos *conceitos* da ontologia, i.e. conceitos da forma $a : C$ ou $C \sqsubseteq D$ ⁶ (para a um indivíduo, e C e D , conceitos), que representam os sistemas jurídicos nos quais estão presentes diferentes tipos de lei. VLSs também apresentam uma relação de precedência, derivada da hierarquia de leis individuais da jurisprudência Kelseniana, conforme visto.

⁶Vale relembrar aqui que iALC considera \sqsubseteq como formador de conceitos.

Assim, podemos criar um conceito *Advogado* no qual valem todos os VLSs representando um advogado, por exemplo. Então, os indivíduos (legais) para os quais esse conceito é válido serão interpretados semanticamente como os mundos em que vale *Advogado*. Vale notar que esses indivíduos não devem ser vistos como advogados do mundo real, sendo apenas abstrações sobre eles.

Na prática, indivíduos legais podem ser vistos como combinações específicas de leis que acabam representando um ser legal artificialmente criado. Por exemplo, podemos supor que *João Silva* passou no Exame da OAB, e é agora um advogado. Então, sabe-se que há um documento legal que diz exatamente esse fato, seja contrato assinado, publicação no Diário Oficial da União ou qualquer outro método, que varia com a legislação de cada local, e podemos representá-lo como $j_a : \text{Advogado}$ (j_a representando nossa abstração artificial sobre o fato de que o sr. Silva é um advogado). Mas, além disso, por ser um cidadão brasileiro, ele tem uma certidão de nascimento. Supondo que temos um conceito representando todo VLSs sobre cidadãos brasileiros, *CidadaoBR*, podemos também concluir que temos $j_b : \text{CidadaoBR}$ (sendo j_b uma diferente sentença legal, representando o fato de o sr. Silva ser brasileiro).

Agora as regras de precedência para indivíduos legais de iALC começam a aparecer explicitamente. Uma vez que, de acordo com [HR16], pelo aspecto intuicionista da lógica um modelo de Kripke nela é uma álgebra de Heyting, todo par de mundos apresenta um *meet* finito (digamos que seja, para j_a e j_b , j_c), relacionado via precedência de VLSs para os outros mundos, que valida ambos os conceitos *Advogado* e *CidadaoBR*. Em outras palavras, dados j_a e j_b , sempre há um mundo j_c tal que $j_a \preceq j_c$ e $j_b \preceq j_c$.



Note que não há mundo (indivíduo legal) que represente fielmente o indivíduo real *João Silva*, já que tudo aquilo com o que lidamos são leis e sentenças legais. Daí, concluímos que, conceitualmente, nunca lidamos com pessoas nem objetos do mundo real, apenas com sentenças legais sobre eles. Isso é um ponto central para a aplicação desta lógica, especialmente ao lidar, na prática, com o Exame da OAB, cujas questões tendem a contextualizar com indivíduos “reais”, e um primeiro instinto seria lidar diretamente com os indivíduos encontrados nessas questões em vez de abstrações sobre eles. No caso de cima, por exemplo, poderíamos considerar um indivíduo apenas, j , para representar o sr. Silva, o que impediria uma *construção*

dos fatos legais sobre ele tendo em mãos novas informações. No capítulo 6 damos exemplos disso.

3.4

Comparando com Lógica Deôntica

Como já mencionado, lógica deôntica enfrenta alguns problemas conceituais. Em [HPT07] são explicados dez dos problemas que ela enfrenta, sobre alguns dos quais discorreremos aqui, destacando como iALC os contorna.

A começar pelos paradoxos CTD, explicados em maior detalhe em conjunto com a solução de iALC em [HR16]. Um destes paradoxos, conhecido como paradoxo de Chisholm, ocorre devido a grupos de quatro. Primeiro, há uma norma que diz ser obrigatório determinado fato ou ação, Op . Daí, há uma segunda que indica o que deve ocorrer caso a primeira norma seja atendida, $O(p \rightarrow q)$. Mas há também uma terceira norma que considera a possibilidade de p não ser obedecido, e portando que nesse caso não se deve realizar q , $\neg p \rightarrow O(\neg q)$. Por fim, há um fato (que, para lógica deôntica é também uma fórmula, assim como as normas) alertando que p não foi realizado, $\neg p$. Da primeira e da segunda normas, conclui-se via o uso do axioma **K** e modus ponens que Oq . Da terceira e da quarta, conclui-se via modus ponens o contrário, $O(\neg q)$, gerando o paradoxo. Como o problema não se apresenta na inconsistência das normas e fatos, ele está na sua representação e uso.

O modo de evitá-los em iALC é simplesmente não vermos normas como proposições, mas como indivíduos legais, e mundos em um modelo de Kripke. Podemos criar um mundo em que não valha p , precedendo tanto um mundo em que vale $\neg p$, representando a terceira norma, quanto os demais mundos representando a primeira e a segunda, nas quais pode valer p , sem haver contradições entre as normas, pois, intuicionisticamente, $\not\models p$ é diferente de $\models \neg p$, e se consegue manter a hereditariedade da semântica da lógica. Outro ponto interessante é que não é possível derivar o axioma **K** em iALC, como mostrado em [HR16], o que impede que tanto o paradoxo de Chisolm como muitos outros baseados neste axioma não se tornem paradoxos em iALC.

Outro problema com lógica deôntica é o Dilema de Jørgensen, introduzido em [Jør38], que lida com o problema de considerar normas como proposições, ou seja, de lhes atribuir valores de verdade. Normas podem ser equiparadas a sentenças imperativas, e não faz sentido que elas tenham valores verdade. O que faz sentido é atribuir verdade à obediência de alguma norma, como, por exemplo, “*João passou na OAB e virou advogado*” para uma norma que diga “*É obrigatório passar na OAB para se tornar advogado*”. Isso permite que possamos realizar argumentos sobre as sentenças, podendo relacionar premissas e conclusões. Ao lidar com normas no

mesmo patamar de afirmações sobre normas, a noção de validade lógica não pode ser mais a usual. O dilema se dá pois se deve decidir se argumentos com normas podem ser válidos ou não. Se forem, é necessário expandir a noção de validade lógica (o que não acontece com lógica deôntica, que mantém a noção de validade usual). Se não puderem ser válidos, então não faz sentido realizar argumentos sobre normas. iALC contorna isso pois não atribui valores-verdade às normas, mas sim as torna indivíduos.⁷

Outro paradoxo nesta mesma linha é o paradoxo de Ross, introduzido em [Ros41], que lida com inferência sobre sentenças imperativas. Em sentenças declarativas, é completamente válida uma inferência do tipo “*O céu é azul. Portanto, o céu é azul ou as paredes são brancas.*”. Inclusive, em dedução natural recebe o nome de \vee -introdução, pois introduz o operador \vee . Ao ser aplicada a normas, a inferência pode perder significado, e.g. “*É obrigatório que a carta seja enviada. Portanto, é obrigatório que a carta seja enviada ou queimada.*”. Nesse caso, o cumprimento da segunda parte da disjunção implica no não cumprimento da primeira, e faz a norma original perder o sentido (acaba, de certo modo, introduzindo algo como $O\phi \rightarrow O(\phi \vee \neg\phi)$). Isso nos faz questionar o que é uma inferência válida, pois estruturalmente a premissa leva à conclusão, mas seu significado nos diz o contrário. iALC contorna esse paradoxo, novamente, pois normas não são proposições, e não precisamos nos preocupar com inferências diretamente sobre normas.

De modo geral, demais problemas apresentam como cerne o fato de normas serem proposições em lógica deôntica, assim como os aqui apresentados, e isso é contornado em iALC devido a ser lógica de descrição, conforme já mencionado.

⁷Esse dilema inclusive dá uma nova luz à lógica deôntica: em vez de lidar com normas, ela lidaria com proposições sobre normas. Isso, claro, requer que mudanças ocorram nas bases dessa lógica.

4

Provas de Correção e Completude

Uma lógica por si só pode modelar algum problema ou domínio, mas podemos também gerar raciocínio sobre ele, por meio de regras que formam um cálculo lógico, que consiste em um conjunto de regras que operam sobre as fórmulas se encadeando com outras aplicações das mesmas regras para gerar teoremas com estruturas mais complexas.

Há várias preocupações quando se cria um cálculo lógico, especialmente pois lógica acaba sendo a base de verificação ou modelagem de diversos domínios. É necessário garantir que os formalismos realizados sejam consistentes e confiáveis para a modelagem esperada, e há determinadas propriedades por meio das quais isso é garantido. Há, por exemplo, a *decidibilidade*, que se refere à possibilidade de o sistema lógico apresentar métodos para decidir se uma fórmula qualquer nele pertence à sua teoria.

Em [HR16] estão descritos um cálculo de sequentes para iALC baseado no de [HPR10], e para o qual são apresentadas provas de correção (*soundness*) e completude (*completeness*). Um sistema é dito *correto* quando toda fórmula que pode ser provada no sistema é válida na lógica (ou seja, faz parte da teoria), e é dito *completo* quando tudo aquilo que é válido pode ser provado. Com isso, temos que provas realizadas com o cálculo de sequentes apresentado para iALC conseguem gerar toda a teoria da lógica.

Implementar no provador de teoremas $L\exists\forall N$ as provas que foram feitas tem alguns objetivos, o primeiro sendo apresentar uma garantia computacional das provas, como forma de corroborar os argumentos dos artigos mencionados acima. O segundo objetivo é servir de base para realizar provas em iALC utilizando esse cálculo de sequentes. Por fim, há o terceiro objetivo de ser um passo que nos aproxima da implementação de um sistema de dedução natural para iALC, o que se aproxima mais do seu uso prático. Há exemplos de árvores de dedução feitas no capítulo [6].

Começaremos por explicar como funciona o provador de teoremas utilizado para a implementação das provas, e em seguida explicaremos como elas foram realizadas, destacando os pontos mais relevantes. A implementação das provas se encontra em <https://github.com/Bpalkmim/iALC>, dividida nos seguintes

arquivos:

- `basics.lean` contém a definição dos tipos básicos utilizados pelos demais arquivos.
- `list.lean` define o tipo `list`, um tipo indutivo utilizado para representar conjuntos de fórmulas para o cálculo de seqüentes.
- `axioms.lean`, `rightRules.lean`, `leftRules.lean`, `structRules.lean` e `promRules.lean` contêm as definições das regras do cálculo de seqüentes, divididas por categoria i.e. `axioms.lean` apresenta os axiomas de identidade e o *ex falso quodlibet*, `rightRules.lean`, as regras da parte direita do seqüente, e assim por diante.
- `completenessProof.lean` contém as derivações dos 5 axiomas do cálculo de Hilbert para iALC utilizando as regras do cálculo de seqüentes.
- `soundnessProof.lean` é formado pelas provas de corretude de cada regra do cálculo de seqüentes de iALC.

O conteúdo de cada arquivo será melhor explicado mais à frente.

4.1

LEAN

Criado por Leonardo Moura (Microsoft Research) em 2013, *LEAN*¹ é um provador de teoremas/linguagem de programação puramente funcional e *open source*, que continua até hoje sendo primariamente desenvolvido pela Microsoft Research. Ele se propõe a ser uma ponte entre ITPs (provadores interativos de teoremas) e ATPs (provadores automatizados de teoremas), por incentivar interação com usuário ao mesmo tempo em que permite que sejam criadas provas completamente automatizadas, especialmente por meio das *táticas* que implementa. É bastante versátil e compete páreo a páreo com outros grandes provadores populares, como por exemplo o Coq², que não possui a irrelevância de provas³ que *LEAN* possui.

LEAN se baseia em uma versão de teoria dependente dos tipos com grande expressividade chamada *Cálculo de Construções* com tipos indutivos. Ele utiliza vários métodos e técnicas para realizar provas de teoremas em teoria dependente dos tipos, como reescrita e simplificação automática de termos, além de inferir tipos de elementos que não foram explicitamente tipados. Em teoria dos tipos, tudo apresenta um tipo e tipos podem ficar mais complexos, i.e. podemos ter

¹<https://leanprover.github.io>

²<https://coq.inria.fr/>

³Irrelevância de provas ocorre quando podemos afirmar que quaisquer duas provas da mesma proposição (ou seja, do mesmo tipo) são iguais.

um tipo A e um tipo B e a partir deles criar o tipo $A \rightarrow B$ que representa as funções de A em B . Em teoria dos tipos dependentes, tipos podem depender de parâmetros, aumentando a complexidade de teoria dos tipos simples e nos ajudando a representar passos dedutivos em função de tipos. Podemos ter, por exemplo, um tipo $ListaA$ para listas genéricas, que depende de um tipo qualquer A . Em definição de funções, essa dependência de tipos é denotada por Π , mantendo o padrão da literatura (esses tipos de parâmetro são chamados de Π -tipos, como visto em [Mar84]).

Um ponto importante para o trabalho aqui realizado é ver proposições como tipos. Com isso, podemos realizar provas genéricas (como as de correção e completude que precisam de elementos genéricos da lógica) utilizando os tipos dependentes para as hipóteses que assumimos. Com isso em mente, podemos considerar as provas como tipos funcionais que levam os tipos das premissas ao tipo da conclusão, via os passos dedutivos, que por sua vez podem ser vistos como provas simples, cujo encadeamento contribui para a tipagem geral da prova. De modo geral, se uma prova *tipa* corretamente, ela está correta.

Além disso, num ponto de vista mais prático, $L\lambda E\lambda N$ apresenta alguns meios para facilitar a geração de provas mais complexas. Um deles é o termo *sorry* que produz uma prova de qualquer coisa que se queira, e se mostra extremamente útil para a criação procedural da prova. Outro é a possibilidade de criarmos tipos indutivos por meio do termo *inductive*, que neste trabalho foi utilizado para levar os *contextos* do cálculo de sequentes em listas recursivas. Mais à frente, conforme necessário, mais desses mecanismos serão explicados.

4.2

Um cálculo de sequentes para iALC

Em [HR16], conforme dito anteriormente, foi apresentado um cálculo de Hilbert e um cálculo de sequentes para iALC. Focaremos aqui no de sequentes, para o qual foram apresentadas provas de correção e completude. Vale notar que o cálculo de sequentes foi apresentado para raciocínio em TBox, ou seja, sem a presença de *nominals* (ou indivíduos) em todos os conceitos apresentados. Para a apresentação do cálculo de sequentes, acrescentamos as regras estruturais e o corte que foram omitidos no artigo por serem análogos aos do cálculo de sequentes para lógica proposicional. A fim de manter consistência com o artigo, a notação utilizada é a seguinte: δ , δ_1 e δ_2 representam fórmulas, i.e. um conceito (como C , $C \sqcap D$, entre outros) ou uma asserção ($x : C$ ou xRy), α e β são conceitos, R é um papel, e Δ é um conjunto de fórmulas. Nas regras de promoção, a notação $\forall R.\Delta$ representa que todos os conceitos em Δ estão sendo restritos universalmente pelo mesmo papel, mantendo as asserções em Δ inalteradas (no caso de $\exists R.\Delta$,

a restrição é existencial). De modo análogo, $x : \Delta$ indica que o nominal x é adicionado apenas aos conceitos em Δ . Um sequente é indicado pelo operador \Rightarrow .

$$\begin{array}{ll}
\frac{}{\Delta, \delta \Rightarrow \delta} Ax - id & \frac{}{\Delta, x : \perp \Rightarrow \delta} Ax - eq \\
\\
\frac{\Delta, xRy \Rightarrow y : \alpha}{\Delta \Rightarrow x : \forall R.\alpha} \forall - r & \frac{\Delta, x : \forall R.\alpha, y : \alpha, xRy \Rightarrow \delta}{\Delta, x : \forall R.\alpha, xRy \Rightarrow \delta} \forall - l \\
\\
\frac{\Delta \Rightarrow xRy \quad \Delta \Rightarrow y : \alpha}{\Delta \Rightarrow x : \exists R.\alpha} \exists - r & \frac{\Delta, xRy, y : \alpha \Rightarrow \delta}{\Delta, x : \exists R.\alpha \Rightarrow \delta} \exists - l \\
\\
\frac{\Delta, \alpha \Rightarrow \beta}{\Delta \Rightarrow \alpha \sqsubseteq \beta} \sqsubseteq - r & \frac{\Delta_1 \Rightarrow \alpha \quad \Delta_2, \beta \Rightarrow \delta}{\Delta_1, \Delta_2, \alpha \sqsubseteq \beta \Rightarrow \delta} \sqsubseteq - l \\
\\
\frac{\Delta \Rightarrow \alpha \quad \Delta \Rightarrow \beta}{\Delta \Rightarrow \alpha \sqcap \beta} \sqcap - r & \frac{\Delta, \alpha, \beta \Rightarrow \delta}{\Delta, \alpha \sqcap \beta \Rightarrow \delta} \sqcap - l \\
\\
\frac{\Delta \Rightarrow \alpha}{\Delta \Rightarrow \alpha \sqcup \beta} \sqcup_1 - r & \frac{\Delta, \alpha \Rightarrow \delta \quad \Delta, \beta \Rightarrow \delta}{\Delta, \alpha \sqcup \beta \Rightarrow \delta} \sqcup - l \\
\\
\frac{\Delta \Rightarrow \beta}{\Delta \Rightarrow \alpha \sqcup \beta} \sqcup_2 - r & \frac{\Delta_1 \Rightarrow \delta_1 \quad \Delta_2, \delta_1 \Rightarrow \delta}{\Delta_1, \Delta_2 \Rightarrow \delta} cut \\
\\
\frac{\Delta, \alpha \Rightarrow \beta}{\forall R.\Delta, \exists R.\alpha \Rightarrow \exists R.\beta} p - \exists & \frac{\Delta \Rightarrow \delta}{\Delta, \delta_1 \Rightarrow \delta} weakening \\
\\
\frac{\Delta \Rightarrow \alpha}{\forall R.\Delta \Rightarrow \forall R.\alpha} p - \forall & \frac{\Delta, \delta_1, \delta_1 \Rightarrow \delta}{\Delta, \delta_1 \Rightarrow \delta} contraction \\
\\
\frac{\Delta \Rightarrow \delta}{x : \Delta \Rightarrow x : \delta} p - N & \frac{\Delta_1, \delta_1, \delta_2, \Delta_2 \Rightarrow \delta}{\Delta_1, \delta_2, \delta_1, \Delta_2 \Rightarrow \delta} permutation
\end{array}$$

Segue versão das regras com *nominals* (focando no *nominal* x):

$$\begin{array}{ll}
\frac{}{\Delta, x : \delta \Rightarrow x : \delta} Ax - id - n & \frac{}{\Delta, y : \perp \Rightarrow x : \delta} Ax - eq - n \\
\\
\frac{\Delta, x : \alpha \Rightarrow x : \beta}{\Delta \Rightarrow x : (\alpha \sqsubseteq \beta)} \sqsubseteq - r - n & \frac{\Delta_1 \Rightarrow x : \alpha \quad \Delta_2, x : \beta \Rightarrow \delta}{\Delta_1, \Delta_2, x : (\alpha \sqsubseteq \beta) \Rightarrow \delta} \sqsubseteq - l - n \\
\\
\frac{\Delta \Rightarrow x : \alpha \quad \Delta \Rightarrow x : \beta}{\Delta \Rightarrow x : (\alpha \sqcap \beta)} \sqcap - r - n & \frac{\Delta, x : \alpha, x : \beta \Rightarrow \delta}{\Delta, x : (\alpha \sqcap \beta) \Rightarrow \delta} \sqcap - l - n
\end{array}$$

$$\frac{\Delta \Rightarrow x : \alpha}{\Delta \Rightarrow x : (\alpha \sqcup \beta)} \sqcup_1 - r - n \qquad \frac{\Delta, x : \alpha \Rightarrow \delta \quad \Delta, x : \beta \Rightarrow \delta}{\Delta, x : (\alpha \sqcup \beta) \Rightarrow \delta} \sqcup - l - n$$

$$\frac{\Delta \Rightarrow x : \beta}{\Delta \Rightarrow x : (\alpha \sqcup \beta)} \sqcup_2 - r - n$$

Aqui ficam excluídas as regras de promoção, estruturais, e \forall -l, \forall -r, \exists -l, e \exists -r, por apresentarem somente uma única versão, já mostrada anteriormente, que apresenta *nominals* naturalmente. As regras estruturais e de promoção fazem menção a qualquer tipo de fórmula, portanto não é necessário criar versões diferentes.

Vale notar que, como a definição dos sequentes aqui apresentam apenas uma fórmula do lado direito, as regras estruturais só valem para o lado esquerdo. Não faz sentido enfraquecer, contrair ou permutar o lado direito quando há apenas uma fórmula. O corte também está adaptado para esse fato. Outro ponto importante é considerar que y deve ser livre tanto para $\forall - r$ quanto para $\exists - l$.

Para implementar esse cálculo em $L\exists\forall N$ foi necessário ver como implementar cada componente: conceitos, papéis, asserções, indivíduos, sequentes, e conjuntos de fórmulas. Conceitos e asserções entram como fórmulas, e em suas diversas formas, sejam disjunções, restrições universais, asserções de conceitos etc. todas entram como construtores do tipo indutivo `Formula`. Considerar todos como esse tipo deixa a notação menos convoluta e permite que tratemos do map (necessário para as sobrecargas de operadores nas regras de promoção) via pattern matching nos diferentes casos do tipo indutivo `Formula`. Indivíduos, ou *nominals*, apresentam o tipo genérico `Type`, pois não são fórmulas em lógica de descrição, assim como papéis e conceitos (para um conceito simples virar uma fórmula, realizamos um *lift* via o construtor `simple`). Conjuntos de fórmulas são representados pelo tipo indutivo `list` como `list Formula`. Sequentes são funções que levam uma lista de fórmulas (lado esquerdo do sequente) e uma fórmula (lado direito) a uma proposição i.e. `list Formula → Formula → Prop`. Por fim, provas levam sequentes (`Prop`) a um tipo: `Prop → Type`. As definições dos tipos no arquivo `basics.lean` são como segue:

```
constant Concept : Type
constant Nominal : Type
constant Role : Type

constant Top : Concept
```

```

constant Bot : Concept

inductive Formula
| simple : Concept -> Formula
| univ : Role -> Formula -> Formula
| exis : Role -> Formula -> Formula
| subj : Formula -> Formula -> Formula
| conj : Formula -> Formula -> Formula
| disj : Formula -> Formula -> Formula
| neg : Formula -> Formula
| elemOf : Nominal -> Formula -> Formula
| relation : Role -> Nominal -> Nominal -> Formula

constant Sequent : list Formula -> Formula -> Prop
constant Proof : Prop -> Type

```

Acima, no tipo indutivo `Formula`, apenas `elemOf` (representando $x : C$) e `relation` (representando xRy) são asserções, enquanto os demais representam conceitos vistos como fórmulas. As restrições universal e existencial permitem que o conceito utilizado não seja simples, por isso foi utilizado `Formula` nas premissas. O mesmo vale para a subjunção e a negação.

Com esses tipos básicos, podemos formar as regras, que levam provas já existentes (premissas) em uma prova nova (conclusão). Segue um exemplo com a regra $\sqcap - r$:

```

constant conj_r {Delta : list Formula} {a b : Concept} :
  Proof (Sequent Delta (Formula.simple a)) ->
  Proof (Sequent Delta (Formula.simple b)) ->
  Proof (Sequent Delta (Formula.conj a b))

```

Essa regra depende de uma lista de fórmulas e dois conceitos, e leva provas de dois sequentes com conceitos simples do lado direito a uma terceira prova, com a conjunção deles no lado direito, mantendo a lista de fórmulas do lado esquerdo do sequente (que deve ser a mesma para todos). Note como os conceitos simples foram elevados à condição de fórmula via o construtor `simple` do tipo indutivo `Formula`.

Com este formato, podemos encadear as regras do cálculo de sequentes a fim de formar provas mais complexas. Como dito anteriormente, as regras estão implementadas nos arquivos `axioms.lean`, `rightRules.lean`, `leftRules.lean`, `structRules.lean` e `promRules.lean`, separadas por categoria: axiomas de identidade e *ex falso quodlibet* estão no primeiro arquivo, regras para a parte

direita do sequente no segundo, regras para o lado esquerdo no terceiro, regras estruturais no quarto, e regras de promoção no quinto.

Para realizar as regras de promoção foi necessário utilizar o map de listas para distribuir a operação desejada para todos os conceitos de dentro da lista, seja ela restrição universal, existencial, ou o uso de *nominals*. Um ponto importante é que é necessário manter asserções do mesmo modo, então a função a ser mapeada deve ser separada em casos. Por isso, o tipo `Formula` foi definido indutivamente em `basics.lean`, o que facilita o pattern-matching para as funções que são utilizadas por map. Segue o exemplo da função passada para a adição de um nominal:

```
def add_nom (x : Nominal) : Formula -> Formula
| (Formula.simple c)      := Formula.elemOf x (Formula.simple c)
| (Formula.univ r f)      := Formula.elemOf x (Formula.univ r f)
| (Formula.exis r f)      := Formula.elemOf x (Formula.exis r f)
| (Formula.subj c d)      := Formula.elemOf x (Formula.subj c d)
| (Formula.conj c d)      := Formula.elemOf x (Formula.conj c d)
| (Formula.disj c d)      := Formula.elemOf x (Formula.disj c d)
| (Formula.neg c)         := Formula.elemOf x (Formula.neg c)
| (Formula.elemOf y c)    := Formula.elemOf y c
| (Formula.relation rel y z) := Formula.relation rel y z
```

Essa função, dado um nominal x , o adiciona a conceitos, mantendo asserções inalteradas. As funções para as restrições universal e existencial apresentam estrutura parecida, só que em vez de um indivíduo x , a elas é dada um papel r . As regras que fazem uso dessas funções se encontram em `promRules.lean` e são justamente as regras de promoção, ou seja, que *promovem* uma operação de fórmula para um conjunto de fórmulas (no caso, se limitando aos conceitos, mantendo as asserções inalteradas). Segue o exemplo da regra de promoção de *nominals*, $p - N$:

```
constant prom_n {Delta : list Formula} {X : Nominal} {delta :
  Formula} :
  Proof (Sequent Delta delta) ->
    Proof (Sequent (list.map (add_nom X) Delta)
      (Formula.elemOf X delta))
```

4.3

Prova de Completude

Primeiro vamos definir completude para o cálculo de sequentes de iALC:

Definição 4.1 *Completude* Seja Θ um conjunto de fórmulas e Γ um conjunto de todos os conceitos, e δ uma fórmula. O sistema é dito completo quando $\Theta, \Gamma \models \delta$ implica em $\Theta \Rightarrow \delta$.

Aqui, a prova de completude se refere apenas ao raciocínio de TBox. A prova para raciocínio de ABox é realizada utilizando o método de modelos canônicos. Para o raciocínio de TBox, como a presença de Γ torna a prova trivial via $Ax - id$, consideramos $\Gamma = \emptyset$.

Para provar que o cálculo de sequentes apresentado é completo, vamos primeiro voltar ao cálculo de Hilbert apresentado em [HR16]. Um cálculo de Hilbert depende de uma base axiomática, a partir da qual são feitas as provas de novos teoremas. Temos também que essa base implementa o raciocínio de TBox, ou seja, decide a relação semântica $\Theta, \emptyset \models C$, com Θ um conjunto de fórmulas e C um conceito puro qualquer. A base para iALC é composta pelos seguintes axiomas (onde IPL é lógica intuicionista proposicional, do inglês *intuitionistic propositional logic*):

0. Todas as instâncias de substituição de teoremas de IPL

$$1. \forall R. (C \sqsubseteq D) \sqsubseteq (\forall R.C \sqsubseteq \forall R.D)$$

$$2. \exists R. (C \sqsubseteq D) \sqsubseteq (\exists R.C \sqsubseteq \exists R.D)$$

$$3. \exists R. (C \sqcup D) \sqsubseteq (\exists R.C \sqcup \exists R.D)$$

$$4. \exists R. \perp \sqsubseteq \perp$$

$$5. (\exists R.C \sqsubseteq \forall R.D) \sqsubseteq \forall R. (C \sqsubseteq D)$$

MP Se C e $C \sqsubseteq D$ são teoremas, então D também é teorema.

Nec Se C é teorema, então $\forall R.C$ também é teorema.

Em 0 estão adaptações dos teoremas de IPL para iALC, por ser de descrição, e não meramente proposicional. De 1 a 5 há os novos axiomas para iALC. *MP* é o *modus ponens*, e *Nec*, o axioma de necessitação.

Podemos utilizar estes axiomas para representar a completude do cálculo de sequentes para iALC considerando que eles formam uma base axiomática para raciocínio de TBox para toda a lógica. A prova dos teoremas em 0 é trivial pois basta utilizar regras proposicionais para fórmulas de iALC. *MP* é derivável das regras de iALC ao utilizar a regra do corte (*cut*). *Nec* é um caso especial da regra $p - \forall$. Assim, basta derivar os axiomas de 1 a 5 com as regras de iALC.

Implementar as regras de completude é relativamente simples, envolvendo realizar provas dos axiomas utilizando as regras de iALC vistas anteriormente. Em *L $\exists\forall N$* , como visto anteriormente, tais provas bastam ter os tipos coerentes, das premissas à conclusão. Como são 5 axiomas, foram implementadas 5 provas em *L $\exists\forall N$* , que se encontram no arquivo `completenessProof.lean` no repositório do github.

Um exemplo de prova é a do axioma 4: $\exists R.\perp \sqsubseteq \perp$. Um ponto importante dele é que ele usa uma versão de $\sqsubseteq -r$ que usa *nominals*. Em [HR16] é dito explicitamente que cada regra do cálculo de sequentes apresenta também uma versão com *nominals*, que também foram todas implementadas, especialmente para as provas de correção mais à frente. Com isso, foi necessário implementá-la como segue, assim como a versão com *nominals* para as demais regras:

```
constant subj_r_n {Delta : list Formula} {X : Nominal} {a b :
  Formula} :
  Proof (Sequent (Formula.elemOf X a :: Delta) (Formula.elemOf X
    b)) ->
    Proof (Sequent Delta (Formula.elemOf X (Formula.subj a b)))
```

A prova do axioma 4, conforme está no artigo, é como segue:

$$\frac{\frac{xRy, y : \perp \Rightarrow x : \perp}{x : \exists R.\perp \Rightarrow x : \perp} \text{ax-efq}}{\Rightarrow x : (\exists R.\perp \sqsubseteq \perp)} \frac{\exists\text{-I}}{\sqsubseteq -r\text{-n}}$$

A implementação é, portanto, uma aplicação de $\sqsubseteq -r\text{-n}$ sobre uma aplicação de $\exists\text{-I}$ sobre um sequente inicial que tem sua prova dada via formação de $ax - efq$. A prova se dá via comparação do tipo do encadeamento dessas regras sobre o sequente inicial com o tipo que esperamos ter, que no caso é uma prova de um sequente com o lado esquerdo $\Delta = \emptyset$ (a lista nil) e o lado direito com a fórmula $x : (\exists R.\perp \sqsubseteq \perp)$. Os tipos das expressões são checados pelo provador via `#check`. Segue como foi implementada esta Prova:

```
constants x y : Nominal
constant r : Role

#print "Axioma 4"
#check subj_r_n
  (exists_l
    (axiom_efq
      (Formula.relation r x y :: nil)
      (Formula.elemOf x (Formula.simple Bot)) y))
```

Essa prova tem como saída o seguinte tipo:

```
Proof
  (Sequent
    nil
    (Formula.elemOf x
      (Formula.subj
        (Formula.exis r (Formula.simple Bot))
```

(Formula.simple Bot))))

As provas para os axiomas 1, 2, 3 e 5 seguem esse mesmo padrão, e estão todas em `completenessProofs.lean`.

4.4

Prova de Correção

Um sistema é dito correto quando toda fórmula que pode ser provada é válida na lógica (ou seja, faz parte da teoria). Em outras palavras:

Definição 4.2 *Correção* Seja Θ um conjunto de fórmulas e Γ um conjunto de conceitos, δ uma fórmula, e C um conceito puro. O sistema é dito correto quando, se $\Theta, \Gamma \Rightarrow \delta$ é provável, então $\Theta, \Gamma \models C$.

A prova de [HR16] demonstra que cada regra do cálculo de sequentes preserva a validade do sequente, i.e. sequentes válidos levam a outros sequentes válidos via as regras a apresentadas. Para isso, é necessário definir o que é um sequente válido.

Definição 4.3 *Sequente válido* Seja \mathcal{I} uma interpretação, Θ um conjunto de fórmulas e Γ um conjunto de conceitos, δ uma fórmula, e C um conceito puro. Sejam também \vec{z} um vetor de indivíduos e $Nom(\Gamma, \delta)$ o vetor com todos os nominals mais externos ocorrendo em cada asserção em $\Gamma \cup \{\delta\}$ (lembrando que C não contém nominal externo). Vale lembrar que nossa notação de sequente utiliza o operador \Rightarrow . Dizemos que $\Theta, \Gamma \models \delta$ quando:

$$\forall \mathcal{I}, \text{ se } (\forall x \in \Delta^{\mathcal{I}}, (\mathcal{I}, x \models \Theta)) , \text{ então } \\ \forall Nom(\Gamma, \delta) (\forall \vec{z} (Nom(\Gamma, \delta) \preceq \vec{z} \text{ implica em } \mathcal{I}, \vec{z} \models \Gamma \Rightarrow \mathcal{I}, \vec{z} \models \delta))$$

Nesta notação, o operador \preceq está sobrecarregado para dois vetores de indivíduos, e $Nom(\Gamma, \delta) \preceq \vec{z}$ significa que $\forall x \in Nom(\Gamma, \delta) \forall y \in \vec{z}, x \preceq y$.

Essa definição diz que, se conseguirmos em qualquer interpretação levar a validade de Θ para a validade de um sequente envolvendo todos os *nominals* que são precedidos pelos *nominals* externos de Γ e δ , então podemos concluir que $\Theta, \Gamma \Rightarrow \delta$ é um sequente válido. Em outras palavras, um sequente é válido quando a validade dos elementos de Θ leva, sob qualquer interpretação, à validade dos conceitos em Γ , implicando via a relação de precedência à validade de δ . O uso da relação de precedência está intimamente ligado aos indivíduos externos de asserções do tipo $x : C$, pois precisa-se garantir que todos os indivíduos que x precede precisem validar C .

Com a definição de um sequente válido em mãos, basta então demonstrar que cada regra do cálculo de sequentes preserva a validade dos sequentes. Vale notar que para essa prova é necessário utilizar os *nominals* nas provas envolvendo subjunção, conjunção e disjunção, pois a versão sem *nominals* já é correta de acordo com semântica de Kripke para IPL (como visto em [HR16]).

Daí seguem algumas definições de tipos necessários para a estruturação da prova: $Nom(\Gamma, \delta)$ é uma função que recebe uma lista de fórmulas, e retorna uma lista com os *nominals* externos de todas elas (que podem ser zero, dando numa lista vazia). \vec{z} é também uma lista de *nominals*. O operador \models é, dado, $\mathcal{I}, w \models \delta$, um tipo funcional que leva uma interpretação (lista de fórmulas), um nominal e uma fórmula a uma proposição (Prop, para servir de entrada para nosso tipo Proof). Há também uma versão de \models utilizando uma lista de *nominals* em vez de um nominal simples, para o caso de $I, \vec{z} \models \delta$, implementada por meio de uma função que leva para a conjunção dos \models que utilizam apenas um nominal, ou seja, leva para $\bigwedge_{z \in \vec{z}} I, z \models \delta$. Estes novos tipos e funções estão implementados em `basics.lean` como segue:

```
constant Interpretation : list Formula [U+FFFD] Prop
constant Model : Interpretation [U+FFFD] Nominal [U+FFFD] list
  Formula [U+FFFD] Prop

-- Funcao que mapeia a validade de um sequente a partir de uma
-- lista de \textit{nominals}
-- para a validade de um sequente a partir da conjuncao das
-- validades para cada
-- nominal
def model_list : Interpretation -> list Nominal -> list Formula
  -> Prop
| I l nil      := ff
| I nil Delta  := tt
| I (x :: l) Delta := (Model I x Delta) ^ (model_list I l Delta)

-- Funcao que busca por uma relacao entre dois nominals numa
-- lista de formulas
def rel_in_formula_list : Role -> Nominal -> Nominal -> list
  Formula -> Prop
| r x y nil      := ff
| r x y ((Formula.relation rel x y) :: l) := tt
| r x y (_ :: l) := in_formula_list r x y l

-- Funcao que busca numa lista de formulas pelas relacoes entre
```



```

-- o nominal da esquerda e os da lista a direita
def prec_general : Role -> Nominal -> list Nominal -> list
  Formula -> Prop
| r x nil Delta      := ff
| r x (y :: l) Delta := (rel_in_formula_list r x y Delta) ^
  prec_general r x l Delta

-- Funcao auxiliar para obter os nominals externos de uma lista
  de formulas
def nom_ext : list Formula -> list Nominal
| nil          := nil
| ((Formula.elemOf y c) :: l) := y :: (nom_ext l)
| (_ :: l)      := nom_ext l

```

Assim, a definição de sequente válido foi implementada do seguinte modo:

```

def valid_seq {prec : Role} {Theta Gamma : list Formula} {Delta :
  Formula} :
Proof (Sequent (Theta ++ Gamma) delta) ->
\forallall I : Interpretation, ((\forallall x : Nominal, Model I x
  Theta) ->
  (\forallall x : nom_ext (Delta :: Gamma),
    (prec_general prec x (nom_ext (Delta :: Gamma)) (Delta ::
      Gamma) ->
      (model_list I (nom_ext (Delta :: Gamma)) Gamma ->
        model_list I (nom_ext (Delta :: Gamma)) (Delta ::
          nil)))
    )
  )
)

```

Ela recebe a prova de um sequente e mostra sua validade. O papel *prec* representa o operador de precedência entre *nominals*, \preceq .

As provas são, portanto, chegar na validade do sequente da conclusão partindo da validade dos sequentes das premissas. Elas estão implementadas no arquivo `soundnessProof.lean` no repositório do github. Um exemplo de prova, a da regra \sqsubseteq -r-n é como segue:

```

constants A B : Formula
constant X : Nominal

#check valid_seq (subj_r_n (Proof (Sequent ((Formula.elemOf X A)
  :: Delta) B)))

```

Vale notar que grande parte da complexidade dessa prova se encontra na definição de o que é um sequente válido, pois ali se encontram todas as relações a serem realizadas entre os diferentes *nominals* envolvidos.

5

Leis Brasileiras

Existem diferentes códigos legais que agem sobre diferentes regiões geográficas. Cada um deles apresenta uma base teórica relacionada com a situação da sociedade em que ele está inserido desde a sua criação e no decorrer de seu desenvolvimento, além de ter diferentes influências da estrutura do texto e na hierarquia entre leis individuais.

As leis brasileiras são organizadas tendo como base a Constituição Brasileira de 1988, que teve grande influência da Constituição Portuguesa de 1976, em especial com relação às doutrinas do direito que a regem.^[1] Há outras influências, porém, como a Lei Fundamental alemã, da qual foi trazido o primado da dignidade da Pessoa Humana, e as Reformas de Base de João Goulart em 1964^[2] que alteraram a função social da propriedade (entre outras características), por exemplo.

Além da Constituição Federal, há vários outros conjuntos de leis (que a ela estão submetidos, por ser o equivalente na legislação brasileira às *groundnorms* da Jurisprudência Kelseniana), como as Constituições Estaduais, o Código Civil, o Código de Defesa do Consumidor, o Estatuto do Idoso, entre outros.

5.1

Organização e Estrutura

Na legislação brasileira, como em muitas outras, há diferentes categorias de leis, de importância e aplicabilidade variada. Caso todas as leis tivessem a mesma importância, seria extremamente difícil legislar de modo a evitar as ditas *antinomias reais*, pois seria mais fácil de uma lei poder contradizer alguma outra com a qual, a princípio, não teria alguma relação direta, e manter em mente todas as possíveis ramificações de todas elas quando expostas umas às outras é inviável.

Conforme descrito por Bobbio em [BS95], há 3 principais modos de solucionar antinomias legais: hierarquicamente (leis fundamentais têm prioridade - *lex superior derogat legi inferiori*), cronologicamente (leis posteriores têm prioridade - *lex posterior derogat legi priori*), ou via especialidade (leis mais específicas apresentam

¹<https://www.conjur.com.br/2016-abr-07/constituicao-portuguesa-chega-ao-s-40-anos-ideologica>

²<https://estado.rs.gov.br/jango-foi-pioneiro-da-transformacao-social-do-pais-afirma-tarso-em-sao-borja>

prioridade - *lex specialis derogat legi generali*). Isso garante manutenção de precedência entre leis estruturalmente, i.e. via Ordenamento Jurídico, o que diminui a chance de ocorrerem antinomias reais, além de facilitar sua identificação, pois a maioria das antinomias fica caracterizada como *aparente*, ou seja, solucionável pelos três critérios citados.

Esses métodos de solução de antinomias geram estruturação e prioridade nos textos jurídicos. Criam-se, portanto camadas hierárquicas de leis, que atuam sobre os mais diversos assuntos e acabam por seguir uma *pré-ordem*. Os grandes grupos de leis brasileiras respeitam uma determinada hierarquia:

1. Leis fundamentais

As mais importantes, que devem preceder todas as demais leis. Aqui se encontram a Constituição Federal, as constituições estaduais, Leis complementares, e Emendas constitucionais, entre outras. Como há vários grupos de leis com mesma hierarquia, é possível pensar que não ocorra realmente superioridade da Constituição Federal ante as demais leis fundamentais, mas isso não necessariamente ocorre.

Como a Constituição Federal é o conjunto principal de leis, mesmo que os outros códigos desta categoria tenham a mesma hierarquia que ela, é necessário que eles não gerem antinomias. No caso das constituições estaduais isso se dá por meio de maior especificidade do que a federal. Já as leis complementares e emendas constitucionais são acréscimos ou alterações realizados diretamente sobre a constituição, então não há ocorrência de antinomias pois a constituição é efetivamente modificada.

Nas demais camadas hierárquicas, são mantidos os critérios de resolução de antinomias. Vale notar que de modo geral não há antinomia entre leis, dado que elas são criadas justamente com esses critérios em mente, e a necessidade de solução de antinomias ditas *aparentes* via os 3 critérios mencionados anteriormente é rara, tornando mais pontuais ainda as antinomias *reais*, que realmente indicam incoerências nos textos jurídicos.

2. Supralegais

As leis supralegais são formadas por tratados internacionais, hierarquicamente inferiores apenas às leis fundamentais do Brasil. Justamente por envolverem direito internacional e relações com demais códigos legais, que não apresentam necessariamente precedência com as leis brasileiras, é necessário garantir a manutenção das leis fundamentais brasileiras, mas sem permitir a influência de leis específicas demais ao Brasil, de modo respeitar a soberania alheia.

3. Ordinárias

Aqui se encontra a maioria das leis brasileiras, além de resoluções, medidas provisórias, e tratados internacionais de menor impacto ou importância, entre outros. Aqui se encontra a Lei 8906, formalizada parcialmente no Apêndice **A** e apresentada a seguir.

4. Proposições

Nesta categoria ficam as propostas e projetos de leis, como os projetos de emenda à constituição (PECs), projetos de decreto legislativo, projetos de lei complementar etc. Ao serem aprovados pelas autoridades responsáveis, tornam-se leis das categorias em que devem estar.

5. Regulamentadoras

As leis regulamentadoras são formadas pelos regimentos internos, decretos, instruções normativas, e portarias.

6. Infralegais coletivas

Referem-se aos acordos coletivos de trabalho e às convenções coletivas de trabalho.

7. Infralegais privadas

Aqui estão localizados os contratos, representando a menor camada hierárquica (e a mais volumosa) das leis brasileiras.

Cada lei em si pode estar seccionada por títulos a fim de organização dos temas que ela aborda, mas isso não é necessário. Um título pode estar dividido em capítulos, e um capítulo por seções. Independentemente das *quebras* que ocorrem, uma lei é composta por diferentes artigos.

Um artigo pode apresentar um *caput* i.e. o texto ou a parte mais importante do artigo, que pode introduzir ou agrupar suas diferentes partes, até mesmo para evitar repetições de linguagem. Um exemplo de artigo com *caput* é o artigo 1º da Lei 8906 (no apêndice **A**), que apresenta como *caput*: *São atividades privativas de advocacia*:. Caso ele não tenha subdivisões, o texto que ali está não recebe o nome de *caput*, sendo apenas o corpo do artigo.

Um artigo pode ser dividido em parágrafos, e caso tenha apenas um, este recebe o nome de *parágrafo único*. Parágrafos podem acrescentar informações novas ao *caput* ou apresentar exceções. Outra divisão de um artigo ou de um parágrafo é por incisos, que têm caráter de listagem. Incisos, por sua vez, podem apresentar alíneas, que têm por objetivo criar sub-listas.

Além das divisões já ditas, uma lei apresenta um preâmbulo com a data que foi sancionada, um pequeno texto com seu tema geral, e indicação de quem a sancionou. Ao final, apresenta uma assinatura. Essas partes não são relevantes para o desenvolvimento deste trabalho, porém.

5.2

Lei 8906

A Lei 8906 é uma Lei Ordinária de 1994 que dispõe sobre o Estatuto da Advocacia e a OAB. Nela estão apresentados os principais direitos e deveres de advogados, em especial no exercício da profissão, além de explicitar o papel e a influência da OAB sobre a advocacia no Brasil. Ela apresenta 87 artigos, separados em 4 Títulos: Da Advocacia, Da Ordem dos Advogados do Brasil, Do Processo na OAB, e Das Disposições Gerais e Transitórias.

No Apêndice A estão formalizados os primeiros 14 (15, ao considerarmos o artigo 7-A) artigos dessa lei, todos referentes ao Título I - Da Advocacia, e cobrindo 3 capítulos: Da Atividade de Advocacia, Dos Direitos do Advogado, e Da Inscrição. Espero que ele possa ser utilizado como referência ao leitor para o que será discutido neste capítulo.

5.2.1

Gerando os Conceitos

Para realizar a formalização de artigos em iALC é importante ter alguns critérios em mente. O primeiro é que, por tratar de modo genérico, o texto jurídico muito dificilmente envolverá instanciação do que foi interpretado³ (que, no caso de lógica de descrição, envolve o uso de indivíduos ou *nominals*, i.e. construções da forma $n : C$, onde n é um nominal e C é um conceito), ou seja, de formalização em ABox. Nas formalizações realizadas até o momento, seja das apresentadas no apêndice A ou das do capítulo 6, não foi necessário instanciar nada daquilo visto no texto da lei,⁴ bastando criar os conceitos necessários e as relações entre eles, dentro da linguagem de iALC.

O segundo critério a se considerar é sobre quais termos encontrados no texto serão os novos conceitos gerados, lembrando sempre que um conceito representa o conjunto de VLSs sobre o termo representado textualmente. No caso da Lei

³A exceção ocorre principalmente nas leis das categorias Reguladoras, Infralegais Coletivas, e Infralegais Privadas, que lidam explicitamente com indivíduos, sem a generalização presente nas leis de camadas superiores. Podemos considerar, por exemplo, um contrato entre duas pessoas físicas específicas, que é um texto legal válido, mas que nomeia as partes diretamente.

⁴Vale notar aqui que é possível, dependendo da situação, interpretar e formalizar utilizando nominals, mas em todos os casos vistos isso não se mostrou necessário, pois sempre foi possível realizar a formalização genérica (ou seja, puramente em TBox), e a formalização genérica induz uma maior automatização do processo.

8906, um conceito trivial é o que agrupa todos os VLS sobre advogados, ao qual damos o nome *Advogado*. Muitos artigos dessa lei fazem referência a *Advogado*, justamente por este ser o tema central dessa lei, como já pode ser visto no artigo 1º, no apêndice **A**. De modo geral, cria-se conceitos para pessoas representando determinados papéis (*DiretorJur*, para aqueles que exercem direção jurídica, no art. 1º inciso II) ou profissões (como *Advogado*, já visto anteriormente), ou em determinado estado (*AdvGestante*, para advogadas gestantes, como no art. 7º-A). Também para ações envolvendo documentos (legais ou não), como *HabeasCorpus*, representando o conjunto de VLSs de impetrações de habeas corpus (art. 1º §1º), e *DivulgacaoAdv*, para a divulgação da atividade da advocacia (art. 1º §3º).

Um terceiro critério é que, por estarmos gerando um agrupamento de fórmulas construtivamente, vamos acabar utilizando conceitos já vistos para utilizar em sentenças nas quais geramos novos. Por exemplo, do *caput* do art. 1º criamos o conceito *Advogado*, e o utilizamos para a criação dos conceitos *PostuladoOPJ* e *PostuladoJE* do inciso I, via um operador de lógica de descrição de acordo com o modo como eles estão relacionados (a escolha do operador será explicada logo mais, em **5.2.2**). Portanto, é importante sempre criar conceitos os mais atômicos possíveis para permitir possíveis futuras relações entre eles, como feito no art. 1º inciso I, onde separamos “a postulação a qualquer órgão do Poder Judiciário e aos juizados especiais;” em *a postulação a qualquer órgão do Poder Judiciário* (*PostuladoOPJ*) e *a postulação aos juizados especiais* (*PostuladoJE*). Com isso, permitimos que leis que serão vistas no futuro que lidem apenas com uma destas postulações possam gerir fórmulas apenas relacionadas ao conceito relevante.

5.2.2

Diferentes Conexões entre Conceitos

Além dos conceitos em si, é importante compreender como eles podem se relacionar, e isso ocorre de variados modos. Essas variações são percebidas linguisticamente, e há certos padrões que podem ser percebidos, especialmente em textos jurídicos.

1. Direitos e Deveres

Muito do que está na lei faz menção a direitos ou deveres de entidades. Para o direito D de uma entidade C , os conectamos do seguinte modo: $C \sqsubseteq D$, explicitando que todos os VLSs para C também são VLSs de D , um *direito* dos indivíduos de C . O contrário não pode ocorrer pois estaríamos restringindo os direitos em D apenas para os indivíduos de C , o que não é

dito na lei explicitamente. Um exemplo está no art. 7º inciso I, em que temos que o exercício livre da profissão da advocacia é um direito de todo advogado, nos levando a $Advogado \sqsubseteq ExercAdvBR$.

De modo análogo, os deveres assumem a forma $\neg Dev \sqsubseteq \neg C$, i.e. a negação do conceito Dev com os VLSs dos deveres implica a negação do conceito C das VLSs sobre aqueles indivíduos que têm o dever de Dev . Consideramos essa forma pois os deveres apresentam caráter restritivo aos indivíduos no conceito C , dado que, ao não realizar o dever, um indivíduo não pode estar em C . No art. 10 §2º, um advogado deve realizar uma inscrição suplementar à inscrição principal na OAB para ser um advogado (neste parágrafo está descrita apenas a inscrição suplementar, pois a principal já foi apresentada anteriormente). Daí segue que $\neg InscricaoSupl \sqsubseteq \neg Advogado$. Pode-se perguntar por que não são ambos tratados do mesmo modo, uma vez que nos deveres, lidamos com a contra-positiva dos direitos. Vale lembrar aqui a discussão sobre a contrapositiva numa lógica intuicionista realizada no capítulo 3, para evidenciar a diferença dessa fórmula para $Advogado \sqsubseteq InscricaoSupl$.

2. Especificação e Generalização

Alguns conceitos podem apresentar sub-conceitos para os quais determinados pedaços da lei valem, portanto é necessário que os relacionemos via subjunção de conceitos. Isso pode ser visto no art. 7º inciso VI alínea d, que faz menção a clientes que dispõem de poderes especiais, sendo que já havíamos criado anteriormente (art. 7º inciso III) o conceito para clientes de modo geral, $Cliente$. Daí, se faz necessário criar o conceito $ClienteEsp$, e conectá-los via $ClienteEsp \sqsubseteq Cliente$. Nesta alínea, então, podemos utilizar $ClienteEsp$ conforme dito no texto, evitando comprometer a estrutura do conceito $Cliente$.

De modo análogo, caso encontremos algum termo que generalize algum conceito já criado, devemos criá-lo e utilizar subjunção de conceitos com o já criado, mas trocando as posições do caso da especificação. No *caput* do art. 6º, criamos o conceito $NivelHierarquicoAdv$ para conter as VLSs que apresentem o mesmo nível hierárquico de um advogado, e o conectamos ao conceito já existente $Advogado$, $Advogado \sqsubseteq NivelHierarquicoAdv$.

3. Agrupamento

De modo muito parecido com a Generalização, o Agrupamento de conceitos também envolve incluir conceitos em outros, a diferença estando na hora da criação dos conceitos: no agrupamento, vemos todos sendo criados no mesmo

fragmento textual. No art. 4º, parágrafo único, temos quatro conceitos novos: *AdvImpedido*, *AdvSuspendido*, *AdvLicenciado*, e *AtivIncompatAdv*,⁵ sendo que os 3 últimos fazem parte do primeiro.

4. Conjunções/Disjunções

Alguns conceitos agem em conjunto em determinadas situações, especialmente via o uso de *e* ou *ou*. No art. 1º §2º, temos que atos e contratos de pessoas jurídicas necessitam de vistos de advogados, nos levando a $(AtoPJ \sqcap ContratoPJ) \sqsubseteq VistoAdvogado$. Alternativamente, a formalização poderia ser dividida em duas: $AtoPJ \sqsubseteq VistoAdvogado$ e $ContratoPJ \sqsubseteq VistoAdvogado$. Há casos mais complexos, especialmente os que acabam envolvendo negação de conceitos, que impedem essa divisão.

Muitas das vezes há sentenças incompletas, especialmente formadas em parte pelo *caput* de um artigo, e em parte pelos incisos. A formalização nesses casos (especialmente levando em conta o caráter de listagem dos incisos) se dá repetindo a frase parcial do *caput* para cada inciso. O art. 7º todo apresenta esse formato, incluindo também frases parciais de incisos para suas respectivas alíneas.

5. Relação entre indivíduos de diferentes conceitos

Certos conceitos têm por definição uma relação com outros conceitos, pois têm implícita a existência de um par de indivíduos que se relaciona. Nesses casos, criamos não apenas um conceito novo, mas também uma relação nova, com as construções de $\forall R.C$ ou $\exists R.C$. De modo geral, \forall envolve relações que implicam em necessidade, e \exists , possibilidade ou direitos, mas mesmo aparentando as características de Deveres ou Direitos, o tratamento é diferente do item 1 desta lista, justamente por internalizar o direito ou dever no uso do quantificador \forall ou \exists .

Um exemplo de uso se encontra no art. 3º §2º, que trata da atividade do estagiário de advocacia regularmente inscrito, que pode ser realizada caso ele esteja sendo supervisionado por um advogado. Daí temos que criar uma relação nova a qual chamaremos *supervisiona*, que liga um advogado (*Advogado*) a um estagiário (*EstagiarioAdv*). Assim, criamos um conceito novo, $\forall supervisiona.EstagiarioAdv$, de todos os que supervisio-

⁵Um ponto interessante do conceito *AtivIncompatAdv* é o fato de o art. 28 (não formalizado) descrever todas as atividades incompatíveis com a advocacia, explicitando um uso extenso da Especificação, o que minimiza a necessidade de alterar fórmulas anteriores, simplesmente acrescentando aquilo que já foi criado. No capítulo 6 mostramos exemplo de uso do art. 28.

nam estagiários de advocacia, que devem ser advogados. Temos, portanto, $(\forall supervisona.EstagiarioAdv) \sqsubseteq Advogado$.

6. Redundância

Alguns textos legais acabam não acrescentando informação relevante do ponto de vista de formalização em iALC, bastando a formalização anteriormente realizada. O art. 7º-A refere-se aos direitos da advogada, e gira em torno de advogadas gestantes, lactantes, que derem à luz, e adotantes. Seguindo o caput e os incisos deste artigo, criamos os conceitos *Advogada*, *AdvGestante*, *AdvLactante*, *AdvAdotante*, e *AdvGestante*, e geramos as inclusões necessárias, além de criar conceitos para o acesso a creche e demais direitos, e realizarmos as amarrações necessárias. Mas, ao chegarmos ao §1º, vemos um texto que diz

“Os direitos previstos à advogada gestante ou lactante aplicam-se enquanto perdurar, respectivamente, o estado gravídico ou o período de amamentação.”

Pela definição de *AdvGestante*, a advogada que não for mais gestante já não pode estar neste conceito, além de a afirmação acima lidar com o conceito de tempo, não necessariamente acobertado por iALC. O mesmo vale para *AdvLactante*. Assim, acabamos por não gerar novas fórmulas com esse parágrafo, pois seria redundante.

7. Alteração de sentenças anteriores

Apesar de tentarmos ao máximo evitar retrabalho, há casos em que não se pode fugir da reescrita de fórmulas anteriores. É o que ocorre com o art. 7º §6º, que altera as fórmulas do inciso II do mesmo artigo.

No inciso II, temos que o escritório ou local de trabalho e os instrumentos de trabalho (entre outros) de um advogado são invioláveis. Isso nos leva a algumas fórmulas, dentre elas $(\forall posse.Escritorio) \sqsubseteq ObjInviolaveisAdv$, indicando que escritórios sob a posse de advogados são invioláveis, e $(\forall posse.Instrumentos) \sqsubseteq ObjInviolaveisAdv$, indicando que instrumentos de trabalho sob a posse de advogados são invioláveis.

No §6º, temos que, caso o advogado esteja envolvido em um crime, a inviolabilidade do local e dos instrumentos de trabalho pode ser quebrada, o que implica mudanças nas fórmulas vistas anteriormente para $(\forall posse.Escritorio) \sqcap \neg Crime \sqsubseteq ObjInviolaveisAdv$ e $(\forall posse.Instrumentos) \sqcap \neg Crime \sqsubseteq ObjInviolaveisAdv$, respectivamente.

A mudança se torna necessária neste caso pois os objetos se tornam invioláveis apenas quando o advogado não tem envolvimento em crime, ou seja, adicionamos uma condição nova para a entrada do local ou dos instrumentos de trabalho em *ObjInviolaveisAdv*. Precisamos, portanto, descartar as fórmulas anteriores pois elas estão em direto desacordo com o que está presente no texto jurídico (agora mais completo), e substituí-las pelas novas, não servindo apenas adicionar as novas, pois isso faria a lógica perder sua monotonicidade.

6

Exemplo de Uso: Exame da OAB

Nesta Seção será explicada a estrutura do Exame da OAB a fim de introduzir 3 exemplos de questões da etapa de múltipla escolha do exame como uma possível aplicação para a formalização de texto legal em iALC apresentada neste documento. Após a apresentação dos exemplos, eles serão discutidos e comparados, de modo a expandir a discussão iniciada em [AHR18].

6.1

Estrutura do Exame

A partir de 2010, os Exames da OAB, que eram regionais, passaram a ser unificados para todo o país. Preparados pela FGV,^[1] por ano são realizados 3 exames no Brasil, e todos apresentam duas etapas: uma primeira, de 80 questões de múltipla escolha, abordando assuntos gerais do direito, e uma segunda, discursiva e específica para a especialidade de escolha de cada candidato. Candidatos que apresentarem uma taxa de acerto superior a 50% na primeira fase passam para a realização da segunda. Focaremos aqui apenas nas questões presentes na primeira fase.

Na primeira fase, cada questão apresenta 4 opções de resposta, com letras de A a D, sendo apenas uma a correta de cada questão. Cada questão, além disso, faz parte de um assunto,^[2] que não é explicitamente mencionado aos candidatos, cabendo a eles deduzir qual é o tema e quais textos utilizar para auxiliá-los a encontrar a resposta correta. A taxa de reprovação tende a ficar por volta de 80% nessa fase ainda.

Para nós, não é interessante apenas responder as questões do exame, mas sim fornecer uma justificativa que corrobore positivamente com a resposta. Por isso buscamos utilizar a abordagem de formalização em lógica do conteúdo da lei e das questões, para no fim termos toda a informação necessária para raciocinar de modo procedural sobre as questões do exame.

¹<https://fgvprojetos.fgv.br/node/135>

²Em [Del+17] há uma listagem dos principais temas que as questões podem apresentar.

6.2

Exemplo 1: Marcas

Neste exemplo utilizamos um sistema de dedução natural para iALC [HPR10], baseado no cálculo de seqüentes introduzido em [HR16]. A questão é a de número 50 do Exame da OAB VII de 2012, na área de Negócios. O assunto é *marcas*, que se relaciona à Lei 9279.³ Conforme visto no capítulo 5, esta lei pode ser definida como a interseção dos conceitos representando cada um de seus artigos, i.e. $Lei_{9279} = Art_1 \sqcap Art_2 \sqcap \dots \sqcap Art_{244}$. O mesmo ocorre nos artigos com relação a seus respectivos incisos e parágrafos. A questão é como segue:

Sobre as marcas, é correto afirmar que

- A) a marca de alto renome é sinônimo de marca notoriamente conhecida.
- B) a vigência do registro da marca é de 5 (cinco) anos, sendo prorrogável por períodos iguais e sucessivos.
- C:CORRETA) é permitida a cessão do pedido de registro de marca, caso o cessionário atenda aos requisitos legais.
- D) a marca de produto ou serviço é aquela usada para identificar produtos ou serviços provindos de membros de uma determinada entidade.

A alternativa A é relacionada aos artigos 125 e 126. O artigo 125 (“*a marca registrada no Brasil considerada de alto renome será assegurada proteção especial, em todos os ramos de atividade*”) pode ser formalizado como $(MarcaBr \sqcap AltoRen) \sqsubseteq EspProtec$, mas não termina aí: como essa é a única menção na lei de uma marca ser de *alto renome* (não há outra definição explícita ou uso do termo), podemos deduzir que apenas marcas Brasileiras podem ser de alto renome ($AltoRe \sqsubseteq MarcaBr$). Já o artigo 126 (“*a marca notoriamente conhecida em seu ramo de atividade [...] goza de proteção especial, independentemente de estar previamente depositada ou registrada no Brasil*”) pode ser formalizado como $NoCo \sqsubseteq (MarcaBr \sqcup \neg MarcaBr)$. Nosso objetivo é mostrar que estes conceitos não são, de fato, sinônimos.

Queremos mostrar que uma marca pode ser notoriamente conhecida e não de alto renome (o que constitui um contraexemplo para a alternativa A). Um simples contra-modelo envolve termos $AltoRe = \emptyset$ e $NoCo = MarcaBr \neq \emptyset$, no qual as únicas marcas notoriamente conhecidas são todas as marcas brasileiras e não há marcas de alto renome.

A alternativa B tangencia o artigo 133, que pode ser formalizado como $r : Dez$, para um registro r de uma marca (“*o registro da marca vigorará pelo prazo de 10 (dez) anos, contados da data da concessão do registro, prorrogável por*

³Texto original em http://www.planalto.gov.br/ccivil_03/Leis/L9279.htm.

períodos iguais e sucessivos”). Isso claramente contradiz o que está escrito em B, $r : \text{Cinco}$. Em iALC, *Dez* e *Cinco* representam os conceitos cujos indivíduos são VLSs com validade de 10 e 5 anos, respectivamente. Aqui, vemos uma instância na qual é necessário utilizar uma *TBox* para expressar este conhecimento de mundo: $\text{Dez} \sqsubseteq \neg \text{Cinco}$ e $\text{Cinco} \sqsubseteq \neg \text{Dez}$, uma vez que, apesar de não ser diretamente relacionado à lei ou à questão, é conhecimento geral. Ainda neste capítulo será discutido esse assunto, após a apresentação dos exemplos. De qualquer modo, a fim de gerar um contra-modelo, cria-se um mundo r , no qual teríamos $r \models \text{Dez}$ e $r \not\models \text{Cinco}$ coexistindo.

A alternativa C relaciona-se ao artigo 134, e é a resposta correta à questão. O artigo diz que “o pedido de registro e o registro poderão ser cedidos, desde que o cessionário atenda aos requisitos legais para requerer tal registro”, e pode ser formalizado como $\text{RequisLegal} \sqsubseteq \text{PedidoRegis}$ and $\text{RequisLegal} \sqsubseteq \text{Registro}$. A alternativa C em si pode ser formalizada como $\text{RequisLegal} \sqsubseteq \text{PedidoRegis}$, que é uma das formalizações já feitas no artigo 134. Já que é a opção correta, seria impossível construir um contra-modelo. Fazemos, então, a seguinte árvore de dedução, constituída de uma simples tautologia, já que a alternativa C é apenas uma das fórmulas no texto da lei:

$$\frac{}{\text{RequisLegal} \sqsubseteq \text{PedidoRegis}} \text{Tautologia}$$

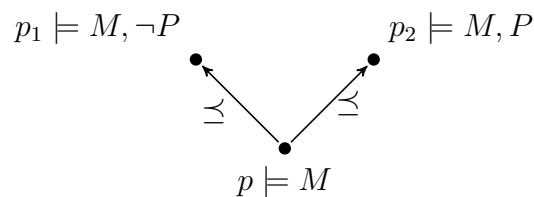
Alternativamente, podemos formalizar aquilo que está escrito no artigo 134 como $\text{RequisLegal} \sqsubseteq (\text{PedidoRegis} \sqcap \text{Registro})$, o que é equivalente. A árvore de dedução neste caso segue (com c sendo um VLS representando um cessionário):

$$\frac{\frac{\frac{[c : \text{RequisLegal}]^1}{c : (\text{PedidoRegis} \sqcap \text{Registro})}}{c : \text{PedidoRegis}}}{c : (\text{RequisLegal} \sqsubseteq \text{PedidoRegis})} 1$$

Por fim, a alternativa D faz referência ao primeiro inciso do artigo 123: “considera-se [...] marca de produto ou serviço: aquela usada para distinguir produto ou serviço de outro idêntico, semelhante ou afim, de origem diversa”. Ela pode ser formalizada como havendo $p_1 : (M \sqcap P)$ e $p_2 : (M' \sqcap P)$ implicando não poder haver precedência para ambos os produtos, sendo M e M' os conceitos válidos para VLSs das marcas M e M' , respectivamente, p_1 e p_2 produtos, e P um conceito genérico representando qualquer propriedade que um produto possa apresentar.

A questão pode ser formalizada como tendo $p_1 : (M \sqcap \neg P)$ e $p_2 : (M \sqcap P)$ implicando haver algum mundo precedendo ambos. Note como não utilizamos M' , já que a questão lida apenas com a mesma entidade, ou seja, com a mesma marca.

A chave para a geração de um contra-modelo aqui é mostrar que, com nossas premissas, é impossível diferenciar (para níveis de modelo, impedir relação de precedência entre os VLSs) um produto de uma marca de outro produto de outra marca, mesmo que ambos apresentem propriedades diferentes. É, portanto, suficiente mostrar que um mundo p_1 no qual temos M e $\neg P$ e um mundo p_2 no qual tem-se $p_2 \models M$ e $p_2 \models P$ coexistindo precedidos por um mesmo mundo, p , o meet de ambos, no qual vale M , tornando explícito no modelo que a lei não pode discernir entre os dois produtos, já que há precedência para ambos.



6.3

Exemplo 2: Deise

Para esta questão, utilizamos um método mais procedural de interpretação e formalização da questão e do texto jurídico. Ao formalizar o texto da lei, sempre lidamos com conceitos de um modo mais geral (i.e. sem indivíduos/VLSs), dado que o texto da lei é, por construção, genérico. Essa etapa se mantém como no exemplo anterior. Mas agora, o que muda é que, para o enunciado e para as alternativas buscamos certos *blocos sintáticos*, sendo que neste exemplo focamos em sintagmas nominais, cada um levando a uma fórmula da forma *sujeito : Objeto* (*sujeito* sendo um VLS em um fato legal geralmente representando o sujeito sintático de uma frase, e *Objeto* sendo o conceito dos VLSs relacionados ao objeto sintático da frase). Com isso em mãos, nosso foco será de encontrar os blocos *relevantes* e utilizá-los para criar contra-exemplos (via a geração de *meets* de indivíduos) a fim de encontrar as respostas corretas para as questões. Para este exemplo, consideraremos apenas os sintagmas nominais relevantes à resolução da questão em si.

Esta questão é a de número 6 do Exame XVII de 2015, cuja área é Ética. O assunto relaciona-se à Lei 8906.⁴ Como visto anteriormente, esta lei pode ser definida como as interseções os conceitos dos artigos, i.e. $Lei_{8906} = Art_1 \sqcap Art_2 \sqcap \dots$. O mesmo acontece com os artigos e seus respectivos incisos e

⁴Texto original em http://www.planalto.gov.br/ccivil_03/Leis/L8906.htm.

parágrafos. A questão é a seguinte (principais blocos sintáticos após um potencial passo de processamento textual estão sublinhados):

Deise é uma próspera advogada e passou a buscar novos desafios,
[ela] sendo eleita Deputada Estadual. Por força de suas raras habilida-
des políticas, [ela] foi eleita integrante da Mesa Diretora da Assembleia
Legislativa do Estado Z. Ao ocupar esse honroso cargo procurou con-
ciliar sua atividade parlamentar com o exercício da advocacia, sendo
seu escritório agora administrado pela filha. Nos termos do Estatuto
da Advocacia, assinale a afirmativa correta.

A) A atividade parlamentar de Deise é incompatível com o [seu] exer-
cício da advocacia.

B:CORRETA) A participação de Deise na Mesa Diretora a torna in-
compatível com o [seu] exercício da advocacia.

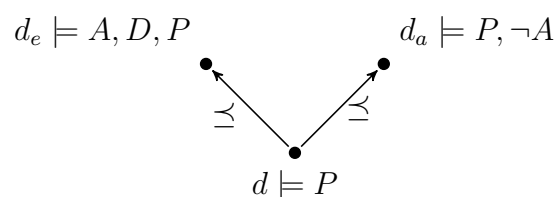
C) A função de Deise como integrante da Mesa Diretora do Parla-
mento Estadual é conciliável com o [seu] exercício da advocacia.

D) A atividade parlamentar de Deise na Mesa Diretora pode ser con-
ciliada com o [seu] exercício da advocacia em prol dos necessitados.

Todas as alternativas são contempladas pelo artigo 28, que diz que “a *advocacia é incompatível, mesmo em causa própria, com as seguintes atividades: I - chefe do Poder Executivo e membros da Mesa do Poder Legislativo e seus substitutos legais; [...]*”, e pode ser formalizada como $Advogado \sqsubseteq \neg Diretoria$ e $Diretoria \sqsubseteq \neg Advogado$,⁵ ou seja, os indivíduos para os quais o conceito *Advogado* é válido não podem preceder indivíduos para os quais *Diretoria* é válido, e vice-versa.

Podemos formalizar o enunciado da questão como $d_e : Advogado$, $d_e : Parlamentar$, e $d_e : Diretoria$, d_e representando nosso conhecimento legal (parcial, vide capítulo 3) sobre Deise.

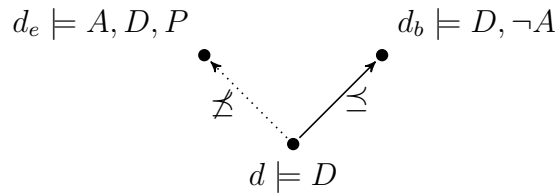
A alternativa A pode ser formalizada como $d_a : \neg Advogado$ e $d_a : Parlamentar$. Isso nos leva a um *meet* d de d_e e d_a que tem *Parlamentar* como um conceito válido. Para fins de legibilidade, nos modelos apresentados os conceitos serão reduzidos às suas iniciais.



⁵O conceito *Diretoria* é válido em todos os VLSs sobre membros de diretorias de assembleias legislativas e o conceito *Advogado* é válido em todo VLS sobre advogados.

Já que d precede um mundo que apresenta *Advogado* como um conceito válido, podemos concluir que é possível ser um Parlamentar e um Advogado ao mesmo tempo, tornando a alternativa A falsa.

A alternativa B é a resposta correta à questão, e é a única que não pode apresentar um contra-modelo. Ela pode ser formalizada como $d_b : Diretoria$ e $d_b : \neg Advogado$ (alternativamente, a negação pode estar em *Diretoria* e o procedimento será análogo). Um *meet* entre d_b e d_e deve conter *Diretoria* como um conceito válido, precedendo o também válido *Advogado* em d_e , claramente contradizendo aquilo que é dito na Lei 8906.



Como não é possível que esse *meet* exista, concluímos que não podemos gerar contra-modelo, e ela é, portanto, a alternativa correta.

A alternativa C pode ser formalizada como $d_c : Diretoria$ e $d_c : Advogado$. Ela é uma negação direta de o que está na alternativa B, e pode-se verificar sua falsidade via uma simples árvore de dedução (note que todas as premissas ou vêm da lei, da questão, ou da alternativa C em si):

$$\frac{\frac{d_c : D \quad D \sqsubseteq \neg A}{d_c : \neg A} \quad [d_c : A]^1}{\frac{d_c : \perp}{\neg (d_c : A)} 1}$$

Uma vez que uma alternativa era a negação direta de outra, não haveria necessidade de criar um contra-modelo para a segunda, uma vez que é claro que ela não é a resposta correta. Além disso, fica evidente que em casos como esse a resposta *deve* ser uma delas. Isso é um forte indicador para que primeiro formalizemos todas as alternativas de uma vez e só aí tentemos realizar deduções sobre elas, em vez de fazer uma a uma.

Por fim, a alternativa D pode ser formalizada como as seguintes fórmulas: $d_d : Diretoria$, $d_d : Advogado$ e $d_d : AdvogadoNecessitados$, com a adicional $AdvogadoNecessitados \sqsubseteq Advogado$, dizendo que aquilo que é válido para um advogado que age em prol dos necessitados deve também ser válido para um advogado em geral. A justificativa é basicamente a mesma da alternativa C, com uma regra dedutiva a mais (e *AdvogadoNecessitados* reduzido para *AN*):

$$\frac{\frac{d_d : D \quad D \sqsubseteq \neg A}{d_d : \neg A} \quad \frac{[d_d : AN]^1 \quad AN \sqsubseteq A}{d_d : A}}{\frac{d_d : \perp}{\neg(d_d : AN)} \quad 1}$$

6.4

Exemplo 3: Luana, Leonardo, e Bruno

Esta questão (questão 4 do Exame da OAB XIX de 2016) já foi mostrada como um exemplo em [Del+17], e de modo similar ao exemplo anterior, envolve a presença de atividades incompatíveis com a prática da advocacia (essencialmente o artigo 28 da Lei 8906). A formalização dessa questão, diferentemente do exemplo da Deise, é muito mais simples, apesar de não aparentá-lo à primeira vista, devido à descentralização das instâncias (lidamos com 3 pessoas, e não mais apenas uma) e ao grau de especificação das histórias de cada uma das pessoas envolvidas. De fato, realizar dedução para as alternativas A, B, e D toma o mesmo formato, que é, inclusive, o mesmo formato utilizado para a alternativa C do exemplo da Deise. A questão é como segue:

Formaram-se em uma Faculdade de Direito, na mesma turma, Luana, Leonardo e Bruno. Luana, 35 anos, já exercia função de gerência em um banco quando se graduou. Leonardo, 30 anos, é prefeito do município de Pontal. Bruno, 28 anos, é policial militar no mesmo município. Os três pretendem praticar atividades privativas de advocacia. Considerando as incompatibilidades e impedimentos ao exercício da advocacia, assinale a opção correta.

A) Luana não está proibida de exercer a advocacia, pois é empregada de instituição privada, inexistindo impedimentos ou incompatibilidades.

B) Bruno, como os servidores públicos, apenas é impedido de exercer a advocacia contra a Fazenda Pública que o remunera.

C:CORRETA) Os três graduados, Luana, Leonardo e Bruno, exercem funções incompatíveis com a advocacia, sendo determinada a proibição total de exercício das atividades privativas de advogado.

D) Leonardo é impedido de exercer a advocacia apenas contra ou em favor de pessoas jurídicas de direito público, empresas públicas, sociedades de economia mista, fundações públicas, entidades paraestatais ou empresas concessionárias ou permissionárias de serviço público.

Além do primeiro inciso do artigo 28, precisamos também dos incisos “V -

ocupantes de cargos ou funções vinculados direta ou indiretamente a atividade policial de qualquer natureza” e “VIII - ocupantes de funções de direção e gerência em instituições financeiras, inclusive privadas”, que geram os pares de fórmulas $Advogado \sqsubseteq \neg Policial$ e $Policial \sqsubseteq \neg Advogado$, e $Advogado \sqsubseteq \neg GerFinan$ e $GerFinan \sqsubseteq \neg Advogado$, respectivamente. Assim podemos resolver completamente a questão.

Conforme dito anteriormente, as deduções para as letras A, B e D ocorrem de modo similar à C do segundo exemplo, bastando trocar as instâncias para se referirem a 3 indivíduos distintos. Basta criar três VLSs lu , bru , e leo , tais que $lu : GerFinan$, $bru : Policial$, e $leo : Diretoria$, para resolver as alternativas respectivamente.

A argumentação para a resposta correta ser a letra C segue por eliminação, apesar de apresentar uma árvore de dedução simples também, formada apenas por um modus ponens, com o adicional que estamos lidando com três situações. Portanto, a árvore se quebra em três, todas com o mesmo formato. Segue o exemplo da árvore de dedução para Bruno:

$$\frac{Policial \sqsubseteq \neg Advogado \quad b : Policial}{b : \neg Advogado}$$

6.5

Discussão dos Exemplos

Desses exemplos tiramos alguns pontos importantes, em especial tendo em vista os critérios levantados em [Ver17](#) sobre a formalização de casos e argumentos jurídicos:

1. Escolher a justificativa adequada

Já no primeiro exemplo tivemos alternativas que puderam ser resolvidas tanto via o uso de árvores de dedução quanto de geração de contra-modelos, mas esse não é necessariamente o caso. Podemos encontrar questões cujas alternativas podem ser resolvidas apenas por um desses métodos (de modo geral, queremos árvores de dedução para as alternativas corretas e contra-modelos para as incorretas).

De fato, conforme visto na alternativa B do Exemplo 2 (Deise), para a alternativa correta é impossível gerar um contra-modelo, e nesse caso, a justificativa foi que era impossível gerar um *meet* dos VLSs apresentados, o que deveria ser sempre possível. Essa justificativa indica a impossibilidade da existência de qualquer contra-modelo para a letra B, mas não é necessariamente o caso que a impossibilidade de gerar um contra-modelo implique

na impossibilidade da existência de um contra-modelo. A chave aqui foi a *justificativa* que se baseia em propriedades de Álgebras de Heyting [Kri65].

2. Lidar com conhecimento geral

Conforme visto na alternativa B do primeiro exemplo, será necessário também lidar com conhecimento de mundo que não está necessariamente ligado a leis. Este ponto foge um pouco do escopo apresentado e nos leva a uma discussão mais geral, sobre a expressividade que iALC apresenta, e sobre a expressividade de que precisamos para poder responder ao Exame da OAB.

Temos nas mãos uma lógica que lida com leis, e um exame que envolve leis, mas que foi criado para (e por) seres humanos, e apresenta implícitos conhecimentos esperados de um ser humano comum, e que nada têm a ver com leis. O quão adequado é o uso de iALC especificamente ao exame da OAB? O quanto podemos esperar de um possível “raciocinador” automatizado sobre esta prova? O que falta? E, quanto ao que falta, é possível representá-lo no mundo de iALC?

Todas essas questões e outras indicam importantes direções para tomar em trabalho futuro, em especial, no âmbito “teórico”, dos limites de representações de leis quando apresentadas a representações de texto e conhecimento humano em geral, e no âmbito mais prático, de questões a se ter em mente na implementação de um resolvidor automatizado para as questões do exame da OAB.

3. Buscar por estruturas linguísticas similares

Como visto no exemplo 2, a extração de informação de um texto apresenta uma tradução em lógica bem mais direta e padronizada quando se busca por padrões específicos de linguagem. Isso efetivamente facilita a categorização de tipos de respostas que podem ser dadas, e que tipo de informação pode estar faltando nas questões para suas opções estarem corretas ou incorretas. Também diminui a chance de alguma informação realmente útil à resolução da questão passar despercebida.

Este ponto também abre bastante espaço para trabalho futuro, em especial sobre quais estruturas linguísticas buscar, e como traduzi-las para sentenças formalizadas em iALC, também levantando questões sobre a expressividade de iALC quando avaliada sob a ótica da linguagem natural.

4. Reutilizar formalizações já realizadas

Conforme fomos respondendo às questões da prova e consultando os textos legais, é importante que vamos adicionando aquilo que for sendo formalizado

a uma base de conhecimento, a fim de evitar retrabalho, tanto de busca quanto de formalização. Isso é útil, como vimos, para questões como as do exemplo 2 e 3, que ambas fazem referência ao mesmo texto legal (no caso, o artigo 28 da Lei 8906).

Daqui segue quase diretamente que é sempre bom priorizar o uso de conceitos já instanciados para posteriores formalizações, conforme discutido no capítulo 5.

7

Considerações Finais

Neste documento, começamos apresentando o estado da arte em formalização de textos legais (especialmente em lógica), e motivamos para a utilização da lógica de descrição intuicionista iALC, que teve seus principais conceitos introduzidos e contextualizados para leis. Daí, para solidificar a motivação para o uso de iALC apresentamos implementações de suas provas de correção e completude, apontando para um modo de estruturar a lógica para uso computacional. Por fim, apresentamos proposta de formalização de leis brasileiras em iALC, além de uma aplicação prática dessa formalização, na forma da fase de múltipla escolha do Exame da OAB.

O tema desta dissertação tem um caráter bastante interdisciplinar, e aqui buscamos apresentar uma visão do direito sob a ótica da informática e da lógica, seja para organização das leis em si ou do raciocínio que pode ser realizado sobre elas.

Uma certeza que temos sobre este trabalho é que ainda há muito mais a ser feito. Tanto com foco na parte linguística do problema, na parte de implementação, ou na parte de metodologia de formalização, temos uma possibilidade de aplicação que guia nossos esforços.

7.1

Estudo e Aprendizado

Para a criação deste documento foi necessário muito estudo em áreas da lógica com as quais não havia tido muito contato. Lógica intuicionista já era mais familiar, assim como um pouco dos modelos de Kripke e das lógicas modais, mas DLs e ontologias se mostraram uma grande e nova carga de conhecimento teórico a ser adquirido. Além disso, foi necessário criar costume com os conceitos adquiridos para aplicação prática, o que nem sempre se traduz diretamente daquilo apreendido na teoria, ao menos a nível cognitivo. No fim, os temas se mostraram bastante ricos e expandiram a ciência da lógica como um todo.

Além disso, devido ao caráter naturalmente interdisciplinar do problema, foi necessário pesquisar nas áreas do direito e da linguística. O nível de aprofundamento nestas áreas, claro, não se compara ao aprofundamento que foi necessário em se

tratando de DLs, mas ainda assim foi significativo. Os estudos sobre ordenamento jurídico, jurisprudência de Kelsen, e legislação brasileira aproximaram e deram forma a assuntos que outrora não consideraria interessantes, ou até mesmo relevantes às áreas de pesquisa da informática.

Outro componente que se provou ser um desafio foi a necessidade de construções mais complexas para representar as provas de iALC em $L\exists\forall N$. A experiência prévia deu base para o formato geral das provas, mas o caráter similar à teoria dos conjuntos (pelo menos no uso prático) de DLs fez necessária a geração e uso de tipos indutivos e manejo de elementos com parametrização de tipos, o que tende a deixar a notação um pouco mais convoluta. Isso, claro, precisava ser feito em algum momento. O saldo positivo desta etapa é o menor retrabalho para implementar um raciocinador utilizando o cálculo de seqüentes para iALC em $L\exists\forall N$ (ou em outra ferramenta que se mostrar útil).

De modo geral, a experiência mostrou-se muito rica para a formação como pesquisador, especialmente ao expandir os horizontes da pesquisa a novas áreas do conhecimento. Também foi muito útil tomar maior dimensão da própria área de pesquisa, e das diferentes vertentes e modos de fazer pesquisa ainda dentro da mesma, sem mencionar as possíveis relações que lógica apresenta com outras áreas, e possíveis aplicações para resolução de problemas mais *tangíveis*, que saem da discussão puramente teórica.

7.2

Trabalho Futuro

Conforme dito anteriormente, a implementação das provas de completude e correção para o cálculo de seqüentes em iALC servem como um primeiro passo para a implementação de um raciocinador automatizado utilizando este cálculo. Apesar disso, conforme vimos no capítulo 6, outro candidato para um raciocinador seria a dedução natural. De qualquer modo, a implementação dos tipos básicos continua sendo útil, bem como as demais definições feitas, como o tipo indutivo de listas, por exemplo. Além disso, as regras do cálculo de seqüentes apresentam uma estrutura similar a boa parte das regras de dedução natural, facilitando a adaptação. Tendo ambas as possibilidades em mãos permitirá melhor avaliação de desempenho e facilidade de uso de ambas, a fim de definir qual apresenta melhor utilidade para o raciocinador. Ter em mãos essas ferramentas dará maior robustez a iALC e permitirá avaliar em maior detalhes seu uso prático.

Aliadas ao raciocinador, as técnicas para formalização das leis estão ainda sendo refinadas, e buscamos torná-las o mais geral e algorítmico quanto for possível, para, no futuro, poder gerar bases de conhecimento mais precisas e diversas para resolução das questões do Exame da OAB, tomando como base o que foi

feito para as provas de correção e completude. Conforme nossa metodologia for evoluindo, avaliaremos quais ferramentas já existentes para DLs em geral poderão ser utilizadas como um *framework* ou base de inspiração para nosso raciocinador.

Apesar de complexo, o raciocinador fará parte de um projeto ainda maior, que envolve mais etapas. Conforme pode ter sido notado, assume-se muito sobre a etapa de processamento de linguagem natural que ocorre anterior à formalização em lógica. Quais estruturas sintáticas são mais úteis? Alguma pode ser ignorada? Como realizar esse processamento? Via gramáticas? Via *machine learning*? Como tornar esta parte coesa com a etapa de formalização em lógica? Essas e outras questões abrem um leque grande para pesquisa futura, possivelmente até sem envolver textos jurídicos.

Do ponto de vista mais teórico, podemos expandir as comparações realizadas com lógica deôntica e com demais lógicas utilizadas para leis, como a lógica *default* utilizada em [Per+17], por exemplo. Com isso, poderemos avaliar de modo mais analítico como (ou se) uma lógica se adequa mais ao domínio de leis que as outras, em especial ao considerarmos a expressividade de cada uma.

Ainda há muito a ser feito, mas temos confiança de que aquilo que já está pesquisado e implementado servirá de base para muitos projetos e pesquisas futuros, envolvendo diversas áreas do conhecimento.

Bibliografia

- [Jør38] Jørgen Jørgensen. ‘Imperatives and Logic’. Em: *Erkenntnis* (1937-38), 7: 288-96.
- [Ros41] Alf Ross. ‘Imperatives and Logic’. Em: *Philosophy of Science* 11.1 (1941), pp. 30–46.
- [Hey56] A. Heyting. *Intuitionism: An Introduction*. 3rd. Amsterdam, Netherlands: North-Holland Publishing, 1956, pp. 101–109.
- [Kri65] S. A. Kripke. ‘Semantical analysis of intuitionistic logic I’. Em: *Formal Systems and Recursive Functions*. Ed. por J. N. Crossley e M. A. E. Dummett. Amsterdam: North-Holland, 1965, pp. 92–130.
- [Pra65] Dag Prawitz. *Natural Deduction: A Proof-Theoretical Study*. Dover Publications, 1965.
- [Bel71] W. Russell Belding. ‘Intuitionistic Negation’. Em: *Notre Dame Journal of Formal Logic* 12.2 (1971), pp. 183–187.
- [Rei80] Raymond Reiter. ‘A Logic for Default Reasoning.’ Em: *Artif. Intell.* 13.1-2 (1980), pp. 81–132. URL: <http://dblp.uni-trier.de/db/journals/ai/ai13.html#Reiter80>.
- [Mar84] Per Martin-Löf. ‘Intuitionistic Type Theory’. Em: (1984). Ed. por Bibliopolis.
- [BS95] N. Bobbio e M.C.C.L. dos Santos. *Teoria do Ordenamento Jurídico*. Ed. Universidade de Brasília, 1995. ISBN: 9788523002763. URL: <https://books.google.com.br/books?id=2n0kAAAACAAJ>.
- [Kel99] Hans Kelsen. *Teoria Pura do Direito*. Trad. por Joffe de Baptista Machado. Martins Fontes, 1999.
- [Con+03] Cleo Condoravdi et al. ‘Entailment, Intensionality and Text Understanding’. Em: *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning - Volume 9*. HLT-NAACL-TEXTMEANING ’03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 38–45. DOI: [10.3115/1119239.1119245](https://doi.org/10.3115/1119239.1119245). URL: <https://doi.org/10.3115/1119239.1119245>.

- [HPT07] Jörg Hansen, Gabriella Pigozzi e Leendert van der Torre. ‘Ten Philosophical Problems in Deontic Logic’. Em: *Normative Multi-agent Systems*. Ed. por Guido Boella, Leon van der Torre e Harko Verhagen. Dagstuhl Seminar Proceedings 07122. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. URL: <http://icr.uni.lu/leonvandertorre/papers/normas07b.pdf>.
- [Wal07] Vern R. Walker. ‘Discovering the Logic of Legal Reasoning’. Em: *Hofstra Law Review* 35.4 (2007). URL: <https://scholarlycommons.law.hofstra.edu/hlr/vol35/iss4/2>.
- [HLP08] Frank van Harmelen, Vladimir Lifschitz e Bruce Porter. *Handbook of Knowledge Representation*. Amstrerdam: Elsevier, 2008, pp. 135–180. ISBN: 978-0-444-52211-5.
- [Pri08] Graham Priest. *Introduction to Non-Classical Logic*. Cambridge University Press, 2008.
- [AG09] Ioan A. Letia e Adrian Groza. ‘Modelling Imprecise Arguments in Description Logic’. Em: *Advances in Electrical and Computer Engineering* 9 (out. de 2009). DOI: [10.4316/aece.2009.03017](https://doi.org/10.4316/aece.2009.03017).
- [Pri09] Graham Priest. ‘Dualising Intuitionistic Negation’. Em: *Principia: an international journal of epistemology* 13.2 (2009), pp. 165–184. ISSN: 1808-1711. DOI: [10.5007/1808-1711.2009v13n2p165](https://doi.org/10.5007/1808-1711.2009v13n2p165). URL: <https://periodicos.ufsc.br/index.php/principia/article/view/1808-1711.2009v13n2p165>.
- [HPR10] Edward Hermann Haeusler, Valéria de Paiva e Alexandre Rademaker. ‘Intuitionistic Logic and Legal Ontologies’. Em: *Proc. JURIX 2010*. IOS Press, 2010, pp. 155–158.
- [Ben14] Martin Mose Bentzen. ‘Action Type Deontic Logic’. Em: *Journal of Logic, Language and Information* 23.4 (dez. de 2014), pp. 397–414. ISSN: 1572-9583. DOI: [10.1007/s10849-014-9205-0](https://doi.org/10.1007/s10849-014-9205-0). URL: <https://doi.org/10.1007/s10849-014-9205-0>.
- [Har+14] Babak Bagheri Hariri et al. ‘Description Logic Knowledge and Action Bases’. Em: *CoRR* abs/1402.0569 (2014). arXiv: [1402.0569](https://arxiv.org/abs/1402.0569). URL: <http://arxiv.org/abs/1402.0569>.

- [HR16] Edward Hermann Haeusler e Alexandre Rademaker. ‘On How Kelsenian Jurisprudence and Intuitionistic Logic help to avoid Contrary-to-Duty paradoxes in Legal Ontologies’. Em: *Proc. Journal of Applied Non-Classical Logics* (2016).
- [Kha+16] Imran Khan et al. ‘Conversion of Legal Text to a Logical Rules Set from Medical Law Using the Medical Relational Model and the World Rule Model for a Medical Decision Support System’. Em: *Informatics* 3 (2016), p. 2.
- [WFP16] Adam Zachary Wyner, Biralatei James Fawei e Jeff Z Pan. ‘Passing a USA National Bar Exam: a First Corpus for Experimentation’. Em: *LREC 2016, Tenth International Conference on Language Resources and Evaluation*. LREC. 2016.
- [Del+17] Pedro Delfino et al. ‘Passing the Brazilian OAB Exam: Data’. Em: *Legal Knowledge and Information Systems* (2017), p. 89.
- [Per+17] Marcos A. Pertieria et al. ‘Towards Formalizing Statute Law as Default Logic through Automatic Semantic Parsing’. Em: *Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts co-located with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017), London, UK, June 16, 2017*. 2017. URL: <http://ceur-ws.org/Vol-2143/paper9.pdf>.
- [Ver17] Bart Verheij. ‘Formalizing Arguments, Rules and Cases’. Em: *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law*. ICAIL ’17. London, United Kingdom: ACM, 2017, pp. 199–208. ISBN: 978-1-4503-4891-1. DOI: [10.1145/3086512.3086533](https://doi.org/10.1145/3086512.3086533). URL: <http://doi.acm.org/10.1145/3086512.3086533>.
- [Wan+17] Shuohang Wang et al. ‘ R^3 : Reinforced Reader-Ranker for Open-Domain Question Answering’. Em: *arXiv preprint arXiv:1709.00023* (2017).
- [AHR18] Bernardo Alkmim, Edward Hermann Haeusler e Alexandre Rademaker. ‘Utilizing iALC to Formalize the Brazilian OAB Exam’. Em: *Proceedings from the EXplainable AI in Law Workshop (XAILA)*. to appear. CEUR-WS, 2018.
- [Cla+18] Peter Clark et al. ‘Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge’. Em: *arXiv preprint arXiv:1803.05457* (2018).

- [Din+18] Jiwei Ding et al. ‘Answering Multiple-choice Questions in Geographical Gaokao with a Concept Graph’. Em: (2018).

A

Apêndice - Lei 8906

Neste apêndice estão listados os primeiros 3 capítulos do primeiro título da Lei 8906, constituídos dos artigos de número 1 a 14 (do 1 ao 5 para o primeiro capítulo, do 6 ao 7 para o segundo, e do 8 ao 14 para o terceiro). Para cada artigo, está uma proposta de sua formalização em iALC. Este apêndice não apresenta maiores discussões sobre as formalizações realizadas, servindo de consulta para o capítulo 5, onde a motivação e categorização de formalizações são debatidas.

Vale relembrar, aqui, da estrutura da lei apresentada: ela é dividida em artigos (organizados em títulos e capítulos, a fim de seccionar as leis por assunto), e os artigos, por sua vez, podem ser divididos em incisos, com a possibilidade de haver um ou mais parágrafos para elucidar ressalvas ou observações sobre seus conteúdos. Temos, portanto, que $Lei_{8906} = Art_1 \sqcap \dots \sqcap Art_{87}$, e que cada artigo toma a forma (para o i -ésimo artigo, com n incisos e m parágrafos):

$$Art_i = \prod_{j=1}^n Inc_j \sqcap \prod_{k=1}^m Par_k$$

Todos os 3 capítulos aqui apresentados fazem parte do título *Da Advocacia*, que foca em advogados e em seu exercício da profissão. O primeiro capítulo, *Da Atividade da Advocacia*, regula as atividades profissionais de um advogado. O segundo, *Dos Direitos do Advogado*, apresenta tudo aquilo a que um advogado (e uma advogada, no artigo 7A) tem direito, enquanto profissional. Já o terceiro, *Da Inscrição*, faz referência à inscrição do advogado, tanto perante a OAB quanto os Conselhos Seccionais de suas regiões de exercício da profissão.

A.1

Capítulo I - Da Atividade da Advocacia

A.1.1

Art. 1º

São atividades privativas de advocacia:

I - a postulação a qualquer órgão do Poder Judiciário e aos juizados especiais;

PostuladoOPJ \sqsubseteq Advogado

PostuladoJE \sqsubseteq Advogado

II - as atividades de consultoria, assessoria e direção jurídicas.

ConsultorJur \sqsubseteq Advogado

AssessorJur \sqsubseteq Advogado

DiretorJur \sqsubseteq Advogado

§ 1º Não se inclui na atividade privativa de advocacia a impetração de habeas corpus em qualquer instância ou tribunal.

HabeasCorpus \sqsubseteq (Advogado \sqcap \neg Advogado)

§ 2º Os atos e contratos constitutivos de pessoas jurídicas, sob pena de nulidade, só podem ser admitidos a registro, nos órgãos competentes, quando visados por advogados.

VistoAdvogado \sqsubseteq Advogado

(AtoPJ \sqcap ContratoPJ) \sqsubseteq VistoAdvogado

§ 3º É vedada a divulgação de advocacia em conjunto com outra atividade.

DivulgacaoAdv \sqsubseteq Divulgacao

DivulgacaoOutra \sqsubseteq Divulgacao

Advogado \sqsubseteq \neg (DivulgacaoAdv \sqcap DivulgacaoOutra)

A.1.2

Art. 2º

O advogado é indispensável à administração da justiça.

Advogado \sqsubseteq AdminJustica

§ 1º No seu ministério privado, o advogado presta serviço público e exerce função social.

AdvMinPriv \sqsubseteq Advogado

AdvMinPriv \sqsubseteq ServicoPublico

AdvMinPriv \sqsubseteq FuncaoSocial

§ 2º No processo judicial, o advogado contribui, na postulação de decisão favorável ao seu constituinte, ao convencimento do julgador, e seus atos constituem múnus público.

AdvProJud \sqsubseteq Advogado

DecisaoJud \sqsubseteq Juiz

AdvProJud \sqsubseteq MunusPublico

§ 3º No exercício da profissão, o advogado é inviolável por seus atos e manifestações, nos limites desta lei.

InviolAtosProfissionais \sqsubseteq Advogado

A.1.3**Art. 3º**

O exercício da atividade de advocacia no território brasileiro e a denominação de advogado são privativos dos inscritos na Ordem dos Advogados do Brasil (OAB)

Advogado \sqsubseteq InscritoOAB

§ 1º Exercem atividade de advocacia, sujeitando-se ao regime desta lei, além do regime próprio a que se subordinem, os integrantes da Advocacia-Geral da União, da Procuradoria da Fazenda Nacional, da Defensoria Pública e das Procuradorias e Consultorias Jurídicas dos Estados, do Distrito Federal, dos Municípios e das respectivas entidades de administração indireta e fundacional.

IntegranteAGU \sqsubseteq Advogado

IntegrantePFZ \sqsubseteq Advogado

IntegranteDP \sqsubseteq Advogado

IntegrantePCJEst \sqsubseteq Advogado (uma para cada estado)

IntegrantePCJDF \sqsubseteq Advogado

IntegrantePCJM \sqsubseteq Advogado (uma para cada município)

§ 2º O estagiário de advocacia, regularmente inscrito, pode praticar os atos previstos no art. 1º, na forma do regimento geral, em conjunto com advogado e sob responsabilidade deste.

(\forall supervisiona.EstagiarioAdv) \sqsubseteq Advogado

A.1.4**Art. 4º**

São nulos os atos privativos de advogado praticados por pessoa não inscrita na OAB, sem prejuízo das sanções civis, penais e administrativas.

\neg Advogado \sqsubseteq AtosNulos

Parágrafo único. São também nulos os atos praticados por advogado impedido - no âmbito do impedimento - suspenso, licenciado ou que passar a exercer atividade incompatível com a advocacia.

AdvImpedido \sqsubseteq \neg Advogado

AdvSuspenso \sqsubseteq AdvImpedido

AdvLicenciado \sqsubseteq AdvImpedido

AtivIncompatAdv \sqsubseteq AdvImpedido

A.1.5**Art. 5º**

O advogado postula, em juízo ou fora dele, fazendo prova do mandato.

Advogado \sqsubseteq Postulado

Postulado \sqsubseteq ProvaMandato

$(PostJuizo \sqcup PostForaJuizo) \sqsubseteq Postulado$

§ 1º O advogado, afirmando urgência, pode atuar sem procuração, obrigando-se a apresentá-la no prazo de quinze dias, prorrogável por igual período.

$Advogado \sqsubseteq AtoSemProcuracao$

$(\forall ate15dias.AtoSemProcuracao) \sqsubseteq Procuracao$

$(\forall ate15dias.AtoSemProcuracao) \sqsubseteq AtoSemProcuracao$

§ 2º A procuração para o foro em geral habilita o advogado a praticar todos os atos judiciais, em qualquer juízo ou instância, salvo os que exijam poderes especiais.

$ProcuracaoGeral \sqsubseteq Procuracao$

$ProcuracaoForoGeral \sqsubseteq AtosJudiciaisRegulares$

$AtosJudiciaisRegulares \sqsubseteq AtosJudiciais$

$AtosJudiciaisEspeciais \sqsubseteq AtosJudiciais$

$Advogado \sqsubseteq AtosJudiciaisRegulares$

§ 3º O advogado que renunciar ao mandato continuará, durante os dez dias seguintes à notificação da renúncia, a representar o mandante, salvo se for substituído antes do término desse prazo.

$(\forall apos10dias.AdvRenunciado) \sqsubseteq \neg Advogado$

$\neg (\exists apos10dias.AdvRenunciado) \sqsubseteq Advogado$

$AdvRenunciado \sqcap AdvSubstituido \sqsubseteq Advogado$

A.2

Capítulo II - Dos Direitos do Advogado

A.2.1

Art. 6º

Não há hierarquia nem subordinação entre advogados, magistrados e membros do Ministério Público, devendo todos tratar-se com consideração e respeito recíprocos.

$Advogado \sqsubseteq NivelHierarquicoAdv$

$Magistrado \sqsubseteq NivelHierarquicoAdv$

$MembroMP \sqsubseteq NivelHierarquicoAdv$

Parágrafo único. As autoridades, os servidores públicos e os serventuários da justiça devem dispensar ao advogado, no exercício da profissão, tratamento compatível com a dignidade da advocacia e condições adequadas a seu desempenho.

$(\forall trataRespAdv.AutoridadeJus) \sqsubseteq Advogado$

$(\forall trataRespAdv.SevidorPublico) \sqsubseteq Advogado$

$(\forall trataRespAdv.ServentuarioJus) \sqsubseteq Advogado$

A.2.2

Art. 7º

São direitos do advogado:

I - exercer, com liberdade, a profissão em todo o território nacional;

Advogado \sqsubseteq ExercAdvBR

II – a inviolabilidade de seu escritório ou local de trabalho, bem como de seus instrumentos de trabalho, de sua correspondência escrita, eletrônica, telefônica e telemática, desde que relativas ao exercício da advocacia;

Advogado \sqsubseteq ObjInviolaveisAdv

(\forall posse.Escritorio) \sqsubseteq ObjInviolaveisAdv

(\forall posse.Instrumentos) \sqsubseteq ObjInviolaveisAdv

(\forall posse.Correspondencia) \sqsubseteq ObjInviolaveisAdv

III - comunicar-se com seus clientes, pessoal e reservadamente, mesmo sem procuração, quando estes se acharem presos, detidos ou recolhidos em estabelecimentos civis ou militares, ainda que considerados incomunicáveis;

ClientePreso \sqsubseteq Cliente

ClienteDetido \sqsubseteq Cliente

ClienteRecolhido \sqsubseteq Cliente

Advogado \sqsubseteq (\forall comunPriv.Cliente)

IV - ter a presença de representante da OAB, quando preso em flagrante, por motivo ligado ao exercício da advocacia, para lavratura do auto respectivo, sob pena de nulidade e, nos demais casos, a comunicação expressa à seccional da OAB;

PrisaoExercAdv \sqsubseteq Prisao

\neg (PrisaoExercAdv \sqcap \neg RepresentanteOAB \sqsubseteq Advogado)

V - não ser recolhido preso, antes de sentença transitada em julgado, senão em sala de Estado Maior, com instalações e comodidades condignas, assim reconhecidas pela OAB, e, na sua falta, em prisão domiciliar;

Advogado \sqsubseteq ((PrisaoDom \sqcup SalaEstMaior) \sqcap Julgamento)

VI - ingressar livremente:

a) nas salas de sessões dos tribunais, mesmo além dos cancelos que separam a parte reservada aos magistrados;

Advogado \sqsubseteq LivreIngrSalaTribunal

b) nas salas e dependências de audiências, secretarias, cartórios, ofícios de justiça, serviços notariais e de registro, e, no caso de delegacias e prisões, mesmo fora da hora de expediente e independentemente da presença de seus titulares;

Advogado \sqsubseteq LivreIngrSalaAud

Advogado \sqsubseteq LivreIngrSalaSec

Advogado \sqsubseteq LivreIngrSalaCart

$\text{Advogado} \sqsubseteq \text{LivreIngrSalaOJ}$

$\text{Advogado} \sqsubseteq \text{LivreIngrSalaSN}$

$\text{Advogado} \sqsubseteq \text{LivreIngrSalaSR}$

$\text{Advogado} \sqsubseteq \text{LivreIngrSalaDel} \sqcap \text{LivreIngrSalaDelForaExp}$

$\text{Advogado} \sqsubseteq \text{LivreIngrSalaPri} \sqcap \text{LivreIngrSalaPriForaExp}$

$\text{Advogado} \sqsubseteq \text{LivreIngrSalaAud}$

c) em qualquer edifício ou recinto em que funcione repartição judicial ou outro serviço público onde o advogado deva praticar ato ou colher prova ou informação útil ao exercício da atividade profissional, dentro do expediente ou fora dele, e ser atendido, desde que se ache presente qualquer servidor ou empregado;

$\text{Advogado} \sqsubseteq \text{LivreIngrRepJudicial}$

$\text{Advogado} \sqsubseteq \text{LivreIngrSerExercAdv}$

d) em qualquer assembleia ou reunião de que participe ou possa participar o seu cliente, ou perante a qual este deva comparecer, desde que munido de poderes especiais;

$(\text{Advogado} \sqcap \text{ClienteEsp}) \sqsubseteq \text{LivreIngrAss}$

$\text{ClienteEsp} \sqsubseteq \text{Cliente}$

VII - permanecer sentado ou em pé e retirar-se de quaisquer locais indicados no inciso anterior, independentemente de licença;

$\text{Advogado} \sqsubseteq \text{DireitoPe}$

$\text{Advogado} \sqsubseteq \text{DireitoSentado}$

$\text{Advogado} \sqsubseteq \text{DireitoSair}$

VIII - dirigir-se diretamente aos magistrados nas salas e gabinetes de trabalho, independentemente de horário previamente marcado ou outra condição, observando-se a ordem de chegada;

$(\forall \text{dirigirseSalas.Magistrado}) \sqsubseteq \text{Advogado}$

IX - (REMOVIDO);

X - usar da palavra, pela ordem, em qualquer juízo ou tribunal, mediante intervenção sumária, para esclarecer equívoco ou dúvida surgida em relação a fatos, documentos ou afirmações que influam no julgamento, bem como para replicar acusação ou censura que lhe forem feitas;

$\text{Advogado} \sqsubseteq \text{EsclarecerVerb}$

$\text{Advogado} \sqsubseteq \text{ReplicarAcus}$

$\text{Advogado} \sqsubseteq \text{ReplicarCens}$

XI - reclamar, verbalmente ou por escrito, perante qualquer juízo, tribunal ou autoridade, contra a inobservância de preceito de lei, regulamento ou regimento;

$\text{Advogado} \sqsubseteq \text{ReclamarInobservLeiVerb}$

$\text{Advogado} \sqsubseteq \text{ReclamarInobservLeiEscr}$

XII - falar, sentado ou em pé, em juízo, tribunal ou órgão de deliberação coletiva da Administração Pública ou do Poder Legislativo;

Advogado \sqsubseteq *FalarOrgaoAdmPub*

Advogado \sqsubseteq *FalarOrgaoPoderLeg*

XIII - examinar, em qualquer órgão dos Poderes Judiciário e Legislativo, ou da Administração Pública em geral, autos de processos findos ou em andamento, mesmo sem procuração, quando não estejam sujeitos a sigilo, assegurada a obtenção de cópias, podendo tomar apontamentos;

Advogado \sqsubseteq *ExamAutosProcNaoSigilo*

Advogado \sqsubseteq *CopiAutosProcNaoSigilo*

XIV - examinar, em qualquer instituição responsável por conduzir investigação, mesmo sem procuração, autos de flagrante e de investigações de qualquer natureza, findos ou em andamento, ainda que conclusos à autoridade, podendo copiar peças e tomar apontamentos, em meio físico ou digital;

ExamAutosFlagInvestigacao \sqsubseteq *ExamAutosProcNaoSigilo*

CopiAutosFlagInvestigacao \sqsubseteq *CopiAutosProcNaoSigilo*

XV - ter vista dos processos judiciais ou administrativos de qualquer natureza, em cartório ou na repartição competente, ou retirá-los pelos prazos legais;

Advogado \sqsubseteq *RetirarProcCartorio*

Advogado \sqsubseteq *RetirarProcRepartComp*

Advogado \sqsubseteq *ExaminarProcCartorio*

Advogado \sqsubseteq *ExaminarProcRepartComp*

XVI - retirar autos de processos findos, mesmo sem procuração, pelo prazo de dez dias;

Advogado \sqsubseteq *RetirarAutosDezDias*

XVII - ser publicamente desagravado, quando ofendido no exercício da profissão ou em razão dela;

Advogado \sqsubseteq *PublicDesagravado*

XVIII - usar os símbolos privativos da profissão de advogado;

Advogado \sqsubseteq *SimbolosPrivAdv*

XIX - recusar-se a depor como testemunha em processo no qual funcionou ou deva funcionar, ou sobre fato relacionado com pessoa de quem seja ou foi advogado, mesmo quando autorizado ou solicitado pelo constituinte, bem como sobre fato que constitua sigilo profissional;

Advogado \sqsubseteq *RecusarDeporEnvolvimento*

Advogado \sqsubseteq *RecusarDeporSigiloProf*

XX - retirar-se do recinto onde se encontre aguardando pregão para ato judicial, após trinta minutos do horário designado e ao qual ainda não tenha com-

parecido a autoridade que deva presidir a ele, mediante comunicação protocolizada em juízo.

Advogado \sqsubseteq *SairPregaoPos30Min*

XXI - assistir a seus clientes investigados durante a apuração de infrações, sob pena de nulidade absoluta do respectivo interrogatório ou depoimento e, subsequentemente, de todos os elementos investigatórios e probatórios dele decorrentes ou derivados, direta ou indiretamente, podendo, inclusive, no curso da respectiva apuração:

Advogado \sqsubseteq *AssistirClienteInvestigado*

Advogado \sqsubseteq *AnularInterr*

Advogado \sqsubseteq *AnularDepo*

Advogado \sqsubseteq *AnularConseqInterr*

Advogado \sqsubseteq *AnularConseqDepo*

a) apresentar razões e quesitos;

Advogado \sqsubseteq *ApresRazoesQuesInterr*

Advogado \sqsubseteq *ApresRazoesQuesDepo*

b) (VETADO).

§ 1º Não se aplica o disposto nos incisos XV e XVI:

1) aos processos sob regime de segredo de justiça;

RetirarProcCartorioSegJust \sqsubseteq \neg *RetirarProcCartorio*

RetirarProcRepartCompSegJust \sqsubseteq \neg *RetirarProcRepartComp*

ExaminarProcCartorioSegJust \sqsubseteq \neg *ExaminarProcCartorio*

ExaminarProcRepartCompSegJust \sqsubseteq \neg *ExaminarProcRepartComp*

2) quando existirem nos autos documentos originais de difícil restauração ou ocorrer circunstância relevante que justifique a permanência dos autos no cartório, secretaria ou repartição, reconhecida pela autoridade em despacho motivado, proferido de ofício, mediante representação ou a requerimento da parte interessada;

RetirarAutosDifRest \sqsubseteq \neg *RetirarAutosDezDias*

RetirarAutosPerma \sqsubseteq \neg *RetirarAutosDezDias*

3) até o encerramento do processo, ao advogado que houver deixado de devolver os respectivos autos no prazo legal, e só o fizer depois de intimado.

RetirarAutosSemDevol \sqsubseteq \neg *RetirarAutosDezDias*

§ 2º O advogado tem imunidade profissional, não constituindo injúria, difamação ou desacato puníveis qualquer manifestação de sua parte, no exercício de sua atividade, em juízo ou fora dele, sem prejuízo das sanções disciplinares perante a OAB, pelos excessos que cometer.

Advogado \sqsubseteq *ImunidadeProfissional*

§ 3º O advogado somente poderá ser preso em flagrante, por motivo de exercício da profissão, em caso de crime inafiançável, observado o disposto no inciso IV deste artigo.

$PresoFlagrante \sqsubseteq Preso$

$CrimeInafiancavel \sqsubseteq Crime$

$Advogado \sqsubseteq (\neg CrimeInafiancavel \sqcup \neg PresoFlagrante)$

§ 4º O Poder Judiciário e o Poder Executivo devem instalar, em todos os juizados, fóruns, tribunais, delegacias de polícia e presídios, salas especiais permanentes para os advogados, com uso e controle assegurados à OAB.

$Advogado \sqsubseteq SalaEspecial$

§ 5º No caso de ofensa a inscrito na OAB, no exercício da profissão ou de cargo ou função de órgão da OAB, o conselho competente deve promover o desagravo público do ofendido, sem prejuízo da responsabilidade criminal em que incorrer o infrator.

$InscritoOAB \sqsubseteq (OfensaProf \sqcup OfensaCargoOAB) \sqcap Desagravo$

§ 6º Presentes indícios de autoria e materialidade da prática de crime por parte de advogado, a autoridade judiciária competente poderá decretar a quebra da inviolabilidade de que trata o inciso II do caput deste artigo, em decisão motivada, expedindo mandado de busca e apreensão, específico e pormenorizado, a ser cumprido na presença de representante da OAB, sendo, em qualquer hipótese, vedada a utilização dos documentos, das mídias e dos objetos pertencentes a clientes do advogado averiguado, bem como dos demais instrumentos de trabalho que contenham informações sobre clientes.

ALTERAÇÃO DO INCISO II, NAS FÓRMULAS MENCIONANDO ESCRITÓRIO E INSTRUMENTOS

$(\forall posse.Escritorio) \sqcap \neg Crime \sqsubseteq ObjInviolaveisAdv$

$(\forall posse.Instrumentos) \sqcap \neg Crime \sqsubseteq ObjInviolaveisAdv$

§ 7º A ressalva constante do § 6º deste artigo não se estende a clientes do advogado averiguado que estejam sendo formalmente investigados como seus partícipes ou co-autores pela prática do mesmo crime que deu causa à quebra da inviolabilidade.

ALTERAÇÃO DO INCISO II, NA FÓRMULA MENCIONANDO CORRESPONDÊNCIAS

$(\forall posse.Correspondencia) \sqcap \neg CrimeCliente \sqsubseteq ObjInviolaveisAdv$

§ 8º (VETADO)

§ 9º (VETADO)

§ 10. Nos autos sujeitos a sigilo, deve o advogado apresentar procuração para o exercício dos direitos de que trata o inciso XIV.

ALTERAÇÃO NO INCISO XIV

$Advogado \sqsubseteq (ExamAutosFlagInvestigacao \sqcap Procuracao)$

$Advogado \sqsubseteq (CopiAutosFlagInvestigacao \sqcap Procuracao)$

§ 11. No caso previsto no inciso XIV, a autoridade competente poderá delimitar o acesso do advogado aos elementos de prova relacionados a diligências em andamento e ainda não documentados nos autos, quando houver risco de comprometimento da eficiência, da eficácia ou da finalidade das diligências.

ALTERAÇÃO NO §10º

$Advogado \sqsubseteq$

$(ExamAutosFlagInvestigacao \sqcap Procuracao \sqcap AcessoLimitado)$

$Advogado \sqsubseteq$

$(CopiAutosFlagInvestigacao \sqcap Procuracao \sqcap AcessoLimitado)$

§ 12. A inobservância aos direitos estabelecidos no inciso XIV, o fornecimento incompleto de autos ou o fornecimento de autos em que houve a retirada de peças já incluídas no caderno investigativo implicará responsabilização criminal e funcional por abuso de autoridade do responsável que impedir o acesso do advogado com o intuito de prejudicar o exercício da defesa, sem prejuízo do direito subjetivo do advogado de requerer acesso aos autos ao juiz competente.

$ImpedirAcessoAutos \sqsubseteq AbusoAutoridade$

$AbusoAutoridade \sqsubseteq Crime$

A.2.3

Art. 7º-A.

São direitos da advogada:

$Advogada \sqsubseteq Advogado$

I - gestante:

$AdvGestante \sqsubseteq Advogada$

a) entrada em tribunais sem ser submetida a detectores de metais e aparelhos de raios X;

$AdvGestante \sqsubseteq EntrSemRaioX$

b) reserva de vaga em garagens dos fóruns dos tribunais;

$AdvGestante \sqsubseteq VagaForum$

II - lactante, adotante ou que der à luz, acesso a creche, onde houver, ou a local adequado ao atendimento das necessidades do bebê;

$AdvLactante \sqsubseteq Advogada$

$AdvAdotante \sqsubseteq Advogada$

$AdvDarLuz \sqsubseteq Advogada$

$AdvLactante \sqsubseteq AcessoCreche$

$AdvAdotante \sqsubseteq AcessoCreche$

$AdvDarLuz \sqsubseteq AcessoCreche$

III - gestante, lactante, adotante ou que der à luz, preferência na ordem das sustentações orais e das audiências a serem realizadas a cada dia, mediante comprovação de sua condição;

$(AdvGestante \sqcap ComprovacaoGest) \sqsubseteq PrefAudiencias$

$(AdvLactante \sqcap ComprovacaoLact) \sqsubseteq PrefAudiencias$

$(AdvAdotante \sqcap ComprovacaoAdot) \sqsubseteq PrefAudiencias$

$(AdvDarLuz \sqcap ComprovacaoDarLuz) \sqsubseteq PrefAudiencias$

IV - adotante ou que der à luz, suspensão de prazos processuais quando for a única patrona da causa, desde que haja notificação por escrito ao cliente.

$(AdvAdotante \sqcap NotifCliente) \sqsubseteq SuspPrazos$

$(AdvDarLuz \sqcap NotifCliente) \sqsubseteq SuspPrazos$

§ 1º Os direitos previstos à advogada gestante ou lactante aplicam-se enquanto perdurar, respectivamente, o estado gravídico ou o período de amamentação.

§ 2º Os direitos assegurados nos incisos II e III deste artigo à advogada adotante ou que der à luz serão concedidos pelo prazo previsto no art. 392 do Decreto-Lei no 5.452, de 1º de maio de 1943 (Consolidação das Leis do Trabalho).

§ 3º O direito assegurado no inciso IV deste artigo à advogada adotante ou que der à luz será concedido pelo prazo previsto no § 6º do art. 313 da Lei no 13.105, de 16 de março de 2015 (Código de Processo Civil).

A.3

Capítulo III - Da Inscrição

A.3.1

Art. 8º

Para inscrição como advogado é necessário:

I - capacidade civil;

II - diploma ou certidão de graduação em direito, obtido em instituição de ensino oficialmente autorizada e credenciada;

III - título de eleitor e quitação do serviço militar, se brasileiro;

IV - aprovação em Exame de Ordem;

V - não exercer atividade incompatível com a advocacia;

VI - idoneidade moral;

VII - prestar compromisso perante o conselho.

$\neg (CapacidadeCivil \sqcap DiplomaGradDirBR$

$\sqcap (TitEleitor \sqcap MilitarBr) \sqcap AprovadoOAB \sqcap \neg AtivIncompatlAdv$

$\sqcap IdoneidadeMoral \sqcap CompromConselho)$

$\sqsubseteq \neg Advogado$

§ 1º O Exame da Ordem é regulamentado em provimento do Conselho Federal da OAB.

$$RegExameOAB \sqsubseteq ConsFedOAB$$

$$ConsFedOAB \sqsubseteq RegExameOAB$$

§ 2º O estrangeiro ou brasileiro, quando não graduado em direito no Brasil, deve fazer prova do título de graduação, obtido em instituição estrangeira, devidamente revalidado, além de atender aos demais requisitos previstos neste artigo.

$$DiplomaGradDirBr \sqsubseteq DiplomaGradDir$$

$$(DiplomaGradDir \sqcap ValidaDiploma) \sqsubseteq DiplomaGradDirBr$$

§ 3º A inidoneidade moral, suscitada por qualquer pessoa, deve ser declarada mediante decisão que obtenha no mínimo dois terços dos votos de todos os membros do conselho competente, em procedimento que observe os termos do processo disciplinar.

$$DoisTercosConselho \sqsubseteq IdoneidadeMoral$$

§ 4º Não atende ao requisito de idoneidade moral aquele que tiver sido condenado por crime infamante, salvo reabilitação judicial.

$$CrimeInfamante \sqsubseteq Crime$$

$$(CrimeInfamante \sqcap \neg ReabJudicial) \sqsubseteq \neg IdoneidadeMoral$$

A.3.2

Art. 9º

Para inscrição como estagiário é necessário:

I - preencher os requisitos mencionados nos incisos I, III, V, VI e VII do art. 8º;

$$\neg(CapacidadeCivil \sqcap (TitEleitor \sqcap MilitarBr)) \sqcap \neg AtivIncompatAdv \sqcap IdoneidadeMoral \sqcap CompromConselho \sqsubseteq \neg EstagiarioAdv$$

II - ter sido admitido em estágio profissional de advocacia.

$$EstagiarioProfAdv \sqsubseteq EstagiarioAdv$$

§ 1º O estágio profissional de advocacia, com duração de dois anos, realizado nos últimos anos do curso jurídico, pode ser mantido pelas respectivas instituições de ensino superior pelos Conselhos da OAB, ou por setores, órgãos jurídicos e escritórios de advocacia credenciados pela OAB, sendo obrigatório o estudo deste Estatuto e do Código de Ética e Disciplina.

$$\neg(InstEnsSup \sqcap AprovacaoOAB) \sqsubseteq \neg EstagiarioProfAdv$$

§ 2º A inscrição do estagiário é feita no Conselho Seccional em cujo território se localize seu curso jurídico.

$$EstagiarioAdv \sqsubseteq InscricaoCSLocalCurso$$

§ 3º O aluno de curso jurídico que exerça atividade incompatível com a advocacia pode frequentar o estágio ministrado pela respectiva instituição de ensino superior, para fins de aprendizagem, vedada a inscrição na OAB.

$AtivIncompatAdv \sqcap BacharelDir \sqsubseteq EstagiarioAcadDir$

$EstagiarioAcadDir \sqsubseteq EstagiarioAdv$

§ 4º O estágio profissional poderá ser cumprido por bacharel em Direito que queira se inscrever na Ordem.

$(BacharelDir \sqcap InscritoOAB) \sqsubseteq EstagiarioProfAdv$

A.3.3

Art. 10

A inscrição principal do advogado deve ser feita no Conselho Seccional em cujo território pretende estabelecer o seu domicílio profissional, na forma do regulamento geral.

$Advogado \sqsubseteq \forall local. (InscricaoPrinSC \sqsubseteq DomProfissional)$

§ 1º Considera-se domicílio profissional a sede principal da atividade de advocacia, prevalecendo, na dúvida, o domicílio da pessoa física do advogado.

$DomProfissional \sqsubseteq DomPFAdv$

§ 2º Além da principal, o advogado deve promover a inscrição suplementar nos Conselhos Seccionais em cujos territórios passar a exercer habitualmente a profissão considerando-se habitualidade a intervenção judicial que exceder de cinco causas por ano.

$\neg InscricaoSupl \sqsubseteq \neg Advogado$

§ 3º No caso de mudança efetiva de domicílio profissional para outra unidade federativa, deve o advogado requerer a transferência de sua inscrição para o Conselho Seccional correspondente.

$MudancaDomProf \sqsubseteq TransfInscricaoSC$

§ 4º O Conselho Seccional deve suspender o pedido de transferência ou de inscrição suplementar, ao verificar a existência de vício ou ilegalidade na inscrição principal, contra ela representando ao Conselho Federal.

$IlegalInscricao \sqsubseteq (\neg InscricaoSuplementar \sqcap \neg TransfInscricaoCS)$

A.3.4

Art. 11

Cancela-se a inscrição do profissional que:

I - assim o requerer;

$ReqCancelInsc \sqsubseteq \neg InscricaoPrinCS$

II - sofrer penalidade de exclusão;

$PenalidadeExclusao \sqsubseteq \neg InscricaoPrinCS$

III - falecer;

Falecido $\sqsubseteq \neg \text{InscricaoPrinCS}$

IV - passar a exercer, em caráter definitivo, atividade incompatível com a advocacia;

AtivIncompativelAdv $\sqsubseteq \neg \text{InscricaoPrinCS}$

V - perder qualquer um dos requisitos necessários para inscrição.

§ 1º Ocorrendo uma das hipóteses dos incisos II, III e IV, o cancelamento deve ser promovido, de ofício, pelo conselho competente ou em virtude de comunicação por qualquer pessoa.

(PenalidadeExclusao \sqcup *Falecido* \sqcup *AtivIncompativelAdv*)

$\sqsubseteq (\text{CancelConselho} \sqcup \text{CancelComunic})$

CancelConselho $\sqsubseteq \neg \text{InscricaoPrinCS}$

CancelComunic $\sqsubseteq \neg \text{InscricaoPrinCS}$

§ 2º Na hipótese de novo pedido de inscrição - que não restaura o número de inscrição anterior - deve o interessado fazer prova dos requisitos dos incisos I, V, VI e VII do art. 8º.

SegInscricao $\sqsubseteq \text{Advogado}$

SegInscricao $\sqsubseteq \text{InscricaoPrinCS}$

$\neg(\text{CapacidadeCivil} \sqcap \neg \text{AtivIncompatAdv} \sqcap \text{IdoneidadeMoral} \sqcap \text{CompromConselho}) \sqsubseteq \neg \text{SecInscricao}$

§ 3º Na hipótese do inciso II deste artigo, o novo pedido de inscrição também deve ser acompanhado de provas de reabilitação.

ALTERAÇÃO DO § 2º, NA TERCEIRA FÓRMULA

$\neg(\text{CapacidadeCivil} \sqcap \neg \text{AtivIncompatAdv} \sqcap \text{IdoneidadeMoral} \sqcap \text{CompromConselho} \sqcap (\text{PenalidadeExclusao} \sqsubseteq \text{ProvasReab})) \sqsubseteq \neg \text{SecInscricao}$

A.3.5

Art. 12

Licencia-se o profissional que:

Licenciado $\sqsubseteq \neg \text{Advogado}$

I - assim o requerer, por motivo justificado;

ReqJustLicenca $\sqsubseteq \text{Licenciado}$

II - passar a exercer, em caráter temporário, atividade incompatível com o exercício da advocacia;

AtivIncompatAdv $\sqsubseteq \text{Licenciado}$

III - sofrer doença mental considerada curável.

DoencaMentalCuravel $\sqsubseteq \text{Licenciado}$

A.3.6**Art. 13**

O documento de identidade profissional, na forma prevista no regulamento geral, é de uso obrigatório no exercício da atividade de advogado ou de estagiário e constitui prova de identidade civil para todos os fins legais.

$\neg \text{DocIdentidadeProf} \sqsubseteq \neg \text{Advogado}$

$\neg \text{DocIdentidadeProf} \sqsubseteq \neg \text{EstagiarioAdv}$

A.3.7**Art. 14**

É obrigatória a indicação do nome e do número de inscrição em todos os documentos assinados pelo advogado, no exercício de sua atividade.

$\text{Advogado} \sqsubseteq \text{DocAssinado}$

Parágrafo único. É vedado anunciar ou divulgar qualquer atividade relacionada com o exercício da advocacia ou o uso da expressão escritório de advocacia, sem indicação expressa do nome e do número de inscrição dos advogados que o integrem ou o número de registro da sociedade de advogados na OAB.

$\text{Advogado} \sqsubseteq \neg \text{AnunciarSemInscNome}$