Universidade de Brasília
Instituto de Ciências Humanas
Programa de Pós-Graduação em Filosofia

Raciocínio e ação: A lógica das transações aplicada à resolução de problemas

Hítalo Fernandes de Oliveira

Brasília-DF
2020

Hítalo Fernandes de Oliveira

Raciocínio e ação: A lógica das transações aplicada à resolução de problemas

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Filosofia da UnB como parte dos requisitos necessários para a obtenção do Título de Mestre em Filosofia.

Universidade de Brasília – UnB

Programa de Pós-Graduação em Filosofia - PPG-FIL

Orientador: Alexandre Costa-Leite

Brasília-DF

2020

Hítalo Fernandes de Oliveira

Raciocínio e ação: A lógica das transações aplicada à resolução de problemas

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Filosofia da UnB como parte dos requisitos necessários para a obtenção do Título de Mestre em Filosofia.

Trabalho aprovado. Brasília-DF, 2020:

Alexandre Costa-Leite
Orientador

Alfredo Roque Freire

Fabien Schang

Brasília-DF
2020

Aos meus pais Hilton Jarbas de Oliveira e Rosa Helena Fernandes de Oliveira.

Agradecimentos

Agradeço a Deus.

Agradeço aos meus pais, Hilton Jarbas de Oliveira e Rosa Helena Fernandes de Oliveira, por tudo que eles me ensinaram, pelas oportunidades que me deram e por sempre estarem comigo, me incentivando em todos os momentos.

Agradeço ao meu orientador, Alexandre Costa-Leite, por todos os conselhos e pela paciência.

Aos professores e colegas que contribuíram com ideias, sugestões e críticas.

A todos meus amigos que compreenderam o meu desaparecimento durante o período em que escrevia este trabalho.

A CAPES pelo apoio financeiro que contribui para a realização deste trabalho.

"Escrito está: "Era no início o Verbo!"
Começo apenas, e já me exacerbo!
Como hei de ao verbo dar tão alto apreço?
De outra interpretação careço;
Se o espírito me deixa esclarecido,
Escrito está: No início era o Sentido!
Pesa a linha inicial com calma plena,
Não se apressure a tua pena!
É o sentido então, que tudo opera e cria?
Deverá opor! No início era a Energia!
Mas, já, enquanto assim o retifico,
Diz-me algo que tampouco nisso fico.
Do espírito me vale a direção,
E escrevo em paz: Era no início a Ação!
(GOETHE, Johann Wolfgang.)

Resumo

Neste trabalho analisamos as relações entre conhecimento, resolução de problemas e lógica formal. O objetivo específico desta dissertação é examinar em que medida a lógica das transações ($\mathcal{TR}$) permite representar os conhecimentos procedural e declarativo, bem como agentes capazes de utilizar esses tipos de conhecimentos para resolver problemas bem definidos. Para realizar tal tarefa, apresentamos no primeiro capítulo algumas definições de problema e de conhecimento, discutimos os principais aspectos envolvidos na resolução de problemas e analisamos a relação entre ações e intenções. No segundo capítulo, apresentamos uma introdução a lógica das transações. No terceiro capítulo, mostramos como a lógica das transações pode ser aplicada no contexto da resolução de problemas e apresentamos algumas ideias que poderiam ser utilizadas para desenvolver linguagens que superassem algumas limitações de $\mathcal{TR}$. Concluímos que apesar de $\mathcal{TR}$ cumprir sua missão de representar o conhecimento e agentes que resolvem problemas bem definidos, existem alguns aspectos que poderiam ser modificados para criar lógicas mais adequados ainda para tal tarefa.

Palavras-chaves: Lógica das Transações. Conhecimento procedural. Conhecimento declarativo. Resolução de Problemas. Ações intencionais.

Abstract

In this paper we analyze the relationship between knowledge, problem solving and formal logic. The specific objective of this dissertation is to examine in what extent the logic of transactions ($\mathcal{TR}$) allows to represent procedural and declarative knowledge, as well as agents capable of using these types of knowledge to solve well-defined problems. To accomplish this task, we present in the first chapter some definitions of knowledge problem, discuss the main aspects involved in problem solving and analyze the relationship between actions and intentions. In the second chapter, we present an introduction to the logic of transactions. In the third chapter, we show how transaction logic can be applied in the context of problem solving and present some ideas that could be used to develop languages that overcome some limitations of $\mathcal{TR}$. We conclude that although $\mathcal{TR}$ fulfills its mission of representing knowledge and agents that solve well-defined problems, there are some aspects that could be modified to create even more adequate logic for this task.

Keywords: l. Transaction Logic. Procedural Knowledge. Declarative Knowledge. Intentional actions.

Sumário

Introdução

A resolução de problemas, dos mais variados tipos, sempre esteve presente em todas as sociedades desde os primórdios da humanidade. Não são apenas os seres humanos que enfrentam problemas, diversas espécies de animais também dependem, para sobreviver, da habilidade dos indivíduos (ou grupos de indivíduos) de enfrentar e solucionar problemas. Entretanto, os seres humanos não se envolvem apenas com problemas práticos, mas também com questões conceituais, lógicas, metafísicas, epistemológicas, éticas etc. e podem utilizar a linguagem como ferramenta na resolução desses problemas.

Para investigar o mundo e sua estrutura, a humanidade desenvolveu diversas disciplinas que tratam de diferentes fragmentos daquilo que nos cerca e a tentativa de resolver problemas é justamente o que está na base do desenvolvimento dessas disciplinas. Além da resolução de problemas, o desenvolvimento de uma disciplina depende da existência diversos mecanismos que dão suporte á resolução de problemas, como por exemplo, políticas de fomento e meios de divulgação dos trabalhos. Mas, no cerne do desenvolvimento de uma disciplina está a resolução de problemas. Se um indivíduo pretende deixar de ser apenas um consumidor de conhecimento e contribuir com suas próprias produções originais, então, para satisfazer essa pretensão, ele precisa reconhecer ou propor problemas em sua área de pesquisa e fornecer tentativas de solução. Casos os praticantes de uma disciplina não realizem esse tipo de atividade, sua área de pesquisa ficará estagnada. Em suma, para alguém contribuir significativamente em uma área de pesquisa é preciso mais que conhecer e dominar um conjunto de declarações e procedimentos acumulados ao longo da história, é preciso atacar os problemas presentes na disciplina. Para desenvolver, por exemplo, uma prática filosófica que contribua com a filosofia, não é suficiente estar familiarizado e dominar certos textos canônicos, pois conhecer a história da filosofia, não é o mesmo que filosofar (ainda que a prática filosófica esteja vinculada ao conhecimento de sua história). De modo similar, o pensamento matemático (para citar mais um exemplo) não se desenvolve com a simples aprendizagem de certos conteúdos, é preciso haver a formação de atitudes e crenças que permitam aos matemáticos representar novos problemas a partir de uma perspectiva matemática e buscar por soluções de acordo com regras aceitas (tacitamente ou não) por seus pares.

Essa dissertação tem como tema geral as relações entre a resolução de problemas, o conhecimento e a lógica formal. Para estudar tais relações precisamos definir o que significa “problema” e o que significa “conhecimento”. Para tal, apresentamos a forma como a noção de problema aparece na literatura, antes de especificar o que nós entendemos por “problema” e com qual classe de problemas vamos lidar nesta dissertação. Os termos “conhece” e “sabe” possuem alguns significados distintos. Esses diferentes significados indicam a existência de três tipos de conhecimento que precisam ser esclarecidos para podermos entender qual o papel de cada um deles em uma resolução de problema. Por fim, também precisamos tratar da lógica.

Como agentes resolvem problemas de forma lógica? Ou melhor, como a lógica pode ser aplicada na resolução de problemas? Como problemas lógicos são resolvidos? Muitas vezes, para para encontrar a solução de problemas fazemos diversas deduções, essas deduções também permitem determinar se um procedimento pode ser aplicado em dada situação[1] e se uma suposta solução está correta. Além disso, uma análise lógica pode provar que para determinado problema não existe uma solução, evitando que os indivíduos percam seu tempo procurando algo que não existe. Além dessas aplicações aos problemas, também existem problemas lógicos propriamente ditos, que gostaríamos de solucionar. Problemas como: dada uma fórmula qualquer, provar se ela é válida em determinada lógica ou, provar se ela é dedutível de uma teoria. Um aspecto relevante da relação entre problemas e lógica é que:

> "[...] mesmo problemas não matemáticos podem ser formulados como teoremas a serem provados. Muitos quebra-cabeças e problemas da vida real que requerem uma análise do senso comum podem em princípio ser colocados em um formalismo lógico, e então resolvidos com técnicas de encontrar provas." (NILSSON, 1971, p. 7)[2]

Nosso objetivo não é estudar as relações entre problemas com a lógica universal. A lógica universal é uma teoria geral da lógica apresentada nos trabalhos de Jean-Yves Béziau[3]. Nestes trabalhos, uma lógica L é definida como $L = \langle F, \vdash \rangle$, ou seja, como um par, onde F é um conjunto e $\vdash$ é uma relação de $\mathcal{P}(\mathbb{F}) \times F$, chamada relação de dedutibilidade. Nesta perspectiva, o conjunto F é tratado de forma abstrata, e por isso, não é preciso especificar a natureza de seus elementos, nem dar um nome para eles. Além disso, para atingir a generalidade desejada nesta teoria, é utilizada uma teoria de conjuntos na qual o axioma da extensionalidade não é válido. Deste modo, $\{a, a\} \vdash b$ e $\{a\} \vdash b$ não precisam ser identificados, o que permite incluir lógicas, nas qual não se pode efetuar a contração.

O objetivo deste trabalho é investigar a aplicação de uma lógica particular, chamada lógica das transações (abreviadamente $\mathcal{TR}$) na resolução de problemas. A principal relação que investigamos nesta dissertação é a forma como $\mathcal{TR}$ pode ser utilizada para representar diversos tipos de conhecimento e dispositivos capazes de utilizar esse conhecimento na resolução de problemas, o que permite fazer da busca pela solução do problema original em uma busca pela prova de teoremas.

---

1 Diversos procedimentos possuem condições que devem ser satisfeitas antes, durante ou após sua execução para que o procedimento possa ser realizado e podemos fazer deduções para verificar se essas condições serão satisfeitas

2 Tradução nossa do original: "[...] *even nonmathematical problems can be formuleted as theorems to be proved. Many puzzles as many real-life problems requiring common-sense analysis can in principle be posed in a logical formalism and then solved by proof-finding techniques."*

3 Como em (BÉZIAU, 1994).

Dentre as questões que abordamos neste trabalho, podemos citar as seguintes: o que significa problema? Qualquer tarefa ou questão, mesmo as triviais, devem ser vistas como problemas? Há alguma definição precisa para o termo "problema " que capture as ideias que temos do termo, ou estamos presos à uma noção informal? O que é uma solução? O que significa "conhecimento"? O que é a lógica das transações? A lógica das transações pode ser aplicada para representar quais tipos de problema? Como podemos representar por meio da lógica das transações os diferentes conhecimentos que um agente utiliza para atacar os problemas?

A proposta aqui é tentar esclarecer as relações entre determinada classe de problemas e o conhecimento, aplicando a lógica das transações. Tal esclarecimento é realizado com base nos seguintes trabalhos: *Mathematical Problem Solving* (SCHOENFELD, 2014), Aspectos de uma Teoria Geral dos Problemas (VELOSO, 1984), Inteligência Artificial (LUGER, 2013), *Problem-solving Methods in Artificial Intelligence* (NILSSON, 1971) e *Transaction Logic Programming* (BONNER; KIFER, 1995).

No primeiro texto, Schoenfeld apresenta uma noção intuitiva de problema e um arcabouço teórico elencando quatro componentes para explicar o comportamento das pessoas quando elas estão resolvendo problemas matemáticos. Os quatro componentes são: o conhecimento do agente sobre o domínio do problema (recursos); as estratégias que ajudam a fazer o problema avançar em momentos de dificuldade (heurística); a capacidade de o agente controlar e monitorar os recursos que possui - como o tempo - de forma eficiente (controle); e as atitudes e crenças do indivíduo que determinam quais ferramentas ele considera relevantes para solucionar o problema (sistema de crenças e atitudes).

Apesar dessa estrutura teórica ter sido proposta para elucidar os aspectos envolvidos no comportamento das pessoas, quando estas estão resolvendo problemas matemáticos (de construção de figuras geométricas, por exemplo), concordamos com o autor e acreditamos que os componentes desse arcabouço teórico se aplicam na resolução de problemas de diversas disciplinas. Algumas habilidades, como, por exemplo, a capacidade de monitorar e avaliar o progresso em tempo real, evitando a perda de tempo prolongada em caminhos infrutíferos, são importantes em qualquer área, tanto em resoluções realizadas por seres orgânicos, quanto naquelas realizadas por máquinas. Além disso, se uma pessoa não acredita que uma ferramenta pode ser útil na resolução de um problema, ela não vai empregá-la. Um filosofo que acredita que formalizar uma teoria faz com que ela perca sua "essência", abre mão de uma ferramenta que pode ajudar a corrigir problemas com suas concepções.

Para realizar um estudo rigoroso sobre as propriedades dos problemas é preciso ter uma ideia clara do que é um problema, entretanto, no segundo texto, Veloso aponta para o fato de que apesar de o termo "problema" ser bastante utilizado, diversos autores costumam tratar este conceito de forma vaga, sem fornecer uma definição precisa para o mesmo. Ao invés de fornecer uma definição precisa, é comum encontrar diversos autores fornecendo exemplos dos tipos de problema que eles querem tratar, na esperança de que esse procedimento seja suficiente para delimitar o assunto sem gerar confusões. Este procedimento apresenta algumas limitações, pois não garante que será possível determinar se um candidato qualquer a problema é de fato um problema. Por exemplo, ao tratar de métodos bastante gerais de resolução de problemas aplicáveis à problemas (bem definidos) em diversos domínios, Nilson diz:

> "Nós ainda não fomos precisos sobre o que significa resolver um problema. [...] Nem fomos precisos sobre o que queremos dizer por problema. Provavelmente ninguém produziu uma definição simples para a palavra *problema* que capture completamente o sentido intuitivo que nós desejamos utilizar aqui. Deste modo, ao invés de tentar fornecer uma definição formal, nós vamos começar nossa discussão comum exemplo típico de problema". (NILSSON, 1971, p. 2-3)[4]

O exemplo paradigmático de problema apresentado por Nillson para introduzir os conceitos fundamentais sobre a resolução de problemas é o chamado jogo do oito[5]. Assim como Nilsson, Kolmogoroff não estava tão preocupado em definir o que eram os problemas que apareciam no seu cálculo de problemas. Ao fornecer uma interpretação para a lógica intuicionista baseada na ideia de problemas (matemáticos), ele diz:

> "Nós não vamos definir o que é um problema, mas iremos explicar este conceito por meio de alguns exemplos. Estes são [exemplos de] problemas: 1. Encontrar quatro inteiros x, y, z, n para os quais a relação $x^n + y^n = z^n$, $n > 2$ é o caso. 2. Provar a falsidade da proposição de Fermat. 3. Desenhar um círculo através de três pontos dados (x, y, z). 4. De posse de uma das raízes da equação $ax^2 + bx + c = 0$, encontrar a outra. [...] Nós acreditamos que depois destes exemplos e explicações, o conceito "problema" e "solução de problemas" podem ser utilizados sem desentendimento." (KOLMOGOROV, 1932, p.2)[6]

Diferente dos trabalhos de Nilson e Kolmogoroff, mencionados acima, que estão voltados para as questões "Qual é o problema?" e "como resolvê-los" e no qual a ideia de

---

[4] Tradução nossa do original: *"We have not yet been precise about what it means to solve a problem.* [...] *Neither have we defined what we mean by problem. Probably no one has produced a simple definition of the word problem that captures completely the intuitive meaning that we wish to use here. So rather than attempt a formal definition we shall begin our discussion by considering a typical example problem"*.

[5] Este jogo é um quebra-cabeça que consiste em um arranjo 3x3, no qual oito peças numeradas podem ser movimentadas. Como um espaço nessa estrutura sempre fica vazio, é possível deslocar as peças ao lado deste espaço para ocupar este espaço. O problema é transformar uma configuração inicial em determinada configuração almejada. Uma solução para o problema é uma sequência de movimentos que realiza tal transformação.

[6] Tradução nossa do original: *"we will not define what a problem is, but will explain this concept through a few examples. These are [examples of] problems: 1. To find four integers x,y,z,n for which the relations* $x^n + y^n = z^n$, $n > 2$ *hold. 2. To prove the falsity of the Fermat proposition. 3. To draw a circle through three given points (x,y,z). 4. Provided that one root of the equation* $ax^2 + bx + c = 0$ *is given, to find the other.* [...] *We believe, that after these examples and explanations, the concepts "problem" and "solution of the problem" can be used without misunderstanding"*.

um "cálculo de problemas" é apresentada, Veloso se volta para as questões "o que é um problema?" e "o que é a solução de um problema?", tendo como objetivo clarificar noções que costumam ser apresentadas de forma imprecisa. Para tal, o autor fornece algumas definições estruturais precisas da noção de problema e define as soluções como certas funções (ou descrições de funções) entre os elementos dessas estrutura.

> "Em matemática, noções como teorema, demonstração, etc. recebem uma formulação precisa, como o objetivo de analisar criticamente esses conceitos e seu alcance, e não com a finalidade de ensinar a demonstrar teoremas. Analogamente, nossa teoria se propõe a dar uma formulação precisas para a noção de problema, solução, etc. visando uma análise crítica, sem pretender, pelo menos em primordialmente, ensinar a resolver problemas." (Veloso, 1984, p. 22)

De acordo com Veloso, sua formulação não se aplica apenas a problemas matemáticos, mas é uma formulação geral aplicável a problemas como: montar um aeromodelo; resolver *puzzles* (como o jogo do oito, mencionado acima); classificar espécies biológicas; problemas envolvendo o raciocínio prático (como o problema do macaco e das bananas[7]; e claro, problemas matemáticos (como provar um determinado teorema e encontrar o centro de uma figura geométrica).

Muitos dos exemplos apresentados na literatura sobre problemas são jogos e quebra-cabeças (*puzzles*). Estes problemas são claros e simples, permitindo ilustrar com facilidade diversos aspectos dos problemas e diversos métodos gerais de resolução de problemas. Outra parte dos problemas apresentados consiste em problemas envolvendo o raciocínio prático, cujo entendimento está ao alcance da maioria dos leitores. Nestes casos, apesar de a solução desses problemas não requerer muito conhecimento especializado, encontrar tais soluções pode ser uma tarefa desafiadora e forçar o solucionador a pensar bastante antes de encontrar uma solução. Nos próximos capítulos desta dissertação, quando este tipo de problema aparecer, o motivo de sua escolha não decorre apenas de sua clareza e simplicidade. Esses problemas aparecem nesta dissertação de filosofia porque acreditamos que, apesar desses domínios serem semanticamente mais simples, muitas das ideias e técnicas envolvidas na resolução desse tipo de problema podem ser generalizadas para domínios semanticamente sofisticados, incluindo a filosofia.

Para resolver problemas matemáticos, provar teoremas lógicos, realizar atividades como desvendar quebra-cabeças (*puzzles*) e dirigir um carro é preciso inteligência. Se um computador consegue realizar esse tipo de atividade, então ele deve possuir um certo grau

[7] O raciocínio prático envolve a escolha dos meios adequados para se chegar a determinado fim que se pretende atingir. Podemos dizer que essa forma de raciocínio é instrumental, no sentido que ela não diz onde ir, ela não estabelece uma meta, apenas o modo como chegar lá, empregando as ferramentas que temos a disposição. O problema do macaco e das bananas costuma ser apresentado como exemplo paradigmático de tal raciocínio. Esse problema consiste na tarefa de o macaco pegar um cacho de bananas que, a princípio, estavam fora de seu alcance.

de inteligência artificial. Em Inteligência Artificial (2013), Lurge apresenta um panorama da inteligência artificial, já em *Problem-solving methods in artificial intelligence* (1971), Nilsson não discute diversos aspectos importantes na área da inteligência artificial[8] como o registro e processamento de informações sensoriais, o acesso e a manutenção de banco de dados, a interpretação da linguagem natural, sistemas especialistas etc., tratando apenas daquelas técnicas de resolução de problemas que aparentam ter uma utilidade geral para resolver problemas bem definidos. Ao estudar os métodos de resolução de problemas empregados na área, ambos os autores constatam que a noção de *procura por tentativa e erro* é utilizada com frequência. Isto é, esses métodos resolvem problemas procurando uma solução em um *espaço do problema*.

Podemos descrever um espaço do problema de forma simplificada como um grafo contendo todos os estágios do problema em algum grau de solução, cujos elos representam os procedimentos que permitem passa de um estado a outro. Um estado pode ser rotulado com um arranjo mostrando uma configuração do jogo dos oito, ou um conjunto com as fórmulas demonstradas na prova de um teorema. De qualquer forma, assume-se que a busca de uma solução é feita conceitualmente, isto é, por meio de representações. Para encontrar uma solução para o jogo dos 8, por exemplo, não é necessário manipular um puzzle concreto, feito de plástico; assim como para encontrar o menor trajeto entre algumas cidades não é preciso viajar, esse processo pode ser feito por meio da manipulação de símbolos. Os problemas filosóficos, que já são de natureza conceitual envolvem a manipulação de representações, sejam elas públicas (na produção de um texto) ou privadas (no pensamento). Sob este viés, para encontrar a solução de um problema é preciso antes, representá-lo. Estes são os dois elementos fundamentais na resolução de problemas na visão tradicional das ciências cognitivas[9]: a representação e a busca. Quem deseja estudar a resolução de problemas pode dividir sua tarefas em duas: 1) investigar formas de representar os problemas, que facilitem a busca de uma solução e 2) investigar formas eficientes de procurar uma solução dada uma certa representação.

Dentre os diversos modos possíveis de representar problemas, vamos tratar de dois mais detalhadamente no capítulo 1 desta dissertação: a representação em um *espaço do problema* (no qual os estados do problema são representados por nós em um grafos) e a abordagem lógica (no qual os estados do problema são representados por conjuntos de

[8] A inteligência artificial é o ramo das ciências da computação que se ocupam com a automação da inteligência. Ela estuda os mecanismos subjacentes ao comportamento inteligente por meio da construção e avaliação de artefatos que representam ou disponham desses mecanismos. Nesse sentido, a inteligência artificial se desenvolveu aproximando-se mais das engenharias que de uma ciência e está menos preocupada com a criação de uma teoria geral da inteligência do que com a utilização de diversas teorias (teoria dos grafos, lógica de predicados etc.) para criar dispositivos capazes de comportamento inteligente.

[9] As ciências cognitivas formam uma área interdisciplinar, envolvendo disciplinas como a filosofia, a psicologia e a inteligência artificial. Ela tem como objeto de estudo os processos envolvidos na formação do conhecimento, como o raciocínio, a percepção, a imaginação, a atenção e a memória.

fórmulas). Nestas representações, a busca por uma solução se converte, respectivamente, no problema de encontrar um caminho conectando o nó inicial do grafo e um determinado nó que satisfaça as condições almejadas e provar certo teorema.

Em *Transaction logic programming*, Bonner e Kifer (1995) apresentam uma extensão conservativa da lógica de primeira ordem clássica (além dos conectivos "clássicos", a lógica das transações possui dois novos operadores: $\otimes$ e $\diamond$), chamada lógica das transações, cujo objetivo é lidar com um mundo em constante mudança, formalmente suas sentenças serão satisfeitas em sequências finitas e discretas de estados (chamados de caminhos). No trabalho citado, eles fornecem uma semântica para a lógica como um todo e um sistema de prova para um fragmento que serve como linguagem de programação, chamado de "fragmento de Horn". Os autores também afirmam que podemos interpretar as sentenças da linguagem de neste fragmento como sentenças imperativa e que as fórmulas podem ser utilizadas para denotar ações. Em decorrência dessas características, nesta lógica, os símbolos de predicados podem representar não apenas diversas propriedades e relações de forma apropriada, mas também ações que tem como efeito alterar o estado das coisas. Para determinar os efeitos que a execução de uma ação pode provocar em um estado, os autores apresentam uma relação de consequência lógica semântica, chamada de relação de consequência execucional.

Por que escolhemos a lógica das transações para subsidiar este trabalho? Em um primeiro momento, essa lógica foi escolhida porque seus desenvolvedores afirmaram: 1) que ela era um formalismo capaz de representar tanto o conhecimento declarativo, quanto o conhecimento procedural de maneira uniforme; 2) nela, os símbolos de predicado representam ações de forma apropriada; e 3) suas sentenças também poderiam ser interpretadas de forma imperativa.

Dado que ambas as formas de conhecimento são necessárias para um agente resolver problemas; que os problemas são solucionados pela execução de série de ações; que os problemas, muitas vezes, são colocados por meio de sentenças imperativas; e que alguns problemas podem ser solucionados seguindo instruções, as informações acima nos pareceram, *prima facie*, promissoras para aplicação de $\mathcal{TR}$ para entender a relação entre conhecimento e resolução de problemas. Além disso, a semântica de $\mathcal{TR}$ é baseada nas noções de *conjunto de dados abstratos* e *caminhos*, permitindo criar uma ponte entre as estratégias de representar o problema por meio de um grafo e representá-lo por meio da lógica de predicados, fornecendo uma solução para algumas limitações de ambas.

Como há outros formalismos que tratam de ações e problemas (como, por exemplo, a lógica dinâmica), podemos nos perguntar se alguma delas seria tão apropriada quanto $\mathcal{TR}$ para esta tarefa de representar o conhecimento e artefatos capazes de resolver problemas. Para Bonner e Kifer (1995), nenhuma delas é tão versátil quanto $\mathcal{TR}$ e elencamos três motivos para justificar essa declaração.

Primeiro, na lógica das transações podemos representar tanto a execução das ações (a sequência de estados que sua execução acarreta) quanto representar a busca por meio de conceitos que permitem apenas encontrar uma solução, sem realizar nenhuma mudança no estado das coisas. Deste modo, podemos levar para dentro deste formalismo a distinção entre a execução real de ações que permitem *solucionar* um problema e a representação das ações que permitem *encontrar* uma solução conceitualmente, sem realizar uma mudança no estado atual do problema. Segundo, nos outros sistemas formais costuma ser difícil atribuir nomes para ações compostas, isto é, não há mecanismos lógicos que facilitem a introdução de nomes para a ações mais complexas definidas a partir de ações mais simples. Entretanto, tal mecanismo é útil para representar o conhecimento procedural que os agentes trazem para enfrentar os problemas. Isso porque, o conhecimento procedural pode ser representado por meio de programas e, portanto, representar tal conhecimento sem este mecanismo seria tão difícil quanto programar sem sub-rotinas. Terceiro, muitas lógicas fazem uma distinção sintática entre os símbolos que representam relações e propriedades por um lado, e os símbolos que representam ações provocam mudanças por outro. Na lógica dinâmica, por exemplo, as ações são representadas por meio de operadores modais e na lógica de predicados ($\mathcal{FOL}$) as ações costumam ser representadas por meio de símbolos funcionais. Em $\mathcal{TR}$ não existe uma distinção sintática entre os símbolos reservados para representar ações e relações; ambas são representadas por símbolos de predicado. Essa uniformidade sintática nos permite utilizar os operadores lógicos para combinar proposições e ações de diversas formas, mixando condições (prévias, dinâmicas e posteriores) que devem ser satisfeitas em um estado com ações que atualizam o estado de coisas em uma única operação.

O tema deste trabalho é relevante não apenas no âmbito da lógica. Isto porque, problemas estão no cerne de todas as disciplinas. O motivo para existência das diversas disciplinas é justamente a pretensão dos seres humanos de solucionar problemas e responder questões que nos angustiam. Como as disciplinas se constituem de problemas e soluções, um entendimento sobre a natureza dos problemas é, portanto, de interesse geral.

Se por um lado, a lógica, o conhecimento e os problemas podem ser objeto de estudo da filosofia e alvo de diversas indagações como: O que é uma lógica? Quantas lógicas existem? O que é um problema? Em que sentido um problema posto intuitivamente na linguagem natural e um problema representado em um formalismo podem ser ditos “equivalentes”? O que significa conhecimento? Quais as condições de possibilidade do conhecimento?, por outro, a lógica também é utilizada na tentativa de resolver problemas filosóficos como uma ferramenta. Neste caso, ela permite formalizar conceitos e teorias, auxiliando na tarefa de fornecer definições precisas e aumentando o rigor da argumentação. Nesta dissertação vamos mostrar um modo como a lógica das transações pode ser aplicada ao problema do conhecimento. Nosso intuito, entretanto, não é o de fornecer uma definição formal para as diversas noções intuitivas de conhecimento - uma teoria formal do conhecimento -

utilizando a linguagem de $\mathcal{TR}$. Antes disso, nosso objetivo é discutir em que medida a lógica das transações pode ser utilizada para representar o conhecimento e dispositivos capazes de utilizá-lo para solucionar problemas. Essa lógica permite representar os tipos de conhecimentos tal qual nós o entendemos ou apenas algo similar? Nesta dissertação, defendemos a seguinte tese: a lógica das transações não é expressiva o suficiente para representar adequadamente dispositivos que *sabem* como resolver problemas, mas permite de representar dispositivos que são *capazes* de resolver problemas. O sentido desta distinção ficará claro no capítulo 2. Essa lacuna decorre de o fato de a linguagem ser expressiva o suficiente para dizer que uma ação A pode ser executada em determinada sequência de estados, mas não se A é executada intencionalmente ou se A é executada sem intenção.

Objetivos

Para esclarecer as relações entre lógica, conhecimento e resolução de problemas acreditamos que os trabalhos mencionados anteriormente forneçam um arcabouço teórico suficiente para encaminhar a discussão sobre a possibilidade de aplicar a lógica das transações na representação do conhecimento e de dispositivos capazes de utilizá-lo na resolução de problemas. A literatura apresentada fornece o subsídio necessário para 1) definir o que são problemas e fornecer uma noção do é conhecimento; 2) explicar o que é a lógica das transações dos pontos de vista da teoria da prova e da semântica; 3) mostrar como a lógica das transações é aplicada para representar o conhecimento procedural e declarativo, bem como dispositivos que utilizam o conhecimento para resolver problemas bem-definidos; e 4) fazer uma análise crítica dessa desta aplicação.

Com o quarto objetivo temos em mente discutir em que grau tal formalismo se mostrou adequado para cumprir a tarefa de representar o conhecimento. Obviamente, uma resposta para esta pergunta depende do que entendemos por “conhecimento”, do que entendemos por “problema” e de qual classe de problemas estamos tratando. Nesta dissertação, defendemos a seguinte tese: a representação fornecida pela lógica das transações não captura adequadamente alguns aspectos dos conceitos de conhecimento (que apresentamos informalmente no capítulo 1). São três os aspectos de nossa teoria informal do conhecimento e sobre problemas que os dispositivos representados no fragmento de $\mathcal{TR}$ analisado deixam a desejar:

1. O caráter dialético do conhecimento (não podemos representar modificações na base de conhecimento de um agente, podemos representar apenas a aplicação do conhecimento, mas não a aprendizagem).

2. O conhecimento de estados intencionais é limitado. Como a linguagem de $\mathcal{TR}$ não é expressiva o suficiente para construir sentença como “Jõao tem a intenção de fazer

A" e "Jõao sabe que P", não podemos, por exemplo, modelar problemas cujos dados e informações envolvem a vida mental de outras pessoas.

3. O caráter cooperativo e a distribuição do conhecimento entre diversos agentes.(Muitos problemas não são enfrentados por um único agente, mas em um sistema no qual diversos agentes situados em um ambiente distribuem as tarefas e trabalham simultaneamente para solucionar problemas. Entretanto, não podemos representara a execução de ações em paralelo por meio dos operadores lógicos de $\mathcal{TR}$).

Para cumprir os objetivos desta dissertação, apresentamos no capítulo 1 (O que são problemas e como resolvê-los?) uma discussão sobre o significado do termos "problema", "solução de um problema"; apresentamos os principais elementos de um problema: a representação e a busca; discutimos o significado de "conhecimento"; e tratamos da questão: "O que significa executar uma ação intencionalmente?".

No capítulo 2 (A lógica das transações) fornecemos uma introdução a lógica das transações, apresentando sua semântica e um sistema de prova para o "fragmento de Horn" da lógica, tendo por base o trabalho de Bonner e Kifer[10]. Também tratamos das questão da interpretação imperativa das fórmulas de transação e sua relação com o conceito de verdade.

No capítulo 3 (Aplicações da lógica das transações ao problema do conhecimento) fornecemos alguns exemplos simples para ilustrar a semântica de $\mathcal{TR}$ e sua aplicação no contexto da resolução de problemas. O primeiro exemplo mais sofisticado mostra como representar o conhecimento procedural por meio de um conjunto de fórmulas, chamado de base de transações. O segundo exemplo mostra como a distinção conceitual entre os resultados e as consequências de uma ação é capturada nesta lógica. O terceiro, mostra como representar a resolução de um problema realizada por tentativa e erro. O quarto exemplo ilustra como uma resolução que utiliza uma heurística geral, chamada de análise dos meios e fins, pode ser representada. O quinto exemplo mostra uma solução para o *frame problem* usando a lógica das transações.

Por fim, concluímos que a lógica das transações fornecer bons *insights* para representar os diversos tipos de conhecimento e sistemas capazes de utilizá-los na resolução de problemas, acreditamos que algumas modificações podem ser utilizadas, para torná-la mais adequadas as ideias apresentadas no capítulo 2. Esperamos que este trabalho forneça uma introdução para a discussão sobre a forma como sistemas que processam informações solucionam problemas.

[10] (BONNER; KIFER, 1995).

1 O que são problemas e como resolvê-los?

Duas noções informais de problema

No uso ordinário da linguagem, o termo "problema" é aplicado às mais variadas coisas. Nós o utilizamos para tratar desde tarefas do dia-a-dia (como conseguir comida ou encontrar o caminho para uma festa) e *puzzles* (como o jogo dos oito e a torre de Hanói), até questões filosóficas (como "o que significa conhecimento?"); de exercícios matemáticos triviais (como encontrar as raízes inteiras de uma equação do segundo grau), até questões matemáticas que nos deixam perplexos (como determinar se a hipótese do contínuo está "correta"). Além desses usos, também podemos dizer "eu tive problema" quando, por exemplo, o pneu do nosso carro fura, ou "eu estou com um problema de saúde" quando estamos doentes. Nestes últimos casos, os problemas não são tarefas e nem questões, mas um defeito e uma doença.

Nesta dissertação, apesar de termos a pretensão de dar ao termo "problema" um escopo bastante amplo, não pretendemos abarcar todos aqueles usos. Não vamos tratar aqui, por exemplo, dos problemas enquanto defeito, doença. Para nós, se algo é um problema, então ele deve ser uma tarefa. Como todos os problemas que estão colocados na forma de uma questão podem ser apresentados como tarefas (a saber, a tarefa de responder a questão corretamente), também podemos abarcar problemas filosóficos sob essa definição de problema. Algumas vezes, vamos preferir manter a apresentação de um problema sob a forma de uma questão, ao invés de apresentá-la como uma tarefa. Independente da forma como é colocado, a noção de problema sempre nos remete a noção de solução. A primeira vista, parece que alguns problemas possuem mais de uma solução e alguns problemas não possuem nenhuma solução. Entretanto, para justificar tal afirmação precisamos definir o que são soluções.

Se todo problema é uma tarefa, então solucionar determinado problema é realizar determinada tarefa. O termo "solução" é ambíguo, pois pode significar tanto o resultado final obtido pela realização da tarefa, quanto o modo para se chegar ao resultado final. No caso do problema ser apresentado como uma pergunta, o termo "solução" é usado tanto para indicar a resposta final, quanto o processo de justificação que leva a resposta. Devemos ficar atentos, portanto, pois a expressão "ter mais de uma solução" pode significar que há mais de um resultado final para um problema ou que há mais de um modo de se chegar ao(s) seu(s) resultado(s) finais. Por exemplo, há diferentes algoritmos que podem ser utilizados para calcular o valor de certa função para determinada entrada, entretanto o resultado final obtido como valor da função será único. Assim como, a tarefa de *chegar em um local específico c* tem como resultado final certa relação, a saber, que o objeto que realize a tarefa esteja no local c, mas há diversos caminhos diferentes que podem ser utilizados para se chegar neste resultado. Por outro lado, a tarefa de encontrar as raízes inteiras de uma equação de segundo grau aplicando determinada sequência de instruções, pode até levar a dois resultados finais, mas só há uma forma de realizar tal tarefa, aquela especificada pelas instruções. Claro, nós podemos encontrar as respostas (as raízes) utilizando um método diferente, mas, neste casos, não estaríamos realizando a tarefa original (que possui como condição imposta que a resposta seja encontrada aplicando um método bem especifico e que só pode ser realizada de um único modo). Note que a tarefa de encontrar as raízes de dada equação e a tarefa de encontrar as raízes, usando um determinado algorítimo não possuem as mesmas condições de satisfação (o número de caminhos que podem ser seguidos no segundo caso é mais restrito), entretanto, essas tarefas possuem as mesmas respostas.

Como alguns problemas não possuem solução, isto pode significar que não há uma resposta para algumas perguntas ou que não dispomos de um meio efetivo para obter a resposta; ou que os efeitos de algumas tarefas são sempre falsos ou que não dispomos um meio para alcançar seus efeitos. Essa indisponibilidade pode ser de natureza lógica, caso a existência de uma solução implique na trivialização da teoria que descreve o domínio do problema, mas geralmente, vamos pensar em uma impossibilidade prática, como por exemplo, por não existir uma solução computacionalmente viável. Para alguns problemas existe claramente uma resposta correta, mas não é possível obtê-la do ponto de vista prático, como por exemplo, a questão "existe uma estratégia vencedora para as peças brancas no jogo de xadrez?", para responder esta pergunta bastaria analisar o conjunto de todas os jogos possíveis, mas, como esse conjunto é muito grande, não conseguimos do ponto de vista prático descobrir qual é a resposta. Isto é, existe uma resposta correta para a questão (a resposta correta é sim ou a resposta correta é não), mas não conhecemos um meio viável de descobrir qual delas é a resposta correta. Para outros problemas, não é tão claro se há uma resposta correta. Se pensarmos na hipótese do contínuo (HC), por exemplo, constatamos que não temos um modelo pretendido para os conjuntos, apenas algumas ideias sobre sua organização e produção e constatamos que a formalização padrão da teoria de conjuntos é consistente tanto com sua afirmação, quanto com sua negação. Neste caso, entender se a hipótese do contínuo está correta esbarra em problemas metafísicos sobre a natureza das estruturas matemáticas. Só a título de ilustração vamos contrapor duas posições sobre este tópico: uma realista e uma construtivista. Se existisse um modelo único e correto do universo dos conjuntos, então haveria uma resposta correta, aquela determinada por esta estrutura. Desta perspectiva, poderíamos argumentar que faltam informações nos axiomas tradicionais, ou que as regras de inferência da lógica de primeira ordem clássica não são todos os procedimentos desejáveis na busca da resposta correta e que é desejável alterar os axiomas axiomas e\ou a lógica subjacente. Em ambos os casos, precisaríamos justificar as alterações. Desta perspectiva, o problema original não estaria tão bem definido, pois faltariam informações (axiomas insuficientes) ou procedimentos (regras de inferência), e seria adequado buscar uma re-caracterização que permitisse solucionar a questão. Por outro lado, se adotarmos uma perspectiva metafisica na qual a teoria de conjunto e as estruturas matemáticas são produtos de nossas faculdades e que, portanto, estariam sujeitas às limitações e imprecisões impostas por nossas limitações, então apesar de a hipótese do contínuo ser verdadeira ou falsa (em cada modelo), nós não somo capazes de fixar o modelo pretendido e, portanto, não haveria necessidade fixar uma resposta correta a partir deste modelo. Desta perspectiva, não precisaríamos de uma mudança na formalização padrão da teoria de conjuntos em primeira ordem para encontrar a resposta correta porque ela simplesmente não existiria. Neste caso, podemos considerar a tarefa de provar HC ou provar não-HC como uma tarefa bem definida, mas que não pode ser realizada, ou seja, estaríamos lidando com um problema para o qual não há solução. Independente da visão metafísica adotada, essas questões tocam nos problemas envolvidos na passagem das ideias e concepções intuitivas (como a de hipótese correta) para uma teorias e concepções formais (como a de prova ou verdade em uma estrutura), na qual os problemas são re-caracterizados para se tornar melhor definidos.

Nesta dissertação, estamos interessados em lidar com problemas bem-definidos. De acordo com Herbert Simon, um problema é bem-definido quando apresenta as seguintes características: há uma critério definido para testar as soluções propostas e este critério pode ser mecanizado; o problema pode ser representado de modo que estejam presentes na representação os estados inicial e almejado do problema, além de todos os outros estados que podem ser considerados durante a tentativa de encontrar uma solução para o problema, bem como procedimentos que alteram os estados do problema; todo o conhecimento que o solucionador possui pode ser representado; se o problema envolve a atuação no ambiente externo, então as definições dos procedimentos e os efeitos, que eles provocam ao serem aplicados, refletem com precisão as mudanças no mundo externo; todas os processos postulados devem requerer apenas quantidades praticáveis de computação, assim como a informação postulada deve estar acessível de forma eficiente, através de uma quantidade praticável de computação. Esses critérios não formam uma definição precisa, já que expressões como “uma quantidade praticável de computação” são definidas apenas em relação ao poder computacional do agente e o tempo disponível para realizar a tarefa. A vagueza nessa definição reflete a ideia de que existe um espectro entre problemas bem-definidos e mal-definidos. (SIMON, 1977).

Como problemas filosóficos podem parecer, pelo menos a princípio, qualquer coisa menos problemas bem definidos - já que são postos primeiramente na linguagem natural, sem nenhum compromisso explícito com uma teoria - podemos nos perguntar se as técnicas utilizadas para resolver problemas bem-definido podem ser aplicadas na filosofia. Tais técnicas podem ser aplicadas nos problemas filosóficos. Entretanto, antes de aplicar essas técnicas, os problemas precisam passar por um tratamento, precisamos re-caracterizá-los.

Isso significa que temos que tentar torná-los mais precisos, definir seus conceitos com a maior clareza possível e estabelecer nossos pressupostos teóricos, bem como qual sistema lógico utilizamos na argumentação. A formalização de teorias e a definição de conceitos por meio de operadores lógicos ou com o suporte de ferramentas matemáticas, tem como objetivo re-caracterizar o problema original definido-o melhor. Essa formalização tem como benefício tornar a proposta de um filósofo mais suscetível a análise crítica, o que permite encontrar falhas e permitir revisões.

Como todas teorias e definições formais partem de uma base intuitiva e informal, podemos nos perguntar em que medida um formalismo "captura" a proposta intuitiva. Não pode haver uma prova rigorosa da equivalência entre uma proposta informal e uma teoria formal. Isso significa que o problema filosófico de saber se uma formalização captura a proposta intuitiva original não pode ser resolvido por meio de uma prova formal. Entretanto, se diferentes formalismos que tentam capturar o mesmo fenômeno intuitivo se revelam equivalentes (e aqui pode haver uma prova rigorosa dessa equivalência), então teremos uma evidência de que ambos os formalismos capturam a classe de fenômenos pretendida. Por exemplo, podemos definir intuitivamente e informalmente as funções computáveis, como as funções que podem ter seus resultados calculados em uma quantidade finita de tempo, seguindo um conjunto preciso de instruções. Historicamente, essa noção intuitiva foi formalizada por meio da definição rigorosa de máquinas de Turing e por meio da definição rigorosa de funções recursivas. A afirmação de que uma função é computável, em sentido informal e intuitivo, se e somente se existe uma máquina de Turing que a computa não pode ser provada rigorosamente. Entretanto, posteriormente se provou que existe uma máquina de Turing que computa determinada função se e somente se essa função é recursiva, e aqui estamos falando de uma prova rigorosa. Como ambos formalismo foram criados visando capturar a mesma classe de fenômenos, então a prova da equivalência dessas definições formais pode ser interpretada como uma evidência em favor da tese que afirma de que ambas definições formais capturam a ideia intuitiva.

Também podemos nos perguntar se qualquer tarefa deve ser chamada de problema. É suficiente que algo seja uma tarefa para ser um problema? O predicado "ser problema" se aplica a todas as tarefas, mesmo àquelas que são triviais, ou apenas àquelas que são difíceis? Dependendo da resposta que dermos a esta pergunta, ficaremos com uma das seguintes noções do que são problemas:

Noção 1: Problemas são quaisquer tarefas (procedimentos).

Noção 2: Problemas são tarefas (procedimentos) difíceis de serem executadas em um dado contexto.

Para alguns autores que coadunam com a primeira definição, todo comportamento é uma instância de resolução e toda atividade cognitiva é fundamentalmente uma resolução de problemas. Isso porque estes autores entendem que todas as tarefas, mesmo as que não apresentam dificuldades e já estão automatizadas seriam problemas. Autores como Anderson, que diz:

> "O que aconteceria se [o solucionador] tivesse resolvido o mesmo problema repetidamente? Em algum momento, toda a solução ficaria englobada em uma única operação, e [o solucionador] iria simplesmente passar pela sequência de etapas necessárias para alcançar a meta. Não mais pareceria intuitivamente uma resolução de problemas, e sim, que [o solucionador] estava executando simplesmente um procedimento aprendido. [. . .] Habitualmente, costumamos empregar a expressão *resolução de problemas* para os episódios originalmente difíceis, como o primeiro esforço [do solucionador] para resolver o problema. Contudo, os episódios subsequentes, mais automáticos, não deixam de ser resolução de problemas." (ANDERSON, 2004, p. 140-141)

Nós preferimos nos afastar da postura desses autores por considerá-la anti-intuitiva (como o próprio Anderson nota) e entender que a dificuldade é um componente que faz parte da noção de problema. Isso não significa que tarefas consideradas triviais não possam ser realizadas durante a resolução de um problema, antes disso, o objetivo de um problema bem-definido é alcançado pela decomposição de tarefas complicadas em tarefas mais simples. Para nós, um problema é uma tarefa difícil, em decorrência das limitações do meios que podem ser utilizados para realizá-la. Optar pela segunda noção acarreta algumas consequências para uma definição formal de problema. Uma delas, é que, como não há critério absoluto de dificuldade, a definição de problema se torna relativa ao solucionador (ou grupo de solucionadores) envolvido com a tarefa. O que é um problema para um agente, pode não ser problema para outro. Dependendo das características do solucionador, algo pode ser considerado uma tarefa trivial (se ele conhece algum esquema para resolvê-lo de antemão), um problema cuja solução está a seu alcance, mas requer que o solucionador pense para descobrir o que deve ser feito; ou mesmo um problema que está mal definido, precisando ser re-caracterizado antes de ser resolvido.

Obviamente, não são todas as propriedade do solucionador que são relevante para determinar se uma tarefa é um problema para ele. Se o solucionador for uma pessoa, diversas propriedades (como a terceira letra de seu nome, o dia de seu nascimento, etc.) são irrelevantes para tal propósito, na maioria dos dos casos. Seguindo o arcabouço proposto por Schoenfeld discutimos nas próximas seções algumas das propriedades que consideramos relevantes.

Existe uma teoria geral de problemas que se aplique a todos os casos que gostaríamos de abarcar sob o rótulo problema e que, ao mesmo tempo, esteja de acordo com nossas intuições? Antes de responder esta pergunta na próxima seção, vamos tentar apontar algumas ideias que norteiam a noção de problema da qual queremos tratar:

- Todo problema é uma tarefa.

- Nem toda tarefa é vista como um problema, apenas aquelas que são difíceis de realizar. Ou seja, uma solução não deve estar disponível previamente para o solucionado (uma tarefa que já está realizada não é mais um problema) e ele não deve conhecer nenhum procedimento que leve diretamente a solução. Se ele estiver familiarizado com tal procedimento, a tarefa não seria um problema, apenas um exercício trivial.

- Todo problema envolve um conjunto de procedimentos que podem ser combinados durante a tentativa de solucioná-lo.

- Todo problema possui a expectativa uma solução, mas as vezes, essa não existe. Por "solução" podemos entender tanto o resultado final da tarefa quanto o modo pelo qual ela pode ser realizada. Deste modo, a noção de problema seria dependente das noções de tarefa e solução.

- Todo problema pressupõe um certo estado inicial, um conjunto de dados iniciais que assumimos. Toda tarefa é realizada a partir de certo estado de coisas.

A segundo ponto fornece uma característica negativa, indicando que a noção de problema é relativa a um solucionador e deve ser definida levando em consideração algum parâmetro. Esse parâmetro na definição de problema, indica a existência de restrições na forma como a tarefa é realizada. Entendemos que solucionar um problema envolve a execução de uma sequência de procedimentos que realizam a tarefa difícil e que encontrar uma solução envolve a manipulação de representações que permitem mostrar o modo como alcançar o resultado. Essa distinção reflete a ideia de que problemas podem ser resolvidos de fato ou que podemos encontrar uma solução por meio de representações. Podemos representar um problema e encontrar uma solução, que pode ser implementada posteriormente realizando de fato a tarefa. Tipicamente, estamos interessados em problemas cuja implementação é irrelevante, ou seja na qual a passagem da representação para sua execução real não apresenta dificuldades ou não têm importância. Por exemplo, em muitos problemas matemáticos envolvendo a construção de figuras geométricas com compasso e régua, o ponto da questão é deduzir o local onde a figura deve ser construída, sendo irrelevante do ponto de vista matemático, a implementação manual do desenho usando régua e compasso. Para realizar uma tarefa, muitas vezes intercalamos buscas por meio de representações e implementações dos resultados até então obtidos. Por exemplo, para provar um teorema por dedução natural (escrevendo uma sequência de fórmulas obtidas pela aplicação de regras de inferência), muitas vezes, pensamos nas fórmulas que parecem úteis e formamos uma ideia de como chegar a meta (manipulando representações mentais) e depois escrevemos as fórmulas (implementando a ideia) e voltamos a alternar entre pensar no caminho que faremos e sua implementação.

Três definições formais de problema

Um trabalho que teve como objetivo desenvolver uma teoria geral dos problemas, tornando a definição de diversos termos ligados a resolução de problemas mais rigorosa foi apresentado por Veloso[1]. Em seu trabalho, o autor usa as perguntas “Quais os dados do problema?”, “Quais são os possíveis resultados?” e “O que constitui uma solução[2] satisfatória?” como guia para apresentar diversas definições de problema, entre elas estão as de: problemas irrestritos, problemas inseridos em contextos e problemas intencionais. Vamos apresentar essas definições e depois discutir se alguma delas captura a nossa noção informal de problema.

Com a resposta da pergunta “quais os dados do problema?”, o autor tinha em mente estabelecer com qual instância particular ou instâncias de um problema mais geral estamos lidando. Por exemplo, o problema de encontrar uma raiz qualquer do polinômio $(3x^4 + 7x^3 + 1)$ é uma instância particular do problema mais geral de encontrar uma raiz de um polinômio com coeficientes reais e de grau arbitrário. No caso do problema particular, os dados do problema indicariam que estamos tratando apenas do polinômio $(3x^4 + 7x^3 + 1)$ e no caso do problema geral, os dados do problemas indicariam que estamos tratando de todos os polinômios com coeficiente real e de grau arbitrário. A resposta a esta pergunta estabelece com quais casos estamos tratando, para quais casos precisamos encontrar um resultado. Estamos interessados em polinômios cúbicos ou de grau arbitrário? Com coeficientes inteiros ou reais? Se ao invés de um problema matemático, estivermos tratando de uma questão como “o que é o conhecimento?”, precisamos antes definir o que significa “conhecimento”. Nós estamos tratando de todos os tipos de conhecimento ou apenas do conhecimento proposicional? De qual conhecimento estamos falando? Se problema for resolver um quebra cabeça como o jogo dos oito, estamos interessados em uma configuração particular especifica ou em configurações iniciais arbitrárias?

Respondendo a pergunta “Quais são os possíveis resultados?” especificamos que tipo de coisa contaria como um possível resultado da tarefa. No caso dos polinômios, queremos raízes reais, inteiras ou complexas? No caso da questão sobre o conhecimento, o que contaria como uma resposta? Queremos como resposta uma proposição derivada de uma teoria dualista, funcionalista, fisicista reducionista, fisicista não-reducionista ou tanto faz? Assim como um polinômio pode não ter uma raiz inteira, mas ter uma raiz real, pode ser o caso, que para o problema do conhecimento não exista, ou pelo menos não esteja disponível, por exemplo, uma teoria metafísica não-reducionista consistente que permita explicar o que é o conhecimento. Claro, que esta analogia não é perfeita, pois a noção de proposição de uma teoria dualista não é tão bem definida como a noção de um número inteiro.

1 (VELOSO, 1984).
2 No sentido de um resultado final satisfatório.

O que constitui uma solução (no sentido de resultado) satisfatória? Às vezes, não estamos interessados em todos os resultados, apenas naquelas que cumprem determinadas condições. A resposta dessa pergunta especifica quais resultados resolvem quais instâncias do problema sob certas condições. Uma condição separa no conjunto dos resultados aqueles que satisfazem certa condição. Um exemplo fornecido por Veloso para esclarecer este aspecto é o seguinte: um polinômio pode ter diversas raízes, estamos interessados na menor, na maior ou em qualquer uma? No problema de achar uma raiz real arbitrária de um polinômio com coeficientes reais e de grau arbitrário, a relação q, ser uma solução satisfatória, é a relação de D para R, definida por: r $\in$ R é um resultado satisfatório para p(x) $\in$ D se e só se o valor de p(x) no ponto r é zero, mais formalmente, (p(x), r) $\in$ q se e somente se $p(r) = 0$. Se, em vez disso, a condição problema fosse o de "achar a menor raiz real de um polinômio com coeficientes reais e grau arbitrário", podemos especificá-lo tomando o mesmo domínio D de dados acima, o mesmo domínio R de resultados e a relação de ser um resultado satisfatório seria agora definida por (p(x), r) $\in$ q' se e somente se $p(r) = 0$ e $r < r'$ para todo r' $\in$ R com $p(r') = 0$. No caso de um problema filosófico, nós esperamos que uma teoria apresente uma única resposta à pergunta "o que é o conhecimento?" e, portanto, não faria sentido separar algumas delas. A partir dessa discussão, Veloso define um problema irrestrito da seguinte forma:

Definição: Um problema irrestrito, P, é uma estrutura $\langle D, R, q \rangle$, onde:

- D é um conjunto não vazio de dados do problema, que especifica as instâncias do problema com as quais estamos lidando. Este conjunto é chamado de domínio dos dados.
- R é um conjunto não vazio de possíveis respostas ao problema, chamado de domínio de respostas.
- q é uma relação binária de D para R, ela é uma extensão para o conceito de *ser um dos resultados que satisfaz a condição do problema*. Essa relação é chamada de condição do problema.

Para Veloso, a solução para um problema P é uma função total σ de D para R, que para cada instância i $\in$ D associa uma resposta $\sigma(i) \in$ R, de modo a satisfazer a condição do problema, isto é, $\sigma : D \to R$ é tal que para todo $i \in D$ tem-se $(i, \sigma(i)) \in q$. Nesse sentido, a solução de um problema geral atribui para cada instância particular do problema um resultado satisfatório. Se um problema tiver mais de um resultado podemos atribuir um conjunto formado por todos eles.

Esta definição é intencionalmente ampla para abarcar diversos tipos de problemas, não só matemáticos. Contudo, essa formulação não é compatível com nossas intuições sobre os problemas. Essa incompatibilidade está ligada ao conceito de solução e decorre por algumas razões. Primeiro, a definição de problema irrestrito se aproxima mais das nossas ideias sobre tarefas, pois não leva em consideração a dificuldade da solução. Na nossa terminologia, o que Veloso chama de problema irrestrito não deveria ser chamado de problema, mas de uma tarefa, pois nossa ideia sobre um problema é que ele é uma tarefa difícil de realizar em certo contexto. Segundo, nós não queremos que a solução de um problema seja o mero emparelhamento entre os dados iniciais do problema e seus resultados, mas algo que descreva um dos modos como essa atribuição se dá. Neste caso, uma solução é melhor entendida como um objeto linguístico que descreve exatamente a série de mudanças que levam dos dados do problema até seus resultados. Terceiro, não estamos interessados em qualquer descrição de função que possa levar dos dados iniciais do problema até seus resultados, mas em um conjunto particular que represente as ações que determinado solucionador é capaz de realizar para solucionar o problema bem definidos, e portanto poderíamos nos limitar a priori às funções computáveis.

Para proibir certas funções, que a priori não são aceitáveis como candidatas a solução, Veloso propõe especificar um contexto para o problema, dado por uma classe A de funções, ditas *funções admissíveis*. Uma função precisa estar no conjunto das admissíveis para poder ser considerada como uma solução. Um problema P no contexto A consiste dos domínios D e R e de uma relação q de D para R, como antes. Entretanto, uma solução para o A-problema P é uma função $\alpha : D \to R$ tal que:

- α é admissível, ou seja, $\alpha \in$ A;
- α satisfaz a condição do problema, ou seja, para todo d $\in$ D (d, α(d)) $\in$ q.

Veloso ressalta dois aspectos dos problemas inseridos em contextos. Primeiro, um A-problema nada mais é que um problema irrestrito; é a definição de solução que é alterada, sendo relativizada à classe A de funções admissíveis. Além disso, os problemas irrestritos são casos especiais de A-problemas em que A consiste em todas as funções, isto é, nenhuma função é descartada a priori. Segundo, para um A-problema determinado, não é necessário se considerar toda a classe A de funções admissíveis, apenas o conjunto A (D, R) das funções de D em R que são admissíveis. Note que a condição do problema descarta alguns resultados e que o contexto faz com que diversas classes de função sejam desconsideradas como soluções.

A definição de problemas inseridos em determinado contexto fornece uma formulação para as noções de problema que não se importam com o modo como o resultado é obtido. Essa formulação trata as soluções como objetos extensionais, como funções, o que não é compatível com a ideia de que uma solução é o *modo* pelo qual os resultados almejados são obtidos a partir de certos dados iniciais e que há diversos modos de se chegar aos mesmos resultados a partir dos mesmo dados (só existe uma função que atribui aos mesmo dados, os mesmo resultados). Para capturar noções de problema para os quais quais a solução não é pensada como um objeto extensional, Veloso define outro tipo de problema, nos quais as soluções são vistas como descrições de funções.

Definição: Um problema intencional, P, é uma estrutura $\langle D, R, q, F, v \rangle$, onde:

- D, R e a relação q de D para R, são como antes;
- F é um conjunto não vazio, chamado domínio de programas (ou de descrições de funções);
- a função v : FxD $\rightarrow$ R é chamada função de avaliação.

A função de avaliação aplica uma determinada descrição de função p $\in$ F a um dado argumento d $\in$ D, fornecendo um resultado v (p, d) $\in$ R. Assim, cada descrição de função p $\in$ F induz uma função v_p : D $\rightarrow$ R, definida pela atribuição, para cada d $\in$ D, v_p(d) = v(p, d). É possível que programas distintos p $\neq$ p' induzam à mesma função $v_p = v_{p'}$, isto é, p e p' são duas descrições diferentes da mesma função $v_p = v_{p'}$. Assim, podemos distinguir p e p' por meio de atributos não extensionais (como, por exemplo, clareza, elegância, eficiência, etc.), que não se aplicam à função induzida.

Seguindo essa linha de raciocínio, Veloso define uma solução para o problema intencional $P = \langle D, R, q, F, v \rangle$ como sendo uma descrição de função p $\in$ F cuja função induzida v_p é uma solução do problema extensional subjacente $P = \langle D, R, q \rangle$, ou seja, para todo d $\in$ D, (d, v(p, d)) $\in$ q. Problemas inseridos em um contexto eliminam a priori determinadas funções julgadas inaceitáveis como soluções, mesmo que elas satisfaçam a condição do problema. Problemas intencionais são aqueles cujas soluções não são meramente funções, mas sim descrições de funções formuladas em uma linguagem apropriada para tanto. Acreditamos que a definição de um problema intencional inserido em um contexto é compatível com noção intuitiva de problema que apresentamos anteriormente. Geralmente, a dificuldade de um problema é capturada por algoritmos que fazem uso da noção de busca em um espaço do problema, ao invés de atacar o prolema diretamente. A seguir vamos tratar das características de uma descrição de programa que ajudam a capturar a ideia de que encontrar um solução é difícil.

Como resolver problemas: representação e busca

Luger (2013) e Nilsson (1971) apontam que a procura pela solução de um problema possui dois componentes principais: a representação e a busca. A função de um sistema de representação é capturar as características relevantes de um certo domínio e disponibilizá-las para um mecanismo de busca. O sistema de representação escolhido deve ser expressivo o suficiente para realizar essa abstração, mas deve ao mesmo tempo garantir que a procura por uma solução entre as representações possa ser realizada de forma eficiente. Os diversos estágios ou estados pelos quais um problema pode passar até ser resolvido podem ser representados por tabelas, matrizes, imagens, arranjos, conjuntos de fórmulas lógicas etc. Escolher uma representação apropriada é fundamental, pois uma escolha ruim pode tornar inviável encontrar uma solução para o problema. Quando se trata de investigar domínios semanticamente complexos, as linguagens lógicas são ferramenta muito úteis porque são bastante expressiva e flexíveis permitindo representar diversos aspectos qualitativos de diversos domínios e as transições entre os estados.

Tendo em mãos uma certa forma de representar os estados do problema, o segundo componente da resolução de problemas é buscar por uma solução. Quando as pessoas estão tentando encontrar a solução de um problema, elas geralmente pensam em diversas alternativas antes de implementar uma que pareça adequada, é essa a ideia que está por trás das técnicas que visam encontrar a solução de um problema por meio de buscas em um *espaço de estados do problema*.

A busca em um espaço de estados é um formalismo que funciona independentemente da estratégia de busca e do sistema de representação particulares escolhidos, servindo como ponto de partida de diversas abordagens de resolução de problemas. Representar o estado de um problema e suas transições por meio de um grafo rotulado[3] permite utilizar a teoria dos grafos para analisar o problema e responder diversas questões, como: o solucionador encontrará uma solução? O solucionador pode ficar presos em loops durante a procura? Se ele encontrar uma solução ela será a ideal?

Nesta forma de modelar uma resolução de problemas, os nós do grafo são utilizados para representar as etapas ou estados da resolução do problemas, tais como o resultado de inferências lógicas ou as diferentes configurações do jogo dos oito. Em um grafo rotulado, os rótulos identificam os estados do problema por meio do sistema de representação escolhido. Os arcos representam os passos que levam de estado um estado a outro, tais como inferências lógicas ou movimentos válidos em um jogo. Um ou mais estados iniciais, que correspondem às informações fornecidas em uma instância do problema, formam a raiz do grafo. A busca de uma solução em um espaço de estados é o processo de procurar um caminho que conecte o(s) estado(s) inicial(is) e um objetivo. Um objetivo pode ser descrito como um estado específico ou apenas por meio de uma propriedade que deve ser satisfeita no estado. Os caminhos no grafo representam as possíveis soluções.

O espaço do problema pode ser explorado em duas direções: a partir dos dados fornecidos em direção ao objetivo ou, então, a partir do objetivos em direção aos dados. Ou seja, a exploração dos caminho pode começar no estado inicial e se estender pelo grafo até que um estado almejado seja encontrado (*bottom-up*) ou começar pelo objetivo e se

[3] Um grafo consiste de uma estrutura formada por um conjunto de nós e um conjunto de arcos que conectam pares de nós. Um grafo rotulado possui um ou mais descritores (rótulos) atribuídos aos seus nós e arcos. Cada arco pode ter uma direção associada. Um caminho ao longo do grafo conecta uma sequência de nós por meio de arcos sucessivos. Uma definição formal de um grafo pode ser encontrada em Luger (2013 p. 70).

estender até que os estado iniciais do problemas sejam encontrado (*top-down*) .

> Na *busca guiada por dados*, algumas vezes chamada de *encadeamento progressivo*, o algoritmo para resolver o problema começa com os fatos fornecidos e um conjunto de movimentos válidos ou regras para a mudança de estado. A busca prossegue pela aplicação de regras aos fatos para produzir novos fatos, que por sua vez, são usados pelas regras para gerar novos fatos. Esse procedimento continua até que (assim esperamos!) ele gere um caminho que satisfaça a condição-objetivo. É possível, também, uma abordagem alternativa: comece com o objetivo que desejamos resolver. Veja quais regras ou movimentos válidos poderiam ser utilizados para gerar esse objetivo e determine que condições devem ser verdadeiras para que eles sejam utilizados. Essas condições se tornam novos objetivos, ou *subobjetivos*, para a busca. A busca continua, atuando de forma regressiva de subobjetivos sucessivos até que (assim esperamos) chegue aos fatos do problema. (LUGER, 2013, p. 73)

A exploração de novos caminhos é realizada pela aplicação de operadores, como os movimentos válidos de um jogo ou regras de inferência em um problema lógico, a partir de estados previamente explorados. Para realizar uma busca exaustiva, a busca no grafo pode ser ordenada em profundidade ou em amplitude. Na busca em profundidade, quando um nó é examinado, todos os seu descendentes são examinados antes de seus irmão e, quando a busca é feita em amplitude, se investiga o grafo nível por nível, quando todos os irmão em um nível tiverem sido explorados, se passa a investigar o próximo nível. O problema é que nem sempre é viável realizar uma busca exaustiva para encontrar uma solução, isso porque, o grafo pode ter um tamanho descomunal, neste caso heurísticas são necessárias para tornar a busca mais inteligente.

Lurge apresenta a seguinte definição para noção de espaço do problema.

Definição: um espaço de estados do problema é uma quadra $\langle N, A, I, O\rangle$, onde:

- N é o conjunto de nós de um grafo. Eles representam os estados de uma resolução de um problema.
- A é o conjunto de arcos entre os nós. Eles correspondem aos passos de uma resolução de problemas.
- I, $\subseteq N \neq \emptyset$, contém o(s) estado(s) iniciais do problema. Eles representam os dados do problema.
- O, $\subseteq N \neq \emptyset$ contém o(s) estado(s) almejado(s).

Uma solução é um caminho no grafo que começa em um nó em I e vai até um nó em O.

As características do solucionador

Nesta seção apresentamos um arcabouço teórico, baseado no trabalho de Schoenfeld (1985), com o objetivo de explicitar alguns aspectos da atividade intelectual das pessoas quando elas estão envolvidas com problemas para os quais elas não dispõem, a princípio, de um esquema prévio para resolvê-lo diretamente, o que faria da tarefa um mero exercício trivial e não um problema, mas para os quais elas possuem recursos suficiente para avançar no problema e encontrar uma solução. Se o que faz de uma tarefa um problema são as limitações do solucionador que impõem restrições ao modo como ele pode solucionar a tarefa, então precisamos responder às seguintes questões: Por que o solucionador falha mesmo tendo um *background* suficiente para encontrar uma solução? De quais recursos ele dispõe? Ou melhor, o que ele sabe sobre o domínio do problema e quais procedimentos ele é capaz de aplicar? A Quais estratégias ele é capaz de recorrer em momentos de dificuldade, quando ele não sabe o que fazer?

Para responder a estas questões Schoenfeld propõe quatro categorias, chamadas: recursos, heurística, controle e sistema de crenças e atitudes, que explicam as características do solucionador que são relevantes na resolução de problemas. Nesta dissertação trataremos apenas das duas primeiras categorias e deixaremos de lado questões do tipo: Como ele gerencia os recursos disponíveis como o tempo? Como as crenças e atitudes dele podem evitar que algumas ferramentas sejam empregadas?

Recursos

Para explicar as diferenças nas performances das pessoas ao tentar resolver a mesma tarefa devemos considerar que, tipicamente, diferentes solucionadores não sabem as mesmas coisas. Eles não compartilham uma base comum com as mesmas intuições, definições, crenças, informações e procedimentos. Como no fundamento da tentativa de um agente solucionar um problema está o conhecimento que este possui, então a primeira coisa que devemos levar em consideração são seus recursos, os tipos de conhecimento que o agente traz consigo.

1.1 Três tipos de conhecimento

Ao analisar os usos das expressões “conhece” e “sabe”, Meyer Luz constata que essas expressões não apontam para um único conceito. Essa análise revela que essas expressões são utilizadas de três formas distintas, indicando a existência de pelo menos três tipos de conhecimento. (LUZ, 2013)

Assim como Luz, podemo nos perguntar, qual é o significado da palavra “sabe” em sentenças como “João sabe andar de bicicleta ” e “Gödel sabe como provar a completude deste sistema lógico". Nestas sentenças, a palavra expressa uma habilidade, uma aptidão dos indivíduos. A sentença "João tem a habilidade para andar de bicicleta" poderia ser utilizada no lugar da primeira mantendo o mesmo sentido. Este tipo de conhecimento se adquire, sobretudo, através da prática e repetição.

Há um segundo modo de utilizar a palavra "sabe". Este uso aparece em sentenças como "Galileu sabe que a Terra gira em torno do sol" e "João sabe que Brasília é a capital do Brasil". Nestas sentenças nós não estamos afirmando que o sujeito possui uma habilidade, mas uma crença "privilegiada". Essas crenças possuem um status "privilegiado" porque, se assumirmos que as sentenças acima são verdadeiras, então além de Galileu e João estarem em condições de fornecer justificações adequadas para sustentar essas crenças, essas crenças devem estar de fato corretas. Este tipo de conhecimento é facilmente transmitido pela linguagem.

Já a expressão "conhece" é encontrada em sentenças como "Freud conhece Londres", "João conhece o atual reitor da universidade" e "Albert conhece a teoria da gravitação universal de Newton". Nestes casos, não usamos a palavra "conhece" nem no sentido de ter uma habilidade, nem no sentido de ter uma crença, e, portanto, podemos distinguir esse tipo de conhecimento dos anteriores. Nestas sentenças, a palavra expressa certa familiaridade dos indivíduos com lugares, pessoas, objetos e coisas. Esse tipo de conhecimento é adquirido quando o indivíduo tem diversas experiências com algo.

Para nós, todo conhecimento é uma relação entre o sujeito do conhecimento (o agente) e aquilo que se conhece. Como há diferentes tipos de coisas que os agentes podem conhecer (proposições[4], habilidades, pessoas, lugares etc.), há diferentes tipos de conhecimento. Por exemplo, quando dizemos que "João sabe como andar de bicicleta", fazemos referência a certa atividade (andar de bicicleta) para a qual João têm aptidão. Neste caso, uma atividade é conhecida por João, no sentido de que João têm aptidão para realizá-la intencionalmente. Já quando dizemos "João sabe que $\sqrt{4} = 2$", fazemos referência a determinada proposição ($\langle\sqrt{4} = 2\rangle$). Neste casos, uma proposição é conhecida por João, no sentido de que João crê corretamente que a proposição é verdadeira e possui uma justificação para tal). E quando dizemos que "João conhece Pedro", fazemos referência, nesta frase, a determinada pessoa (Pedro), da qual afirmamos que João possui certa familiaridade. Em cada um desses casos a palavra "conhecimento" possui um significado diferente e expressa um tipo diferente de conhecimento.

Chamamos de conhecimento proposicional a relação que se estabelece entre os sujeitos do conhecimento e as proposições. Já a relação entre os agentes e as habilidades, chamaremos de conhecimento procedural. Nesta dissertação tratamos, sobretudo, de dois tipos de conhecimento: o conhecimento proposicional (ou declarativo) e o conhecimento

[4] Uma proposição é o pensamento expresso por uma sentença declarativa e que pode ser verdadeira ou falsa

procedural, abordando superficialmente o conhecimento por familiaridade.

Conhecimento por familiaridade

Neste trabalho não vamos apresentar uma definição de conhecimento por familiaridade. Nesta seção, nos limitamos a fornecer alguns esclarecimentos sobre o uso da expressão “conhece”. Isso porque, temos interesse em tratar deste tópico apenas na medida em que para um sujeito conhecer as coisas (por familiaridade) é preciso ter experiências com elas, e essa experiência permite ao sujeito manifestar as outras duas formas de conhecimento. Isto é, por meio da experiência com as coisas nos tornamos conscientes de certas informações e podemos executar ações envolvendo essa coisas. A experiência com objetos nos permite tanto conhecer certas proposições, quanto utilizá-los na execução de tarefas.

Neste trabalho, assumimos um viés dialético para entender a relação entre experiência e conhecimento. Acreditamos que a experiência é resultado da interação recíproca entre nosso sistema perceptivo que captura informações do mundo ao nosso redor e de esquemas impostos pelos sujeitos da experiência. Usando a terminologia de Piaget (1954), *assimilamos* os fenômenos de acordo com nossos esquemas atuais e *acomodamos* nossos esquemas as demandas dos fenômenos. Deste modo, nossa experiência com o mundo não é passiva e neutra (como acreditam certos empiristas), nem os esquemas que organizam nossa experiência do mundo estão imunes a revisão (como acredita certo racionalismo). Nosso contato com o mundo é ativo e interpretativo. Ele é interpretativo porque experienciamos o mundo através de certos esquemas prévios (categorias, conceitos, teorias e procedimentos) e é ativo porque estamos revisando esses esquemas constantemente para atingir uma maior adaptação ao mundo.

Usamos o termo esquema para descrever tanto a estrutura que organiza nossas experiências do mundo, quanto a estrutura que nos permite atuar no mundo, modificando o estado das coisas. Nossa experiência se torna inteligível porque ela está organizada por meio de categorias gerais como objeto, propriedade, relação e ação. Essas categorias se dividem em diversos conceitos diferentes: nas diversas propriedades (ser azul, estar sentado, pertencer, etc.), relações (ser maior, estar entre, etc.) e ações (abrir, chegar, provar, etc.). Devemos notar que em certas circunstâncias estes conceitos adquirem uma extensão. A extensão de um conceito é o objeto (ou conjunto de objetos) ao qual ele se aplica em dado contexto. Por exemplo, se limitarmos o contexto a sala onde eu estava no momento em que comecei a escrever esta frase, o conceito de *ser azul* se aplicava a certo conjunto de objetos. Este conjunto inclui objetos como minha jaqueta, minha garrafa de água, minha pasta etc. isto é, naquele contexto todos esses objetos eram azuis. Nós não precisamos pensar que a extensão de um conceito aparece apenas em contextos estáticos, isto é, em um instante, mas também em intervalos. Podemos pensar que os conceitos possuem uma extensão durante eventos que possuem uma duração. Em um contextos no qual as mudanças são permitidas, a extensão das propriedades e das relações é alterada por meio da realização de ações envolvendo certos objetos. Por exemplo, a cor de minha garrafa deixaria de ser azul se ela fosse pintada completamente de vermelho. Neste caso, iniciaríamos em um estado no qual a minha garrafa estava na extensão da propriedade *ser azul* e terminaríamos em um estado no qual ela não estaria mais na extensão de tal propriedade, mas estaria na extensão da propriedade *ser vermelho*. E durante essa sequência de estados, a garrafa estaria na extensão da ação *ser pintada de vermelho*. Não são apenas as ações que ocorrem em sequências de estados, os objetos também possuem propriedades e relações durante uma sequência de eventos. Durante a pintura (e não apenas em cada instante) ela possui a propriedade de ser de alumínio e, portanto, nessa sequência de estados ela está na extensão da propriedade *ser de alumínio*. Neste exemplo, apesar de a extensão de *ser azul* ter sido alterada, o conceito de *ser azul* (a ideia que temos do que é ser azul) não foi modificada pela realização da pintura da garrafa de vermelho, apenas a extensão do conceito foi modificada. O termo "ação" não costuma ser utilizado para falar de mudanças na definição de um conceito, mas de sua extensão.

Nem sempre as informações que percebemos sobre o mundo estão de acordo com nosso esquemas prévios e nem sempre nossos esquemas de ação provocam os efeitos intencionados. Dessa falta de ajuste surge a necessidade de revisar os esquemas atuais. Essa necessidade pragmática de ajustar os esquemas ao mundo pode levar a redefinição dos conceitos, abandono de certas categorias e revisão até dos fundamentos de uma teoria. Como veremos posteriormente, na lógica das transações não são representadas as mudanças de tais esquemas, apenas a utilização de uma base de conhecimento que se assume fixa.

Para nós, o conhecimento por familiaridade pressupõe uma experiência pessoal com o objeto primário que se conhece. Vamos apontar a ideia por trás das noções de "experiência pessoal" e "objeto primário" com alguns exemplos. Se a sentença "João conhece Niagara Falls" é verdadeira, então João esteve nessa catarata, ao invés de apenas ver uma foto ou vídeo do lugar. Isto é, ele teve uma experiência com o objeto primário e não com uma representação do objeto (um objeto secundário como uma foto, vídeo ou gravação). Outro exemplo, a verdade da sentença "João conhece Pedro" implica que ambos entraram em contato e não apenas que João tenha visto uma gravação de pedro. Neste último caso, João poderia até ter conhecido Pedro pela internet, em uma videoconferência ou rede social, ainda assim, diríamos que João estava conversando com o próprio Pedro e não que estava assistindo e interagindo com uma gravação do Pedro. A experiência deve ser pessoal em dois sentidos. Primeiro, conhecer uma coisa é diferente de já ter ouvido falar dela de de saber informações sobre ela. Um biógrafo, por exemplo, pode saber muito a respeito da pessoa sobre quem escreve, sem nunca a ter conhecido. Segundo, não é suficiente estar familiarizado com algo para conhecê-lo. Existem situações nas quais interagimos com frequência pessoas lugares e objetos até adquirir certa familiaridade, mas esta interação é superficial e produz conhecimento. Por exemplo, uma pessoa que faça a mesma viagem com frequência, passando sempre por determinada cidade, mas sem parar para conhecê-la pode adquirir familiaridade com a cidade, com suas ruas, o movimento etc.

Quanto a natureza daquilo que se conhece, algumas vezes, se aponta que esse conhecimento se aplica apenas a particulares concretos, mas não a universais e tipos. Entretanto, nós usamos sentenças como "João conheceu a injustiça bem jovem", "João conhece o novo modelo do Iphone" e "João conhece o perdão" no sentido de João está familiarizado com essas coisas e não apenas que ele ouviu falar ou que sabe o significado dos termos "injustiça" "Iphone" e "perdão". Por isso acreditamos que podemos conhecer por familiaridade estruturas abstratas.

Conhecimento Proposicional (declarativo)

Neste trabalho, chamamos de conhecimento proposicional (ou declarativo), o tipo de conhecimento que relaciona agentes e proposições. As proposições são pensamentos sobre os estado das coisas e, portanto, este é o conhecimento que um sujeito possui sobre o estado das coisas (das propriedades e das relações entre as coisas).

Esse tipo de conhecimento visa descrever tanto fatos de algum fragmento do "mundo real" (como o fato de que a terra possui um único satélite natural), quanto situações hipotéticas (como o que aconteceria se as calotas polares derretessem) e mundos fictícios (como no caso de relatos literários). Usualmente expressamos o conhecimento proposicional de um agente por meio de sentenças da forma "i sabe que P", onde "i" denota um agente e "P" expressa uma proposição. Por exemplo:

- Giuseppe sabe que todo número natural possui sucessor.
- Lineu sabe que nenhum mamífero é uma ave.

O conhecimento proposicional de um agente é tradicionalmente, identificado com o conjunto das crenças verdadeiras e justificadas desse agente. Isto é, nesta perspectiva cada uma das seguintes condições são necessárias e o conjunto delas é suficientes[5] para que um sujeito (i) conheça determinada proposição (P):

1. O agente i deve acreditar em P, isto é, acreditar que P corresponde ao estado das coisas. O conhecimento implica crença. Se i sabe que P, então i acredita que P. Não é possível, por exemplo que alguém saiba que $1 + 1 = 2$, mas não acredite nisso ao mesmo tempo. Entretanto, o conhecimento não requer um grau de crença absoluto, é possível acreditar em algo sem ter certeza, pois nem sempre temos justificações fortes para nossas crenças.

[5] Edmund Gettier apresenta dois contra exemplos a tese de que conhecimento é crença verdadeira justificada. Este contra-exemplos evidenciam o fato de que as três condições tomadas em conjunto não são suficientes para o conhecimento. (GETTIER, 1963).

2. P deve ser verdadeira. Uma crença em uma proposição falsa não é conhecimento, portanto, embora crer em P seja necessário, a condição 1 não é suficiente para haver conhecimento, é preciso que P seja de fato verdadeira.

3. A crença verdadeira em P deve estar justificada. Ter uma crença verdadeira não é suficiente para saber algo, o sujeito deve ter razões para acreditar que P é verdadeira. Uma justificação adequada para tal pode ser entendida de forma fraca ou forte. Quando a evidência é suficiente para garantir logicamente a verdade de P, temos uma justificação forte e quando a evidência apenas endossa a possibilidade de P ser verdadeira, temos uma justificação fraca.

De acordo com a definição tradicional, i sabe que P se, e somente se:

1. O agente i acredita que P.

2. P é verdadeira.

3. A crença de i em P está justificada.

As sentenças sobre os estados epistêmicos dos indivíduos e essa teoria tradicional do conhecimento podem ser formalizadas a partir de uma lógica modal, onde os verbos crer e conhecer são representados por diferentes operadores modais. Nessa dissertação entretanto, não pretendemos definir esses conceitos e esta teoria em uma linguagem modal, mas utilizar uma lógica para representar diferentes tipos de conhecimento, dispositivos que utilizam esses conhecimentos e os estágios pelos quais uma tarefa passa, avaliando o grau de sucesso da lógica nessa empreitada.

Nosso conhecimento sobre o estado das coisas (suas propriedades e relações) é tão relevante quanto nossa capacidade de modificar tais estados. Se somos capazes de realizar ações que atualizam o estado das coisas, e de representar essas ações, então é proeminente entender esse tipo de ação se articula do ponto de vista lógico-conceitual com propriedades e relações[6] podendo modificar suas extensões[7].

Conhecimento Procedural

Além de pensar representações corretas de fragmentos do mundo, também sabemos como agir em diversas situações. Com o intuito de atingir certos objetivos somos capazes de realizar diversos procedimentos independente de termos (ou não) a capacidade de fornecer uma descrição adequada do procedimento que estamos realizando. Nesta dissertação, nós identificamos o conhecimento procedural de um agente com o conjunto de procedimentos

[6] Propriedade e relações em sentido intencional.

[7] Relações no sentido da teoria de conjuntos, o conjunto de tuplas ao qual as propriedades e relações se aplicam

que ele têm aptidão para realizar intencionalmente. Esse conjunto pode incluir habilidades como dirigir; andar de bicicleta; multiplicar; jogar xadrez; provar determinados teoremas etc. e requer que diversas operações sejam sequenciadas, modificando o estado das coisas.

O conhecimento procedural pode envolver a manipulação de diversos tipos de coisas, desde objetos ordinários (como cadeiras, pedras, teclados etc.), até objetos simbólicos, que representam objetos, conceitos, situações e pensamentos. Para realizarmos com sucesso a manipulação destes objetos, não é necessário sermos capazes de descrever corretamente o procedimento que estamos realizando. Não somos capazes, por exemplo, de descrever os diversos movimentos que fazemos para manter o equilíbrio ao andar de bicicleta, nem precisamos estar cientes das regras gramaticais e de inferência com as quais agimos de acordo ao argumentar em português. Muitas vezes, sabemos como agir, antes de adquirir um entendimento conceitual da situação e do que estamos fazendo. Nesse sentido, a ação precede o entendimento. Além disso, saber certas instruções não implica em ser capaz de executá-las. Portanto, não é necessário, nem suficiente conhecer certas instruções para sermos capazes de realizá-las. No entanto, mesmo que o próprio agente não saiba descrever os procedimentos que é capaz de realizar intencionalmente, nós que estamos interessados em investigar o conhecimento dos agentes, por meio da criação e análise da representação de dispositivos que sabem como de resolver problemas, precisamos ter uma forma de representar tais habilidades. E a forma de representar tal conhecimento se dá pela descrição destes em alguma linguagem apropriada.

Tanto o conhecimento proposicional quanto o procedural são disposições. Disposições são tendências que “carregamos” conosco o tempo todo, mas que só se manifestam em situações apropriadas. Nós não estamos o tempo todo conscientes daquilo que sabemos, nem estamos exercendo a todo instante as habilidade para as quais temos aptidão. Quando estou dormindo profundamente, por exemplo, eu não deixo de saber que $2+2 = 4$ nem como andar de bicicleta, pois possuo estes conhecimentos como disposições. O conhecimento proposicional se manifesta quando estamos conscientes de um pensamento e podemos fazer uso dele no raciocínio e no controle racional da ação, já o conhecimento procedural se manifesta por meio de performances e podemos fazer uso dele para realizar certos objetivos.

O conhecimento procedural de um agente pode ser expresso por meio de sentenças da forma “i sabe como A” e “i sabe A ”, onde “i” é um agente e “A” expressa uma ação. Por exemplo:

- João sabe como transferir dinheiro de sua conta para a de Pedro.
- Pedro sabe como chegar em casa, sem passar pela EPTG.

Nos parágrafos anteriores identificamos o conhecimento procedural de um agente com o conjunto de procedimentos que ele tem aptidão para realizar intencionalmente. Por esta definição, um agente i sabe como realizar um procedimento A se, e somente se, o agente i têm aptidão para realizar A intencionalmente. Essa definição requer alguns esclarecimentos.

Primeiro vamos esclarecer o que entendemos por aptidão. Um agente i deve ter a capacidade[8] de realizar A com sucesso nas circunstâncias apropriadas se ele sabe como fazer A. Entretanto, a mera capacidade de i realizar A não é suficiente para que i saiba como realizar A. Isso porque, para saber como realizar um procedimento é preciso ter a capacidade de realizá-lo de forma rotineira. Por exemplo, o fato de um jogador ter conseguido resolver um cubo mágico apenas por tentativa e erro, sem nenhuma estratégia implica que ele é capaz de resolvê-lo, mas esse sucesso eventual não é suficiente para garantir que ele saiba como resolver cubos mágicos porque é provável que ele vá falhar na maioria das vezes que empregar essa "estratégia". Por aptidão para realizar um procedimento entendemos justamente a capacidade de um agente realizar um procedimento com sucesso de forma rotineira. Como a aptidão é uma disposição, um agente só consegue manifestá-la em situações apropriadas e não em qualquer situação. Nós não esperamos que um agente consiga manifestar sua habilidade de andar de bicicleta se ele não tiver com uma bicicleta, ou que alguém consiga nadar se estiver no meio do deserto. Nesse sentido, o conhecimento procedural depende dos instrumentos e meios apropriados para se manifestar.

Segundo, o agente i ter a capacidade realizar A de forma rotineira (ter aptidão para A), não implica que i saiba como fazer A. Isso porque, o agente i pode ser capaz de realizar A apenas de maneira fortuita ou acidental. Por exemplo, o fato de uma criança pequena pegar o telefone para brincar e ligar acidentalmente para o corpo de bombeiros diversas vezes, não fornece uma justificação para afirmar que ela sabe como ligar para os bombeiros, pois se tratam de ligações acidentais e não de atos intencionais. Este exemplo mostra que a criança é capaz de ligar para os bombeiros rotineiramente, não que ela *sabe* como fazê-lo. Portanto, o conhecimento procedural requer que o agente tenha aptidão para realizar o procedimento de forma intencional. Obviamente, ter a intenção de fazer A não é suficiente para saber como fazer A, pois se um agente i sabe como fazer A, então i consegue traduzir sua intenção em ações bem sucedidas nas circunstâncias apropriadas. Ainda nos resta esclarecer algumas questões: O que é uma intenção? E o que significa realizar uma ação intencionalmente? Vamos deixar essas questões de lado até a próxima seção.

Quando existe um procedimento que resolve certa classe de problemas e ele pode ser descrito com clareza e precisão por meio de um conjunto finito de instruções, e além disso, sempre que elas forem seguidas de forma mecânica levarem em um tempo finito à

[8] A mera possibilidade lógica de um agente realizar um procedimento A não implica que ele seja *capaz* de realizá-lo. Estamos interessados naquilo que um agente pode efetivamente fazer dadas suas limitações físicas e computacionais e que estão estabelecidas em um certo contexto.

solução de qualquer instância do problema, dizemos que há um algoritmo para resolver este problema é que o problema é computável. Essa noção intuitiva de computável pode ser definida com precisão por meio de Máquinas de Turing e Funções recursivas para problemas como calcular funções nos naturais e decidir se uma expressão pertence a uma linguagem (e muitos problemas podem ser representados desta forma). Um exemplo de procedimento algorítmico de decisão é a construção de tabelas da verdade para decidir se uma fórmula qualquer é um teorema da lógica proposicional clássica. Esse método mecânico sempre termina indicando, se a fórmula for um teorema, que ela, ou se ela não for, indicando que ela não é um teorema.

Para problemas sem métodos de decisão, ainda pode haver procedimentos não-algorítmicos que podem ser empregados rotineiramente na tentativa de resolvê-los. Neste casos, entretanto, não há garantias de que o procedimento encontrará uma solução em um tempo finito para todas as instâncias do problema. Um exemplo de procedimento não-algorítmico deste tipo é usar o método do tablô para determinar se uma fórmula é teorema da lógica de primeira ordem clássica. Esse procedimento será eficaz sempre que a fórmula em questão for de fato teorema, mas caso a fórmula não seja o procedimento pode nunca terminar.

Muitas tarefas com as quais um agente se depara não são inteiramente novas para ele. Isso permite ao agente desenvolver esquemas estereotipados, de modo que ao identificar uma dessa tarefas, ele pode realizá-la diretamente seguindo um desses esquemas pré-definidos, sem precisar procurar por uma forma de realizá-la. Diversas e ferramentas e noções foram desenvolvidas para tentar capturar e explicar esse fenômeno, como: *chunks*, *frames*, *sistemas de produção* e *scripts*. Um *script* (roteiro), por exemplo, é um formalismo desenvolvido para tentar representar esse tipo de conhecimento acessado automaticamente por agentes que se deparam com uma tarefa familiar. Nestes casos, a tarefa se desenrola sem maiores complicações, sendo um mero exercício e não um problema. Um problema é uma tarefa para a qual o solucionador não possui um esquema pré-definido e por esse motivo precisa procurar por uma forma de realizá-la.

Nem todo nosso conhecimento procedural pode ser entendido considerando-o exclusivamente como um esquema estereotipado. Para as tarefas que requerem procurar por uma forma de realizá-las há diversas técnicas que podem ser empregadas, isso porque, muitos problemas não possibilitam utilizar um algoritmo de força bruta para encontrar uma solução. Para estes problemas que são dispendiosos computacionalmente, é possível incorporar diferentes estratégias heurísticas que guiam a procura. Vamos tratar brevemente das estratégias heurística na próxima seção.

Heurísticas

Diversos problemas bem-definidos possuem um espaço de busca muito grande porque o número de estados gerados a cada nível cresce exponencialmente. Desse modo, não é possível (do ponto de vista prático) efetuar uma busca exaustiva percorrendo todos os caminhos que poderiam levar ao objetivo. Para determinar quais caminhos vão ser explorados e em qual ordem, existem diversas estratégias que podem ser adotadas. Heurística é o nome dado as estratégias práticas que enfrentam a complexidade do problema guiando a busca através dos caminhos mais promissores, mas que são potencialmente falíveis. Por exemplo, problemas que são representados na lógica de predicados e cujas transições são inferências lógicas requerem esse tipo de estratégias porque não é possível examinar todas as inferências lógicas que podem ser realizadas, precisamos "adivinhar" aquelas que são relevantes.

De acordo com Luger, a palavra "heurísticas" tem origem grega e significa algo como *descoberta*. Polya, em *How to solve it* (POLYA, 1957), empregou o termo ao retomar o estudo das regras e métodos de descoberta. Neste livro, Polya apresenta um guia de regras práticas empregadas por matemáticos na solução de problemas. Essas regras auxiliam na compreensão do problema e na escolha do que fazer quando não se sabe ao certo o que deveria ser feito.

Há dois tipos de estratégia heurísticas, as fracas e as fortes. As fracas possuem um escopo de aplicação geral e as fortes se aplicam em áreas específicas do conhecimento. Como exemplos de heurísticas fracas, que costumam ser empregadas por seres humanos, podemos citar uma técnica conhecida como *subida pela montanha*, na qual se escolhe o operador que reduz a maior diferença entre o estado atual e o almejado; e a uma técnica conhecida como *análise dos meio e fins* que permite buscar as condições que permitem aplicar operadores que eventualmente estejam indisponíveis no momento. Como exemplos de técnicas heurísticas específicas na matemática podemos citar a exploração de casos particulares como o seguinte: se no problema houver um parâmetro n inteiro, podemos calcular os casos particulares com os valores 1, 2 e 3 identificando um padrão e verificar por indução se o padrão vale para todo n.

O objetivo desta seção não é descrever algoritmos de busca em grafos que utilizam heurísticas podem ser implementados, mas somente apontar que dada a vastidão de informações e procedimentos aos quais temos acesso, é preciso ter um critério que selecione aquilo que é relevante permitindo ao conhecimento avançar tão longe quanto possível. No capítulo 3 mostramos como a análise dos meios e fins pode ser representada na lógica das transações.

Ações e intenções

O objetivo deste capítulo é apresentar uma breve discussão sobre as ações, acontecimentos e sobre as intenções. Tratamos ao longo das próximas páginas principalmente das seguintes questões: o que é significa palavra "ação"? O que é uma intenção? E, o que significa realizar uma ação intencionalmente?

Verbos de ação são aqueles que expressam ações, acontecimentos, estados e atividades mentais. Alguns exemplos de verbo de ação são: "ir", "cair", "querer" e "perceber". Alguns filósofos, como Davidson[9], reservam o termo "ação" para indicar tudo que um agente faz intencionalmente e apenas isso. Acontecimentos naturais (como o movimento dos planetas ou o crescimento das plantas) e movimentos corporais involuntários (como espirrar e cair) não são classificados como ações por Davidson. De modo semelhante, para Searle[10], uma ação consiste de dois componentes, uma intenção e o movimento corporal, sendo que a intenção é vista como causa do movimento corporal.

Neste trabalho, quanto usarmos o termo "ação" vamos pensar em um sentido mais abrangente que filósofos citados anteriormente, incluindo tanto acontecimentos que ocorrem sem a intenção de um agente e atividades mentais, quanto as ações propriamente ditas, que são realizadas intencionalmente e atividades mentais. Além disso, vamos argumentar que para explicar o que significa "executar uma ação intencionalmente", não precisamos lançar mão de uma relação de causa e efeito entre a intenção de fazer a ação e a ação, como Searle propõe, pois podemos defendemos que para definir o que é fazer uma ação A intencionalmente basta articular os conceitos de fazer A e ter a intenção de fazer A utilizando uma conjunção lógica com uma semântica próxima a da conjunção clássica.

Nos próximo parágrafos vamos utilizar a distinção, apresentada por Von Wright[11], entre ações como conceitos genéricos e ocorrências particulares de ações. Exemplos de conceitos genéricos de ação são viajar, comer, fumar e assassinar. Um exemplo fornecido por Wright de ocorrência de ação individual é o assassinato é de César por Brutus, outro exemplo, é minha viajem para o Canadá em 2013. Um conceito genérico é definido a partir de conceitos primitivos e operações lógicas.

A seguir, apresentamos alguns exemplos que ilustram como na atuação do ser humano estão envolvidas ações propriamente ditas (realizadas intencionalmente), acontecimentos (que ocorrem sem a intenção de uma agente) e atividades mentais. Considere a seguinte situação. Imagine que João tenta frear, mas acaba atropelando alguém em um acidente automobilístico durante uma noite chuvosa. Neste caso, a ocorrência da ação de atropelar alguém é um acontecimento, algo realizado sem uma intenção, pois se trata de um acidente. Nessa situação também ocorreu uma ação intencional, pois assim que

[9] (DAVIDSON, 2001).
[10] (SEARLE; WILLIS et al., 1983).
[11] (WRIGHT, 1981).

João percebeu a vítima, ele tentou frear o carro. A tentativa de frear o carro foi uma ação realizada intencionalmente. Neste exemplo, a ocorrência da chuva foi um acontecimento não intencional. Também podemos imaginar uma situação diferente na qual João atropela a vítima deliberadamente, pois esta era de um desafeto seu. Neste caso, a ocorrência da ação de atropelar é realizada intencionalmente. Portanto, para nós, algumas ações, em quanto conceitos, podem ser aplicadas em ocorrências que possuem modos de intencionalidade diferentes. Na primeira situação, o conceito indicado por “atropelar alguém” é aplicado à ocorrência de uma ação intencional, mas na segunda situação, esse conceito é aplicado à ocorrência de uma ação não intencional. Uma expressão como *fazer uma cesta de três pontos*, pode ser aplicada em ações propriamente ditas e acontecimentos. No primeiro caso, ela é feita intencionalmente e no segundo sem intenção (como no caso de alguém tentar passar a bola para um companheiro, mas acabar fazendo a cesta). Portanto, uma ação, em quanto conceito, não precisa determinar se as ocorrências da ação serão realizadas intencionalmente ou não. Isso não significa que as ações, enquanto conceitos, não possam ser construídas de modo a determinar a intencionalidade. Por exemplo, ao conceito indicado pela expressão “atropelar alguém intencionalmente” aplica-se apenas ocorrências intencionais. De modo geral, uma ação pode ser definida a partir da combinação de conceitos de ações intencionais, não intencionais ou que não determinam o modo de execução quanto a intencionalidade. Quando nós, afirmamos que uma ação A pode ser realizada de modo intencional ou de modo não intencional não queremos dizer que uma ação pode ser intencional e não intencional se descritas de modo distintos. Para nós, se A é uma ação, então a ação de fazer A intencionalmente é diferente da ação de fazer A sem intenção (tanto se pensarmos nessas ações enquanto conceitos quanto ocorrências) entretanto, as duas últimas ações podem ser vistas como sendo do mesmo tipo de ação (fazer A). Deste modo, fazer A intencionalmente implica em fazer A, assim como fazer A sem intenção implica em fazer A, mas, a volta não é válida, pois fazer A não implica em fazer A intencionalmente, nem implica em fazer não intencionalmente.

Para definir as ações, no sentido de especificar as formas como elas podem ser realizadas, nós não podemos elencar apenas um conjunto de movimentos corporais de um agente, porque as mudanças ambientais também fazem parte da definição, já que alguém poderia realizar o mesmo movimento corporal e mas não realizar a ação. Alguém, por exemplo, poderia errar a cesta, caso estivesse em um posição diferente. Neste caso, a ação envolvida (fazer a cesta) depende tanto do fato de ela ter sido lançada pelo agente - independente do movimento corporal específico realizado - quanto do fato de a bola passar pelo aro - independente do trajeto específico percorrido pela bola. Em algumas situações, também também vamos precisar utilizar algumas condições para definir uma ação. Para definir os movimentos válidos de uma peça no jogo de xadrez, não basta indicar a forma de deslocamento que ela pode fazer, pois um jogador não pode realizar um movimento que o coloque seu próprio e portanto um movimento só é válido caso o deslocamento da peça não coloque seu próprio rei em xeque.

Além das ações propriamente ditas e dos acontecimentos, que atualizam o estado das coisas, também estamos interessados em alguns atividades mentais como: perceber, ver, pensar e raciocinar. Diferente das ações propriamente ditas, que são procedimentos para manipular o mundo intencionalmente, e dos acontecimentos que são mudanças no mundo não intencionais, as atividades mentais são métodos de obtenção de informação. Enquanto as ações propriamente ditas nos permitem atuar no mundo, esses procedimentos nos permitem acessá-lo. Muitas vezes, as ações que realizamos são casos híbridos, nos quais o método de obtenção de informação são combinados com ações que atualizam o estado das coisas.

Como apontado por Von Wright[12], as ações provocam dois tipos de mudança no mundo, aquelas que estão presentes na própria definição da ação (os resultados de uma ação) e aquelas que aparecem como efeitos colaterais de uma ação (suas consequências). A conexão entre uma ação e seus resultados é intrínseca, já a relação entre uma ação e suas consequências é extrínseca. O seguinte exemplo ajuda a ilustrar esta distinção: se a ação de abrir uma garrafa é executada, é um efeito (intrínseco) que a garrafa esteja aberta no final da ação. Caso uma garrafa não esteja aberta ao final da ação, a ação executada não poderia ser a ação de abrir uma garrafa. Pode acontecer também que abrir a garrafa tenha como consequência (extrínseca) que o líquido dentro dela derrame. A relação entre o derramamento do líquido e a ação de abrir a garrafa pode até ser causal, mas é extrínseca. Uma ação pode ser definida em termos dos seus resultados e condições de aplicação, já as consequências de uma ação geralmente ocorrem algum tempo depois de sua realização e é impossível saber de antemão todas as consequências de uma ação.

Também precisamos chamar a atenção para outra ambiguidade no significado do termo “efeito”. Por “efeito” de uma ação podemos entender tanto o processo de mudança correspondente a ação, a sequência de estados provocados pela ação, quanto apenas o estado final que ação prova. Deste modo, podemos pensar no efeito que abrir uma uma garrafa provoca tanto como o estado no qual a garrafa está aberta e vazia, quanto um processo envolvendo diversos estágios no qual a tampa vai sendo retirada e pelo qual a água se esvai pouco a pouco a pouco. Esse ponto retoma a discussão presente no capítulo 2 sobre ambiguidade do termo “solução”, que pode significar tanto o resultado final obtido pela realização de uma tarefa, quanto modo de obtê-lo. Isso porque um problema é um tipo de tarefa e solucionar um problema é executar a tarefa.

As transações são uma categoria que incluem os métodos de obtenção de informação, as ações propriamente ditas, acontecimento e combinações híbridas desses elementos. O termo “transação” é utilizado com dois nichos diferentes. No primeiro caso, no qual não estamos interessados, o termo indica acordos e compensações comerciais. No segundo caso,

[12] (WRIGHT, 1981).

o termo é utilizado no contexto das ciências da computação, mais especificamente na área que estuda banco de dados (a representação, armazenamento, acesso e manipulação de dados). É no segundo sentido que o termo é utilizado por Bonner e Kifer[13] e por nós nesta dissertação. Nesse contexto, uma transação é uma ação atômica composta por consultas a um banco de dados (métodos de obtenção de informação) e atualizações no banco de dados (métodos que modificam o banco de dados). A ideia das transações é aglutinar uma série de ações em uma única ação atômica, o que permite tratar como indivisível algo que a principio não era. Apesar de a definição de uma transação poder envolver a composição de diversas ações mais simples, uma transação é vista do ponto de vista de sua execução como um "átomo", isto é, ela possui a garantia lógica-conceitual de que é completamente executada ou que falha sem provocar nenhuma mudança estado. Por exemplo, uma tarefa como *fazer um bolo* é, a princípio, realizada gradualmente e poderíamos dizer que foi executada parcialmente se pararmos em um estado no qual o bolo está não está pronto. Entretanto, se pensarmos em uma transação, então, a partir de um estado inicial, haverá apenas duas possibilidades para interpretar a situação: a ação é completamente executada ou a ação falha completamente. Se uma transação falha, temos garantia de que não sairemos do estado inicial e se ela é cometida temos garantia de que chegaremos ao estado final, não há uma terceira opção, como por exemplo, a ação ser executada parcialmente. Deste modo, a execução de uma transação é vista como algo indivisível e, portanto, se ação de *fazer um bolo* é vista como uma transação, então não faz sentido do ponto de vista conceitual dizer que ela foi executada parcialmente, deixando o mundo em algum estado intermediário entre o estado inicial e o estado final da transação. No próximo capítulo, veremos que a lógica das transações foi construída com o objetivo de fazer corresponder a noção de execução de uma transação com a valoração de uma sentença em um sequência de estados. Na lógica das transações os diversos modos de aglutinar as ações são definidos por meio de operações lógicas, de modo que cada operador lógico articula os procedimentos a sua maneira. Como todos as ações em $\mathcal{TR}$ são vistos como transações podemos tratar a execução de procedimentos que a princípio seria decomponível como se ela fosse indivisível. A partir de agora nós vamos usar nesta dissertação o termo "ação" para tratar de transações.

1.2 Ações intencionais

Assim como as crenças e os desejos, as intenções são estado mentais que se direcionam para algo. Mas, diferente de um desejo que também pode se direcionar para objetos e pessoas, uma intenção se direciona apenas para ações. Por exemplo, eu posso desejar um novo celular (um objeto) ou desejar viajar para Paris (usualmente uma ação propriamente dita), mas, se eu tenho uma intenção, então ela é uma intenção de fazer algo.

[13] (BONNER; KIFER, 1995).

Eu posso, por exemplo, ter a intenção de viajar para Paris (uma ação propriamente dita) ou ter a intenção de pesar 70 kg (um acontecimento). Note que apesar de agentes poderem ter a intenção de pesar 70 kg, pesar 70 kg não é algo que se realize intencionalmente. Ninguém responderia às perguntas “O que você está fazendo?” e “O que você costuma fazer?” dizendo “eu estou pesando 70 kg” e “eu costumo pesar 70 kg”. Como pesar 70 kg não é algo que se faça intencionalmente, então pela nossa definição de conhecimento procedural apresentada anteriormente, não é possível que alguém saiba como pesar 70kg. Entretanto, é possível que alguém saiba o que fazer para pesar 70 kg; que alguém seja capaz de pesar 70kg; e que alguém saiba como fazer algo que tenha como consequência deixá-lo com o peso 70 kg. Assim, para satisfazer a minha pretensão de pesar 70 kg deve acontecer algo ou eu devo fazer algo que tenha como consequência me deixar com 70 kg. Portanto, uma intenção é satisfeita se e somente se a ação propriamente dita representada no conteúdo da intenção é executada intencionalmente ou se o acontecimento intencionado ocorre. Deste modo, tudo que pode satisfazer a intenção de uma ação propriamente dita é a realização intencional dessa ação. Note que podemos desejar fazer algo sem ter a intenção de o fazer. Por exemplo, apesar de desejar fumar um cigarro, eu poderia não ter a intenção fumar um cigarro se estivesse tentando parar de fumar. E um médico pode ter a intenção de amputar a perna de alguém sem desejar amputá-la.

Todas ações propriamente ditas são realizadas intencionalmente. Mas, em alguns casos, a intenção surge antes da ação começar a ser executada. Por exemplo, geralmente, a intenção de votar em determinado candidato existe muito antes do eleitor votar no candidato. Independente de haver ou não uma intenção prévia, sempre há uma intenção que acompanha às ações propriamente ditas. Uma ação pode ser definida por meio de muitos efeito não relacionados. Por exemplo, atirar em alguém pode ter dois efeitos: matar o alvo e descarregar a arma. Ao planejar uma linha de ação para atingir um objetivo, precisamos levar em consideração que as ações podem ter dois tipos de efeito: os intencionais e os não intencionais. Descarregar a arma é normalmente um efeito não intencionado da ação de atirar e matar alguém geralmente é o efeito intencionado. O planejamento de um agente deve levar essa distinção em consideração, caso contrário, são produzidos planos indesejáveis, como atirar em alguém quando o objetivo é descarregar a arma.

Como apontado por diversos filósofos entre eles Searle[14], mesmo que uma ação correspondente à uma intenção ocorra, esta realização pode não satisfazer a intenção. Isso porque a realização da ação deveria ocorrer de “forma apropriada”, isto é, a ação deveria ser realizada intencionalmente. Essa característica não possui paralelo com outros estados mentais. Se eu desejar ficar rico, meu desejo é satisfeito não importa o modo como eu tornei rico e, se eu acredito que está chovendo e de fato está chovendo, então minha crença é satisfeita não importa o que fez chover ou como a chuva foi provocada. Mas, no caso

[14] (SEARLE, 2003).

da intenção existe uma série de contra exemplos, elencados por Searle, que mostram que não basta que ação ocorra, é preciso que ela seja realizada intencionalmente para que a intenção seja satisfeita. Nos baseamos nestes exemplos para enfatizar que uma realizar uma ação intencionalmente não significa o mesmo que ter uma intenção prévia seguida da ação correspondente, nem fazer duas coisas paralelas (ter a intenção enquanto realiza uma ação correspondente) e, que afirmar que a intenção deve causar a ação correspondente não ajuda a definir adequadamente o que significa realizar uma ação intencionalmente.

Suponha que João tenha intenção de matar seu tio e que ele de fato o mate, ainda assim pode ocorrer de sua intenção não ser satisfeita. Imagine que João tinha uma intenção prévia de matar seu próprio tio, mas que com o passar do tempo essa intenção acaba desaparecendo e que João mate seu tio posteriormente, quando já não havia mais nenhuma intenção. Nós, obviamente, não diríamos que ele matou seu tio intencionalmente neste caso. Isso porque para realizar uma ação intencionalmente não é suficiente que o agente tenha uma intenção prévia, a intenção deve em algum sentido "acompanhar " a ação. Os próximos exemplos ilustram a tentativa de esclarecer o sentido de "acompanhar". Isto é, devemos investigar como uma intenção e uma ação correspondente devem estar articuladas conceitualmente para podermos afirmar que a ação é executada intencionalmente.

Podemos pensar que para uma ação ser realizada intencionalmente é suficiente ter a intenção enquanto a ação correspondente acontece em paralelo. Mas, imagine a seguinte variação do exemplo anterior, no qual João tem a intenção de matar seu próprio tio. Suponha ainda, que ao dirigir seu carro João se distraia com o celular, atropelando e matando um pedestre, que por coincidência era seu tio. João matou seu tio intencionalmente? Nossa intuição diz que não. Mas nesta situação, João tinha a intenção de matar enquanto matava seu tio. Por que dizemos que ele não o matou intencionalmente? Um primeiro olhar revela que a intenção de matar não contribui em nada com a realização da morte. Uma tentativa de dizer como a intenção deve se articular com a ação corresponde para que a ação seja realizada intencionalmente consiste em dizer que a ação deve causar a ação. Ou seja, uma ação A é executada intencionalmente se e somente se a ação é executada enquanto há uma intenção de fazer A e está intenção contribui causalmente com a ação. O seguinte contra exemplo mostra que essa tentativa de definir o que significa fazer uma ação intencionalmente também não está correta.

Imagine que João tem a intenção de matar seu próprio tio atropelando-o e que esteja dirigindo pensando na forma como vai fazê-lo. Esses pensamentos deixam João nervoso e o distrai a ponto de fazer com ele atropele e mate acidentalmente um pedestre, que por acaso era o tio de João. Neste caso, é verdade que ele tinha intenção de matar seu tio, que ele o matou e que sua intenção de matar o tio contribui causalmente com a morte, mas não é verdade que ele matou seu tio intencionalmente. O que aconteceu? Poderíamos tentar resolver esse problema redefinindo a noção de causa, mas nós não acreditamos que este é o caminho adequado. Para nós, apelar para noção de causa é um remédio muito forte para uma enfermidade que não é grave. Nós defendemos que para uma ação A ser realizada intencionalmente é suficiente ter a intenção de fazer A e fazer A. Mas, essa definição não teria como um contra-exemplo a segunda situação apresentada anteriormente? Na verdade, não. O problema todo aparece por falta de clareza no significado da conjunção "e" conectando a expressões "ter a intenção de fazer A" e "fazer A", que é utilizada com sentidos diversos na linguagem ordinária. Por exemplo, quando nós encontramos em uma receita a expressão "misture os ingredientes da cobertura *e* a despeje sobre a massa que está na forma" ou quando dizemos "eu fui a feira *e* comprei chuchu " podemos perceber que a conjunção destacada possui um sentido temporal: primeiro misture os ingredientes e, então despeje na massa; e primeiro eu fui a feira e, então comprei chuchu. Entretanto, quando alguém diz "eu tenho 17 anos e eu gosto de jogar futebol", não há qualquer sentido temporal na conjunção, de tal forma que é possível comutar as expressões sem alterar o sentido. Nas próximas seções vamos mostrar como duas conjunções (uma chamada de serial e a outra de "clássica") são formalizadas na lógica das transações e quais as condições de verdade e satisfação de expressões que as contém. Se a conjunção for interpretada de forma serial ($\otimes$) ou paralela ($|$)[15], então, de fato, fazer A e ter a intenção de fazer A não significa o mesmo que fazer A intencionalmente, entretanto se a conjunção "e" for interpretada como uma forma de combinar a ação e a intenção em uma única operação cujas possibilidades de execução são mais restritas se comparadas com a ação e a intenção individualmente ($\wedge$), então teremos o significado desejado.

Em português a articulação entre um verbo de ação e um advérbio não precisa de um conectivo para ser expressa, podendo ser indicada pela flexão do verbo. Por exemplo, eu posso dizer "eu vou para casa pass*ando* pela via estrutural" ao invés de dizer "eu vou para casa *e* passo pela via estrutural". Esse tipo de articulação também pode ser indicada por meio de diversos conectivos como em "eu vou para casa *sem* passar pela via estrutural". Essa sentença possui o mesmo sentido que "eu vou para casa e não eu passo pela via estrutural". *Chegar em casa* e *passar pela via estrutural* são ações não-determinísticas, isto é que podem ser realizadas de diversas maneiras. Assim como o conceito de *chegar em casa* não determina qual caminho deve ser feito, apenas que no final do trajeto se esteja em casa, o conceito de *passar pela via estrutural* não determina de o local de origem do agente, nem o local onde o agente estará ao término da ação. A expressão "chegar em casa passando pela via estrutural" não indica duas ações diferentes que ocorrem em paralelo, mas uma única ação cujos modos de execução são mais restritos que os conceitos que participam de sua definição. Do mesmo modo, a expressão *fazer uma ação A intencionalmente* não indica duas coisas que ocorrem em paralelo, mas uma única ação. Essa ação intencional tem

[15] Nesta dissertação não trataremos da execução de transações paralelas. Uma extensão da lógica das transações que lida com estas questões por meio do conectivo | pode ser encontrado em BONNER e KIFER. (BONNER; KIFER, 1996)

um modo de execução mais restrito que ação isoladamente, já que uma ação poderia ser realizada intencionalmente ou sem intenção e ter a intenção pode ou não ser acompanhada de sua ação correspondente. A seguir veremos que como a conjunção "clássica" da lógica das transações é utilizada para definir uma única ação mais determinísticas ligando dois conceitos. Apesar de na lógica das transações existir um conectivo que capture a articulação entre um verbo e o advérbio de modo desejada, acreditamos que a lógica das transações não é uma linguagem expressiva o suficiente para exprimir adequadamente "ter a intenção de fazer A", assim como ela não é expressiva o suficiente para exprimir sentenças como "alguém que sabe p", onde p é uma proposição. Porém, assim como é possível modelar agentes que sabem p, é possível modela que tem a intenção de fazer A.

Essa seção expôs nossas intuições e ideias sobre diversos tópicos: tipos de conhecimento, intenções, ações e problemas. Nosso objetivo nas próximas seções não é desenvolver uma teoria formal conectando essas ideias, mas mostrar um modo de utilizar a lógica das transações para representar. Isto é nós não vamos dizer o que são os tipos de conhecimento, ações etc. em uma linguagem formal tentando criar uma teoria consistente par descrever esses elementos, mas mostrar como essa linguagem pode ser utilizada para definir ações (como conceitos e ocorrências), seus resultados e consequências, representar os conhecimentos procedural e declarativo por meio de fórmulas o uso destes conhecimentos para resolver tarefas difíceis.

2 A lógica das transações

Enquanto na lógica clássica fazemos asserções sobre um mundo estático, na lógica das transações tentamos capturar a ideia de que o mundo é dinâmico - formalmente, as fórmulas são valoradas em sequências discretas de estados, chamadas de caminhos. Essa lógica foi desenvolvida com o objetivo de fornecer uma ferramenta capaz de capturar a forma como sistemas simbólicos relativamente arbitrário podem ser interpretados e atualizados. Nós sabemos que existem diversas formas de identificar os dados de um problema em cada estágio de resolução. É possível utilizar sistemas simbólicos como as linguagens naturais, matrizes, arranjos, conjuntos de fórmulas etc. Ao invés de nos comprometermos com um sistema de representação especifico, vamos representar os estados por meio conjuntos de tipo abstrato, isto é não vamos indicar, principio, a natureza de seus elementos, mas vamos assumir que existe um procedimento que atribui à cada identificador uma teoria de primeira ordem clássica. Portanto, além das fórmulas que tipicamente esperamos encontrar em uma lógica, na lógica das transações, uma parte da linguagem é tratada como um parâmetro. Um caminho será uma sequência desses identificadores, cuja natureza não precismos especificar, a princípio. Na semântica, o objetivo é interpretar as sequências desses identificadores de estados, de modo a permitir dizer quais ações e relações vigoram nesses caminhos. Nessa lógica, os predicados podem representar tanto ações, quanto relações e propriedades. Grosso modo, a ideia é que a verdade de uma sentença em um caminho significa informalmente que ação representa pela sentença pode ser executada ao longo daquele caminho, isto é, sua execução pode transformar o estado inicial no estado final, passando por diversos estágios intermediários.

Mas, como determinar quais fórmulas são satisfeitas em cada sequência de estados? Ou melhor, como interpretar essas sequências? O que as fórmulas representam? Quais as condições de verdade das fórmulas atômicas e complexas? Bonner e Kifer[1] afirmam que as fórmulas de transação podem ser interpretadas tanto de forma declarativa, quanto imperativa. O que eles pretendem dizer com isso? O objetivo das próximas seções é responder as questões acima fornecendo exemplos que conectam a linguagem da lógica das transações com a linguagem natural. A lógica das transações tem por objetivo formalizar alguns aspectos da execução de transações. Antes de mostrar como isso é feito, vamos analisar a sintaxe e o significado intuitivo em linguagem natural dessas fórmulas.

[1] (BONNER; KIFER, 1997).

Sintaxe

O alfabeto para gerar uma linguagem da lógica das transações consiste nos seguintes símbolos:

- Um conjunto infinito enumerável de símbolos funcionais $\mathcal{F}$. As constantes são vistas como símbolos funcionais 0-ários.
- Um conjunto infinito enumerável de variáveis $\mathcal{V}$. (Usaremos letras maiúsculas para representar variáveis ou palavras iniciadas com letras maiúsculas).
- Um conjunto infinito contável de símbolos de predicado $\mathcal{P}$. Os símbolos de predicado 0-ários são vistos como constantes proposicionais.
- Os conectivos lógicos $\vee$, $\wedge$, $\otimes$, $\diamond$, $\neg$. Outros conectivos podem ser definidos a partir destes.
- Os quantificadores $\forall, \exists$.
- Símbolos adicionais, como "(", ")", e ",".

Termos

Os termos são definidos recursivamente de maneira usual.

As fórmulas de transação

A lógica das transações estende a lógica clássica de primeira ordem com dois novos conectivos, $\otimes$ e $\diamond$, chamados de conjunção serial e executor hipotético.

Definição (Fórmula de transação): uma atômica é uma expressão da forma $p(t_1, \cdots, t_n)$, onde $p \in \mathcal{P}$ é um símbolo de predicado n-ário e $t_1, \cdots, t_n$ são termos. Uma atômica é uma fórmula de transação e se ϕ e ψ são fórmulas de transação, então as seguintes expressões também são:

- $\phi \vee \psi$, $\phi \wedge \psi$, $\neg\phi$, $\phi \otimes \psi$ e $\diamond\phi$.
- $(\forall X)\phi$ e $(\exists X)\phi$, onde X é uma variável.

Há um aspectos para os quais gostaríamos de chamar a atenção dos leitores novamente. Além das fórmulas acima, a lógica das transações pressupõe a existência de conjunto de identificadores de estados e assume que existe algum procedimento que converte esses identificadores em uma teoria de primeira ordem clássica. Essa teoria são todos os dados disponíveis naquele estado do problema[2].

Para ilustrar a expressividade dessa linguagem, apresentamos alguns exemplos de fórmulas, baseados em Bonner (1997a), e seu significado em português. Os dois primeiros exemplos utilizam apenas conectivos clássicos e não apresentam nenhuma novidade.

- Um dia é uma segunda, uma terça, uma quarta... ou um domingo:

$$\forall X[dia(X) \leftrightarrow seg(X) \vee ter(X) \vee qua(X) \vee qui(X) \vee sex(X) \vee sab(X) \vee dom(X)] \quad (2.1)$$

- Um dia não pode ser ao mesmo tempo segunda e terça:

$$\forall X[dia(X) \rightarrow \neg(seg(X) \wedge ter(X))] \quad (2.2)$$

Os dois próximos exemplos usam a conjunção serial para fazer asserções sobre a ordem e estrutura de fenômenos temporais.

- Uma semana é uma sequência de sete dias consecutivos começando no domingo, indo até sábado e passando por segunda, terça, quarta, quinta e sexta:

$$\forall X[semana(X) \leftrightarrow (dom(X) \otimes seg(X) \otimes ter(X) \otimes qua(X) \otimes qui(X) \otimes sex(X) \otimes sab(X))] \quad (2.3)$$

- Segundas e quartas não são consecutivas:

$$\forall X[\neg(seg(X) \otimes qua(X))] \quad (2.4)$$

Os exemplos acima ilustram uma leitura declarativa de $\mathcal{TR}$. Na leitura declarativa, as sentenças representam proposições e são utilizadas para fazer asserções sobre os objetos do universo do discurso em um mundo em mudança. No fragmento de Horn da lógica das transações, que é um fragmento que serve como linguagem de programação, as fórmulas podem ser interpretadas de forma imperativa. Por exemplo, a sentença atômica $saque(quantia_a, conta_b)$ pode ser lida como uma declaração de que determinada quantia é sacada de determinada conta ou como um comando para sacar uma determinada quantia de dinheiro ($quantia_a$) de uma determinada conta bancária ($conta_b$). Em ambas as leituras, as sentenças são satisfeitas em sequencias de estados que representam o saque. No estado inicia seria dado um valor para o saldo da $conta_b$ e no estado final após o saque, seria um dado que o valor do saldo diminuiu a $quantia_a$. Um aspecto pragmático importante na satisfação

[2] Uma justificativa para considerar cada identificador de estado como um conjunto de tipos abstratos e pressupor um procedimento que atribui a cada identificador um conjunto de fórmulas de primeira ordem, ao invés de considerar cada identificador de estado como um conjunto de fórmulas de primeira ordem pode ser encontrada em (BONNER; KIFER, 1995).

de uma sentença imperativa é que os estados que determinam se a sentença será satisfeita ainda não foram realizados e dependem da execução das ações futuras para determinar se ela será satisfeita ou não. Esse não é o casos com as sentenças declarativas, pelo menos na medida em que não forem feitas declarações sobre o sobre o futuro. Por exemplo, a verdade de uma sentença como "João é educado." depende da série de comportamentos que joão apresentou até momento, mas a satisfação[3] de uma ordem "João, seja educado!", dependerá do que João vai fazer posteriormente ao proferimento da ordem. A importância da leitura imperativa na lógica das transações está no fato de que há uma forma de encontrar uma sequência de estados a partir da qual a tarefa é executada, quando, a princípio, só se conhece o estado inicial a partir do qual a ordem deve ser executada e as definições das ações disponíveis e a tarefa a ser realizada. Se a ação expressa pela ordem for não-determinística, haverá diferentes caminhos que realizam essa tarefa, mas todos eles podem ser extraídos a partir do sistema de prova.

Como apontado por Rezk[4] mesmo os conectivos que tem uma interpretação similar à clássica podem ser interpretados em termos imperativos. Por exemplo, a fórmula $\varphi \vee \psi$ significa intuitivamente em termos procedurais "faça φ ou faça ψ" e a fórmula $\neg\phi$ significa "não faça ϕ". Enquanto a conjunção serial fornece uma forma de sequenciar ações, a conjunção "clássica" $\wedge$ restringe o não determinismo de uma ação, forçando a satisfazer alguma condição. Por exemplo, $\varphi \wedge \psi$, significa intuitivamente em termos procedurais "faça φ de tal forma a fazer ψ também". Uma fórmula hipotética, $\diamond\phi$, representa uma ação na qual ϕ é testada hipoteticamente para verificar se ela pode ser executada a partir do estado atual, mas sem que a ação seja realizada, em termos procedurais ela significa "faça ϕ hipoteticamente" e será satisfeita se e somente se for possível executar ϕ a partir do estado atual. As implicações, como em $\phi \leftarrow \psi$[5] podem ser entendidas como declarações que estabelecem ϕ como o nome de uma ação complexa e ψ um modo de realizá-la. Intuitivamente, essa fórmula significa "é suficiente fazer ψ para fazer ϕ". As implicações "clássicas" permitem introduzir sub-rotinas na linguagem de programação lógica.

2.1 Semântica

Vamos apresentar aquelas ideias que acreditamos estar na origem do desenvolvimento da lógica das transações e que nos parecem ser as mais fundamentais. Elas são seis: a existência de estados dos quais a valoração das sentenças dependem; em cada estado exite um conjunto de dados que restringe as estruturas que poderemos atribuir quando formos estabelecer uma interpretação; a lógica das transações visa certa independência

[3] Preferimos usar o termo "satisfação" ao invés de "verdade", pois o conceito de verdade, não se aplica a sentenças imperativas. Mas podemos dizer que uma sentença imperativa é satisfeita.

[4] (REZK; KIFER, 2011).

[5] Que também pode ser escrita como $\psi \rightarrow \phi$, pois ambas as fórmulas são equivalentes a $\neg\psi \vee \phi$. A grafia para a esquerda é comumente utilizada na linguagem de programação ProLog.

do sistema de representação escolhido para representar os dados disponíveis em cada estado; a lógica das transações também visa certa independência em relação ao conjunto de atualizações elementares escolhidas; a existência de um conjunto definindo as ações e conceitos que restringem a forma de atualizar os estados e o que pode ser verdade neles; e a distinção entre dados e fatos derivados.

Assim como na lógica dinâmica, a lógica das transações começa com uma semântica baseada na ideia de estados, na qual as ações podem provocar a transição de um estado a outro. Entretanto, em $\mathcal{TR}$ uma sentença não é valorada em um estado, mas em sequências de estados. Além disso, na lógica das transações, proposições e ações podem ser representadas por símbolos de predicado, enquanto na lógica dinâmica, existe uma distinção sintática obrigatória. Outra diferença entre essas lógicas e que aos identificadores de estado na lógica dinâmica $(w_1, w_2, w_3, \cdots)$, podem ser vistos como estruturas de primeira ordem arbitrárias, já os identificadores de estados (como por exemplo, conjuntos de fórmulas de primeira ordem) em $\mathcal{TR}$ representam os dados que temos do problema, e esses dados limitam as estruturas de primeira ordem que podemos atribuir às sequências de comprimento 1 contendo esses estados.

Quando uma ação é executada, a situação pode mudar de um estado inicial, D_1, para um estado final, D_n. A sequência de estados que uma ação provoca é representada por meio de uma sequência discreta de estados $\langle D_1, \cdots, D_n \rangle$ com comprimento n, onde $n \geq 1$, chamada de caminho de execução da transação. A princípio, as fórmulas que são satisfeitas em um caminho podem representar tanto ações que seriam capazes causar transição dos estados em questão, quanto proposições verdadeiras naquela sequência de estados.

Por meio da expressão "independência do sistema de representação" queremos dizer que a lógica das transações permite que estes dados sejam representados por diversos sistemas simbólicos ao pressupor que existe um procedimento para interpretar o sistema de representação escolhido em termos de uma teoria de primeira ordem. Isso torna possível interpretar os dados de tabelas, grafos, arranjos e teorias lógicas com diferentes semânticas, desde clássicas até não-monotônicas. Para atingir tal generalidade, a lógica das transações trata o procedimento que realiza essa conversão como um parâmetro.

Um problema análogo ocorre com as atualizações elementares, assim como não há uma único sistema simbólico para representar os dados em cada estágio do problema, não há um pequeno conjunto de atualizações elementares que seja ideal para definir todas ações mais complexas. Dessa forma, ao invés de definir as atualizações elementares diretamente na semântica da lógica da transações, ou autores Bonner e kifer preferiram as atualizações elementares como parâmetros.

Se, por exemplo, cada identificador de estado for uma teoria relacional clássica (um conjunto de sentenças atômicas) uma atualização elementar pode mudar a teoria inserindo ou deletando sentenças atômicas, o que corresponde a alteração da extensão de uma propriedade ou relação. Já se os identificadores de estado forem teorias axiomatizada clássica (um conjunto finito de sentenças), uma atualização pode inserir ou deletar fórmulas complexas resolvendo possíveis conflitos entre a fórmula nova e as antigas. Mas, existem diversas maneiras de realizar tais operações, e diferentes formas correspondem a diferentes ideias por detrás dessas operações. Katsuno e Mendelzon[6] afirmam que há duas grandes categorias as quais as transições das teorias pertencem (as atualizações$_1$ e as revisões)[7]. A ideia por trás das atualização$_1$ é que os estado de coisas mudou e nossa teoria-variável deve ser atualizada$_1$ para acompanhar o novo estado de coisas. A ideia por trás das revisões é que apesar de o estado de coisas não ter sido alterado, novas informações sobre este estado foram adquiridas e nossa teoria precisa ser revista para acomodar as informações antigas e novas. Desta forma, que revisar e atualizar$_1$ uma teoria com uma fórmula são ações elementares diferentes.

É possível distinguir os dados em cada estado do problema e os fatos que um solucionador pode derivar a partir do conhecimento que possui. A“visão” que um agente dotado de conhecimento possui em um estado do problema é a união dos dados com os fatos que ele pode derivar desses dados. Isto é, a “visão” que um agente possui é representada pela união dos dados determinados pelo identificador de estados com o conjunto de fórmulas que representam o conhecimento de um agente.

Muitas vezes, nós não queremos um estado de coisas seja atualizado arbitrariamente porque as atualizações devem ser feitas de acordo com as capacidades e limitações do agente que queremos modelar. Neste caso, estaremos interessados nos modelos das fórmulas que representam essas capacidades.

Operações elementares

Vamos tratar agora de como as ideias de independência da escolha do sistema de representação e da escolha das atualizações elemetares foram concretizadas na lógica das transações. Em $\mathcal{TR}$, os símbolos de predicado podem indicar ações e as constantes indicam objetos com os quais essas ações podem ser realizadas. Como as ações complexas são definidas a partir de um conjunto de ações elementares por meio dos operadores lógicos, precisamos mostrar como as ações elementares são estabelecidas em $\mathcal{TR}$. Há dois tipos de ações elementares: os métodos de obtenção de informação primitivos que interpretam os dados do problema e as atualizações primitivas. Uma forma de definir essas operações elementares, seria construí-las na própria semântica de $\mathcal{TR}$. Entretanto, um problema apontado por Bonner e Kifer[8] com essa abordagem é que ela comprometeria a lógica

6 (KATSUNO; MENDELZON, 2003).

7 O termo “atualização” é utilizado em dois sentidos: para falar de qualquer mudanças nas teorias e estados e para falar de uma categoria particular de atualização. Estamos utilizando o termo “atualização” para falar de modo geral e “atualização$_1$” para uma categoria específica.

8 (BONNER; KIFER, 1995).

com uma teoria particular de atualizações elementares e com um sistema particular de representação dos estados. Um dos objetivos daqueles autores era apresentar uma lógica para combinar as ações primitivas em ações complexas independentemente da escolha das ações primitivas. A semântica da lógica das transações é tal que permite acomodar diversos conjuntos elementares de atualizações e diversos sistemas de representação para os dados que ela atualiza. Para atingir tal generalidade, a lógica das transações assume que as operações elementares são especificadas por meio de dois parâmetros, chamados de oráculo de dados e oráculo de transição. Esses oráculos são procedimentos dos quais não precisamos, a princípio, saber como eles funcionam. Os oráculos atuam como "caixas preta" permitindo que diferentes ações elementares sejam "plugados" na teoria de $\mathcal{TR}$.

Os dados em cada estado são representados formalmente por meio de um identificador de estado e as mudanças de estados são representadas por meio de sequências de identificadores de estado. Em $\mathcal{TR}$ um identificador de estado é um conjunto de tipo abstrato que representam os dados disponíveis naquele estágio de resolução do problema. Dizemos que estes identificadores são conjuntos de tipo abstrato porque não há, a princípio, necessidade de especificar o sistema de representação utilizado nesses identificadores, nem como funciona o procedimento que permite interpretar esses identificadores atribuindo um significado para eles, onde "atribuir um significado" tem o sentido de atribuir um conjunto de sentenças em uma linguagem de primeira ordem. Quando falamos em "dado" estamos pensando em suposições que assumimos em determinado estado, mas não queremos aproximar essa noção de "dado" com concepções epistemológicas que ratificam a capacidade de a percepção fornecer dados brutos (algo não interpretado) ou qualquer coisa que serviria para fundamentar o conhecimento empírico. Esses dados são apenas as coisas que se assume como verdadeiras em um estado.

O oráculo de dados atribui a cada identificador de estado uma teoria de primeira ordem, contribuindo com o aspecto estático da semântica da lógica das transações e o oráculo de transição atribui a certos pares ordenados de estados uma sentença atômica, contribuindo com aspecto dinâmico da semântica. Isso porque, as estruturas atribuídas para interpretar as sequências de comprimento 1 e 2 deverão satisfazer as sentenças que esse oráculos indicarem. A utilização de parâmetros permite separar a especificação das ações elementares da lógica que as combina. Para Bonner e Kifer[9], essa separação possui dois grandes benefícios: (1) ela permite desenvolver uma lógica para combinar e programar transações sem compromisso com uma teoria particular das atualizações elementares e (2) ela permite acomodar diversos sistemas simbólicos para representar os estados.

[9] (BONNER; KIFER, 1995).

Oráculo de dados

Os dados de um problema podem ser representados de diversas formas, como por exemplo: um teoria de primeira ordem, uma tabela, gráficos etc. Deste modo, precisamos passar da representação do dado, da forma como ele está materialmente identificado para o conjunto das sentenças dadas como verdadeiras naquele estado em uma teoria de primeira ordem clássica. Essa tarefa de traduzir o sistema de representação para uma teoria de primeira ordem clássica é realizada pelo oráculo de dados. Formalmente, um oráculo de dados, $\mathcal{O}^{\mathcal{D}}$, é uma função de um conjunto de identificadores de estados em conjuntos de fórmulas de primeira ordem. Intuitivamente, se i é um identificador de estado, então $\mathcal{O}^{\mathcal{D}}(i)$ é o conjunto considerado como todas as fórmulas dadas como verdadeiras naquele estado e apenas elas. Usualmente, os autores escolhem como identificadores de estados conjuntos de fórmulas de primeira ordem, pois é fácil descrever oráculos em termos dessas fórmulas. Entretanto, também é possível utilizar outros tipos de identificadores de estado, como tabelas, por exemplo, e descrever oráculos que retornam o "conteúdo lógico" desses identificadores.

Oráculo de transição de estados

Ao invés de comprometer $\mathcal{TR}$ com um conjunto fixo de transições elementares, Bonner e Kifer preferiram tratar estas como um parâmetro. A tarefa de especificar as atualizações primitivas que serão satisfeitas em certos pares de estado é realizada pelo oráculo de transição. Formalmente, um oráculo de transição, $\mathcal{O}^{\mathcal{T}}$, é uma função de um conjunto de pares ordenados de identificadores de estados em conjuntos de sentenças atômicas. Essas sentenças atômicas podem representam a ações que podem causar a mudança do estados representada pelo par de estados.

A lógica das transações, portanto, vem parametrizada com um par de mecanismo. Cada instância concreta de oráculo define as ações elementares a sua maneira, e com qual conjunto de identificadores de estado vamos lidar. Essa abordagem que separa a especificação das operações elementares da lógica que as combina, torna $\mathcal{TR}$ muito flexível para combinar ações e para computar os efeitos de sua execução.

Oráculo relacional

Nesta dissertação vamos utilizar um tipo particular de oráculo bastante simples, chamado de oráculo relacional.

Oráculo Relacional: Um identificador de estado D é um conjunto de sentenças atômicas de uma dada linguagem $\mathcal{L}$ e o oráculo relacional simplesmente retorna essas fórmulas. Portanto, $\mathcal{O}^{\mathcal{D}}(D) = D$. Além disso, para cada símbolo de predicado p em D, o oráculo relacional define dois novos predicados $p.ins$ e $p.del$, representando a inserção e exclusão do predicado p. Formalmente: $p.ins(X_1, \cdots, X_n) \in \mathcal{O}^{\mathcal{D}}(D_1, D_2,)$ sse $D_2 = D_1 \cup \{p(X_1, \cdots, X_n)\}$. Analogamente, $p.del(X_1, \cdots, X_n) \in \mathcal{O}^{\mathcal{D}}(D_1, D_2,)$ sse $D_2 = D_1 - \{p(X_1, \cdots, X_n)\}$. Os símbolos de predicado p, $p.ins$ e $p.del$ são símbolos ordinários de uma $\mathcal{TR}$, a diferença entre eles, é que os dois últimos são citados pelo oráculos de transição.

Estrutura de caminhos

Estruturas de caminhos é o nome dado às estruturas para interpretar uma $\mathcal{TR}$-linguagem. Uma linguagem $\mathcal{L}$ para a lógica das transações é composta por um conjunto de fórmulas de transação e por um conjunto de identificadores de estado. O domínio do oráculo de dados determina o conjunto de identificadores de estados, cujas sequências vamos interpretar. O conjunto de caminhos em uma linguagem $\mathcal{L}$ será denotado por $Paths(L)$.

Essa estrutura possui um conjunto de objetos, o universo do discurso, uma função que interpreta os símbolos funcionais e uma função de interpretação dos caminhos. O universo do discurso é o mesmo em todos os estados e a interpretação dos símbolos funcionais é "rígida". Para realizar a interpretação de um caminho não basta mostrar quais sentenças são satisfeitas nesses caminho, pois essas sentenças também devem possuir uma interpretação. Por isso, a função de interpretação atribui uma estrutura semântica de primeira ordem "clássica" para cada caminho π em $Paths(L)$. As estruturas clássicas especificam quais fórmulas atômicas são satisfeitas em cada caminho π e como interpretá-las. Apesar de não estarmos lidando com atualizações de teorias paraconsistentes, é preciso ter algum mecanismo para impedir que o sistema lógico se torne inútil quando contradições aparecem. Para lidar com possíveis contradições uma estrutura especial denotada por $\top$ é adicionada a classe das estruturas clássicas e por isso utilizamos as aspas no parágrafo acima. Essa estrutura possui a propriedade de satisfazer todas as fórmulas de primeira ordem e é utilizada para permitir que a interpretação dos caminhos seja uma função total atribuindo um significado mesmo no casos de a estrutura atribuída precisar satisfazer uma teoria inconsistente. Nós não vamos nos aprofundar sobre este tópico, porque vamos focar em casos nos quais as contradições não aparecem. A definição formal das estrutura de caminhos assenta na definição das estruturas semânticas "clássicas". Como veremos, a verdade na estruturas de primeira ordem "clássicas" desempenha na interpretação de $\mathcal{L}$ um papel similar ao papel que o pertencimento a um conjunto desempenha na interpretação de uma linguagem de primeira ordem clássica. A satisfação de uma sentença atômica p em um caminho depende de ser verdadeira na estrutura "clássica" atribuída ao caminho.

Definição (Estruturas de Caminho): Seja $\mathcal{L}$ uma linguagem para $\mathcal{TR}$, com um conjunto $\mathcal{F}$ de símbolos funcionais. Uma estrutura de caminhos, M, para interpretar

$\mathcal{L}$, é uma tripla $\langle \mathbf{U}, I_{\mathcal{F}}, I_{path} \rangle$, onde:

- $\mathbf{U}$ é um conjunto de objetos, o universo do discurso.
- $I_{\mathcal{F}}$ é uma interpretação dos símbolos funcionais em $\mathcal{L}$. Ela atribui para cada símbolo funcional em $\mathcal{F}$ um função $\mathbf{U}^n \longmapsto \mathbf{U}$

Dados U e $I_{\mathcal{F}}$, seja $Estruturas(\mathrm{U}, I_{\mathcal{F}})$ o nome que denota o conjunto de todas as estruturas semânticas para $\mathcal{L}$ na forma $\langle \mathrm{U}, I_{\mathcal{F}}, I_p \rangle$ e da estrutura $\top$, onde I_p é uma função que interpreta os símbolos de predicado em $\mathcal{P}$ como relações em $\mathbf{U}$ de aridade apropriada.

- I_{path} é um mapa que atribui para cada caminho em $Paths(L)$, uma estrutura em $Estruturas(\mathrm{U}, I_{\mathcal{F}})$. Lembrando que $Estruturas(\mathrm{U}, I_{\mathcal{F}})$ contém a estrutura $\top$ e que $\models^{``c"}$ denota a relação de satisfação clássica ou a estrutura $\top$ satisfazendo todas as fórmulas clássicas. Com as seguintes restrições:
  - Cumplicidade com o oráculo de dados: $I_{path}(\langle D \rangle) \models^{``c"} \phi$ para todo $\phi \in \mathcal{O}^{\mathcal{D}}(D)$, o que significa que uma "visão" do estado deve ser um modelo de todos os dados, ou melhor, de tudo que o oráculo de dados considerar verdadeiro.
  - Cumplicidade com o oráculo de transições: $I_{path}(\langle D_1,\ D_2 \rangle) \models^{``c"} p$ sempre que $p \in \mathcal{O}^{\mathcal{T}}(D_1, D_2)$.

A função I_{path} especifica quais atômicas são satisfeitas em cada caminho e como interpretá-las. Intuitivamente, essas fórmulas denotam ações ou fatos que ocorrem nesses caminhos.

Satisfação em caminhos

Antes de definir as condições de satisfação das fórmulas em um caminho, é conveniente definir o conceito de corte em um caminho.

Definição (Corte em um caminho): Seja um caminho $\pi = \langle D_1, \cdots, D_n \rangle$, um corte em π é qualquer par de subcaminhos, π_1 e π_2, tal que, $\pi_1 = \langle D_1, \cdots, D_i \rangle$ e $\pi_2 = \langle D_i, \cdots, D_n \rangle$ para algum $i (1 \leqslant i \leqslant n)$. Neste caso, escrevemos $\pi = \pi_1 \circ \pi_2$. Em outras palavras, $\pi = \pi_1 \circ \pi_2$ sse π_1 é um prefixo de π, π_2 é um sufixo de π e o último estado do subcaminho π_1 é o primeiro estado do subcaminho π_2.

Definição (Satisfação em Caminhos): Seja $\mathrm{M} = \langle \mathbf{U}, I_{\mathcal{F}}, I_{path} \rangle$ uma estrutura de caminhos, seja π um caminho arbitrário, e v uma função que atribui para cada variável um elemento no domínio. Então:

1. Caso base:
   $\mathrm{M},\pi \models_v p(t_1, \cdots, t_n)$ sse $I_{path}(\pi) \models_v^{``c"} p(t_1, \cdots, t_n)$, para toda atômica $p(t_1, \cdots, t_n)$.

2. Negação:
   M,$\pi \models_v \neg\phi$ sse não é o casos que M,$\pi \models_v \phi$.

3. Conjunção clássica:
   M,$\pi \models_v \phi \wedge \psi$ sse M,$\pi \models_v \phi$ e M,$\pi \models_v \psi$.

4. Disjunção clássica:
   M,$\pi \models_v \phi \vee \psi$ sse M,$\pi \models_v \phi$ ou M,$\pi \models_v \psi$.

5. Conjunção serial:
   M,$\pi \models_v \phi \otimes \psi$ sse M,$\pi_1 \models_v \phi$ e M,$\pi_2 \models_v \psi$ para alguma cisão $\pi_1 \circ \pi_2 = \pi$.

6. Quantificador Universal:
   M,$\pi \models_v \forall(X)\phi$ sse M,$\pi \models_u \phi$ para toda atribuição de variável u que concorda com v exceto em x.

7. Quantificador Existencial:
   M,$\pi \models_v \exists(X)\phi$ sse M,$\pi \models_u \phi$ para alguma atribuição de variável u que concorda com v em todas as atribuições exceto em x.

8. Operador hipotético de possibilidade:
   M,$\langle D\rangle \models_v \diamond\phi$ sse existe um caminho, π, começando no estado D, tal que M,$\pi \models_v \phi$ é o caso.

Se M,$\pi \models_v \phi$, então nós dizemos que ϕ é satisfeita no caminho π da estrutura M. Alguns pontos sobre a definição acima merecem ser considerados:

- O caso base tenta capturar a intuição por detrás da execução de uma ação ao longo de um caminho. Ou da ocorrência de um fato durante o caminho. A satisfação de uma atômica $p(t_1, \cdots, t_n)$ no caminho π significa que a ação p pode ser executada ao longo de π quando aplicada aos objetos $(t_1, \cdots, t_n)$ ou que uma propriedade\relação se aplica aos objetos $(t_1, \cdots, t_n)$ durante o evento representado pela sequência de estados.

- A semântica dos conectivos "clássicos" é definida de forma familiar. Para esses conectivos, a satisfação de uma fórmula depende apenas da estrutura atribuída ao caminho.

- A satisfação de uma fórmula com conectivo $\otimes$ em um caminho não depende apenas deste caminho, mas também de seus sub-caminhos, isto é, das estruturas atribuídas a estes sub-caminhos. Intuitivamente, a semântica de $\otimes$ diz que uma transação serial é executada se e somente se todas suas sub-transações são executadas. Deste modo, ela aglutina duas ações em uma única transação.

- A conjunção $\phi \wedge \psi$ não especifica a execução paralela de duas transações ϕ e ψ. Ao invés disso, ela representa uma única transação cujo modo de execução é mais restrito que as ações conceituais que a define. Intuitivamente, $\phi \wedge \psi$ significa "faça ϕ de tal forma que ψ seja feita".

- Para caminhos de comprimento 1, os conectivos $\otimes$ e $\wedge$ são equivalentes. Disso se segue que em caminhos de comprimento 1, toda ocorrência de $\otimes$ em uma fórmula de transação pode ser trocada por $\wedge$ formando uma fórmula equivalente. Em caminhos com comprimentos maiores que 1, esses conectivos estendem a semântica da conjunção de modos diferentes.

- Intuitivamente, a satisfação de $\diamond\phi$ significa que é possível começar a execução de ϕ a partir do estado atual. Formalmente, isso significa que ϕ é satisfeita ao longo de algum caminho partindo do estado D. Fórmulas com este operador são satisfeitas imediatamente, isto é, apenas em caminhos de comprimento 1.

- Outros operadores como a implicação "clássica" ($\rightarrow$), a bi-implicação "clássica" ($\leftrightarrow$), a implicação serial para a esquerda ($\Leftarrow$) e a implicação serial para a direita ($\Rightarrow$) podem ser definidos. A implicação serial para esquerda, $\psi \Leftarrow \phi$, é definida como $\psi \oplus \neg\phi$ e significa intuitivamente que "a transação ϕ deve ser imediatamente precedida da transação ψ. Já a implicação serial para a direita, $\psi \Rightarrow \phi$ é definida como $\neg\psi \oplus \phi$ e significa intuitivamente que "a transação ϕ deve ser imediatamente seguida da transação ψ". Além disso, podemos definir uma constante proposicional que é satisfeita em todos os caminhos de comprimento 1 e apenas neles em qualquer estrutura de caminhos. Essa constante é chamada **state** e pode ser definida a partir do operador hipotético da seguinte forma:

$$\textbf{state} \equiv_{Def} \diamond(\phi \vee \neg\phi)$$

Lema: Se ϕ é uma fórmula de primeira ordem, então M,$\pi \models \phi$ sse $I_{path}(\pi) \models^{``c''} \phi$ para toda estrutura de caminhos M e todo caminho $\pi$, onde $\models^{``c''}$ denota a satisfação clássica, ou a satisfação na estrutura $\top$. *Prova:* por indução na complexidade de ϕ.

Lema: se ϕ e ψ são duas fórmulas de transação, então M,$\langle D\rangle \models \phi \otimes \psi$ sse M,$\langle D\rangle \models \phi \wedge \psi$, para toda estrutura M e todo estado D. *Prova:* Segue-se do fato que $\langle D\rangle \circ \langle D\rangle$ é o único corte no caminho $\langle D\rangle$.

Modelos

Uma estrutura de caminhos atribui uma estrutura semântica de primeira ordem (tradicional ou $\top$), $I_{path}(\pi)$, para cada caminho π, especificando as ações que podem ser cometidas ao longo deste caminho ou fatos que vigoram nesses caminhos. Em estruturas arbitrárias, as estruturas atribuídas ao caminho π é independente de seus sub-caminhos. Isso significa que não temos nenhum conhecimento sobre a forma como as ações se articulam. A ideia de Bonner e Kifer é que quando houver conhecimento (procedural ou declarativo), tal conhecimento é representado por um conjunto de fórmulas de transação e estaremos interessados apenas em seus modelos, não em estruturas quaisquer. Nós acreditamos que esta ideia está na direção correta, mas divergimos dos autores porque acreditamos que é necessário aumentar a expressividade da estrutura atribuídas às sequências de estados usando alguma lógica modal, de modo a permitir que sentenças como "i tem a intenção de fazer A" possam ser formuladas de forma apropriada.

Definição (Modelos de fórmulas de transação): Uma estrutura de caminhos, M, é um modelo de uma fórmula de transação ϕ, se e somente se $\mathrm{M}, \pi \models \phi$ para todo caminho π, neste caso, escrevemos $\mathrm{M} \models \phi$. Uma estrutura de caminhos é modelo de um conjunto de fórmulas se e somente se é modelo de todas as fórmulas pertencentes ao conjunto.

Nos modelos, a estrutura semântica atribuída pela função I_{path} não é arbitrária, pois são criadas dependências entre a estrutura atribuída a um caminho e seus sub-caminhos. Por exemplo, seja $\langle \mathbf{U}, I_{\mathcal{F}}, I_{path} \rangle$ um modelo para $p \leftarrow (q \otimes r)$, e π qualquer caminho. Se $\pi_1 \circ \pi_2$ é um corte em π, então I_{path} exibe uma dependência entre 3 estruturas semânticas diferentes atribuídas a 3 caminhos diferentes:

$$\text{Se } I_{path}(\pi_1) \models^{\text{``}c''} q \text{ e } I_{path}(\pi_2) \models^{\text{``}c''} r, \text{ então } I_{path}(\pi) \models^{\text{``}c''} p$$

Intuitivamente, isto significa que p é um nome para o procedimento $q \otimes r$.

Relação de Consequência (Execucional)

Uma estrutura de caminhos, pode atribuir às sequências de estados estruturas "clássicas" que satisfazem fórmulas representando propriedades e ações que não tem qualquer envolvimento com os conceitos que pretendemos modelar. Por exemplo, um caminho pode representar a mudança da posição de um bloco, mas nada impede que uma interpretação atribua a este caminho uma estrutura na qual uma fórmula indicando, por exemplo, a cor de um outro bloco qualquer, também seja satisfeita. Para filtrar essas atribuições e prover uma tratamento lógico para a noção de execução das transações Bonner e Kifer desenvolveram a relação de consequência execucional. Como dito anteriormente, nós não estamos interessados em estruturas arbitrárias, mas naquelas que são modelo das fórmulas que visam representar o conhecimento dos agentes. Quando um conjunto de fórmulas de transação P, chamada de base de transação, é utilizado sobretudo para representar o conhecimento de um agente, vamos chamá-lo de base de conhecimento.

Definição (Consequência execucional): Seja P um conjunto de fórmulas, ϕ

uma fórmula e $D_0, D_1, \cdots D_n$ identificadores de estados. Então, a declaração

$$P,\ D_0,\ D_1,\ \cdots,\ D_n\ \models\ \phi \tag{2.5}$$

é verdadeira, se e somente se M,$\langle D_0, D_1, \cdots, D_n\rangle \models \phi$ para todo modelo M de P.

Intuitivamente, essa declaração diz que a execução bem sucedida uma transação ϕ pode realizar a mudança do estado D_0 para D_1, $\cdots$, para D_n. Formalmente, ela diz que em todos os modelos de P existe um caminho $\langle D_0, D_1, \cdots, D_n\rangle$ que satisfaz a fórmula ϕ. Essa declaração pode ser lida assim, "sob a base de conhecimento P, a transação ϕ pode realizar a mudança do estado D_0 para o estado D_n, passando pelos estados intermediários $D_1, \cdots, D_n - 1$". Note que nessa relação a consequência ϕ específica a tarefa que deve ser realizada, a base de transação apresenta as ações enquanto conceitos e as diferentes sequencias de identificadores especificam uma ocorrência da ação. Mesmo nos casos em que o caminho tem comprimento 1 podemos pensar em ϕ como representando uma ação ao invés de uma proposição, mas apenas em sentido amplo de ação, que incluiria métodos de obtenção de informação.

Relacionada com a declaração (2.5), está as seguinte declaração:

$$P,\ D_0 - -- \ \models\ \phi \tag{2.6}$$

que é verdadeira apenas se existe um caminho começando em D_0 para o qual a declaração (2.5) é verdadeira. Usualmente, nós conhecemos apenas o estado atual de tal forma que definir a execução de uma transação por meio de (2.5) nem é sempre apropriado. A versão de consequência apresentada em (2.6) permite omitir os estados intermediários e finais. Informalmente, (2.6) diz que a transação ϕ pode ser executada a partir do estado D_0. Formalmente, a declaração acima é verdadeira se e somente se existe um caminho começando com D_0 em todo modelo M de P que satisfaz ϕ. Quando (2.6) não é verdadeira, podemos dizer que ϕ falha, pois não pode ser executada a partir de D_0.

A seguir apresentamos um sistema de prova que permite demonstrar enunciados da forma:

$$P,\ D_0,\ D_1,\ \cdots,\ D_n\ \vdash\ \phi$$

quando conhecemos, a princípio, apenas o estado inicial D_0. Este sistema de prova é correto e completo com a relação de consequência execucional (2.5) para um fragmento que apresentaremos a seguir e permite computar os caminhos que uma execução de ϕ poderia gerar. Deste modo, este sistema de prova não testa se um caminhinho dado de antemão pode corresponder a execução de uma ação, mas computa os caminhos que uma ação ϕ tal como definida pelos oráculos e pela base de transação P podem provocar, ou seja, intuitivamente, a construção de teoremas corresponde a execução das ações.

Fragmento de Horn e Programação Lógica

As tentativas de gerar um sistema de prova simples que fosse correto e completo e que pudesse ser aplicado a todas as fórmulas de transação foram frustradas. Entretanto, $\mathcal{TR}$ possui um fragmento bastante relevante para o qual Bonner e Kifer desenvolveram um sistema de prova correto e completo. Este fragmento estudado exaustivamente por Bonner e Kifer é chamado de fragmento de Horn e pode ser utilizado como uma linguagem de programação. Vamos definir nas próximas o que é uma fórmula de Horn serial-$\diamond$, o que é a condição de Horn serial-$\diamond$ e o que é um programa de Horn serial-$\diamond$. Com essas definições em mente vamos mostrar no próximo capítulo como o conhecimento é representado na lógica das transações e como ela representa dispositivos que utilizam esse conhecimento para realizar tarefas triviais e resolver problemas.

Definição (objetivo serial-$\diamond$): Estas fórmulas são utilizadas para representar as tarefas que podem ser requisitadas.

- Uma fórmula atômica é um objetivo serial-$\diamond$.
- Se $\phi_1, \ldots, \phi_n$ são objetivos seriais-$\diamond$, então $\phi_1 \otimes \ldots \otimes \phi_n$ são objetivos seriais-$\diamond$.
- Se ϕ é um objetivo serial-$\diamond$, então $\diamond\phi$ também é um objetivo seriais-$\diamond$.

Definição (Fórmulas de Horn serial-$\diamond$):

- Um objetivo serial-$\diamond$ é uma fórmulas de Horn serial-$\diamond$.
- Um objetivo serial-$\diamond$ existencial é uma fórmula da forma $\exists X_1, \ldots \exists X_n \phi$, na qual $x_1, \ldots X_n$ é uma lista com todas as variáveis em ϕ e ϕ um objetivo serial-$\diamond$.
- Uma regra serial-$\diamond$ de Horn é uma fórmula de transação da forma $\forall X_1, \ldots \forall X_n (a \leftarrow \phi)$, na qual b é uma fórmula atômica e ϕ é um objetivo serial-$\diamond$. Em uma regra todas todas as ocorrências de variáveis estão quantificadas universalmente.
- Uma base de transação serial-$\diamond$ de Horn é um conjunto finito de regras seriais-$\diamond$ de Horn.

As regras seriais-$\diamond$ de Horn podem ser utilizadas para definir ações propriamente ditas, acontecimentos e conceitos. Uma base de transação serial-$\diamond$ de Horn pode ser utilizada para representar as ações enquanto categorias enquanto os caminhos representam ocorrências individuais dessas ações.

Definição (Condição de Horn serial relacional): Uma base de transações, P, um oráculo de dados, $\mathcal{O}^{\mathcal{D}}$, e uma fórmula de transação, ϕ, satisfazem a condição serial de Horn restrita sse:

1. P é uma conjunto de regras serialis-$\diamond$ de Horn.
2. $\mathcal{O}^{\mathcal{D}}$ é um oráculo relacional.
3. ϕ é um objetivo serial-$\diamond$ existencial.

Note que ao estabelecer uma condição que requer a utilização do oráculo relacional provocamos uma grande limitação na expressividade dos dados que estão disponíveis em cada estado. Bonner e Kifer apontam que se estivéssemos interessados apenas nos casos nos quais as condições de Horn serial são satisfeitas, poderíamos utilizar uma semântica mais simples que aquela apresentada anteriormente, na qual consideraríamos apenas a satisfação em arcos (pares de estados), ao invés de em caminhos (sequências de estados). Uma semântica baseada em arcos, entretanto, teria limitações porque não permitira expressar restrições que devem ser satisfeitas durante a execução das transações por meio da conjunção "clássica".

2.2 O sistema de inferência

Vamos apresentar o sistema de inferência desenvolvido por Bonner e Kifer[10] que se aplica aos casos de Horn serial relacional. Esse sistema, que possui um axiomas e 4 regras de inferência para manipular sequentes da forma $P,\ D_0 - -- \ \vdash\ \phi$, é correto e completo com relação aos fragmentos que assumem as condições apontadas acima. A primeira representa a aplicação das definições na base de conhecimento; a segunda representa a realização de métodos de obtenção de informação elementares, isto é, o acesso aos dados; o terceira representa a realização das transições elementares; a quarta representa realizações hipotéticas das transações, isto é que verificam se é possível executar uma ação, mas sem realizá-las. O axioma representa uma ação que sempre é possível de ser realizada a partir de qualquer estado.

Esse sistema de prova permite verificar se declarações das formas $P,\ D_0 - -- \ \vDash\ \phi$ são verdadeiras e que declarações da forma $P,\ D_0,\ D_1,\ \cdots,\ D_n\ \vDash\ \phi$ são verdadeiras, mesmo quando só conhecemos, a principio, P, o estado inicial D_0 e a tarefa proposta[11]. Informalmente, a primeira declaração diz que a transação ϕ *pode* ser executada com sucesso a partir do estado D_0, pois existe um caminho pelo qual ela pode ser realizada. A segunda, que diz que a transação ϕ *pode* ser executada pelo caminho $D_0,\ D_1,\ \cdots,\ D_n$. É possível extrair das provas geradas por este sistema de inferência um caminho especifico caso a tarefa possa ser realizada, neste sentido, as tarefas são realizadas, geramos o caminho de

10 (BONNER; KIFER, 1995, p. 136-137)

11 É possível ampliar os sistema proposto como novas regras para lidar também com as situações nas quais os estado conhecido (atual) é o último estado do caminho D_n. Neste caso, poderíamos capturar a noção de execução reversa de uma ação. Mas, este sistema é suficiente para os fins aqui propostos.

execução de uma transação, na medida em que construímos provas de teoremas usando o sistema.

No parágrafo acima, nós utilizamos a expressão "*pode*" porque ϕ pode ser uma ação não determinística, e que, portanto, também poderiam ocorrer mudanças especificadas por caminhos diferentes daquele. O sistema de prova permite encontrar todos os caminhos de execução de uma ação. Como estamos lidando com o caso de serial de Horn relacional, a transação ϕ é um objetivo serial-$\diamond$ existencial, isto é, uma sentença da forma $\exists X_1, \ldots \exists X_n \phi$, na qual cada ϕ é um um objetivo serial-$\diamond$. Para não poluir a notação não vamos escrever a série de quantificadores, mas manteremos um $(\exists)$ para lembrar que se trata de um objetivo serial-$\diamond$ existencial. A teoria de $\mathcal{TR}$ recorre aos oráculo de transição e de dados, que fornecem a semântica dos estados e das transições elementares. O sistema $\mathcal{I}^{\diamond}$ relacional faz uso deles oráculos para acessar e atualizar o estados, que no nosso casos será o oráculo relacional.

Definição (Sistema de inferência $\mathcal{I}\diamond$ relacional): se P é uma base de conhecimento, então $\mathcal{I}_{\diamond}$ é o seguinte sistema de axiomas e regras de inferência, onde D e D_i são quais identificadores de estado:

Axiomas:

$$P,\ D_0 ---\ \vdash\ () \tag{2.7}$$

Regras de inferência: nas regras a seguir a e b são fórmulas atômicas, e ϕ e são metas seriais.

Aplicando a definição das transações: Suponha que $a \leftarrow \phi$ é uma regra em P cujas variáveis foram renomeadas de modo que a regra não tenha variáveis compartilhadas com $b \otimes rest$. Se a e b unificam com mgu σ, então:

$$\frac{P, D --- \vdash (\exists)(\phi \otimes rest)\sigma}{P, D --- \vdash (\exists) b \otimes rest} \tag{2.8}$$

Consultando os dados dos estados: se $b\sigma$ e $rest\ \sigma$ não compartilham variáveis, e $\mathcal{O}^{\mathcal{D}}(D) \models^c (\exists) b\sigma$, então:

$$\frac{P, D --- \vdash (\exists)\ rest\ \sigma}{P, D --- \vdash (\exists) b \otimes rest} \tag{2.9}$$

Realizando transições elementares: se $b\ \sigma$ e $rest\ \sigma$ não compartilham variáveis, então:

$$\frac{P, D_2 --- \vdash (\exists) rest\ \sigma}{P, D_1 --- \vdash (\exists)\ b \otimes rest} \quad se\ \mathcal{O}^{\mathcal{T}}(D_1, D_2) \models^c (\exists)\ b\sigma \tag{2.10}$$

Introduzindo modalidades: se $test\ \sigma$ e $rest\ \sigma$ não compartilham variáveis, então:

$$\frac{P, D --- \vdash (\exists) test\ \sigma \quad P, D --- \vdash (\exists) rest\ \sigma}{P, D --- \vdash (\exists)(\diamond test) \otimes rest} \tag{2.11}$$

Em $\mathcal{I}_\diamond$ relacional os axiomas descrevem uma ação "vazia", que por definição não faz nada, isto é, sua execução não muda o estado atual, e sempre pode ser executada. O símbolo "( )" é uma abreviação para $\diamond(\phi \vee \neg\phi)$. Cada regra de inferência consiste de dois sequentes, um acima do outro, e significa que se o sequente de cima pode ser inferido, então o sequente de baixo também pode ser inferido. Utilizamos a expressão *rest* para representar a sub-fórmula que não está sendo focada no momento.

Podemos entender essas regras da seguinte forma:

1. A primeira regra de inferência permite aplicar as definição das transações. No primeiro caso, a regra diz, grosso modo, que se b é definida por ϕ, e se $\phi \otimes rest$ se sucede de D, então $b \otimes rest$ também se sucede de D. O uso

2. A segunda regra de inferência lida com métodos de obtenção de informação (consultas) elementares que não provocam mudanças de estado. A regra diz, grosso modo, que se b é um dado de D, e se *rest* se sucede de D, então $b \otimes rest$ também se sucede a partir de D. Intuitivamente, b é uma condição que é satisfeita em D, e por isso pode ser acoplada como uma teste em uma nova transação que se sucede de D.

3. A terceira regra de inferência trata das transições elementares. Ela diz que se b pode modificar a base de dados D_1 para D_2, e se *rest* se sucede a partir de D_2, então $b \otimes rest$ se suscetível de ser executa a partir de D_1. Intuitivamente, b pode ser acoplada na frente de *rest*, para fazer uma ação que começa a partir de D_1 ao invés de D_2.

4. A quarta regra de inferência introduz a modalidade. Como dito, na primeira parte do trabalho, uma ação pode ser realizada de fato, mas também é possível verificar os efeitos de uma ação por meio de representações sem cometê-las de fato. Essas regras permitem combinar duas ações que poderiam ser executadas a partir de um estado D em uma única ação, convertendo uma delas em uma execução hipotética.

O objetivo principal deste sistema de prova é permitir encontrar deduções de um sequente dado. Para construir uma prova podemos utilizar duas estratégias bem conhecidas *bottom-up* (aplicando as regras a partir de uma axioma até alcançar o sequente almejado) e *top-down* (a partir do objetivo, aplicando as regras ao inverso até chegar a uma axioma). Quando uma estratégia *top-down* é escolhida para provar os sequentes, podemos extrair o caminho de execução da transação da prova. Se uma estratégia *bottom-up* for escolhida, vamos poder extrair o caminho de execução em ordem reversa, correspondendo a execução em modo reverso da transação.

A princípio, a correção deste sistema de inferência nos permite concluir declarações da forma $P,\ D_0 - -- \ \models\ (\exists)\phi$. Essa declaração diz que uma ação pode ser executada a partir de D. Essa declaração entretanto não nos diz quais são os estados intermediários e final, para isso é preciso concluir declarações da forma $P,\ D_0, \ldots,\ D_n\ \vdash\ (\exists)\phi$. Em particular, dado o estado D_0, nós queremos deduzir os outros estados. Essa dedução corresponde a execução normal de ação ϕ respectivamente.

Para que em uma prova não sejam inseridos sequentes irrelevantes, o que atrapalharia a extração da sequência de estados exata que corresponde a execução da transação, Bonner e Kifer utilizam uma noção de prova mais restrita de prova, chamada de dedução execucional. Uma dedução execucional será uma árvore binária, na qual cada nó é um sequente, as folhas são axiomas e os nós internos são derivados de seus filhos. Um dos ramos dessa árvore corresponde a porção da transação que é executada de fato. Os outros ramos correspondem as sub-transações que são executadas hipoteticamente, sem realizar nenhuma mudança de fato.

Definição (Dedução execucional em $\mathcal{I}\diamond$): Seja P uma base de transação. Uma dedução execucional em $\mathcal{I}\diamond$ de uma fórmula $(\exists\phi$ é uma árvore binária de sequentes que satisfaz as seguintes condições:

1. A raiz da árvore é um sequente da forma $P,\ D---\ \models\ (\exists)\phi$, para algum identificador de estado D.

2. As folhas da árvore são axiomas, isto é, um sequente da forma $P,\ D'---\ \models\ ()$, para algum identificador de estado D'.

3. Cada nó que não é uma folha pode ser ser derivado de seus filhos, pela aplicação de uma das regras do sistema $\mathcal{I}\diamond$, isto é o nó é o consequente de uma das regras para o qual seus filhos são o antecedente.

Em cada regra do sistema de inferência existe um consequente envolvendo uma fórmula *rest* e em cada antecedente existe exatamente um sequente com uma fórmula *rest*. Essa fórmula representa a parte da transação que é realizada de fato. Nas três primeiras regras só há um antecedente e ele foi realizada de fato. Já na regra 4 estão envolvidos dois antecedentes um foi realizado de fato e o outro é hipotético. É possível traçar uma caminho da raiz até uma folha se movendo apenas pelos nós que foram realizados de fato. Intuitivamente, o ramo de nós realizados de fato corresponde as atualizações reais dos estados, enquanto nos outros ramos correspondem a atualizações meramente possíveis de estados. Vendo quando as regras do grupo 3, que são as regras que lidam as atualização, são aplicadas na dedução do ramo realizado de fato podemos extrair a sequência de estados que são o caminho de execução da transação. Uma prova da correção e da completude deste sistema para a relação de consequência execucional é fornecida por Bonner e Kifer. (BONNER; KIFER, 1995)

3 Aplicações da lógica das transações à resolução de problemas

A seguir apresentamos alguns exemplos mostrando como a lógica das transações pode ser aplicana à resolução de problemas. Em todos os exemplos, nós assumimos o oráculo relacional, mas nos primeiros exemplos não nos limitaremos as condições de Horn serial-$\diamond$ porque queremos mostrar a expressividade da lógica como um todo. Apesar de realistas, os primeiros exemplos não pretendem descrever com precisão uma situação. Nosso objetivo com esses exemplos é ilustrar as ideias básicas por trás da semântica dos conectivos lógicos apresentados. Esta seção segue a apresentação da lógica das transações encontrada em Bonner e Kifer[1], mas com exemplos de nossa autoria.

Lembramos aos leitores que para descrever a execução de uma transação formalmente, a lógica das transações utiliza declarações da seguinte forma, que expressam uma forma de consequência lógica chamada consequência execucional:

$$P,\ D_1,\ \cdots,\ D_n \models \psi \tag{3.1}$$

Onde, P é um conjunto de fórmulas de transação, ψ é uma fórmula de transação e cada D_i é um identificador de estado (no caso do oráculo relacional, serão sentenças atômicas). Intuitivamente, P é um conjunto com definições de procedimentos e conceitos complexos. Esse conjunto descreve as ações enquanto conceitos. Esse conjunto, chamado de base de transação, e é utilizada sobretudo para representar, por meio de uma linguagem lógica, os conhecimentos declarativo e procedural que um gente pode utilizar durante a execução de uma tarefa, mas também pode ser utilizado para definir acontecimentos e modelar as consequências de uma ação.

A sequência $D_1,\ \cdots,\ D_n$ representa os efeitos que uma execução particular da transação ψ pode provocar no estado atual D_1, se a ação ψ for executada, passando de D_1 para D_2, etc. até D_n.

Métodos de obtenção de informação e atualizações

Em $\mathcal{TR}$ as transações incluem combinações de métodos de obtenção de informação e atualizações. As consultas são ações que obtém informações de um estado sem modificar o estado atual. Na verdade, qualquer predicado que seja consequência lógica de um caminho de comprimento 1 pode ser vista como uma consulta. As consultas incluem as fórmulas hipotéticas, que representam métodos utilizados para obtenção de estados possíveis. As

[1] (BONNER; KIFER, 1995).

consultas são expressas formalmente por declarações da forma:

Type your text

$$P,\ D_1 \models \psi \tag{3.2}$$

um caso particular de (4.1), no qual o comprimento da sequência é 1. Isto significa que a consulta ψ pode ser executada no caminho $\langle D_1 \rangle$. Ela será executada com sucesso se e somente se a proposição ψ é dada como verdadeira em D (um dado do problema em D, que foi estabelecido por meio do oráculo de dados) ou se ψ é derivada das proposições dadas como verdadeira em D a partir de P. Por exemplo, suponha que em um estado identificado por D seja um dado que p, ou seja, que $p \in \mathcal{O}^{\mathcal{D}}(i)$. Isso garante que uma consulta p seja executada com sucesso no caminho $\langle D \rangle$ de comprimento 1. Suponha ainda que a base de conhecimento, P, contenha a regra $h \leftarrow p$. Isso faz com que o agente também possa executar com sucesso um consulta h para saber que a proposição h é verdadeira.

Na lógica das transações precisamos fazer uma distinção entre os dados em um estado identificado por D e uma sequência de estados $\langle D \rangle$ com comprimento 1. Na relação de consequência execucional apresentada acima, nós assumimos que estamos falando de uma sequência de estados, mas não escrevemos os símbolos "$\langle$" e "$\rangle$" para não poluir a notação. D é um identificador de estado, um conjunto de dados abstratos, ao qual o oráculo de dados atribui um conjunto de fórmulas de primeira ordem. A palavra "dado" vem do latim e significa dar, e portanto, os dados são informações assumidas como verdadeiras no estado. Portanto, cada estado D está associado com algum conjunto (que pode ser vazio) de fórmulas (limitadas a linguagem de primeira ordem) dadas como verdadeiras e que representam os dados do problema. Já o caminho $\langle D \rangle$ representa uma "visão" do estado D, isto é o conjunto com as proposições derivadas a partir da união dos dados em D com o as informações na base de de transação (P). Essa diferença é utilizada para distinguir a "visão" que um agente possui de uma estado do problema, dos dados disponíveis naquele estado. Apenas um caminho $\langle D \rangle$ é consultado, o que significa que um agente não acessa os dados diretamente. Na lógica das transações, a "visão" $\langle D \rangle$ vai ser alterada apenas se o estado atual for modificado. Isso porque a base de transações P que pode ser utilizada para representar o conhecimento dos agentes é um parâmetro fixo da relação de consequência execucional.

Um programa para modelar um agente resolvendo problema inclui pelo menos duas partes: uma base de transações (P) e um identificador de estado, representando os dados do problema atais (D). Podemos imaginar a base de transações como uma memória de longo prazo, cujo conteúdo persiste pelo tempo, isto é, nós assumimos que ela não varia durante toda a a atividade de resolução de problema. As sequências de estados correspondem ao aspecto dinâmico da tarefa que está sendo realizada e D é o estado atual. Um agente pode acessar certos aspectos no estado atual, conjecturar sobre o que seria o caso se certas ações tomassem curso, sem de fato realizá-las, ou engajar em alguma ação modificando as relações e propriedades dos objetos.

Ao modelar um agente que resolve problemas na lógica das transações os procedimentos de atualização fazem uma distinção entre os fatos dados (os dados de um problema) e os fatos derivados a partir do conhecimento de um agente, mas os método de obtenção de informação não. Apenas um estado D é diretamente atualizado e apenas o caminho $\langle D \rangle$ é consultado. Por exemplo, um agente com uma base de conhecimento P pode modificar o estado do problema de D_1 para D_2. Mas, ao aplicar um método de obtenção de informação a "visão" que esse agente possui é das consequência de $P \cup D$. A impossibilidade de atualizar a base de transação é uma das limitações que a lógica das transações possui para representar agentes capazes aprender enquanto solucionam problemas. Isso porque nenhuma transação pode alterar a base de transações. Claro, que se quisermos podemos escolher outro conjunto de fórmulas para P para representar um agente mais informado, mas isso não é o mesmo que modelar uma ação que resulta na aquisição de conhecimento. Alguém poderia argumentar que esta critica não está correta, pois os novos conhecimentos aprendidos podem ser representadas na sequência de estados. Entretanto, essa linha de defesa não funciona para o conhecimento procedural que precisa da expressividades das fórmulas de transação, como as que são permitida na base de transações. Além disso, o caráter transitório dos estados parece não ser compatível com o caráter duradouro da aprendizagem.

Se estivermos no fragmento de $\mathcal{TR}$ que lida apenas com métodos de obtenção de informação (e fatos), isto é se P e ϕ são fórmulas de primeira ordem, então:

$$P,\ D \models \phi \quad \text{sse} \quad P \wedge \mathcal{O}^{\mathcal{D}}(D) \models^{c} \phi \tag{3.3}$$

O teorema acima, no qual "$\models^c$" denota a relação de consequência clássica, mostra que as consultas podem ser expressas na lógica clássica.

Vamos tratar agora como $\mathcal{TR}$ estende a lógica clássica para o conhecimento procedural e mudanças de estado. Enquanto as consultas não alteram o estado do problema, as atualizações são ações que mudam o estado do problema. O tipo mais simples de atualização, em $\mathcal{TR}$, é chamada de atualização elementar. Todas as atualizações elementares são atômicas no sentido de que elas não podem ser decompostas em atualizações mais simples e justamente por este motivo, elas são representadas com fórmulas atômicas. Em uma interpretação, as atualizações elementares possuem um valor satisfação e podem ter um efeito colateral no estado do problema. Para representar a execução de atualizações elementares é representada formalmente por meio da relação de consequência execucional:

$$P,\ D_1,\ D_2 \models u \tag{3.4}$$

Essa declaração diz que a ação u é uma atualização que pode atualizar o estado do problema de D_1 para D_2. Note que as atualizações possuem a mesma natureza sintática que as consultas, ambas são representadas por símbolos de predicados. Vamos utilizar os predicados atômicos $p.ins$, para representar a inserção de objetos (ou tuplas de objetos) na extensão de p e $p.del$ para representar a exclusão de objetos (ou tuplas de objetos) na extensão de p, em conformidade com o oráculo relacional.

Por exemplo, suponha que p é um símbolo de predicado binário para a posição de uma peça em um tabuleiro de xadrez. Então, as atômicas $p.del(h4, pc_1)$ e $p.ins(h5, pc_1)$ são atualizações elementares. Intuitivamente elas significam "remova a peça$_1$ da posição $h4$" e "insira a peça$_1$ na posição $h5$". Uma execução de $p.del(h4, pc_1)$ leva de um estado D para um estado $D - \{p(h4, pc_1)\}$. E inversamente, uma ocorrência da ação $p.ins(h5, pc_1)$ faz com que uma situação mude de um estado D para $D \cup \{p(h5, pc_1)\}$. Esse comportamento é expresso formalmente pelas seguintes declarações, que são verdadeiras para qualquer base de conhecimento P e identificador de estado D:

$$P,\ D,\ D - \{p(h4, pc_1)\} \models\ p.del(h4, pc_1)$$

$$P,\ D,\ D \cup \{p(h5, pc_1)\} \models\ p.ins(h5, pc_1)$$

Bonner e Kifer[2] advertem que as operações de inserção e exclusão não estão construídas na semântica da lógica das transações. Não há, a princípio, qualquer conexão intrínseca entre os predicados p, $p.del$ e $p.ins$ e essa mudança de estados. A escolha desses nomes paras as operações foi arbitrária. Todos eles são predicados ordinários de $\mathcal{TR}$ e a conexão entre eles é estabelecida por meio do oráculo de transição de estados. O que está presente na semântica e teoria da prova de é a habilidade de consultar tais oráculos, quaisquer que eles sejam.

Nas próximas seções mostraremos por meio de exemplos simples a ideia de cada operador.

Composição serial

Uma forma de combinar transações é sequenciá-las, isto é executar uma depois da outra. Por exemplo, alguém poderia tirar uma peça de sua posição e colocá-la em uma nova posição. Para sequenciar as transações há um operador binário, $\otimes$, chamado de conjunção serial. A fórmula $\phi \otimes \psi$ denota uma única transação complexa definida a partir das transações ϕ seguida da transação ψ.

Em um contexto relacional, onde p é um predicado binário e $p(h4, pc_1)$ e $p(h5, pc_1)$ são predicados representando a posição de um peão em um tabuleiro de xadrez, a expressão $p.del(h4, pc_1) \otimes p.ins(h5, pc_1)$ denota uma única transação que possui dois efeitos. Intuitivamente ela significa "primeiro, remova a peça$_1$ da posição $h4$ e, então insira peça$_1$ na posição $h5$". Suponha que no estado D seja dado que $p(h4, pc_1)$. Uma execução da ação acima faz com que o problema mude do estado D, para $D - \{p(h4, pc_1)\}$ para

[2] (BONNER; KIFER, 1995).

$D - \{p(h4, pc_1)\} \cup \{pc(h5, pc_1)\}$. Isto pode ser expresso formalmente por:

$P,\ D,\ D - \{p(h4, pc_1)\}\ ,\ D - \{p(h4, pc_1)\} \cup \{p(h5, pc_1)\} \models p.del(h4, pc_1) \otimes p.ins(h5, pc_1)$

Testes e condições

De acordo com Bonner e Kifer,[3] a conjunção "clássica" é central na semântica de $\mathcal{TR}$, pois ela permite modelar uma série de restrições que uma transação deve satisfazer para ser executada. Por exemplo, é possível requerer que todo estado intermediário satisfaça uma restrição ou proibir certos resultados. É possível colocar consultas que atuam como testes em qualquer lugar em sequência de atualizações. Deste modo, é possível monitorar o progresso da transação e forçá-la a abortar se determinada condição não for cumprida durante a execução.

Condição Prévia

Podemos definir uma condição prévia para iniciar uma ação colocando uma consulta no início da transação. Para ilustrar essa situação imagine uma situação na qual para arrancar um carro manual precisamos estar na primeira marcha. Essa transação pode ser executada apenas se a condição prévia for satisfeita inicialmente.

Seguiremos com exemplo do xadrez para exemplificar esse possibilidade. Assumindo que a atômica $p(h4, pc_1)$ é uma consulta (isto é, não muda os estado do problema), então a expressão $p(h4, pc_1) \otimes p.del(h4, pc_1) \otimes p.ins(h5, pc_1)$ denota um teste seguido duas atualizações. Intuitivamente, ela significa, "teste $p(h4, pc_1)$ é o caso, então retire pc_1 da posição $h4$, e então insira pc_1 na posição $h5$". Podemos retirar uma peça de uma posição apenas se ela estiver lá. Em outras palavras, a transação pode ser executada atualizado o estado D, para $D - \{p(h4, pc_1)\}$ e para $D - \{p(h4, pc_1)\} \cup \{p(h5, pc_1)\}$, apenas se apenas se o teste $p(h4, pc_1)$ puder ser satisfeito no caminho inicial $\langle D \rangle$ do problema. Formalmente:

$$\text{Se } P,\ D \models p(h4, pc_1)$$

$$\text{Então } P,\ D, D - \{p(h4, pc_1)\}\ , D - \{p(h4, pc_1)\} \cup \{p(h5, pc_1)\} \models$$
$$p(h4, pc_1) \otimes p.del(h4, pc_1) \otimes p.ins(h5, pc_1)$$

Condição posterior

Ao colocar um teste no final de uma transação podemos definimos uma condição posterior que deve ser atingida para que a transação seja executada. A transação será realizada modificando o estado atual do problema apenas em um caminho no qual teste final é bem sucedido. A condição posterior é útil para eliminar a ocorrência ações que possuam efeitos colaterais indesejáveis.

[3] (BONNER; KIFER, 1995).

Para continuar com o exemplo do jogo de xadrez, imagine agora, uma movimentação do peão que coloca seu próprio rei em xeque, pelas regras do xadrez, tal movimento não é válido. Se leg é uma consulta que verifica se seu próprio rei não está em xeque, então a expressão $p(h4, pc_1) \otimes p.del(h4, pc_1) \otimes p.ins(h5, pc_1) \otimes leg$ denota um teste seguido de duas atualizações seguidas de um teste, intuitivamente, ela significa "teste se $p(h4, pc_1)$ é o caso e, se for, retire pc_1 da posição $h5$ e insira pc_1 na posição $h5$, contanto que seu próprio rei não fique em xeque". Em outras palavras, essa ação é executada se $p(h4, pc_1)$ é satisfeita no caminho $\langle D \rangle$, então este estado muda para $D - \{p(h4, pc_1)\}$ e, então para o estado $D - \{p(h4, pc_1)\} \cup \{p(h5, pc_1)\}$, mas apenas se leg for satisfeita na última parte do caminho. Se não existir um caminho formada a partir do estado inicial que satisfaz todas essas operações, então a jogada não pode ser realizada. Formalmente:

$$\text{Se } P,\ D \models p(h4, pc_1)$$

$$\text{e } P,\ D - \{p(h4, pc_1)\} \cup \{p(h5, pc_1)\} \models leg$$

$$\begin{aligned}\text{Então } P,\ D,\ D - \{p(h4, pc_1)\}\,,\ D - \{p(h4, pc_1)\} \cup \{p(h5, pc_1)\} \models \\ p(h4, pc_1) \otimes p.del(h4, pc_1) \otimes p.ins(h5, pc_1) \otimes leg\end{aligned}$$

Alguém poderia se perguntar se não deveríamos verificar o xeque antes de movimentar a peça, pois caso a condição posterior não seja cumprida ficaríamos com a peça em posição irregular. Entretanto, isto não é correto. Devemos lembrar que a execução de uma transação é atômica e só pode ser cometida de forma integral. Isto é, uma fórmula formada pela conjunção serial não representa duas ações consecutivas, mas uma única ação atômica. caso a condição posterior não possa ser satisfeita, então nenhuma mudança modificando o estado atual é realizada. Em termos computacionais poderíamos dizer que mudanças computadas até então sofreriam um *rollback*. Em outras palavras, se a condição posterior não pode ser satisfeita, nada ocorre, nenhuma mudança é realizada e a peça permanece na posição inicial.

Execução Hipotética

Antes realizar uma jogada válida é conveniente descobrir se ela traria algum prejuízo, por exemplo, ao deixar uma peça valiosa exposta. Para verificar os efeitos de uma ação antes de executá-la podemos utilizar um operador hipotético. Suponha que que predicado sec, é uma consulta que indica que nenhuma peça do jogador está sob ameaça e que $move(h5, pc_1)$ representa uma jogada válida, que move a peça pc_1 para a posição $h5$. Antes de realizar está jogada, podemos verificar suas consequências e só realizar o movimento caso ele não exponha nenhuma peça. Essa tarefa pode ser representada pela fórmula $\diamond[move(h5, pc_1) \otimes sec] \otimes move(h5, pc_1)$.

$$\text{Se } P,\ D,\ \cdots,\ D_n \models move(h5, pc_1)$$

$$\text{e } D_n \models sec$$

$$\text{então } P,\ D,\ \cdots,\ D_n \models move(h5, pc_1) \otimes sec$$

$$\text{então } P,\ D \models \diamond[move(h5, pc_1) \otimes sec]$$

$$\text{portanto } P,\ D,\ \cdots,\ D_n \models \diamond[move(h5, pc_1) \otimes sec] \otimes move(h5, pc_1)$$

Transações não-determinísticas

A disjunção e o quantificador existencial podem ser utilizados para compor transações não-determinísticas. Por exemplo, a fórmula $\phi \vee \psi$ significa "faça a transação ϕ ou faça a transação ψ". E a fórmula $\exists X\ \phi(X)$ significa "faça a transação $\phi(c)$ para algum c".

Suponha que antes do jogo de xadrez começar, os jogadores decidam jogar uma moeda para saber quem começara. Se as fórmulas $cara$ e $coroa$ representam os possíveis resultados do lance, então a fórmula $ins.cara \vee ins.coroa$ representa uma transação não-determinística para jogar a moeda. Os agentes não têm controle do resultado da transação e esta pode ser executada por diversos caminhos diferentes a partir do estado inicial. Uma teoria da prova deve derivar cada um desses possíveis caminhos, mas apenas um deles será realizado arbitrariamente. Se essa transação é realizada a partir de um estado D, então o estado final do problema será $D + \{cara\}$ ou $D + \{coroa\}$. Formalmente:

$$P,\ D, D \cup \{coroa\} \models ins.cara \vee ins.coroa$$

$$P,\ D, D \cup \{cara\} \models ins.cara \vee ins.coroa$$

são declarações verdadeiras para qualquer base de transação P e cada uma diz que uma ocorrência de $ins.cara \vee ins.coroa$ *pode* ser realizada de uma forma. Considere agora, que em determinado estado D do jogo de xadrez, um dos jogadores possui apenas 2 opções de peças (pc_1, pc_2) que ele pode movimentar, mas como ele está com preguiça de pensar, ele prefere escolher uma peça arbitrariamente. Seja m um predicado unário representado a propriedade de poder ser movida e t um predicado unário representando a propriedade de ser tocada. A fórmula $\exists X[m(X) \otimes t.ins(X)]$ representa uma transação na qual um objeto com a propriedade m é selecionado arbitrariamente e, então esse objeto é tocado (colocado na extensão da propriedade t). Neste caso, quando a consulta é bem sucedida e existe um objeto que satisfaz a propriedade ela também retorna o nome desse objeto para ser utilizado nas operações seguintes. Note que o conjunto de alternativas não determinísticas vai depender dos dados disponíveis. Formalmente, para toda peça a, tal que $m(a)$:

$$\text{Se } P,\ D \models m(a)$$

$$\text{então } P,\ D,\ D \cup \{t(a)\} \models \exists X[m(X) \otimes t.ins(X)]$$

No exemplo, no estado D as fórmulas $m(pc_1)$ e $m(pc_2)$ são dadas como verdadeira. Então há duas escolhas arbitrárias, que resultam em duas atualização diferente. Formalmente, as duas declarações abaixo são são verdadeiras para quaisquer base P.

$$P,\ D,\ D \cup \{t(pc_1)\} \models \exists X[m(X) \otimes t.ins(X)]$$

$$P,\ D,\ D \cup \{t(pc_2)\} \models \exists X[m(X) \otimes t.ins(X)]$$

Regras

Regras são fórmulas da forma $p \leftarrow \phi$, (ou equivalente $\phi \rightarrow p$) fechadas universalmente, onde p é uma fórmula atômica e ϕ é um objetivo serial$\diamond$ de Horn. Ela é uma abreviação de $p \vee \neg\phi$ e pode ser lida como "faça p ou não faça ϕ. Como colocado por Bonner e kifer,[4] essa fórmula também pode ser lida como "para fazer p é suficiente fazer ϕ". Essa última leitura fornece uma forma de criar sub-rotinas, tornando mais fácil utilizar $\mathcal{TR}$ como linguagem de programação. Por exemplo, na regra $p(X) \leftarrow \phi$, o predicado p é um nome de um procedimento, a variável X é um parâmetro e a fórmula ϕ é a definição do procedimento. Nós assumimos que todas as regras possuem um fecho universal. Portanto, toda regra escrita, por exemplo como:

$$p(X) \leftarrow \phi$$

é uma abreviação para:

$$\forall X[p(X) \leftarrow \phi]$$

Vamos refinar o nosso exemplo anterior definindo uma regra para jogar uma moeda. Seja X uma variável percorrendo o conjunto das moedas. A primeira regra diz que para jogar uma moeda é suficiente fazer com que ela de cara, já segunda regra diz que é suficiente fazer com que ela de coroa:

$$j(X) \leftarrow ins.cara(X)$$
$$j(X) \leftarrow isns.coroa(X)$$

Essa regra é equivalente a:

$$j(X) \leftarrow ins.cara(X)\ \vee\ isns.coroa(X)$$

[4] (BONNER; KIFER, 1995).

Formalmente, se m_1 é uma moeda e essas são as únicas regras definido definindo $j(X)$ em P, então ambas as declarações abaixo são verdadeiras para qualquer P.

$$P, \cup \{cara(m_1)\} \models j(m_1)$$
$$P, \cup \{coroa(m_1)\} \models j(m_1)$$

Isso significa que embora a moeda sempre vá aterrissar de um dos dois lados não podemos saber de antemão qual será o resultado final.

Restrições

Enquanto a disjunção clássica gera não-determinismo na transação, a conjunção clássica restringe o não-determinismo de uma transação. Isto é, a transação $\phi \wedge \psi$ é mais determinística que ϕ ou ψ por si mesmas. Isso porque uma execução de $\phi \wedge \psi$ deve ser uma execução permitida de ϕ e uma execução permitida de ψ. A declaração "vá até o shopping" ∧ "não passe pelo Eixo Monumental" tem o mesmo significado o mesmo que "vá até o shopping sem passar pelo Eixo Monumental". Existem diversos caminho pelos quais alguém poderia executar a ação de ir até o shopping a partir do estado atual, a conjunção restringe essa possibilidade eliminando algumas alternativas. A conjunção serial não faz com que a transação seja executada como duas transações concomitantes. Ao invés disso, ela combina as duas transações em uma única transação mais restrita.

Base de transação ou base de conhecimento

Para representar o conhecimento que um agente pode utilizar durante uma tarefa, utilizamos um conjunto de regras. Essas regras, além de representar o conhecimento declarativo e procedural pode defnir acontecimentos e e consequências das ações.

Exemplo[5] [**Movendo blocos**]: As regras abaixo simulam possíveis habilidades de um agente mover blocos em um mundo de blocos. Os identificadores de estado neste exemplo são definidos em termos de dois predicados: *sobre*(X, Y), que diz que o bloco x está sobre o bloco y; $l(x)$, que diz que x está livre, ou seja que não há blocos sobre x. As regras abaixo definem três ações que o agente é capaz de realizar para mudar o estado do mundo. Cada ação é avaliada da esquerda para a direita, e a transação falha se qualquer de suas premissas falhar. O fecho universal está omitido para não poluir a notação.

$$mover(X, Y) \leftarrow pegar(X) \otimes colocar(X, Y)$$

$$colocar(X, Y) \leftarrow X \neq Y \otimes l(Y) \otimes l.del(Y) \otimes sobre.ins(X, Y)$$

$$pegar(X) \leftarrow l(X) \otimes sobre(X, Y) \otimes sobre.del(X, Y) \otimes l.ins(Y)$$

[5] Este exemplo é fornecido por Bonner e Kifer. (BONNER; KIFER, 1995)

As ações básicas são $pegar(X)$ e $colocar(X, Y)$ e estabelecem como "pegar o bloco X" e como "colocar o bloco X no topo do bloco Y". Elas são definidas a partir de consultas elementares e atualizações elementares. E representam ações que têm como resultado a inserção ou retirada de objetos (ou tuplas) da extensão de propriedades (relações) no mundo dos blocos. A outra regra combina essas ações básicas em uma ação mais complexa. A regra $colocar(X, Y)$ diz como fazer para colocar o bloco X sobre Y, essa ação pode ser realizada apenas se X for diferente de Y e se Y estiver livre. É importante notar que este exemplo envolve consultas e atualizações, por exemplo, a atômica $l(Y)$ é um teste que precisa ser bem sucedido para que a transação como um todo seja cometida.

Suponha que a base de transação P contém as três regras acima. E que o agente tenha a tarefa[6] de pegar um bloco qualquer ($\exists X pegar(X)$). Como o bloco não foi especificado, se trata de uma tarefa não-determinística. De modo que ele pode escolher qualquer um dos blocos disponíveis para pegar. Suponha ainda que o identificador de estado D_0 represente um estado inicial com três blocos, no qual o bloco a está sobre o bloco c e o bloco b está sozinho. Se ele escolher pegar o bloco a, então a situação deveria mudar dos estados D_0 para D_1 para D_2, onde:

$$D_0 = \{l(a), l(b), sobre(a, c)\}$$
$$D_1 = \{l(a), l(b)\}$$
$$D_2 = \{l(a), l(b), l(c)\}$$

A seguinte tabela, adaptada de Bonner e Kifer, apresenta uma dedução execucional na qual o agente escolhe pegar o bloco a. Nessa tabela cada sequente se segue do sequente imediatamente abaixo por meio de uma regra de inferência. cada inferência envolve uma unificação da atômica mais a esquerda da fórmula que estamos processando com o estado, com a base de transação ou com uma transição elementar. Ao unificação da atômica l(x) com a atômica l(a) no estado D_0 na construção dessa prova representa a escolha do agente em pegar o bloco a. A construção de outra prova representaria uma escolha diferente e a ocorrência diferente da ação.

Regra	Unificador	Sequente
2.8		$P,\ D_0 - -- \ \vdash\ (\exists) pegar(X)$
2.9	$\{\}$	$P,\ D_0 - -- \vdash (\exists)[l(X) \otimes sobre(x, y) \otimes sobre.del(x, y) \otimes l.ins(y)$
2.9	$\{x \backslash a\}$	$P,\ D_0 - -- \ \vdash\ (\exists)[sobre(a, y) \otimes sobre.del(a, y) \otimes l.ins(y)$
2.10	$\{y \backslash c\}$	$P,\ D_0 - -- \ \vdash\ (\exists)[sobre.del(a, c) \otimes l.ins(c)]$
2.10	$\{\}$	$P,\ D_1 - -- \ \vdash\ l.ins(c)$
	$\{\}$	$P,\ D_2 - -- \ \vdash\ ()$

[6] Isto não é um problema porque há um esquema para fazer tal tarefa diretamente.

Vamos tratar agora de um exemplo de tarefa que mostra como diferenciar os resultados (efeitos que contribuem com a definição da ação) e as consequências de uma ação (efeitos que são engatilhados, após a excussão da ação). O processo de distinguir resultados e consequências não é um processo objetivo, pois, na hora de definir a forma como uma ação é realizada, podemos considerar que alguns efeitos são claramente resultados e outros são claramente consequências, mas também existem efeitos que ficam em uma zona nebulosa. Esse problema, chamado de problema da ramificação está diretamente relacionado com o problema de especificar o que não muda após a realização de uma ação chamado de *frame problem.* A tentativa de declarar em uma teoria que nada além dos efeitos diretos da ação muda impossibilita que as ramificações de uma ação (suas consequências indiretas) sejam consideradas, o que requer ignorar completamente os efeitos indiretos que uma ação. Em $\mathcal{TR}$, o *frame problem* é solucionado no nível das operações elementares por meio do oráculo de transição que determina exatamente o que muda e o que não muda quando uma atualização é realizada, como todas as transações são definidas em termos das atualizações elementares, essa solução se propaga para os níveis superiores, entretanto, ainda é possível inserir efeitos indiretos para uma ação.

Exemplo[7] [**As consequências de mover blocos**]: No exemplo anterior, nós definimos uma ação para pegar blocos na qual era necessário que o bloco estivesse livre, agora, vamos definir uma ação onde tal condição não é necessária. Vamos considerar que essa ação levanta o bloco em questão como um resultado direto e como consequência levanta todos os blocos acima dele. Isto é, quando um bloco é levantado, todos os blocos acima dele também são. Para descrever essa consequência utilizamos uma terceira regra.Também vamos definir uma nova ação para colocar correspondente. Para tal indicaremos o estado dos bloco com mais um predicado "lev" indicando que o bloco está levantado.

$$colocar(X, Y) \leftarrow X \neq Y \otimes l(Y) \otimes l.del(Y) \otimes sobre.ins(X, Y) \otimes lev.del(x)$$

$$pegar(X) \leftarrow sobre(X, Y) \otimes sobre.del(X, Y) \otimes l.ins(Y) \otimes lev(X)$$

$$lev(X) \leftarrow sobre(X, Y) \wedge lev(Y)$$

Como apontado por Bonner e Kifer, para qualquer base de transação P com as regras acima, se $D = \{l(a), l(b), sobre(a, c)\}$ é o estado inicial da execução da tarefa de $pegar(c)$, então a declaração seguinte é verdadeira para qualquer c:

$$\text{Se } P,\ D - - - D' \vDash pegar(c)$$

$$\text{então } P,\ D' \vDash lev(a) \wedge lev(c)$$

[7] Este exemplo é fornecido por Bonner e Kifer. (BONNER; KIFER, 1995)

Isto é, ao pegar o bloco c, ele será levantado e como o bloco a está sobre o bloco c, ele também será levantado. Se o bloco c for colocado sobre outro bloco, o bloco a também deixará de estar levantado. Conseguir tratar das ramificações no formalismo é útil para lidar com ações que possuem múltiplos, possivelmente infinitos, efeitos colaterais, sem que seja necessário descrever na definição da ação todos seus efeitos. Por exemplo, se a posição de um objeto for um dado do problema, para definir uma ação que muda a posição de um objeto basta eliminar a sentença atômica que representa posição antiga do objeto e, então inserir um predicado representando a nova posição. Além disso, se houver uma regra na base de transações que permite dada as posições de dois objetos calcular a distancia entre eles, então a distancia entre o objeto movido e todos os outros objetos vai ser modificada automaticamente. Deste modo, é possível modificar infinitas relações por meio de uma ação que pode ser descrita que faz apenas duas modificações explicitas na base de dados.

Os exemplos anteriores ilustraram tarefas simples, vamos começar a tratar agora de exemplos nas quais a tarefa apresenta certa dificuldade. Em tarefas rotineiras o planejamento não desempenha uma papel significante, já em tarefas problemáticas, o planejamento ocupa grande parte da resolução de problemas. Na lógica das transações é possível representar essa etapa. Mas, se não tivermos nenhum tipo de plano, ainda podemos tentar resolver um problema por tentativa e erro sem nenhuma estratégia mais inteligente. Nesses casos, para alcançar um objetivo, é possível combinar as ações disponíveis de forma arbitrária até que, por acaso, o objetivo seja alcançado. Para representar a capacidade de um agente resolver um problema por tentativa erro, considere uma base de conhecimento com n ações básicas $a_1, \dots a_2$. Não é preciso levar em consideração todas as ações, pois as ações mais complexas que as básicas serão uma combinação das básicas de toda forma. Além dessa ações nós definimos uma tentativa como qualquer sequência de ações básicas. A capacidade de resolver um problema por tentativa e erro pode ser representada acrescentando na base de transações as seguintes regras, onde a é qualquer ação e tentativa é qualquer sequência de ações:

$$\begin{aligned}
&tentativa \leftarrow a \otimes tentativa \\
&tentativa \leftarrow a \\
&a \leftarrow a_1 \\
&a \leftarrow a_2 \\
&\dots \\
&a \leftarrow a_n
\end{aligned}$$

Suponha que o objetivo dele seja fazer com que ϕ seja verdade. A tarefa que um agente pode realizar é tentar até que o objetivo seja cumprindo. Essa tarefa é representa pela fórmula $tentativa \otimes \phi$, que pode ser executada através de todos os caminhos terminando em um estado D_n qualquer, tal que $P, D_n \models \psi$. Se não for possível atingir um estado

D_n no qual $D_n \models \psi$, é porque essa tarefa não pode ser realizada, ou seja, o problema não possui solução nas condições impostas. Na vida real, um agente provavelmente não executaria a tarefa descrita acima mais uma ação parecida $tentativa \otimes \phi \vee tentativa$, na qual ele poderia para em um estado diferente do inicial, mas sem alcançar o objetivo.

O próximo exemplo mostra como representar um agente capaz de utilizar uma estratégia mais sofisticada análise dos meios e fins. A análise dos meios e fins procura eliminar as diferenças entre os estado atual e o estado meta, mas não abre mão de uma ação se ela não puder ser aplicada imediatamente. A principal característica dessa heurística é que ela procura desbloquear ações que não estão em condições de serem executadas. Nos exemplos anteriores, quando as condições de uma ação não eram satisfeitas, a ação simplesmente não era executada, agora, vamos descrever planos que visam liberar as ações que tinham condições bloqueadas. Na base de conhecimento não vamos definir apenas ações, mas planos também. A diferença entre uma ação (enquanto conceito) e um plano é que um plano possui um caráter mais abstrato que a ação.

Exemplo[8] [**Analisando os meios para mover blocos**]: As regras abaixo simulam a análise do meios. No primeiro exemplo do mundo dos blocos era necessário que um bloco estivesse livre para ser movido, caso ele não estivesse livre, o agente não era capaz movê-lo. Este exemplo ilustra um caso no qual o agente pode tentar liberar um bloco antes de movê-lo. Para isso, nós introduzimos um novo predicado $obter.l(X)$ para obter a liberação do bloco X. Com esse predicado se define uma nova versão de $mover(X, Y)$ chamada de *obter.sobre*. A ação indicada por esse predicado não falha se o bloco não estiver liberado, pois pode realizar uma serie de ações para liberar o bloco e então, movê-lo. Essa nova capacidade é representada em uma base de transações com as regras:

$$obter.sobre(X, Y) \leftarrow obter.l(X) \otimes obter.l(Y) \otimes mover(X, Y) \tag{3.5}$$

$$obter.l(X) \leftarrow sobre(Y, X) \otimes obter.l(Y) \otimes mover(Y, Z) \otimes Z \neg X \tag{3.6}$$

$$obter.l(X) \leftarrow l(X) \tag{3.7}$$

$$mover(X, Y) \leftarrow pegar(X) \otimes colocar(X, Y) \tag{3.8}$$

$$colocar(X, Y) \leftarrow X \neq Y \otimes l(Y) \otimes l.del(Y) \otimes sobre.ins(X, Y) \tag{3.9}$$

$$pegar(X) \leftarrow l(X) \otimes sobre(X, Y) \otimes sobre.del(X, Y) \otimes l.ins(Y) \tag{3.10}$$

As novas regras fornecem planos abstratos para atingir um objetivo. As regras (3.6) e (3.7) estabelecem um plano recursivo: liberando o bloco de cima, antes de liberar um bloco

[8] Este exemplo é fornecido por Bonner e Kifer. (BONNER; KIFER, 1995)

e terminando a recurção quando o bloco finalmente é liberado. A ação $obter.sobre(X, Y)$ tenta liberar os blocos X e Y antes de mover X para cima de Y.

Nós definimos anteriormente conhecimento procedural como a aptidão para realizar uma ação intencionalmente. De modo que se tornou necessário investigar qual articulação conceitual entre executar uma ação A e ter a intenção de fazer A que nos permitisse dizer que a ação A é executada intencionalmente. Alguns autores que se debruçaram sobre a relação entre as ações e as intenções apresentaram alguns contra-exemplos aparentemente contrários a tese de que fazer A intencionalmente significa o mesmos que ter a intenção de fazer A e fazer A. entretanto, se olharmos para a semântica do conectivo $\wedge$, podemos perceber que ela captura a ideia que gostaríamos, fugindo dos contra-exemplos que tratam o conectivo "e " como uma articulação entre dois processos diferentes sendo realizadas em paralelo. O problema que se apresenta agora é o de saber se é possível expressar adequadamente na lógica das transações procedimento como "ter a intenção de fazer A" que deve ser satisfeita no mesmo caminho que a ação A. Antes de prosseguir, vamos ver um exemplo no qual planos são criados planos envolvendo uma ação com um efeito intencional e um efeito não intencional.

Exemplo[9] [**frame problem**]: Para ilustrar o poder expressivo da lógica das transações, Bonner e Kifer mostram como é possível aplicá-la em um caso que se mostrou bastante elusivo no passado. O seguinte relato policial foi utilizado parar elencar uma série dificuldades envolvendo o raciocínio temporal:

> "Em 1985, um individuo desconhecido carregou a arma e, após uma pequena pausa, descarregou em Freud."(BONNER; KIFER, 1995, p. 77)[10].

O problema era concluir a partir do relato acima que Freud estava morto e que a arma não havia descarregado miraculosamente durante a pausa. A solução que a lógica das transações oferece para simular o evento acima é o seguinte: Utilizamos quatro predicados para descrever a situação em cada estado: "*carre*", que significa que a arma está carregada; "*desca*" que significa que a arma está descarregada; "*vivo*", que significa que Freud está vivo; e "*morto*", que significa que Freud está morto. Também são definidas cinco ações, *carregar*, *esperar*, *atirar*, *descarregar* e *morrer*:

$carregar \leftarrow desca.del \otimes carre.ins$

$esperar \leftarrow pausa$

$atirar \leftarrow carre \otimes descar \otimes morrer$

[9] Este exemplo baseado no fornecido por Bonner e Kifer. (BONNER; KIFER, 1995)

[10] Tradução nossa do original: *" In 1985, an unknown individual loaded a gun and, a brief pause, discharged into Freud."*.

$descarregar \leftarrow carre.del \otimes desca.ins$

$morrer \leftarrow vivo.del \otimes morto.ins$

Onde *pausa* é uma transição elementar tal que $pausa \in \mathcal{O}^{\mathcal{T}}(D, D)$, para todo identificador de estado D. Deste modo, a execução da transação $carregar \otimes esperar \otimes atirar$ resulta na morte Freud. Suponha que o estado inicial $D_0 = \{desca, vivo\}$, $D_1 = \{vivo\}$, $D_2 = \{vivo, carre\}$, $D_3 = \{\}$, $D_4 = \{desca,\}$, $D_5 = \{desca, morto\}$. A seguinte tabela apresenta uma dedução execucional mostrando a execução de $carregar \otimes esperar \otimes atirar$. Nessa tabela cada sequente se segue do sequente imediatamente abaixo por meio de uma regra de inferência e nós ignoramos a unificação pois não há variáveis presentes.

Regra	Sequente
2.8	$P,\ D_0 - -- \ \vdash\ carregar \otimes esperar \otimes atirar$
2.10	$P,\ D_0 - -- \vdash desca.del \otimes carre.ins \otimes esperar \otimes atirar$
2.10	$P,\ D_1 - -- \ \vdash\ carre.ins \otimes esperar \otimes atirar$
2.10	$P,\ D_2 - -- \ \vdash\ esperar \otimes atirar$
2.8	$P,\ D_2 - -- \ \vdash\ atirar$
2.9	$P,\ D_2 - -- \ \vdash\ carre \otimes vivo \otimes carre.del \otimes vivo.del \otimes desca.ins \otimes morto.ins$
2.9	$P,\ D_2 - -- \ \vdash\ vivo \otimes carre.del \otimes vivo.del \otimes desca.ins \otimes morto.ins$
2.10	$P,\ D_2 - -- \ \vdash\ carre.del \otimes vivo.del \otimes desca.ins \otimes morto.ins$
2.10	$P,\ D_1 - -- \ \vdash\ vivo.del \otimes desca.ins \otimes morto.ins$
2.10	$P,\ D_3 - -- \ \vdash\ desca.ins \otimes morto.ins$
2.10	$P,\ D_4 - -- \ \vdash\ morto.ins$
	$P,\ D_5 - -- \ \vdash\ ()$

Nós construímos esta dedução a partir de uma estratégia do tipo *top-down*, isto é começamos com o sequente que queríamos de provar ($P,\ D_0 - -- \ \vdash\ carregar \otimes esperar \otimes atirar$) e aplicamos as regras de inferência ao reverso, até encontrar o axioma $P,\ D_5 - -- \ \vdash\ ()$. Deste modo, foi possível extrair um caminho de execução da transação $carregar \otimes esperar \otimes atirar$, isto é, a sequência de estados que uma ocorrência da ação pode realizar. Para realizar essa operação, basta extrair o estado inicial e os estados nos consequentes da regra 2.10, que aplica as definições de transição elementar. Neste exemplo, o caminho de execução é $D_0, D_1, D_2, D_2, D_1, D_3, D_4, D_5$. Caso a construção da prova utilizasse uma estratégia do tipo *bottom-up*, teríamos a sequência reversa, representando uma execução ao reverso da transação. Este sistema de prova deduz um sequente $P,\ D_0 - -- \ \vdash\ \psi$, cujo caminho de execução é $D_0, D_1, \cdots, D_n$ se e somente se $P,\ D_0, D_1, \cdots, D_n \ \models\ \psi$.

Conclusão

Bonner e Kifer afirmam que a lógica das transações permite representar formalmente o conhecimento declarativo e o procedural e agentes que utilizam esses conhecimentos para resolver problemas. Neste modelo, os agentes são representados como um programa lógico capaz de gerenciar bancos de dados. Nos propomos investigar ao longo desta dissertação em que medida essa declaração estava correta.

A primeira dificuldade com a qual nos deparamos é que não existe um critério objetivo para determinar se um formalismo captura ideias e noções vagas. Além disso, também percebemos que apesar da vasta literatura sobre problemas não há muita discussão sobre o que são problemas, pois a maior parte dela se volta para tarefa de determinar como eles são resolvidos. Como exceção a essa regra encontram-se os trabalhos de Veloso que visam definir o que são problemas. Percorremos apenas um destes trabalhos para encontrar uma definição que nos pareceu compatível com nossas ideias e concepções do que são problemas. Nesta concepção um problema é definido como uma estrutura formada por um conjunto de dados abstratos indicando o conjunto das instâncias do problema com o qual estamos lidando, um conjunto de dados de tipo abstrato indicando o conjunto dos possíveis resultados, e uma relação atribuindo a cada instância do problema os resultados que satisfazem as condições do problema, um conjunto com descrições das funções e uma função que avalia se determinada descrição induz uma função que atribui para toda instância do problema um resultado satisfatório. As soluções de um problema são os programas que induzem uma função que satisfaz a condição do problema.

Também procuramos definir informalmente o que significa conhecimento. Apontamos que existem três tipos de conhecimento e que estávamos interessados em dois: o declarativo e o procedural. O conhecimento declarativo é definido classicamente como crença verdadeira justifica e definimos o conhecimento procedural como aptidão para realizar tarefas intencionalmente. Essa definição levou a questão de entender a relação entre as ações e a intenção e de determinar o significa "fazer algo intencionalmente". Propusemos que ao interpretar a conjunção "e" usando o mesmo sentido que o cognitivo "$\wedge$" apresenta na semântica da lógica das transação evitaríamos que os contra-exemplos típicos servissem como evidencia contrária a tese que define "fazer A intencionalmente" como "fazer A *e* ter a intenção de fazer A". Isso porque na semântica de $\mathcal{TR}$ este cognitivo não conecta duas coisas ocorrendo em paralelo, mas aglomera em uma única ação mais determinística dois conceitos, contribuindo para determinar o modo como eles devem ser executados.

O segundo capítulo, foi dedicado a fornecer uma introdução a lógica das transações, expondo sua semântica e teoria da prova. A relação de consequência dessa lógica visa capturar a noção de execução de uma ação. De modo que as fórmulas de sua linguagem podem ser vistas como representando ações ou proposições. Ao interpretá-las como ações, cada conectivo lógico serve para combinar as ações de diferentes maneiras. Essa linguagem permite modelar a “execução real” de uma ação e por meio de conectivos modais modelar um procedimento que simula a execução de uma ação sem cometê-la de fato.

No capítulo três, apresentamos alguns exemplos simples para ilustrar como a lógica é utilizada para representar o conhecimento e agentes capazes de utilizar o conhecimento para solucionar problema. Apresentamos um exemplo de um agente executando uma tarefa simples; um agente procurando atingir por tentativa e erro um objetivo; uma agente empregando a heurística conhecida como análise dos meios e fins; e uma situação envolvendo o *frame problem* e uma solução que a lógica das transações permite.

Vamos apontar agora uma sugestão que acreditamos poderia ser realizada para tornar o sistema mais apropriado para representar a resolução de problemas. Essa sugestão envolve a noção de cognição distribuída. De acordo com Dutra, a noção de cognição distribuída se desenvolveu a partir da análise de situações sociais nas quais a solução de problemas requer compartilhamento e processamento coletivo de informações. Fazem parte de um sistema de cognição distribuída os agentes pertencentes a determinada equipe que esteja voltada para o cumprimento de uma tarefa, ou resolução de problema, bem como os equipamentos e dispositivos não humanos que essa equipe possui. Como a solução é alcançada apenas coletivamente e não pode ser alcançada apenas por um indivíduo isolado, então, tanto o processo cognitivo; as informações obtidas; a posse das informações prévias, conhecimento dos procedimentos prévios e as tarefas estão distribuídos. (DUTRA, 2018).

Para representar uma resolução de problemas distribuída é útil estender a lógica com um operador que represente a execução de duas ações em paralelo, de modo a simular agentes que trabalhem simultaneamente. Tal lógica foi desenvolvida por Bonner e Kifer em um trabalho do qual não analisamos nesta dissertação[11].

Vamos finalizar esta dissertação apontando alguns vieses e limitações presentes nesta pesquisa. Não podemos deixar de notar o quanto foi perdido ao passarmos da semântica para a teoria da prova. Perdemos bastante do poder expressivo e nos limitamos a exemplificar estados relacionais (conjuntos de atômicas), ao invés de teorias de primeira ordem arbitrárias. Acreditamos que para representar uma intenção, seria melhor atribuir a cada caminho uma estrutura modal e não uma estrutura de primeira ordem, ao mesmo tempo em que fosse permitido que oráculo de dados interpretasse cada estado por meio de uma teoria modal. Essas estruturas modais deveriam ser tais que houvesse um operador modal para representar a intenção de fazer algo. E determinarias as condições de satisfação de uma fórmula com esse operador de forma análoga a que foi feita com os conectivos

[11] (BONNER; KIFER, 1996).

“clássicos” por Bonner e Kifer. Uma fórmula modal será satisfeita em um caminho se e somente se ela for satisfeita na estrutura modal atribuída aquele caminho na interpretação. De qualquer forma a elaboração de uma teoria modal para a intenção é tópico para outro trabalho. Nosso objetivo aqui era investigar em que medida a lógica das transações é uma linguagem capaz de representar o conhecimento procedural e declarativo. Com isso não queremos dizer que a lógica das transações é absolutamente incapaz de representar o conhecimento procedural, nela podemos representar os conhecimentos declarativos e procedural de um agente, bem como a utilização deles. Entretanto, não podemos representar adequadamente expressões do tipo “João sabe que P” e “João tem a intenção de fazer P”, essa lacuna torna difícil representarmos problemas que envolvem o conhecimento sobre a vida mental de outros agentes.

Algumas questões importantes das quais abordamos superficialmente nessa dissertação precisam de uma tratamento mais apurado. Essas questões envolvem as contribuições do sujeito do conhecimento e do mundo na construção do conhecimento. No modelo proposto, um agente não acessa nenhum dado diretamente, primeiro porque eles são interpretados em termos de uma linguagem, e segundo, porque um agente sempre “vê” esses dados a partir de um esquema, de tal modo, que nesse processo não é possível distinguir claramente o que é um dado e o que é derivado das teorias do agente. Apesar dessas limitações, concluímos que a lógica das transações é um formalismo bastante útil e flexível para representar o conhecimento procedural e diversos aspectos resolução de problemas. Entretanto, como a adequação é uma questão de grau também oferecemos algumas sugestões que conjecturamos serviriam para desenvolver um sistema formal ainda mais apropriado para tal tarefa.

Referências

ANDERSON, J. R. *Psicologia cognitiva e suas implicações experimentais*. [S.l.]: LTC Ed., 2004. Citado na página 15.

BÉZIAU, J.-Y. Universal logic. *Logica*, v. 94, p. 73–93, 1994. Citado na página 2.

BONNER, A. J.; KIFER, M. *Transaction logic programming (or a logic of declarative and procedural knowledge)*. [S.l.], 1995. Citado 16 vezes nas páginas 3, 10, 36, 43, 46, 47, 56, 59, 61, 64, 65, 68, 69, 71, 73 e 74.

BONNER, A. J.; KIFER, M. Concurrency and communication in transaction logic. In: *JICSLP*. [S.l.: s.n.], 1996. p. 142–156. Citado 2 vezes nas páginas 39 e 78.

BONNER, A. J.; KIFER, M. Results on reasoning about updates in transaction logic. In: SPRINGER. *Workshop on (Trans) Actions and Change in Logic Programming and Deductive Databases*. [S.l.], 1997. p. 166–196. Citado na página 41.

DAVIDSON, D. *Essays on actions and events: Philosophical essays*. [S.l.]: Oxford University Press on Demand, 2001. v. 1. Citado na página 33.

DUTRA, L. d. A. *Autômatos geniais. A mente como sistema emergente e perspectivista*. [S.l.]: Brasília: Editora da UnB, 2018. ISBN 9788523012229. Citado na página 78.

GETTIER, E. L. Is justified true belief knowledge? *Analysis*, JSTOR, v. 23, n. 6, p. 121–123, 1963. Citado na página 27.

KATSUNO, H.; MENDELZON, A. O. On the difference between updating a knowledge base and revising it1. *Belief revision*, Cambridge University Press, v. 29, p. 183, 2003. Citado na página 46.

KOLMOGOROV, A. On the interpretation of intuitionistic logic. *Mathematische Zeitschrift*, v. 35, p. 58–65, 1932. Citado na página 4.

LUGER, G. F. *Inteligência Artificial. Tradução de Daniel Vieira*. [S.l.]: São Paulo: Pearson Education do Brasil, 2013. Citado 2 vezes nas páginas 3 e 22.

LUZ, A. M. Conhecimento e justificação: problemas de epistemologia contemporânea. *Dissertatio. Pelotas, RS*, 2013. Citado na página 23.

NILSSON, N. J. Problem-solving methods in. *Artificial Intelligence*, McGraw-Hill, 1971. Citado 3 vezes nas páginas 2, 3 e 4.

POLYA, G. *How to solve it 2nd edition*. [S.l.]: Princeton University Press, 1957. Citado na página 32.

REZK, M.; KIFER, M. Reasoning with actions in transaction logic. In: SPRINGER. *International Conference on Web Reasoning and Rule Systems*. [S.l.], 2011. p. 201–216. Citado na página 44.

SCHOENFELD, A. H. *Mathematical problem solving*. [S.l.]: Elsevier, 2014. Citado na página 3.

SEARLE, J. R. *Rationality in action*. [S.l.: s.n.], 2003. Citado na página 37.

SEARLE, J. R.; WILLIS, S. et al. *Intentionality: An essay in the philosophy of mind*. [S.l.]: Cambridge university press, 1983. Citado na página 33.

SIMON, H. A. The structure of ill-structured problems. In: *Models of discovery*. [S.l.]: Springer, 1977. p. 304–325. Citado na página 13.

VELOSO, P. A. Aspectos de uma teoria geral de problemas. 1984. Citado 2 vezes nas páginas 3 e 17.

WRIGHT, G. H. V. On the logic of norms and actions. In: *New studies in deontic logic*. [S.l.]: Springer, 1981. p. 3–35. Citado 2 vezes nas páginas 33 e 35.