

Utilização de lógica temporal-deontica para representação do comportamento de um agente racional no controle de um veículo autônomo

Vithor Tozetto Ferreira
 Universidade Tecnológica Federal do Paraná
 Av Monteiro Lobato, s/n - Km 04 CEP 84016-210
 Ponta Grossa, Brasil
 vithorferreira@alunos.utfpr.edu.br

Gleifer Vaz Alves
 Universidade Tecnológica Federal do Paraná
 Av Monteiro Lobato, s/n - Km 04 CEP 84016-210
 Ponta Grossa, Brasil
 gleifer@utfpr.edu.br

RESUMO

Os veículos autônomos (VA's) são uma nova tecnologia que pode se tornar realidade num futuro próximo. Esta tecnologia ainda gera preocupação para muitos, já que é difícil demonstrar o quão seguros e confiáveis esses veículos podem ser. Por esse motivo, é necessário desenvolver formas de assegurar o comportamento correto de tais veículos antes de permitir seu uso nas cidades. Considerando que um VA pode ser modelado por meio de um agente racional, é possível utilizar os operadores da lógica temporal-deontica para especificar propriedades de comportamento do agente responsável pelo controle de alto nível do VA. Neste trabalho é discutida a aplicação da lógica temporal-deontica para representar o comportamento de um agente racional em alguns cenários próprios de um VA.

Palavras-chave

Lógica Temporal-Deontica; Agentes; Veículos Autônomos

ABSTRACT

The autonomous vehicles (AVs) are a new technology which may become a reality in the near future. This technology still generates uneasiness for many people, since it is difficult to demonstrate how secure and reliable those vehicles can be. Therefore, it is necessary to develop means to assure the correct behavior of such vehicles before allowing their use in the cities. Considering that an AV can be modeled on a rational agent, it is possible to use the temporal-deontic logic to specify behavior properties of the agent responsible by the high level control of the AV. In this paper we discuss the application of temporal-deontic logic for representing the behavior of a rational agent in some typical AV scenarios.

Keywords

Temporal-deontic Logic; Agents; Autonomous Vehicles

1. INTRODUÇÃO

O grande avanço tecnológico dos últimos anos ocasionou o desenvolvimento de uma nova onda de tecnologia da informação, onde computadores estão presentes em quase todos os locais e se tornaram essenciais para a vida cotidiana. Diversas tarefas humanas estão sendo adaptadas para que um computador possa executá-las, facilitando e melhorando o funcionamento de diversos cenários. Essa substituição ainda gera certa desconfiança no público geral, em relação a quais tarefas realmente são seguras de se atribuir a um computador. Entre elas, a tarefa de controlar um veículo é um motivo de preocupação para muitos, devido ao grande foco que diversas empresas dão a essa tecnologia.

Uma das grandes preocupações em relação aos veículos autônomos (VA's) é a interação que eles têm com vidas humanas. Esse problema é uma questão ética, sobre como uma máquina irá reagir em certas situações, e sobre como interpretar as ações do VA. A discussão sobre esse tema geralmente é relacionada ao famoso dilema do bonde [6], onde existem dois caminhos, uma alavanca para desviar os trilhos e um bonde desenfreado. Em um desses caminhos, cinco pessoas estão posicionadas, e serão atingidas pelo bonde. No outro caminho, existe apenas uma pessoa. Se a alavanca for puxada, o caminho do bonde será desviado e ele atingirá apenas uma pessoa. Se nada for feito, as cinco pessoas no caminho principal serão atingidas. Qual das atitudes seria considerada a mais ética? Não agir, causando a morte de cinco pessoas, ou agir e ser responsável pela morte de uma pessoa?

Existe uma versão do dilema do bonde adaptada ao contexto de VA's, a *Moral Machine*, desenvolvida pelo MIT [1, 7]. Nesse teste existem diferentes cenários críticos, onde algum tipo de colisão envolvendo o VA é inevitável, e neste experimento cabe ao usuário tomar a decisão que julgue adequada. As opiniões sobre o que é ético nesse teste são divergentes, pois cada pessoa irá valorizar um aspecto diferente da situação antes de sua escolha. Em uma situação real de risco na condução de um veículo, o motorista irá agir impulsivamente, pois em muitas situações não há tempo para uma pessoa tomar uma decisão antes de agir.

O grande problema dessa discussão é que um VA não age impulsivamente. O VA segue sua programação, seu conjunto

de regras. Seria correto considerar que um acidente envolvendo VA's foi premeditado? Pois, o VA, baseado em suas regras, fez o que estava programado para fazer. A culpa é do programador? Essas questões devem ser tratadas de alguma forma antes dos VA's serem utilizados abertamente nas cidades.

Uma das formas de ver esse problema é que, a programação do VA deve estabelecer um conjunto de regras para o mesmo, e que o VA decidirá de forma autônoma sobre como proceder. Para estabelecer essas regras é necessário a utilização da lógica. A lógica formal permite a representação de diversos cenários, considerando possíveis comportamentos. Ao usar representações formais, o conjunto de regras pode ser testado e documentado, tornando explícita a forma que o VA irá reagir ao seu ambiente. Quando o comportamento do VA é conhecido, é possível estabelecer regras de segurança para seu uso, regulamentando a utilização dessa tecnologia e permitindo que ela seja acessível ao público.

Para se obedecer esse conjunto de regras, é necessário um controlador que interprete essas regras e as utilize na condução do VA. Esse é o papel do agente racional. Um agente é uma entidade que pode agir de forma autônoma, e que tem a capacidade de interpretar e agir no ambiente que está situado [10]. Um agente racional é capaz de raciocinar sobre si mesmo e o ambiente, e pode tomar decisões autônomas para atingir seus objetivos. O agente racional pode receber esse conjunto de regras, e as utilizará para tomar decisões sobre a condução do VA. Dessa forma, o VA terá um controlador capaz de interpretar as percepções do ambiente, decidir o melhor rumo das ações e agir eficientemente, enquanto respeita as regras estabelecidas na sua criação.

Para tentar assegurar que o mecanismo de controle de um VA irá agir conforme sua respectiva programação de alto nível é possível utilizar a verificação formal. A verificação formal serve para realizar testes no conjunto de regras formais que será aplicado no agente racional do VA. Esses testes percorrem os caminhos possíveis que as regras têm, garantindo que não existem falhas nessas regras que podem resultar em um evento indesejado.

Neste artigo são apresentados conceitos de agentes inteligentes e lógicas modais, em especial lógica: temporal e deôntica. O objetivo da apresentação desses conceitos é mostrar como essas lógicas podem ser utilizadas na representação adequada do comportamento de um agente racional na condução de um VA. Por sua vez, o trabalho aqui apresentado é parte de um projeto de pesquisa que tem como principal objetivo o desenvolvimento de soluções para controle de VA's por meio de agentes inteligentes e verificação formal. Consequentemente, almeja-se que em trabalhos futuros seja possível aplicar o formalismo de uma lógica **temporal-deôntica** na verificação formal do comportamento de agentes inteligentes.

Na continuação deste artigo, tem-se na Seção 2, uma explicação das características de um agente e do funcionamento do raciocínio de um agente racional. Após, a Seção 3 possui uma explicação sobre as divisões da lógica abordadas neste artigo, e seus operadores. Em seguida, na Seção 4 são demonstrados alguns exemplos da utilização dos operadores em argumentos lógicos. Finalmente, na Seção 5, são apresentadas as considerações finais em conjunto com as próximas etapas de realização do trabalho.

2. AGENTES

O termo agente possui diferentes significados dependendo de sua aplicação e do entendimento do autor. Conforme Wooldridge e Jennings [11], a forma mais geral de classificar um agente é como uma entidade situada em certo ambiente, seja esse ambiente virtual ou não, e que tal entidade pode executar ações autônomas em seu ambiente para atingir seus objetivos. Agentes também são relacionados às propriedades de autonomia, habilidade social, reatividade e proatividade. Existem diversas variações de agente, e explicar em detalhes cada uma delas não é o foco deste artigo.

Os agentes realmente interessantes para este artigo são os agentes racionais. Um agente racional é aquele que percebe seu ambiente através de sensores, que possui um modelo do ambiente, que pode raciocinar sobre si mesmo em relação ao ambiente e a outros agentes, e que, baseado em suas próprias conclusões, executa ações para modificar seu ambiente [8]. Um exemplo de agente racional seria uma pessoa qualquer. Por exemplo, se uma pessoa está saindo de casa, e percebe que o céu está cheio de nuvens escuras, a ação racional é levar um guarda-chuva [12]. Quando chover, a pessoa poderá se proteger, e sua ação de levar um guarda-chuva terá sido correta, evitando que a pessoa ficasse encharcada.

Entretanto, caso não chova, uma ação desnecessária foi executada, seria o ato de carregar um guarda-chuva incorreto? Se não chover, não há motivos para se carregar um guarda-chuva. Porém, observando esse cenário racionalmente, percebe-se que é melhor carregar um guarda-chuva como precaução do que fazer o contrário e acabar se molhando. Essa atribuição de racionalidade a um agente foca em balancear o comportamento reativo e proativo, balancear a percepção, deliberação e ação para evitar o desperdício de recursos, e permitir que agentes interajam socialmente compartilhando recursos enquanto equilibram seus interesses e os da comunidade [8].

Para se implementar um agente racional, é preciso interpretar as percepções dos sensores do agente e utilizá-los para o processamento e planejamento de ações. Neste artigo, destaca-se o modelo BDI. A sigla se refere as palavras *Belief, Desire and Intention*, ou Crenças, Desejos e Intenções. Esse modelo ajuda a dividir os aspectos do raciocínio de um agente, separando os conceitos de deliberação, percepção e ação, permitindo ao agente decidir o que deve ser feito, como deve ser feito, e então fazer. Nesse modelo, as Crenças são as informações que o agente possui sobre seu ambiente, obtidas através de seus sensores e outros meios de entrada de informação. Os Desejos e as Intenções influenciam na tomada de decisão e planejamento de ações do agente, já que esses atributos se referem aos objetivos do agente, sobre o que deve ser feito e como deve ser feito para atingir seus objetivos.

3. ASPECTOS LÓGICOS

Para definir as ações de um agente de forma abstrata, geralmente são utilizadas expressões que caracterizam atitudes humanas, como acreditar, achar, perceber, desejar. Descrever as ações de um agente com essas expressões facilita a compreensão de seu funcionamento e permite que o raciocínio do mesmo seja explicado de forma simples. Entretanto, existem maneiras formais de se explicar o comportamento de um agente, utilizando argumentos lógicos.

Esses argumentos lógicos vêm da lógica formal, utilizada para gerar proposições (livres de ambiguidades) que facilitam o entendimento de certas expressões, permitindo repre-

sentar formalmente sistemas de ações de diversas áreas, incluindo a computação. Em um agente racional, é importante a utilização da lógica modal, que contém diversos operadores para a representação da modalidade das ações, ou seja, indicar a forma que uma ação acontece ou a característica de tal ação. Os operadores característicos da lógica modal são: \Box e \Diamond , respectivamente “necessariamente” e “possivelmente”. Dentre as modalidades desta lógica, é necessário destacar aquelas de grande importância para a compreensão deste artigo. As subseções a seguir contêm explicações sobre a Lógica Temporal Linear e a Lógica Deontica, e também uma breve explicação sobre como utilizar esses operadores em uma ferramenta para verificação de agentes.

3.1 Lógica Temporal Linear

Existem diferentes tipos de lógica temporal. Aqui se descreve (brevemente) a Lógica Temporal Linear (LTL). Essa lógica possui operadores utilizados para a definição de ações em função de tempo, permitindo, por exemplo, definir argumentos que são verdadeiros em um período de tempo e falsos em outro [5].

A LTL é útil para um agente racional por permitir a ordenação de ações em um plano e por auxiliar na previsão de possíveis ações necessárias no futuro. Os operadores comuns dessa lógica são, considerando que φ e ψ representam eventos diferentes: $\bigcirc\varphi$, “próximo”, representa que φ é verdadeiro no próximo momento de tempo, em um sentido de ordenação de ações; $\bullet\psi$, “prévio”, representa que ψ é verdadeiro no momento anterior de tempo, em um sentido de ordenação de ações; $\Box\varphi$, representa que φ é sempre verdadeiro no futuro; $\Diamond\psi$, representa que ψ é verdadeiro em algum momento no futuro; $\varphi\mathcal{U}\psi$, “until”, representa que φ vai permanecer verdadeiro do momento atual até o momento onde ψ passa a ser verdadeiro; $\varphi\mathcal{W}\psi$, “while”, representa que φ irá permanecer verdadeiro enquanto ψ for verdadeiro.

3.2 Lógica Deontica

A Lógica Deontica é uma parte da lógica modal que lida com obrigações, permissões e proibições. Essa lógica trabalha com o dever de cometer ou não certo ato, e como obrigações e permissões influenciam no pensamento antes de agir. Os operadores dessa lógica atribuem conceitos como “você deve cometer” ou “você tem permissão para cometer” as ações [9].

O uso da lógica deontica foi motivado pela ideia de que seus operadores, em conjunto com os operadores já utilizados atualmente, podem auxiliar na previsão de novos comportamentos e reações que um agente pode apresentar em situações críticas. A LTL, já utilizada na verificação formal de agentes, permite o planejamento sequencial de ações. Com a aplicação da lógica deontica é possível identificar e reagir a situações onde esses planos podem não ser tão adequados, tornando o controle do agente sobre o VA mais dinâmico em situações críticas.

A aplicação desses conceitos em um agente racional permite a criação de regras que o agente deve seguir e de situações onde o agente terá permissão para infringir algumas dessas regras tentando evitar uma situação de risco. Os operadores dessa lógica são, considerando que φ representa uma ação qualquer: $O\varphi$, “obligation”, representa a obrigação de fazer φ ; $P\varphi$, “permission”, representa a permissão de fazer φ ; $F\varphi$, “forbidden”, representa a proibição de fazer φ .

3.3 Linguagem para especificação de propriedades

AJPF (*Agent Java PathFinder*) é uma ferramenta para verificação formal, através de *model checking*, para programas escritos em APLs (Agent Programming Languages) [3]. A arquitetura da ferramenta consiste do *model checker* JPF (*Java PathFinder*), de um conjunto de classes AJPF para avaliar o processo, *parsers*, tradutores, e de uma PSL (*Property Specification Language*). Um programa escrito em uma das APLs suportadas é, primeiramente, traduzido para sua representação AIL (*Agent Infrastructure Layer*) e embutido ao objeto de controle da AJPF. A representação AIL é criada a partir de um agente, levando em consideração: uma base de crenças, uma biblioteca de planos e um conjunto de intenções. Dessa forma, a representação AIL se torna um modelo do comportamento do agente, modelo que pode ser utilizado para *model checking*. Esse modelo é então testado conforme uma dada PSL. A PSL utilizada pelo AJPF se baseia em LTL, com a adição de modalidades para as características do agente (*e.g.*, objetivos, crenças, etc). Em [3], é definida uma gramática PSL, aqui (parcialmente) definida como segue:

$$\phi ::= \mid B_{ag} f \mid G_{ag} f \mid I_{ag} f \mid P(f) \mid \phi \wedge \phi \mid \phi U \phi \mid \Box \phi \mid \Diamond \phi$$

4. UTILIZAÇÃO DA LÓGICA TEMPORAL-DEONTICA

Conforme descrito na Introdução, o principal objetivo deste trabalho é evidenciar a utilização de uma lógica que mescle operadores da lógica temporal e deontica, para principalmente representar o comportamento de um agente racional, quando na condução de um VA.

Para esse fim, essa seção descreve alguns exemplos de tal utilização. Para auxiliar em tal representação, tem-se a descrição de um trabalho relacionado. Em [4], os autores apresentam a utilização da ferramenta de verificação formal, AJPF [3], a qual é aplicada para verificar formalmente os planos de um agente racional escritos na linguagem de programação de agentes GWENDOLEN [2]. Por meio da ferramenta AJPF [3] é possível escrever propriedades utilizando uma lógica temporal. As propriedades definidas via lógica temporal podem ser formalmente verificadas usando *Model Checking*. E assim assegurar que o comportamento do agente racional ocorre conforme especificado.

Os autores, em [4], descrevem diferentes cenários, dentre esses, um cenário onde uma colisão é inevitável. O que, de certa forma, faz uma sutil referência ao dilema do bonde (ver Seção 1). Especificamente, considera-se que quando um agente percebe a eminência de uma colisão, ele procura escolher a ação que representar menor dano (físico) ao veículo. Para tal, considera-se que no ambiente existem três diferentes níveis de danos físicos: leve, moderado e alto.

Considerando esses cenários foram escritas propriedades (por meio da lógica temporal) para especificar e validar o comportamento do agente. A partir dessas propriedades usando operadores temporais, neste trabalho são elaboradas novas propriedades fazendo uso, não apenas dos operadores da lógica temporal, mas igualmente de operadores deonticos, considerando que: $B(\varphi)$ é verdadeiro se o agente **acredita** que φ é verdadeiro; $G(\varphi)$ é verdadeiro se o agente tem como **objetivo** fazer φ ; $P(\varphi)$ é verdadeiro se o agente tem **permissão** para fazer φ ; $F(\varphi)$ é verdadeiro se o agente está **proibido** de fazer φ ; $Per(\varphi)$ é verdadeiro se o agente

percebe φ como verdadeiro.

Anteriormente, na Subseção 3.2 o operador de obrigação foi apresentado. Contudo, ele não será utilizado na descrição das propriedades que seguem. Isso deve-se ao fato de que, no momento, “obrigar” um agente a fazer uma ação foge do objetivo da aplicação dos outros operadores temporais-deônticos, que visam dinamizar e flexibilizar o comportamento do agente. O conceito de obrigação é muito rígido, pois simplesmente impossibilita que o agente realize certas ações, se tornando inadequado para a representação estabelecida neste trabalho.

No ambiente usado pelo agente criado em [4], considera-se que o agente pode mover-se nas direções norte, leste, oeste e sul. Assim, na especificação a seguir considera-se que há obstáculos nas direções leste, oeste e norte. E que o agente desloca-se vindo do sul, e que não há como ele retornar por essa direção. Sempre que o agente percebe obstáculos em todas as direções de desvio, o agente acredita que uma colisão é inevitável:

$$\Box(\text{Per}(\text{obstaculo_norte}) \wedge \text{Per}(\text{obstaculo_leste}) \wedge \text{Per}(\text{obstaculo_oeste})) \Rightarrow \text{B}(\text{colisao_inevitavel})$$

Quando o agente acredita que uma colisão é inevitável, seu próximo objetivo é selecionar um obstáculo para colidir:

$$\text{B}(\text{colisao_inevitavel}) \Rightarrow \bigcirc \text{G}(\text{selecionar_obstaculo})$$

Sempre que o agente acredita que uma colisão é inevitável e que existe um obstáculo que oferece risco de dano baixo, o agente tem permissão para colidir com o obstáculo de dano baixo:

$$\Box(\text{B}(\text{colisao_inevitavel}) \wedge \text{B}(\text{obstaculo_baixo_existe})) \Rightarrow \text{P}(\text{colisao_baixo})$$

Sempre que o agente acredita que uma colisão é inevitável e que não existe um obstáculo que oferece risco de dano baixo, mas acredita que existe um obstáculo que oferece risco de dano moderado, o agente tem permissão para colidir com o obstáculo de dano moderado:

$$\Box(\text{B}(\text{colisao_inevitavel}) \wedge \sim \text{B}(\text{obstaculo_baixo_existe}) \wedge \text{B}(\text{obstaculo_moderado_existe})) \Rightarrow \text{P}(\text{colisao_moderado})$$

A seguinte propriedade descreve que sempre que um agente perceber que o semáforo na via está com sinalização em vermelho, então o agente é proibido de seguir adiante na via.

$$\Box(\text{Per}(\text{senal_vermelho}) \Rightarrow \text{F}(\text{seguir_adiante}))$$

Contudo, em situação emergenciais pode ser necessário reescrever a propriedade acima, como segue:

$$\Box((\text{Per}(\text{senal_vermelho}) \wedge \text{B}(\text{situacao_emergencial}) \wedge \text{B}(\text{caminho_seguro})) \Rightarrow \text{P}(\text{seguir_adiante}))$$

Nesta segunda versão, a propriedade representa que sempre que um agente perceber o semáforo com sinalização vermelho, mas se o agente acreditar que há uma situação emergencial (e.g., o agente está conduzindo uma ambulância com paciente) e também acreditar que atravessar o sinal vermelho é seguro (o agente acredita que pode atravessar a via

sem atingir um obstáculo), então o agente tem permissão para seguir adiante na via.

5. CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado um estudo inicial para a aplicação e representação de operadores da lógica temporal-deôntica em especificações formais, com o objetivo de aumentar o poder de representação das especificações de um agente racional na condução de um VA em trabalhos futuros. Acredita-se que operadores da lógica deôntica podem acrescentar significado semântico relevante na representação de cenários, conforme previamente descrito na Seção 4.

Mas, naturalmente para atingir tal objetivo faz-se necessário delinear as seguintes metas, quanto a continuidade deste trabalho.

- Definição de lógica temporal-deôntica com a formalização dos operadores que serão necessários para a adequada representação.
- Incorporação de operadores deônticos na linguagem de especificação da ferramenta AJPF.
- Aplicação de especificações formais de um agente racional usando a nova representação com operadores deônticos.
- Comparações entre as diferentes especificações, temporal \times temporal-deôntica.

6. REFERÊNCIAS

- [1] J.-F. Bonnefon, A. Shariff, and I. Rahwan. The social dilemma of autonomous vehicles. *Science*, 352(6293):1573–1576, 2016.
- [2] L. A. Dennis and B. Farwer. Gwendolen: A bdi language for verifiable agents. In *AISB Convention, University of Aberdeen*, 2008.
- [3] L. A. Dennis, M. Fisher, M. P. Webster, and R. H. Bordini. Model Checking Agent Programming Languages. *Automated Software Engineering*, 19(1):5–63, Mar. 2012.
- [4] L. E. R. Fernandes, V. Custodio, G. V. Alves, and M. Fisher. A rational agent controlling an autonomous vehicle: Implementation and formal verification. *Electronic Proceedings in Theoretical Computer Science*, 257:35–42, sep 2017.
- [5] M. Fisher. Temporal representation and reasoning. *Foundations of Artificial Intelligence*, 3:513–550, 2008.
- [6] P. Foot. The problem of abortion and the doctrine of double effect. 1967.
- [7] I. Rahwan. Moral machine, <http://moralmachine.mit.edu/>, 2016.
- [8] A. S. Rao and M. Wooldridge. Foundations of rational agency. In *Foundations of rational agency*, pages 1–10. Springer, 1999.
- [9] G. H. Von Wright. Deontic logic. *Mind*, 60(237):1–15, 1951.
- [10] M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [11] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152, 1995.
- [12] M. J. Wooldridge. *Reasoning about rational agents*. MIT press, 2000.