

GUILHERME FIRMINO MENDES

**FILTRAGEM DE PERCEPÇÕES EM
AGENTES BASEADA EM
OBJETIVOS E NO MODELO DE
REVISÃO DE CRENÇAS
DATA-ORIENTED BELIEF REVISION
(DBR)**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná como requisito parcial para a obtenção do título de Mestre em Computação Aplicada.

Curitiba PR
Novembro de 2015

GUILHERME FIRMINO MENDES

**FILTRAGEM DE PERCEPÇÕES EM
AGENTES BASEADA EM
OBJETIVOS E NO MODELO DE
REVISÃO DE CRENÇAS
DATA-ORIENTED BELIEF REVISION
(DBR)**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná como requisito parcial para a obtenção do título de Mestre em Computação Aplicada.

Área de concentração: *Engenharia de Sistemas Computacionais*

Orientador: Cesar Augusto Tacla

Co-orientador: Gustavo Giménez Alberto Lugo

Curitiba PR
Novembro de 2015

Mendes, Guilherme Firmino

FILTRAGEM DE PERCEPÇÕES EM AGENTES BASEADA EM OBJETIVOS E NO MODELO DE REVISÃO DE CRENÇAS DATA-ORIENTED BELIEF REVISION (DBR). Curitiba, 2015. 108p.

Dissertação - Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Computação Aplicada.

1. Filtro de Percepções 2. Relevância de Percepções 3. Agentes BDI. I. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Computação Aplicada II-t.

*Esta folha deve ser substituída pela ata de defesa devidamente assinada,
que será fornecida pela secretaria do programa após a defesa.*

Aos meus amados filhos e esposa...

Agradecimentos

Agradeço à Thalita, minha esposa, pelo apoio incondicional e essencial para que esta jornada tivesse sucesso. Sem ela, este trabalho não existiria.

Agradeço aos meus filhos, que mesmo sendo muito novos para entender, foram obrigados a abrir mão de parte do convívio com seu pai para que o trabalho pudesse ser completado.

Agradeço ao meu orientador, que com muita paciência e dedicação, guiou-me para que este projeto chegasse ao seu final.

Agradeço aos membros da banca, suas críticas foram essenciais para um salto de qualidade do documento.

Resumo

Em cenários onde agentes inteligentes atuam e percebem dados de ambientes com cada vez mais informações, identificar somente as percepções relevantes aos objetivos pode ser crucial para que o ciclo de raciocínio do programa seja realizado em tempo hábil. Como uma solução a este problema, o presente trabalho criou um modelo de filtragem de percepções baseado no modelo DBR (*Data-oriented Belief Revision*) para ser aplicado em agentes BDI (*Belief Desire Intention*). Para isto, o trabalho estendeu e formalizou parte dos conceitos do modelo DBR tornando-o computacionalmente aplicável. Entre as contribuições deste trabalho estão a extensão da definição dos processos de Foco (seleção de dados percebidos) e Esquecimento de dados inativos; a definição e formalização de modelos de cálculo da Relevância de percepções, que permitem filtrar ou descartar dados em função dos planos do agente e de suas valorações de importância; e as definições dos modelos de armazenamento de Dados Inativos capazes de suportar diferentes cenários de utilização de agentes BDI. O resultado foi um filtro de percepções, genérico e automatizado, orientado aos objetivos de agentes BDI. Para operacionalizar o modelo criado, ele foi implementado na plataforma de desenvolvimento de agentes Jason. Análises empíricas foram realizadas para avaliar a corretude e os impactos relacionados ao tempo de processamento após a aplicação do modelo. Os experimentos realizados indicaram que o modelo de filtragem de percepções proposto neste trabalho contribui no desempenho computacional de agentes expostos a ambientes com muitos ruídos.

Palavras-chave: Filtro de Percepções, Relevância de Percepções, Agentes BDI.

Abstract

In scenarios where intelligent agents act and perceive data of environments with much information, identify only perceptions relevant to goals can be crucial for the agent reasoning cycle to be performed in time. As a solution to this problem, this work creates a model to filter perceptions based on DBR (Data-oriented Belief Revision) model to be applied at BDI (Belief Desire Intention) agents. In order to do it, this work has extended and formalized some of the DBR model concepts making it applicable in computer programs. Among this work contributions are the extension and definition of the processes Focus (selection of perceived data) and Oblivion of inactive data; definition and formalization of perception Relevance models, calculations that allow to filter or discard data based on agent plans and their importance values; definition of Inactive Data storage models able to support different usage scenarios of BDI agents. The result was a generic and automated perception filter oriented to the goals of BDI agents. To operationalize the model, it was implemented in the agent development platform Jason. Empirical analysis have been done to assess the correctness and identify the impact on the processing time after the model application. The results indicate that the perception filtering model proposed in this work contributes to the computational performance of agents exposed to environments with much noise.

Keywords: Perception Filtering, Perception Relevance, BDI Agents.

Lista de Figuras

2.1	Processo epistêmico no modelo DBR.	9
3.1	Escopo de trabalho no DBR.	17
3.2	Planos do cenário de demonstração dos Dados Inativos. Fonte: Autor	24
3.3	Planos do cenário de demonstração dos Dados Inativos, após execução.	25
3.4	Modelo DIP, conjuntos de informações durante a deliberação. Fonte: Autor	25
3.5	Modelo DIPP, conjuntos de informações durante a deliberação.	27
3.6	Modelo DIT, conjuntos de informações durante a deliberação.	28
4.1	Ciclo de interpretação so Jason com a inclusão do DBR.	32
4.2	Sintaxe em AgentSpeak para definição do valor objetivo.	35
4.3	Diagrama de classes do <i>Datum</i>	36
4.4	Diagrama de classes dos Dados Ativos e Inativos.	37
4.5	Diagrama de classes do agente DBR.	37
5.1	Gráfico (ruídos x tempo) comparativo dos tempos com N=1	46
5.2	Gráfico (ruídos x tempo) comparativo dos tempos com N=50	47
5.3	Gráfico (ruídos x tempo) comparativo dos tempos com N=100	48
5.4	Gráfico (ruídos x tempo) comparativo dos tempos com N=200	49
5.5	Gráfico (ruídos x $\frac{MO}{MF}$) comparativo das execuções.	51
A.1	Componentes BDI.	62
A.2	Ciclo de interpretação do Jason.	65

Lista de Tabelas

2.1	Axiomas da operação de contração do AGM.	6
2.2	Axiomas da operação de revisão do AGM.	7
2.3	Comparação DBR x AGM aplicado a agentes.	13
3.1	Aplicação dos modelos de cálculo de Relevância	21
3.2	Instanciação de um processo de Esquecimento.	29
5.1	Comparação dos tempos com N=1.	45
5.2	Comparação dos tempos com N=50.	46
5.3	Comparação dos tempos com N=100.	47
5.4	Comparação dos tempos com N=200.	48
5.5	Experimentos agrupados pela arquitetura com filtro	49
5.6	Experimentos agrupados pela arquitetura original	50
5.7	Comparação $\frac{MO}{MF}$ das execuções.	50
B.1	Lista dos tempos das execuções dos experimentos N=1 com filtro	76
B.2	Lista dos tempos das execuções dos experimentos N=50 com filtro	77
B.3	Lista dos tempos das execuções dos experimentos N=100 com filtro	78
B.4	Lista dos tempos das execuções dos experimentos N=200 com filtro	79
B.5	Lista dos tempos das execuções dos experimentos N=1 sem filtro	80
B.6	Lista dos tempos das execuções dos experimentos N=50 sem filtro	81
B.7	Lista dos tempos das execuções dos experimentos N=100 sem filtro	82
B.8	Lista dos tempos das execuções dos experimentos N=200 sem filtro	83

Lista de Abreviações

AGM	Alchourron, Gardenfors e Markinson
BDI	Belief Desire Intention
DBR	Data-oriented Belief Revision
DIP	Dados Inativos Permanentes
DIPP	Dados Inativos Parcialmente Permanentes
DIT	Dados Inativos Temporários

Sumário

Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Abreviações	xix
1 Introdução	1
1.1 Problema	1
1.2 Hipótese	1
1.3 Objetivos	2
1.4 Contribuição	3
1.5 Organização do documento	3
2 Revisão de Crenças em sistemas multiagentes	5
2.1 AGM	5
2.1.1 Funcionamento básico	6
2.1.2 Extensões e críticas	7
2.2 DBR	7
2.2.1 Isolamento da revisão de crenças	7
2.2.2 Dados e Crenças	8
2.2.3 Propriedades dos Dados	8
2.2.4 Processo epistêmico	9
2.2.5 Contradições	12
2.2.6 Princípios e parâmetros	12
2.3 AGM <i>versus</i> DBR	12
2.4 Conclusão	13
3 Filtro de percepções baseado no DBR aplicado a agentes BDI	15
3.1 Problemas em aberto e escopo do trabalho	15
3.2 Processo de Foco	16
3.2.1 Modelos de cálculo da Relevância	19
3.2.2 Dados Inativos com Relevância	22
3.3 Modelos de Dados Inativos	22

3.4	Processo de Esquecimento dos Dados Inativos	29
3.5	Conclusão	29
4	Aplicação do filtro de percepções na plataforma Jason	31
4.1	Impacto do filtro de percepções no Jason	31
4.2	Especialização do algoritmo para cálculo da Relevância	34
4.3	Extensão da estrutura de dados do Jason	35
4.4	Trabalhos correlatos aplicados ao Jason	38
4.4.1	Automatização da revisão de crenças em agentes Jason	38
4.4.2	Aplicação do AGM no Jason	39
4.4.3	Filtros de percepção no Jason	40
4.5	Conclusão	40
5	Experimentos	43
5.1	Descrição do problema sintético	43
5.1.1	Materiais de experimentação	44
5.1.2	Configuração do filtro	44
5.2	Experimentos	45
5.2.1	Experimentos com $N = 1$	45
5.2.2	Experimentos com $N = 50$	46
5.2.3	Experimentos com $N = 100$	47
5.2.4	Experimentos com $N = 200$	48
5.3	Análises dos resultados	49
5.4	Conclusão	52
6	Conclusão e trabalhos futuros	53
6.1	Conclusão	53
6.1.1	Implementação	55
6.1.2	Trabalhos futuros	55
A	Plataforma de desenvolvimento Jason	61
A.1	Agentes BDI e o interpretador AgentSpeak	61
A.2	Componentes de um agente no Jason	63
A.3	Ciclo de raciocínio de um agente no Jason	65
B	Informações sobre os experimentos	69
B.1	Planos dos agentes do problema sintético criado	69
B.1.1	Planos do agente $N=1$	69
B.1.2	Planos do agente $N=50$	69
B.1.3	Planos do agente $N=100$	70
B.1.4	Plano do agente $N=200$	72
B.2	Tempo das execuções dos experimentos	76
B.2.1	Modelo filtro de percepções com $N=1$	76
B.2.2	Modelo filtro de percepções com $N=50$	77
B.2.3	Modelo filtro de percepções com $N=100$	78
B.2.4	Modelo filtro de percepções com $N=200$	79
B.2.5	Modelo original com $N=1$	80

B.2.6	Modelo original com N=50	81
B.2.7	Modelo original com N=100	82
B.2.8	Modelo original com N=200	83

Capítulo 1

Introdução

1.1 Problema

O modelo BDI (Belief Desire Intention) de [Bratman, 1999] é a abordagem predominante para o desenvolvimento de agentes racionais, caracterizada por implementar a tríade "crenças, desejos e intenções" de estados mentais, ela usa destes conceitos para resolver problemas computacionais inspirada na cognição humana. Um agente BDI percebe o ambiente, raciocina considerando seu estado mental, delibera acerca da ação a ser executada e então atua sobre ele, repetindo o ciclo até que seus objetivos sejam alcançados ou até que ele desista.

Implementações do modelo BDI facilitam o desenvolvimento de agentes que necessitam responder rapidamente a mudanças no ambiente e têm objetivos a longo prazo. Robôs multifuncionais autônomos e que atuam em ambientes dinâmicos podem tirar proveito desta facilidade, utilizando o modelo BDI como implementação de seus controles. Porém, devido ao seu custo computacional quando aplicados em determinados cenários, implementações que utilizam modelos BDI são preteridas na área de robótica e nos sistemas de simulação.

Como exemplo destes cenários, estão os simuladores de futebol de robôs que, a partir do recebimento de muitas percepções, necessitam filtrá-las e priorizar suas ações, pois precisam de uma deliberação rápida para evitar a assincronia entre os estímulos do ambiente e o ciclo de raciocínio do agente. Outra instância do problema são Veículos Aéreos Não Tripulados (VANTs) com diversos objetivos que exploram e atuam em ambientes com muitos ruídos. Em um modelo BDI sem filtragem de percepções, estes ruídos são considerados no processo de deliberação e adicionam um custo computacional desnecessário. Este custo adicional afeta o tempo de autonomia do VANT, uma vez que o uso do processador consome a carga da bateria.

1.2 Hipótese

Em relação ao desempenho, um dos problemas do modelo BDI, é a falta de controle sobre as percepções. Um agente pode ser inundado por informações irrelevantes que podem causar uma sobrecarga de dados impedindo-o de executar raciocínios em tempo hábil em diversos tipos de problemas. Além de comprometer o desempenho computacional, esta falta de filtragem das percepções não é realística quando comparamos com a percepção humana. A atenção de um humano é limitada, não consegue captar todas as informações de um ambiente, ele tende a focar sua atenção de acordo com seus objetivos.

Devido a limitação de espaço para guardar informações e a necessidade de processar rapidamente suas tarefas, a seletividade de percepções observada no comportamento humano deve ser replicada em agentes BDI, esta seletividade deve ser intrínseca e transparente ao comportamento do agente, deve fazer parte do seu ciclo de raciocínio. Um modelo BDI com filtros eficientes e confiáveis de percepções, que considera a relevância do dado em relação aos objetivos do agente, pode viabilizar que agentes deliberativos possam ser aplicados em cenários onde o consumo e o tempo de processamento são críticos.

Para isto, o filtro de percepções deve: i) eliminar percepções irrelevantes; e ii) limitar a quantidade máxima de percepções que participarão da deliberação do agente. A eliminação de percepções irrelevantes reduz a quantidade de crenças e, por consequência, isto reduz o espaço de busca de outros processos do modelo BDI. A limitação da quantidade máxima de percepções também reduz o número de crenças, porém, neste caso, o comportamento do agente pode ser afetado, uma vez que crenças relevantes aos objetivos do agente deixam de ser consideradas em detrimento ao desempenho.

Para incorporar ao modelo BDI um filtro de percepções, este trabalho fundamenta-se em estudos de cognição orientada a objetivos e de dinâmica de crenças individuais realizadas por pesquisadores que atuam na interface de ciências cognitivas e tecnologias. [Paglieri and Castelfranchi, 2004] apresenta o modelo de revisão de crenças Data-Oriented Belief Revision (DBR) que, entre outras tarefas cognitivas, propõe processos que, em conjunto, podem viabilizar um filtro de percepções automatizado e genérico.

O modelo DBR é uma alternativa ao AGM (iniciais de Alchourron, Gardenfors e Markinson), apresentado em [Alchourron et al., 1985]. O modelo DBR, ao contrário do modelo AGM, provê uma forte conexão da revisão de crenças com outras tarefas cognitivas do agente, considerando-a como parte do processo epistêmico e não como uma atividade isolada. Porém, o modelo DBR ainda é bastante abstrato e, para que se torne aplicável, necessita de esforços na formalização e extensão de seus conceitos.

A informação, no modelo DBR, é separada em quatro estados mentais: Dados Inativos, Dados Ativos, Crenças Inativas e Crenças Ativas. O processo epistêmico do modelo é executado em oito passos: Atualização da Informação, Foco, Seleção de Crenças, Raciocínio, Mapeamento do Dado, Recuperação, Armazenamento e Esquecimento. No modelo DBR, os dados têm quatro propriedades epistêmicas: Relevância, Credibilidade, Importância e Atratividade. É baseado nos valores destas propriedades que os processos alteram a condição do dado, movimentando-o entre os estados mentais do agente. Por exemplo, se um dado tem Relevância suficiente, então o processo de Foco alterará sua condição de Dado Inativo para Dado Ativo.

Essa dissertação propõe que, aplicados ao modelo BDI, os novos estados mentais e os processos de Foco e Esquecimento propostos pelo DBR podem funcionar como um filtro de percepções confiável e genérico, permitindo que somente informações com algum grau de relevância em relação aos objetivos do agente sejam consideradas crenças. As informações não relevantes são tratadas como simples fatos observados, não devendo onerar o restante do ciclo de raciocínio do agente.

1.3 Objetivos

O objetivo desse trabalho é formalizar e estender parte do modelo DBR: os estados mentais, a propriedade (do dado) Relevância e os processos de Esquecimento e Foco ao ponto de torná-los aplicáveis em agentes que utilizam o modelo BDI. Fornecendo, deste modo, um

modelo de filtro de percepções genérico e automatizado para ser utilizado em agentes deliberativos.

Os objetivos específicos desse trabalho são:

1. Definir e identificar o impacto da inclusão do processo de Foco no ciclo de raciocínio de um agente BDI.
2. Definir e aplicar modelos de cálculo da Relevância dos dados.
3. Definir e identificar o impacto da inclusão do processo de Esquecimento no ciclo de raciocínio de um agente BDI.
4. Criar os novos estados mentais no ciclo de raciocínio de agentes BDI, adequando sua arquitetura com a diferença entre Dados e Crenças.
5. Analisar o ganho computacional do modelo em diferentes cenários.

1.4 Contribuição

Este trabalho contribui com a criação de um modelo de filtro de percepções orientado aos objetivos de agentes BDI. Para isto, é necessário formalizar o modelo DBR, mais que isto, é preciso estender parte de seus conceitos. Os processos de Foco e Esquecimento são formalizados, estendidos e adaptados à arquitetura BDI tornando-se aplicáveis computacionalmente.

Também são propostos três modelos de cálculos da Relevância do dado, criados a partir da formalização dos conceitos da qualidade e quantidade do dado. Além disto, são definidos três modelos de armazenamento de Dados Inativos. Os modelos propostos, em conjunto, são utilizados para configurar instâncias do filtro de percepção capazes de atuar em diferentes cenários.

Para operacionalizar e analisar o modelo de filtro de percepções aqui proposto, ele é aplicado ao Jason. O Jason é uma plataforma de desenvolvimento de agentes BDI, a qual é apresentada sucintamente no Apêndice A e detalhada em [Bordini et al., 2007]. Esta aplicação também gerou contribuições, já que a arquitetura da plataforma foi adaptada para comportar o filtro de percepções.

1.5 Organização do documento

No Capítulo 2 são explicados os modelos de revisão de crenças AGM e DBR, o primeiro é o trabalho fundamental sobre revisão de crenças, enquanto o segundo é uma alternativa ao primeiro, no qual fundamenta-se o presente trabalho. A partir dos conceitos abordados no Capítulo 2, o modelo de filtro de percepções proposto por esta dissertação é apresentado e formalizado no Capítulo 3, com sua formalização. No Capítulo 4 é tratada a integração do modelo proposto no Capítulo 3, utilizando o Jason, a plataforma de desenvolvimento de agentes BDI. No Capítulo 5 são apresentados experimentos baseado no modelo de filtragem de percepções definido nos capítulos anteriores. Finalmente, o Capítulo 6 apresenta as conclusões desse trabalho.

Capítulo 2

Revisão de Crenças em sistemas multiagentes

A revisão de crenças tem o objetivo de prover um modelo formal para que agentes cognitivos mudem o estado de suas "mentes", em outras palavras, um modelo para definir como os agentes tratarão novas crenças que contrapõem ou complementam crenças atuais.

Para exemplificar as consequências de uma nova crença, Pavlos Peppas em [van Harmelen et al., 2007] descreveu o seguinte drama: "Philippa, uma grega de 19 anos estudante da Universidade de Patras, descobriu que Nikos e Angela não são seus verdadeiros pais, ela foi adotada de um orfanato em São Paulo quando tinha seis meses de idade. A novidade chocou Philippa, muito do que ela acreditava significar sua vida estava errado. Depois de recuperar-se do choque inicial, ela começou a colocar seus pensamentos em ordem: então Alexandros não é seu primo biológico. Também não herdou os olhos castanhos (de quem ela acreditava ser) de sua avó e não precisa mais se preocupar em desenvolver pressão alta por causa do histórico desta doença em ambos os lados da família. Além do mais, provavelmente tenha irmãos biológicos morando no Brasil e, caso se aprofunde na busca, poderia conseguir a cidadania Brasileira, o que facilitaria a realização daquela longa viagem planejada para a América Latina".

Este foi um cenário típico de uma revisão de crenças: um agente racional recebe uma nova informação e isto o faz mudar suas crenças. No exemplo dado, a nova informação contradiz seu estado de crença inicial e o agente necessita descartar algumas de suas antigas crenças antes de acomodar crenças condizentes a nova informação. Ele também precisa aceitar as consequências que podem resultar da interação entre esta nova informação com as crenças remanescentes.

2.1 AGM

O estudo do processo de revisão de crenças começou da década de 1980, sendo que o artigo amplamente considerado como o marco de nascimento desta área de pesquisa é o trabalho seminal de Alchourron, Gardenfors e Markinson, [Alchourron et al., 1985], que até os dias de hoje continua sendo o modelo dominante de revisão de crenças, denominado de modelo AGM.

2.1.1 Funcionamento básico

No modelo AGM as crenças são representadas por sentenças de alguma linguagem formal. As crenças contidas em um agente são um conjunto (K) destas sentenças, sendo assumido que este conjunto é fechado sob consequência lógica, isto é, toda sentença ϕ que é consequência lógica de K também pertence a K .

A teoria AGM define três tipos de mudanças na base de crenças:

1. Expansão: dada uma crença p consistente com a base K , simplesmente se adiciona a crença p à K resultando em um conjunto de crenças fechado sob consequência lógica. O resultado da expansão de K pela adição de p é denotado por $K + p$.
2. Contração: remove uma crença p do conjunto de crenças e modifica K para que continue fechado sob consequência lógica. Para isto, deve retirar outras crenças de K . O resultado da contração de uma base de crenças K pela retirada da crença p é denotado por $K - p$.
3. Revisão: dada uma crença p inconsistente com a base K , adiciona-se p ao conjunto K , porém, para manter a consistência de K algumas sentenças preexistentes em K devem ser removidas. O resultado da revisão da base K pela adição de p é denotado por $K \diamond p$.

A operação de expansão é única ($K + p$), para as operações de contração e revisão existem seis axiomas básicos e dois axiomas complementares para cada uma das operações. As tabelas 2.1 e 2.2 listam os postulados considerando Cn o operador de consequência lógica; p e q sentenças lógicas.

Tabela 2.1: Axiomas da operação de contração do AGM.

Axiomas de contração básicos	
Fechamento	$(K - p) = Cn(K - p)$
Inclusão	$(K - p) \subseteq K$
Vacuidade	Se $p \notin Cn(K)$ então $(K - p) = K$
Sucesso	Se $p \notin Cn(\emptyset)$ então $p \notin (K - p)$
Extensão	Se $Cn(p) = Cn(q)$ então $(K - p) = (K - q)$
Recuperação	$K \subseteq (K - p) + p$,
Axiomas de contração complementares	
Sobreposição conjuntiva	$(K - p) \cap (K - q) \subseteq K - (p \wedge q)$
Inclusão conjuntiva	Se $p \notin K - (p \wedge q)$ então $K - (p \wedge q) \subseteq K - p$

Fonte: Adaptado de [Fermé and Hansson, 2011].

Tabela 2.2: Axiomas da operação de revisão do AGM.

Axiomas de revisão básicos	
Fechamento	$(K \diamond p) = Cn(K \diamond p)$
Inclusão	$(K \diamond p) \subseteq K + p$
Vacuidade	Se $\neg p \notin Cn(K)$ então $(K + p) = (K \diamond p)$
Sucesso	$p \in (K \diamond p)$
Extensão	Se $Cn(p) = Cn(q)$ então $(K \diamond p) = (K \diamond q)$
Consistência	Se $p \notin Cn(X)$ então $(K \diamond p) \notin Cn(X)$
Axiomas de revisão complementares	
Super-expansão	$K \diamond (p \wedge q) \subseteq (K \diamond p) + q$
Sub-expansão	Se $\neg q \notin (K \diamond p)$ então $(K \diamond p) + q \subseteq K \diamond (p \wedge q)$

Fonte: Adaptado de [Fermé and Hansson, 2011].

2.1.2 Extensões e críticas

Depois do trabalho seminal de [Alchourron et al., 1985], outros trabalhos surgiram propondo novos operadores (além dos básicos adição, contração e revisão) e extensões na representação do conjunto de crenças. O trabalho de [Fermé and Hansson, 2011] faz uma revisão destes trabalhos, fornece uma visão geral sobre as diferentes áreas de pesquisa conectadas ao modelo AGM e aponta os trabalhos que criticam determinados conceitos do modelo.

Um dos conceitos criticados, principalmente quando aplicado à agentes BDI, está relacionado à representação de seu conjunto de crenças: com fecho dedutivo, tamanho infinito e a inviabilidade de manter informações importantes sobre crenças não aceitas ou contraídas. [Paglieri and Castelfranchi, 2004] é um dos trabalhos que criticam esta representação do conjunto de crenças aplicada no modelo AGM, o trabalho é uma das origens do modelo DBR.

2.2 DBR

Os trabalhos de [Paglieri and Castelfranchi, 2004] e [Paglieri, 2004] esboçam um modelo em que a revisão de crenças é investigada como uma função específica de um processo cognitivo de estados epistêmicos, integrando modelos formais de revisão de crenças em uma teoria mais compreensiva, provendo conexões sistemáticas com as outras tarefas cognitivas, ou seja, removendo o isolamento da revisão de crenças com outras tarefas cognitivas. Nos trabalhos, os autores propuseram um modelo conceitual de revisões de crenças chamado modelo DBR (*Data-oriented Belief Revision*).

2.2.1 Isolamento da revisão de crenças

A revisão de crenças tem sido abordada como uma atividade isolada e desconexa com outras tarefas cognitivas: como o raciocínio para inferência, comunicação e argumentação. Para [Paglieri and Castelfranchi, 2004], apesar da grande variedade de modelos de revisão de

crenças, eles isolam esta tarefa do processo epistêmico de agentes cognitivos, tratando-a como um assunto separado.

Para [Paglieri, 2004], o problema de integração da revisão de crenças pode ser separado em duas partes: a primeira é a necessidade de se incluir a revisão de crenças no processamento epistêmico global do agente, investigando seus relacionamentos com outras informações internas do agente. A segunda é aprimorar o entendimento da relação da revisão de crenças com outras tarefas cognitivas, sejam individuais ou coletivas.

Existem trabalhos nas áreas de lógica como o de [Pollock and Gillies, 2000] e [Seegerberg, 1999]; inteligência artificial como [Galliers, 1992] e [Alechina and Logan, 2002]; e ciências cognitivas como [Paglieri and Castelfranchi, 2004] que visam superar este isolamento, ampliando a integração da revisão de crenças com outras tarefas cognitivas.

2.2.2 Dados e Crenças

O modelo DBR considera duas categorias básicas de informação: dados e crenças. A primeira corresponde a fatos capturados pelo agente. A segunda, um subconjunto da primeira acrescido das crenças inatas (as que já nascem com o agente), são informações que o agente considera como base confiável para ações, decisões e tarefas específicas de raciocínio, como previsões e explicações.

Um agente cognitivo pode ter conhecimento de um conjunto de dados que em um determinado momento não são importantes para o cumprimento de seus objetivos atuais. Porém, com a chegada de outros dados (ou o mesmo dado originado de uma fonte mais confiável), ele pode mudar seu entendimento em relação aquelas que inicialmente não foram consideradas.

O inverso também pode acontecer: uma crença pode ser recusada caso um novo dado prove-a incorreta ou uma fonte mais confiável forneça um dado mais plausível. Para isto, dados não transformados em crenças devem ser preservados por um modelo formal de revisão de crenças, que poderá utilizá-los para futuras revisões já que eles contém propriedades epistêmicas significantes como: o momento em que o dado foi percebido e razão para sua recusa.

A diferença entre dados e crenças pode parecer básica e evidente, porém é quase que completamente ignorada nos estudos de revisão de crenças, com poucas exceções como em [Boutilier, 1998]. O modelo original do AGM sequer cita dados, reduzindo o processo a somente o tratamento de crenças. Em oposição ao modelo AGM, o modelo DBR considera a diferenciação entre dados e crenças crucial para o entendimento de como a informação é processada no ciclo de raciocínio de um agente.

2.2.3 Propriedades dos Dados

No modelo DBR, os dados são aceitos ou rejeitados baseados em suas propriedades, ou seja, nas possíveis razões cognitivas para que se tornem crenças. Estas propriedades são definidas a seguir, seja ϕ um dado:

1. r^ϕ é a *Relevância* do dado, uma medida da utilidade pragmática do dado, como a quantidade e o valor dos objetivos que dependem daquele dado. Uma extensão deste conceito e uma abordagem mais formal é feita na seção 3.2.1
2. c^ϕ é a *Credibilidade* do dado, uma medida da quantidade e valores de todos os outros dados que suportam um novo dado, em contraste com os que o conflitam, sejam externos

ou internos. O trabalho de [Morveli-Espinoza et al., 2015] estende o conceito e aplica a Credibilidade em um modelo de tomada de decisão cooperativa para agentes inteligentes.

3. i^ϕ é a *Importância* do dado, uma medida da conectividade epistêmica, como a quantidade de outros dados e seus valores que o agente terá que revisar caso um novo dado seja aceito.
4. a^ϕ é a *Atratividade* do dado, uma medida do apelo motivacional, como a quantidade e os valores dos objetivos que são alcançados caso a agente passe a crer naquele dado.

2.2.4 Processo epistêmico

A Figura 2.1 mostra as categorias epistêmicas, os processos responsáveis pelo fluxo dos dados e as propriedades dos dados utilizados em cada processo.

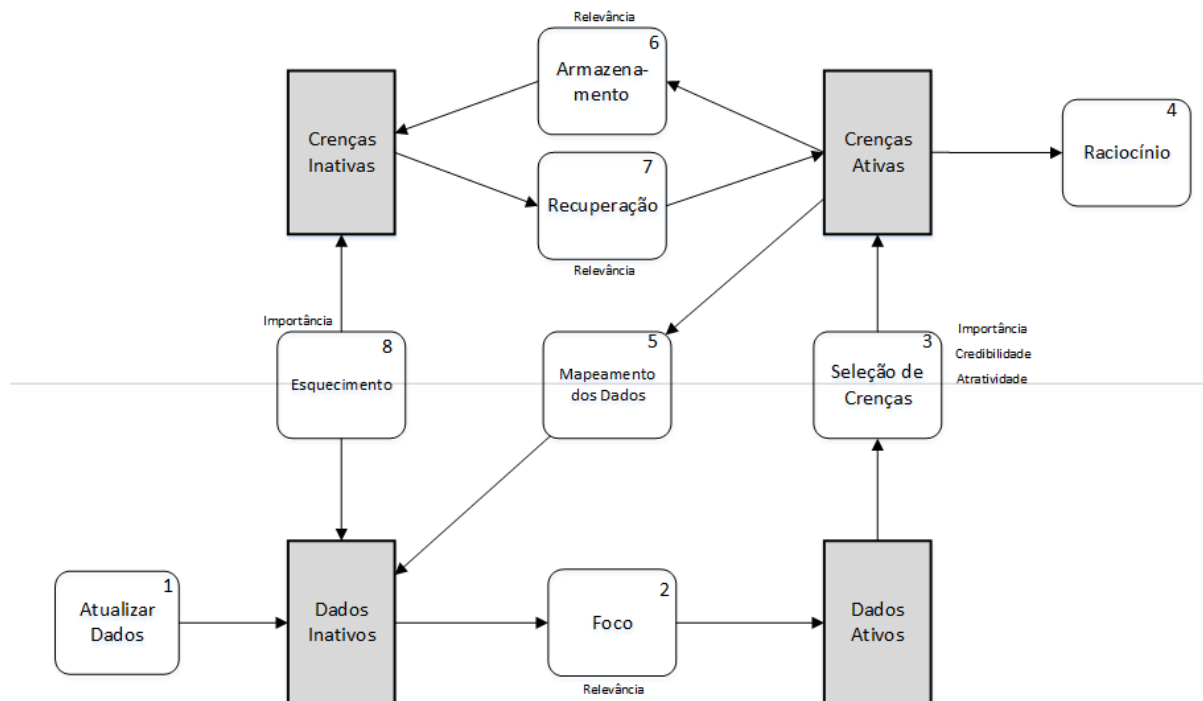


Figura 2.1: Processo epistêmico no modelo DBR.

Fonte: Adaptado de [Paglieri, 2004]

O modelo DBR diferencia dados e crenças em quatro categorias epistêmicas: Dados Inativos, Dados Ativos, Crenças Ativas e Crenças Inativas:

1. D_I representa os Dados Inativos, é o conjunto de dados ϕ que estão sujeitos ao processo de Foco. Os dados deste conjunto são percepções recentes, que ainda não passaram pelo processo de Foco ou percepções antigas, que já passaram pelo processo de Foco e não foram selecionadas como candidatas a crenças.
2. D_A representa os Dados Ativos, é o conjunto de dados ϕ com uma Relevância mínima para os objetivos atuais do agente, são dados que serão submetidos ao processo de Seleção de Crenças.

3. C_A representa as Crenças Ativas, é o conjunto de crenças que foram selecionadas pelo processo de Seleção de Crenças e que são utilizadas no processo de Raciocínio do agente. Uma Crença Ativa é um ϕ que tem r^ϕ , c^ϕ , i^ϕ e a^ϕ suficientes para ser considerada no ciclo atual de raciocínio do agente.
4. C_I representa as Crenças Inativas, são crenças que já tiveram certa Relevância para algum objetivo do agente e no momento atual não são necessárias para o Raciocínio. Porém, o agente pode se comprometer com um novo objetivo que dependa de uma Crença Inativa para ser atingido e ela pode ser reusada. Os dados deste conjunto não precisam submeter-se novamente ao processo de Seleção de Crenças, uma vez que já se provaram confiáveis.

As operações de Atualização da Informação, Foco, Seleção de Crenças, Mapeamento do Dado, Raciocínio, Recuperação, Armazenamento e Esquecimento são responsáveis pela pelas mudanças de categorias entre dados e crenças. O funcionamento resumido de cada item é descrito a seguir:

1. Atualização dos dados: O agente atualiza seus dados percebendo o ambiente ou se comunicando com outros agentes. Percepções são transformadas em estruturas de dados ϕ (que representam a visão que o agente tem do ambiente) e adicionadas ao conjunto D_I .
2. Foco: Este processo é aplicado em D_I , todos os elementos do conjunto têm seu r^ϕ calculado e podem ser transformados em D_A . Dados com r^ϕ insuficiente são mantidos em D_I . Seja $rel(x)$ a função que retorna o valor binário (verdadeiro ou falso) representando se ϕ tem Relevância para o agente (uma extensão do conceito e a formalização do processo de Foco é demonstrada na seção 3.2):

$$\forall \phi \in D_I, \text{ Se } rel(\phi) \text{ então } D_A \leftarrow D_A \cup \{\phi\} \wedge D_I \leftarrow D_I - \{\phi\} \quad (2.1)$$

O Foco também é defendido em [Alechina and Logan, 2010], pois evita custos computacionais desnecessários, já que a avaliação do dado e a Seleção da Crença não devem ser aplicadas em todos os dados disponíveis no agente.

3. Seleção de Crenças: Considerando os dados relevantes que estão no conjunto de Dados Ativos, este processo tem a função de selecioná-los e transformá-los em Crenças Ativas, de acordo com os atributos c^ϕ , i^ϕ e l^ϕ .

O modelo considera um sistema matemático para representar a Seleção de Crenças do agente, incluindo uma condição de confiança C , um limiar k e uma função de cálculo da força F . C e k juntos expressam o requerimento mínimo para que um dado seja selecionado como uma crença. A função F assinala um valor de força para as crenças aceitas. Tanto C como F são funções matemáticas tendo por argumentos uma combinação não vazia entre Credibilidade, Importância e Atratividade, porém os argumentos não necessitam serem idênticos para C e F . Considerando que f^ϕ representa a crença ϕ com força f , [Paglieri and Castelfranchi, 2004] propõem a seguinte formulação de Seleção de Crenças:

$$\text{Se } C(c^\phi, i^\phi, l^\phi) \leq k \text{ então } \phi \notin C_A \quad (2.2)$$

$$\text{Se } C(c^\phi, i^\phi, l^\phi) > k \text{ então } C_A \leftarrow C_A \cup \{\phi\} \text{ com } f^\phi = F(c^\phi, i^\phi, l^\phi) \quad (2.3)$$

As configurações de C , F e k são individuais e podem variar de acordo com o agente.

4. Raciocínio: As Crenças Ativas são utilizadas para a deliberação do agente. Após os processos de Foco e Seleção de Crenças, o agente tem em C_A uma base confiável para deliberar sobre seus objetivos e atuar no ambiente.
5. Mapeamento do Dado: Durante seu raciocínio, o agente pode criar novos dados, que serão adicionados ao conjunto D_I e serão considerados no próximo ciclo do processo epistêmico. Seja D' o conjunto dos novos dados criados pelo raciocínio do agente:

$$\forall \phi \in D', D_I \leftarrow D_I \cup \{\phi\} \quad (2.4)$$

6. Armazenamento: Depois da conclusão de um objetivo, um ϕ pertencente a C_A pode deixar de ser relevante para os objetivos correntes do agente, neste caso a Crença Ativa ϕ é retirada do conjunto e armazenada em C_I :

$$\text{Se } \neg rel(\phi) \text{ então } C_I \leftarrow C_I \cup \{\phi\} \wedge C_A \leftarrow C_A - \{\phi\} \quad (2.5)$$

7. Recuperação: É o processo inverso do Armazenamento, um ϕ pertencente a C_I pode voltar a ser Relevante para um novo objetivo do agente, neste caso a Crença Inativa ϕ volta para o conjunto C_A .

$$\text{Se } rel(\phi) \text{ então } C_A \leftarrow C_A \cup \{\phi\} \wedge C_I \leftarrow C_I - \{\phi\} \quad (2.6)$$

8. Esquecimento: É o processo de apagar a informação permanentemente, ela é removida e só pode ser recuperada se percebida novamente. A Importância e o tempo em que ela foi utilizada pela última vez são as propriedades que devem ser consideradas no esquecimento de uma informação. Seja $esq(x)$ a função que retorna o valor binário (verdadeiro ou falso) representando se ϕ pode ser esquecido pelo agente:

$$\forall \phi \in D_I, \text{ Se } esq(\phi) \text{ então } C_I \leftarrow C_I - \{\phi\} \quad (2.7)$$

$$\forall \phi \in C_I, \text{ Se } esq(\phi) \text{ então } D_I \leftarrow D_I - \{\phi\} \quad (2.8)$$

Uma extensão do conceito e a formalização do processo de Esquecimento dos Dados Inativos é apresentado na seção 3.4.

Os atributos Credibilidade, Importância e Atratividade, determinam o resultado do processo de Seleção de Crenças: a transformação de Dados Ativos em Crenças Ativas e a atribuição de uma força a esta crença; enquanto a Relevância é crucial no Foco, selecionando um conjunto de Dados Ativos. Enquanto Relevância e Atratividade dependem da comparação entre dados e objetivos, Credibilidade e Importância dependem basicamente das relações estruturais entre os dados.

2.2.5 Contradições

No modelo DBR, os estados epistêmicos são finitos e sem fecho dedutivo. Não existem garantias contra contradições, mas em vez disso, o modelo é capaz de capturar duas diferenças em relação à consistência: 1) contradições implícitas e explícitas, 2) contradições entre dados e entre crenças. Agentes são suscetíveis a manter certo número de inconsistências implícitas. Por exemplo, crenças que podem derivar em uma contradição, porém o agente ainda não necessita delas, e até que a contradição permaneça implícita, o agente não terá problemas com isto, basta ignorá-las.

Contradições só precisam ser tratadas caso cheguem ao nível de crenças, ou seja, se o processo de seleção aceitar dois dados contrastantes. Isto é raro, já que a credibilidade tem um papel importante na Seleção de Crenças, e dados contrastantes tem Credibilidades inversamente proporcionais. Por exemplo, o agente *ag1* recebe o dado p , via comunicação, do agente *ag2*, em seguida, *ag1* percebe (usando seus próprios sensores) o dado $\neg p$ no ambiente. Neste cenário *ag1* tende a dar mais credibilidade ao dado $\neg p$, pois foi obtido utilizando uma forma por ele considerada mais confiável, diminuindo o valor de r^p e descartando p durante o processo de Seleção de Crenças, vide a equação (2.2).

Porém, em certas circunstâncias, como um agente que enfatiza a Importância em vez da Credibilidade, pode acontecer que o processo de seleção escolha duas crenças contraditórias entre si. Neste caso, a contradição é resolvida através de raciocínio, como por exemplo, aplicando um axioma para rejeitar uma ou ambas as crenças contraditórias.

2.2.6 Princípios e parâmetros

O modelo distingue princípios e parâmetros: Princípios são naturalmente genéricos e qualitativos, definem as características comuns de processamento de qualquer agente. Parâmetros são individuais e quantitativos, especificando a forma na qual cada agente aplica os princípios universais do modelo.

A parte social e cognitiva do modelo DBR é representado por seus princípios, enquanto a variação individual é feita através da parametrização. Por exemplo, a diferenciação entre dados e crenças é um princípio universal, enquanto a instanciação do processo de Foco é uma parametrização, já que o cálculo da Relevância pode ser diferente entre agentes: um pode considerar somente a quantidade de objetivos dependentes de uma percepção, enquanto outro considera o quão importante é cada objetivo dependente da percepção.

2.3 AGM versus DBR

Inicialmente o modelo DBR diferencia a base de crenças da base de fatos. Para ele, é necessário que um agente tenha razões para transformar um dado em crença, este processo de seleção é onde acontece a integração da revisão de crenças com outras tarefas cognitivas do agente. Conexões que inexistem quando o modelo AGM é aplicado em agentes que raciocinam.

Enquanto o modelo AGM tem sua base crenças com fecho dedutivo e, consequentemente, possivelmente infinita, o modelo DBR mantém sua base finita e sem fecho dedutivo. Isto resolve dois problemas do AGM quando aplicados a agentes que raciocinam: um computacional e outro conceitual.

O problema computacional diz respeito ao tamanho da base de crenças e a complexidade de mantê-la logicamente fechada. Uma vez que todas as consequências lógicas de uma crença pertencem à base de crenças do agente, esta pode crescer exponencialmente enquanto revisada pelo modelo AGM. No modelo DBR o seu crescimento é linear, condicionado ao processo de seleção. Em regra, não existe custo computacional no cálculo das consequências lógicas da crença quando na expansão ou contração da base de crenças (as crenças são simplesmente adicionadas ou retiradas da base). A exceção acontece quando dados contraditórios se tornam crenças.

A falta de consistência prévia entre as crenças pode gerar contradições, porém, isto não é um problema para o agente enquanto a contradição permanecer implícita. Quando encontrada, ou seja, quando o agente realmente precisar utilizar as crenças contraditórias porque elas são necessárias para a realização de um objetivo, a inconsistência se tornará explícita e o modelo DBR a tratará pontualmente.

O problema conceitual do modelo AGM é relativo ao agente conhecer todas as consequências lógicas de uma nova crença adicionada em sua base, o que caracteriza uma situação irreal, uma vez que significa que o agente é logicamente onisciente.

Outra principal diferença entre os dois é o fato do modelo AGM descartar definitivamente crenças não utilizadas (contraídas) enquanto o modelo DBR as mantém na base de dados, uma vez que esta contém informações epistêmicas importantes relacionadas ao motivo que os dados não são selecionados para se tornarem crenças. Na tabela 2.3 é possível comparar as diferenças mais importantes entre os dois modelos.

Tabela 2.3: Comparação DBR x AGM aplicado a agentes.

DBR	AGM
Modelo conceitual complexo	Modelo conceitual simplificado
Fatos são diferentes de crenças	Fatos são crenças
Revisão de crenças conectada com outras tarefas cognitivas do agente	Revisão de crenças como atividade isolada
Base de crenças sem fecho dedutivo	Base de crenças com fecho dedutivo
Base de crenças finita	Base de crenças infinita
Base de crenças inconsistente	Base de crenças consistente
Seleção de crenças	Não existe seleção de crenças
Crenças são temporariamente rejeitadas	Crenças são definitivamente rejeitadas
Esquecimento de acordo com propriedades dos dados e crenças	Crenças são esquecidas nas operações de Contração e Redução

Fonte: Autor

2.4 Conclusão

Este capítulo apresentou brevemente o modelo AGM, seminal modelo de revisão de crenças que iniciou e permitiu a união dos estudos do tópico entre áreas de inteligência artificial, lógica e ciências cognitivas. O modelo também é responsável pela fundamentação do modelo DBR, uma vez que este nasceu como uma alternativa àquele.

Com o objetivo de aprofundar o entendimento do modelo DBR, preparando o modelo para uma maior contribuição (próximo capítulo), o presente trabalho proveu uma maior formalização (fórmulas 2.1 até 2.8) aos processos criados por [Paglieri and Castelfranchi, 2004] e [Paglieri, 2004]. O capítulo também apresentou uma comparação entre os dois modelos (outra contribuição do presente trabalho), identificando as diferenças mais importantes entre os modelos quando aplicados no ciclo de raciocínio de agentes cognitivos.

Para que seja possível explorar, analisar e mensurar as qualidades e defeitos do modelo DBR é necessário transformá-lo em um modelo computacionalmente aplicável. Para isto, é necessária uma maior extensão dos seus conceitos: formalização computacional de seus processos, estruturas e atributos. Assim como é necessário sua adequação no modelo BDI, alterando a lógica e o fluxo de seus processos. Parte disto é feito no próximo capítulo, com a criação do filtro de percepções orientado aos objetivos do agente, fundamentado no modelo DBR.

Capítulo 3

Filtro de percepções baseado no DBR aplicado a agentes BDI

Considerando as características do processo epistêmico Foco e da propriedade Relevância, o presente trabalho defende que eles podem funcionar como um filtro de percepções orientado aos objetivos do agente. Isto porque a Relevância é uma função matemática que considera a relação do dado percebido com os objetivos do agente. Enquanto o Foco é o procedimento computacional executado antes da formação da base de crenças que, dependendo do valor retornado pelo cálculo da Relevância, considera ou descarta uma percepção.

Até aqui, os trabalhos referentes ao modelo DBR são bastante conceituais, o presente trabalho é a primeira tentativa de tornar a parte do modelo relacionada com a filtragem de percepções computacionalmente aplicável. Para isto, são necessárias contribuições para refinar os conceitos já existentes (processo de Foco e Esquecimento) e novas contribuições em aspectos não explorados pelos trabalhos até aqui (diferentes modelos de cálculo de Relevância e de armazenamento de Dados Inativos).

3.1 Problemas em aberto e escopo do trabalho

A aplicação do modelo DBR na arquitetura BDI abre um leque de problemas a serem explorados quando incorporados ao ciclo de raciocínio de um agente, os mais evidentes são listados a seguir:

1. Componentes estruturais:
 - (a) Estruturas de dados que representem o dado e suas propriedades
 - (b) Separação entre Dados e Crenças
2. Processos relacionados com as percepções:
 - (a) Processo do Foco (filtro de percepções orientada aos objetivos)
 - (b) Modelos para determinar a Relevância
 - (c) Processo de Esquecimento dos Dados Inativos
3. Processos relacionados com a revisão de crenças:

- (a) Processo de Seleção de Crenças
- (b) Modelos para determinar a Credibilidade
- (c) Modelos para determinar a Importância
- (d) Modelos para determinar a Atratividade
- (e) Modelos de tratamento de inconsistências na base de crenças
- (f) Processo de Esquecimento dos Crenças Inativas

Os componentes do modelo DBR a serem estendidos e formalizados para a criação do modelo de filtro de percepções são os componentes estruturais (representação de dados, a segregação de dados e crenças e seus impactos no modelo BDI) e processos relacionados com as percepções (inclusão dos processos de Foco e Esquecimento no raciocínio do agente). A propriedade Relevância, por ser fundamental para o processo de Foco, também está no escopo do presente trabalho. Os demais processos do modelo DBR não fazem parte do escopo da modelagem do filtro de percepções proposto pelo presente trabalho.

Os itens do modelo DBR formalizados e estendidos por esse trabalho podem ser identificados pelos retângulos sem as bordas pontilhadas na Figura 3.1. A atualização dos dados e o raciocínio são processos já existentes no modelo BDI, os processos de Foco e Esquecimento são novos e alteram consideravelmente o ciclo de raciocínio do agente. Os estados mentais Dados Inativos e Dados Ativos também são novidades nos modelos BDI.

Na sequência deste capítulo, para definir o modelo de filtro de percepções, será formalizada a separação entre Dados e Crenças, então serão definidos a fórmula e o algoritmo que correspondem ao funcionamento do processo de Foco aplicado na arquitetura BDI. Modelos de cálculo serão propostos e formalizados para que forneçam o valor de Relevância do dado para o processo de Foco. Modelos de armazenamento de Dados Inativos serão propostos para atender diferentes cenários de atuação dos agentes e, finalmente, o processo de Esquecimento será definido e formalizado para tratar dos dados não relevantes ao agente.

3.2 Processo de Foco

O processo de Foco sugerido por [Paglieri and Castelfranchi, 2004] depende do cálculo da Relevância do dado percebido. Este atributo é medido considerando a *quantidade* e uma *valoração*¹ dos objetivos impactados se o dado for considerado no ciclo de raciocínio. Se um dado é relevante para o agente, então ele é considerado como um Dado Ativo, caso contrário, é considerado um Dado Inativo. Um Dado Ativo se tornará uma crença caso tenha f^ϕ suficiente para ser aceito pelo processo de Seleção de Crenças. A definição de [Paglieri and Castelfranchi, 2004] para o processo de Foco é superficial, para que seja possível a criação de agentes práticos utilizando o modelo DBR, o presente trabalho estende a definição e a formaliza. Sejam:

1. ϕ o dado recebido pelo agente (seja por comunicação, percepção do ambiente ou adicionado pela execução de um plano).
2. r^ϕ o valor da Relevância de ϕ .

¹Valoração tem a conotação de emitir juízo sobre algo - no caso, atribuir um valor 'subjetivo' ao objetivo

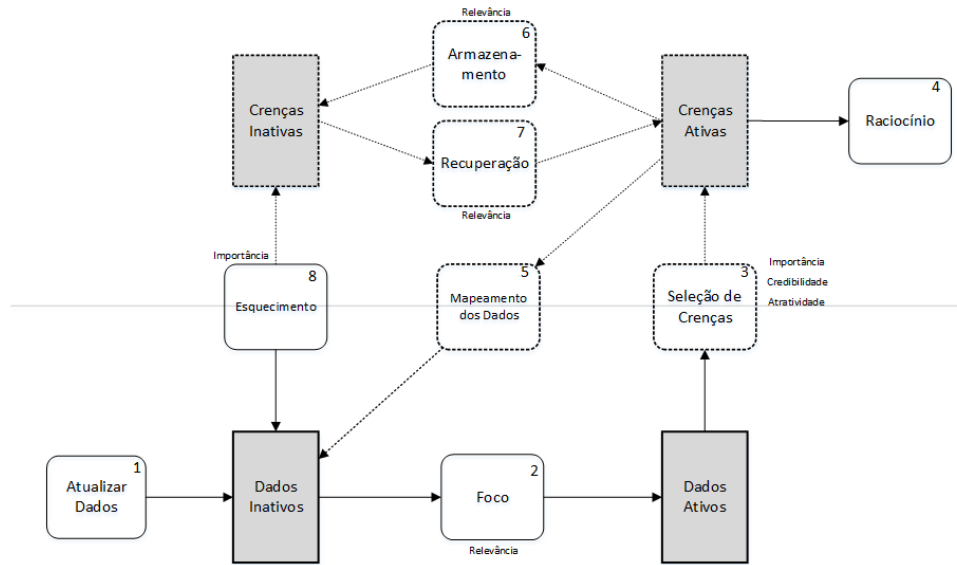


Figura 3.1: Escopo de trabalho no DBR.

Fonte: Adaptado de [Paglieri, 2004]

3. l o valor mínimo considerado para que um dado seja relevante para o agente.
4. D_I o conjunto de Dados Inativos, com $r^\phi < l$.
5. D_A o conjunto de Dados Ativos, com $r^\phi \geq l$.
6. P é o conjunto de planos correntes do agente.

Dados Ativos, Dados Inativos, Crenças Ativas e Crenças Inativas são categorias epistêmicas mutuamente exclusivas. A teoria abaixo (fórmulas 3.1 até 3.9) é usada para distinguir os possíveis estados epistêmicos de um dado recebido pelo agente:

Predicados: $\{dado_inativo/1, dado_ativo/1, crenca_ativa/1, crenca_inativa\}$

Axiomas:

1. Um dado inativo não é um dado ativo:

$$\forall \phi (dado_inativo(\phi) \Rightarrow \neg dado_ativo(\phi)) \quad (3.1)$$

2. Um dado ativo não é um dado inativo:

$$\forall \phi (dado_ativo(\phi) \Rightarrow \neg dado_inativo(\phi)) \quad (3.2)$$

3. Uma crença inativa não é uma crença ativa:

$$\forall \phi (crenca_inativa(\phi) \Rightarrow \neg crenca_ativa(\phi)) \quad (3.3)$$

4. Uma crença ativa não é uma crença inativa:

$$\forall \phi (crenca_ativa(\phi) \Rightarrow \neg crenca_inativa(\phi)) \quad (3.4)$$

5. Um dado não é uma crença:

$$\forall \phi (dado_inativo(\phi) \vee dado_ativo(\phi) \Rightarrow \neg crenca_ativa(\phi) \wedge \neg crenca_inativa(\phi)) \quad (3.5)$$

6. Uma crença não é um dado:

$$\forall \phi (crenca_inativa(\phi) \vee crenca_ativa(\phi) \Rightarrow \neg dado_inativo(\phi) \wedge \neg dado_ativo(\phi)) \quad (3.6)$$

7. Um dado que está em utilização no processo de Foco é um dado existente no conjunto de Dados Inativos:

$$\forall \phi (foco(\phi)) \Rightarrow dado_inativo(\phi) \quad (3.7)$$

8. Um dado que está em utilização no processo de Seleção de Crenças é um dado existente no conjunto de Dados Ativos:

$$\forall \phi (selecao(\phi)) \Rightarrow dado_ativo(\phi) \quad (3.8)$$

9. Um dado que está em utilização no processo de Esquecimento é um dado existente no conjunto de Dados Inativos ou Crenças Inativas:

$$\forall \phi (esquecimento(\phi)) \Rightarrow dado_inativo(\phi) \vee crenca_inativa(\phi) \quad (3.9)$$

O processo de Foco é iniciado considerando o conjunto D_I do ciclo de raciocínio corrente do agente, cada elemento do conjunto tem sua r^ϕ calculada e comparada com l , caso r^ϕ seja menor que o limiar l , então ϕ é mantido em D_I , caso contrário, é adicionado em D_A e removido de D_I :

$$\forall \phi \in D_I, \text{ Se } r^\phi >= l, \text{ então } D_A \leftarrow D_A \cup \{\phi\} \wedge D_I \leftarrow D_I - \{\phi\} \quad (3.10)$$

O valor de l é um parâmetro definido para cada agente e demonstra o quanto um agente quer focar seu raciocínio, um l com um valor alto significa que o agente irá considerar um número restrito de percepções em cada ciclo de raciocínio, enquanto um l com valor zero significa que o agente irá considerar toda percepção como relevante. A seção 3.2.1 exemplifica como a Relevância interage com o l .

O Algoritmo 4 define o processo de Foco executado em cada iteração do ciclo de raciocínio do agente:

Algoritmo 1 $foco(l, P, D_I, D_A)$

Fonte: Autor.

```

1: for all  $\phi \in D_I$  do
2:    $r_\phi \leftarrow relevancia(P, \phi)$ 
3:   if  $r_\phi \geq l$  then
4:      $D_I \leftarrow D_I - \{\phi\}$ 
5:      $D_A \leftarrow D_A \cup \{\phi\}$ 
6:   end if
7: end for

```

A inclusão do processo de Foco modelo BDI, além de simular uma tarefa do comportamento humano, pode ajudar na diminuição do custo computacional do ciclo de raciocínio do agente. São dois os motivos para isto: 1) somente dados relevantes participarão do processo de revisão de crenças; 2) os processos de geração de opções e filtro de intenções do modelo BDI terão um espaço de busca reduzido, uma vez que a base de crenças contém somente dados relevantes.

A melhoria no custo computacional da inclusão do Foco não é um resultado absoluto, depende do problema a ser resolvido. Um agente que percebe muitos dados e poucos são relevantes aos seus objetivos tem um proveito do Foco muito maior que um agente cujo ambiente só proporciona informações relevantes. No segundo caso, o Foco tem um impacto negativo ao custo computacional, já que irá realizar uma tarefa desnecessária ao ciclo de raciocínio. Estes casos são analisados no Capítulo 5 deste trabalho.

3.2.1 Modelos de cálculo da Relevância

Para [Paglieri and Castelfranchi, 2004] e [Paglieri, 2004], a Relevância de um dado é medida considerando a quantidade e a valoração dos objetivos impactados se o dado for considerado no ciclo de raciocínio. A quantidade de objetivos é uma contagem do número de vezes que determinado dado aparece na biblioteca de planos do agente. A valoração do objetivo é um atributo qualitativo, determina a importância daquele plano em relação aos outros planos do agente.

A Relevância do dado é um número inteiro retornado de uma função que considera sua quantidade e sua valoração em relação aos planos do agente. Conforme definido a seguir:

1. ϕ o dado recebido pelo agente (seja por comunicação, percepção do ambiente ou adicionado pela execução de um plano).
2. r^ϕ o valor da Relevância de ϕ .
 - (a) q^ϕ é um atributo da Relevância, representa o número de planos dependentes² de ϕ
 - (b) V_ϕ é um atributo da Relevância, representa o conjunto de valores dos objetivos dos planos dependentes de ϕ
3. D_I o conjunto de Dados Inativos, com $r_\phi < l$.

²Para que um dado ϕ seja dependência de um plano p , significa que ϕ é necessário para que p seja executado. Ou seja, p somente é executado se $\phi \in C_A$

4. D_A o conjunto de Dados Ativos, com $r_\phi \geq l$.
5. p é um plano do agente.

O cálculo de r_ϕ é realizado em dois passos: 1) verifica-se se ϕ pode ser unificado logicamente com algum elemento de um plano; 2) caso positivo, q_ϕ é incrementado e o valor do objetivo do plano é adicionado no conjunto V_ϕ . O Algoritmo 2 apresenta a Relevância:

Algoritmo 2 relevancia(P, ϕ)

Fonte: Autor.

```

1: for all  $p \in P$  do
2:   if unificar( $\phi, p$ ) then
3:      $q^\phi \leftarrow q^\phi + 1$ 
4:      $v^\phi \leftarrow$  valor do objetivo
5:      $V_\phi \leftarrow V_\phi \cup v^\phi$ 
6:   end if
7: end for
8: return calcular( $V_\phi, q^\phi$ )

```

Após calculados q_ϕ e V_ϕ tem-se a informação sobre a quantidade e a valoração dos planos que dependem de ϕ , então é necessário aplicar um modelo de cálculo (*calcular*(V_ϕ, q^ϕ)) para definir o valor de r_ϕ . A escolha deste modelo depende do cenário de atuação do agente e do problema abordado, por exemplo, um agente pode considerar a qualidade dos planos relevantes em detrimento a quantidade de planos relevantes, pois assim garante que seus principais objetivos serão executados com antecedência. Abaixo, como exemplos, três modelos para este cálculo são propostos:

1. Maior valor em V_ϕ : Este modelo considera o maior valor do conjunto V_ϕ como sendo o valor da Relevância de ϕ , isto significa que o valor da Relevância do dado é tão grande quanto o maior valor de objetivo dos planos dependentes de ϕ :

$$r^\phi = \max_{v \in V_\phi} v \quad (3.11)$$

2. Somatório de V_ϕ : O valor da Relevância é o somatório do conjunto de valores de objetivos, o dado que é uma dependência de múltiplos planos tem seu valor de Relevância igual a soma de todos os valores de objetivo dos planos afetados.

$$r^\phi = \sum_{v \in V_\phi} v \quad (3.12)$$

3. Quantidade de r^ϕ : Neste modelo, o valor da Relevância é a quantidade de objetivos dependentes do dado. O dado que é uma dependência de múltiplos planos tem seu valor de Relevância igual a quantidade de planos afetados.

$$r^\phi = q^\phi \quad (3.13)$$

O cenário a seguir tem o objetivo de evidenciar a diferença entre os modelos de cálculo de r^ϕ propostos: Em um determinado ciclo de raciocínio do agente, tem-se o seguinte estado:

1. Limite mínimo do valor da Relevância, $l = 4$
2. Dados Inativos, $D_I = \{d1, d2, d3, d4, d5, d6, d7\}$
3. Planos do agente ³,
 $P =$

1	@plano1[valor_objetivo(5)] +gatilho : d1 <- acao1.
2	@plano2[valor_objetivo(4)] +gatilho : d2 <- acao2.
3	@plano3[valor_objetivo(3)] +gatilho : d2 & d3 & d4 & d5 <- acao3.
4	@plano4[valor_objetivo(2)] +gatilho : d3 & d4 <- acao4.
5	@plano5[valor_objetivo(1)] +gatilho : d2 & d3 & d5 <- acao5.
6	@plano6[valor_objetivo(1)] +gatilho : d2 & d3 & d6 <- acao6.

Considerando este cenário, aplica-se o Algoritmo 1 (Foco), a instanciação das variáveis após o término do algoritmo é representada na Tabela 3.1, M1 representa o modelo "Maior valor em V_ϕ ", M2 representa o modelo "Somatório de V_ϕ " e M3 representa o modelo "Quantidade de r^ϕ ". Em cinza, na tabela, as relevâncias superiores ao limite mínimo de 4.

Tabela 3.1: Aplicação dos modelos de cálculo de Relevância

ϕ	V_ϕ	q^ϕ	r^ϕ em M1	r^ϕ em M2	r^ϕ em M3
d1	{5}	1	5	5	1
d2	{4,3,1,1}	4	4	9	4
d3	{3,2,1,1}	4	3	7	4
d4	{3,2}	2	3	5	2
d5	{3,1}	2	3	4	2
d6	{1}	1	1	1	1
d7	{}	0	0	0	0

Fonte: Autor

O dado $d2$, por exemplo, é dependência dos planos *plano2*, *plano3*, *plano5* e *plano6*. No modelo M1 sua Relevância é igual a 5, pois é o maior valor do conjunto V_ϕ . No modelo M2, sua Relevância é a soma do conjunto V_ϕ , ou seja, 9. No modelo M3, $d2$ tem Relevância 4, resultado da quantidade de planos dependentes deste dado. Os dados $d1$, $d3$, $d4$, $d5$ e $d6$ seguem os mesmos padrões de cálculos para os três modelos. Por não ter planos dependentes, o dado $d7$ não tem Relevância para o agente em nenhum dos modelos.

Considerando o limite l do cenário e aplicando o modelo M1, somente os dados $d1$ e $d2$ são relevantes, já que seus r^ϕ são maiores ou iguais ao l imposto. Considerando o modelo

³Nos quais: *gatilho* representa um evento (uma percepção ou uma comunicação de outro agente) que causa o disparo do plano. As sentenças $di...dn$ representam uma condição lógica de disparo do plano, são crenças presentes na base D_A . As fórmulas *acaoi* representam as ações executadas nos planos. O texto *@planoi* representa o nome do plano e [valor_objetivo(n)] sua valoração v^ϕ .

M2, os dados $d1$, $d2$, $d3$, $d4$ e $d5$ tem Relevância para o agente. Para o modelo M3, somente os dados $d2$ e $d3$ tem Relevância suficiente para se tornarem Dados Ativos.

O modelo M1 privilegia o valor dos objetivos do agente. O dado $d1$ é dependência de apenas um dos planos do agente, porém é aquele com a maior valoração e, conseqüentemente, com a maior Relevância quando se utiliza o modelo M1. Agentes que, em algum momento do seu ciclo de raciocínio, necessitam priorizar a execução de determinados planos, podem utilizar este modelo para selecionar os dados que são dependências dos planos com as maiores valorações.

O modelo M2 tende a considerar tanto a valoração quanto a quantidade, podendo privilegiar dados que são dependências de muitos planos ($d2$, $d3$) àqueles que são dependências de planos com valores de objetivos mais altos ($d1$). Já o modelo M3 considera somente a quantidade de planos que dependem de um determinado dado (neste caso, $d2$ e $d3$ têm os maiores valores), desconsiderando a valoração (V_ϕ). Em cenários onde os planos do agente tem a mesma priorização (como em planos sem valoração), o modelo M3 é recomendado.

3.2.2 Dados Inativos com Relevância

Existem dois possíveis cenários onde dados com alguma Relevância são desconsiderados do processo de raciocínio corrente do agente. O primeiro é quando o dado não atinge o l mínimo para ser considerado um dado ativo (fórmula 3.10). O segundo é ao término do processo de Foco, quando dados com Relevância maior que l (Dados Ativos) estarão disponíveis para a Seleção de Crenças, o cálculo da condição C (fórmula 2.2) pode considerar que um Dado Ativo não tenha um valor C suficiente para se tornar crença. Desconsiderar crenças que tenham Relevância maior que zero pode influenciar o processo de raciocínio do agente.

Quando desconsiderado pelo processo de Foco ou de Seleção de Crenças, o dado volta a ser um Dado Inativo, e como o agente irá tratá-lo nos próximos ciclos de raciocínio dependerá do modelo de armazenamento de Dados Inativos escolhido: Dados Inativos Persistentes (DIP), Dados Inativos Parcialmente Persistentes (DIPP) ou Dados Inativos Temporários (DIT), detalhados na seção 3.3. Quando utilizado o modelo DIP, o dado terá novas chances de se tornar crença nos próximos ciclos de raciocínio do agente, pois será considerado novamente pelo processo de Foco, até seu Esquecimento. No caso da utilização dos modelos DIPP ou DIT, o dado não poderá mais se tornar uma crença, a não ser que seja re-percebido pelo agente.

3.3 Modelos de Dados Inativos

O modelo de armazenamento dos dados inativos pode afetar o ciclo de raciocínio do agente. Um Dado Inativo que permaneça no conjunto D_I mesmo que não tenha sido (re)percebido no ciclo atual pode alterar o processo de deliberação do agente. Isto porque este dado será considerado no processo de Foco até que seja esquecido, podendo se tornar uma crença que não foi percebida na iteração atual do ciclo.

Quando um agente opta por considerar verdadeira uma crença obtida em outros ciclos, sua base de crenças pode se tornar inconsistente com o estado atual do ambiente. Porém, existem cenários em que agentes não precisam manter esta característica, eles podem considerar percepções antigas. Nestes casos a utilização de um modelo que considera permanentemente os Dados Inativos pode ser utilizada. Em situações onde um novo plano é incluído (via comu-

nicação ou deliberação interna), os Dado Inativos do agente são recalculados e dados antes sem Relevância podem se tornar Dados Ativos.

A partir de suas características epistêmicas e do problema a ser resolvido, o programador do agente deve escolher o modelo de armazenamento de D_I . Esse trabalho propõem três modelos genéricos, que garantem o filtro de percepções de acordo com os objetivos do agente, para o tratamento de percepções não Relevantes:

1. Dados Inativos Persistentes (DIP): As informações são mantidas no conjunto de Dados Inativos, podendo ser consideradas pelo processo de Foco e só serão removidas com o processo de Esquecimento. Este modelo é correto para agentes que podem ou devem considerar dados anteriores (percebidos em ciclos anteriores) em seu processo de deliberação corrente. Também é útil para agentes cuja revisão de crenças utilize dados anteriores para tratar inconsistências da base de crença.
2. Dados Inativos Parcialmente Persistentes (DIPP): Os dados percebidos em ciclos anteriores e não (re)percebidos no ciclo corrente continuam fazendo parte do conjunto de Dados Inativos, porém só podem ser utilizados para consultas que não afetarão diretamente a deliberação do agente. Por exemplo, o agente pode utilizar um dado para resolver uma inconsistência da base de crenças. Neste modelo, um Dado Inativo não se tornará um Dado Ativo, ou seja, não será considerado no processo de Foco, a não ser que seja (re)percebido no ciclo de raciocínio corrente do agente.

Para [Paglieri and Castelfranchi, 2004], inconsistências na base de crenças são tratadas após o processo epistêmico de Seleção de Crenças, os modelos de armazenamento que mantêm dados no conjunto D_I podem servir como mais uma base (C_A , C_I são exemplos das outras bases) de consulta para o tratamento de inconsistências.

Um dado em D_I pode auxiliar na tomada de decisão de um agente que encontra inconsistências em sua base. Por exemplo, um determinado dado sem Relevância suficiente para se tornar crença pode servir de suporte à crença que gerou uma inconsistência em C_A . Contudo, o modelo original do DBR não prevê um processo para consulta dos dados em D_I a partir da Seleção de Crenças, este processo deve ser criado e adequado ao ciclo de raciocínio do DBR para, de fato, tornar D_I uma base auxiliar ao tratamento de inconsistências das crenças do agente.

3. Dados Inativos Temporários (DIT) : Os dados não Relevantes são desconsiderados ao final do ciclo de raciocínio corrente. O modelo é proposto para cenários onde o valor epistêmico do Dado Inativo é irrelevante em outros ciclos, uma vez que os dados são desconsiderados no mesmo ciclo que são percebidos. Outra aplicação deste modelo são em cenários onde o tempo de resposta da deliberação do agente é crítica ou os recursos de memória são escassos.

A figura 3.2 contém um plano de execução de um programa que é utilizado para demonstrar o funcionamento da aplicação de cada um dos modelos. O *goalX*, quando disparado, executa a ação *act1*, enquanto o *goalY*, utilizando uma ação interna, adiciona o plano *@plan1* na biblioteca de planos e em seguida o executa. O agente atua no ambiente em 3 ciclos: no primeiro os dados $d1, d2, d3$ e $d4$ são percebidos; no segundo os dados percebidos são $d5, d6$ e $d7$; e no terceiro ciclo nenhum dado é percebido do ambiente.

```

-----
@goalX
+d3 : d4
  <- act1.
-----

@goalY
+d5 : d6
  <- .add_plan("@goalZ +d2 : true <- act3.");
  act2.

```

Figura 3.2: Planos do cenário de demonstração dos Dados Inativos.

Fonte: Autor

A Figura 3.4 demonstra os estados dos conjuntos de informação quando o modelo DIP é utilizado para resolver o problema, a sequência dos acontecimentos de cada ciclo é descrita abaixo:

1. Primeiro ciclo: O agente percebe os dados $d1, d2, d3$ e $d4$. Como não existem objetivos relacionados com as percepções $d1$ e $d2$, o processo de Foco os considera Dados Inativos, enquanto $d3$ e $d4$ são considerados dados ativos que por sua vez são incluídos na base de crenças pelo processo de Seleção. Neste ciclo, $d3$ dispara o gatilho do evento de $@goalX$ e seu contexto é unificado porque $d4$ existe na base de crenças. A ação $act1$ é realizada no ambiente.
2. Segundo ciclo: Neste passo o agente percebe os dados $d5, d6$ e $d7$. O conjunto de Dados Inativos contém as percepções não Relevantes do passo anterior e a base de crenças contém as Relevantes. O processo de Foco adiciona $d7$ no conjunto de Dados Inativos. Por serem dependências do $goalY$ as percepções $d5$ e $d6$ são Relevantes, se tornam Dados Ativos e, na sequência, crenças. A base de crenças neste momento desconsidera $d3$ e $d4$ pois foram crenças percebidas no ciclo anterior. Neste ciclo o $goalY$ se torna uma nova intenção e sua execução cria $@goalZ$, adicionando um novo plano na biblioteca de planos e atua no ambiente executando a ação externa $act2$. A figura 3.3 demonstra a nova biblioteca de planos do agente depois da criação de $@goalZ$.
3. Terceiro ciclo: Não há percepções neste passo, o conjunto de Dados Inativos contém os dados não Relevantes dos passos anteriores, desta vez o processo de Foco passa a considerar $d2$ como uma crença Relevante, já que ela faz parte do gatilho de $@goalZ$, este dado se torna um Dado Ativo e, na sequência, uma crença, disparando $@goalZ$ que executa a ação externa $act3$.


```

-----
@goalX
+d3 : d4
  <- act1.

-----

@goalY
+d5 : d6
  <- .add_plan("@goalZ +d2 : true <- act3.");
  act2.

-----

@goalZ
+d2 : true
  <- act3.
  
```

Figura 3.3: Planos do cenário de demonstração dos Dados Inativos, após execução.

Fonte: Autor

		Conjunto de percepções	Conjunto de Dados Inativos	Conjunto de Dados Ativos	Base de Crenças
1º Ciclo	Perceber Ambiente	d1,d2, d3,d4			
	Foco		d1,d2	d3,d4	
	Seleção Crenças		d1,d2		d3,d4
2º Ciclo	Perceber Ambiente	d5,d6 d7	d1,d2		d3,d4
	Foco		d1,d2,d7	d5,d6	d3,d4
	Seleção Crenças		d1,d2,d7		d5,d6
3º Ciclo	Perceber Ambiente		d1,d2,d7		d5,d6
	Foco		d1,d7	d2	d5,d6
	Seleção Crenças		d1,d7		d2

Figura 3.4: Modelo DIP, conjuntos de informações durante a deliberação.

Fonte: Autor

Quando comparado com o modelo DIP, o processo de raciocínio (descrito abaixo) e os conjuntos de informações (Figura 3.5) do agente são diferentes quando o modelo DIPP é aplicado ao problema descrito:

1. Primeiro ciclo: Neste passo o agente tem o mesmo comportamento do modelo DIP.
2. Segundo ciclo: Neste passo o agente percebe os dados $d5, d6$ e $d7$. O conjunto de Dados Inativos contém as percepções não Relevantes do passo anterior e a base de crenças contém as Relevantes. O processo de Foco adiciona $d7$ no conjunto de Dados Inativos e marca $d1$ e $d2$, sinalizando-os como dados percebidos em outro ciclo. O restante do processo de raciocínio executado neste passo é igual ao descrito no modelo DIP.
3. Terceiro ciclo: Não há percepções neste passo, o conjunto de Dados Inativos contém os dados não Relevantes dos passos anteriores, porém, por se tratarem de percepções de ciclos anteriores, não os considera no processo de Foco corrente. Com isto, $d2$ não é transformada em Dado Ativo e não se torna crença, nunca disparando $@goalZ$ e, consequentemente, não executando $act3$.

	Conjunto de percepções	Conjunto de Dados Inativos	Conjunto de Dados Ativos	Base de Crenças
1º Ciclo	Perceber Ambiente	d1,d2, d3,d4		
	Foco		d1,d2	d3,d4
	Seleção Crenças		d1,d2	d3,d4
2º Ciclo	Perceber Ambiente	d5,d6 d7	d1,d2	d3,d4
	Foco		<u>d1,d2</u> ,d7	d5,d6
	Seleção Crenças		<u>d1,d2</u> ,d7	d5,d6
3º Ciclo	Perceber Ambiente		<u>d1,d2,d7</u>	d5,d6
	Foco		<u>d1,d2,d7</u>	d5,d6
	Seleção Crenças		<u>d1,d2,d7</u>	

Figura 3.5: Modelo DIPP, conjuntos de informações durante a deliberação.

Fonte: Autor

O modelo DIT difere do modelo DIPP na quantidade de informação armazenada (Figura 3.6) nos seus conjuntos de informações, mas seus processos de deliberação são idênticos:

1. Primeiro ciclo: Neste passo o agente que utiliza o DIT tem um comportamento similar ao dos agentes que utilizam DIP e DIPP. A diferença é que o conjunto de Dados Inativos é esvaziado após a Seleção de Crenças.
2. Segundo ciclo: O agente percebe os dados $d5, d6$ e $d7$. O conjunto de Dados Inativos está vazio e a base de crenças contém as percepções Relevantes do ciclo anterior. O restante do processo de raciocínio executado neste passo é similar ao descrito nos modelos DIP e DIPP, com a exceção que o DIT faz a limpeza do conjunto de Dados Inativos durante a Seleção de Crenças.
3. Terceiro ciclo: Como não há percepções e o conjunto de Dados Inativos está vazio, o processo de Foco não executa ação alguma e a Seleção de Crenças elimina as crenças

não percebidas. Com isto, seguindo o mesmo comportamento do modelo DIPP, *d2* não é transformada em Dado Ativo e não se torna crença, nunca disparando *@goalZ* e, conseqüentemente, não executando *act3*.

		Conjunto de percepções	Conjunto de Dados Inativos	Conjunto de Dados Ativos	Base de Crenças
1º Ciclo	Perceber Ambiente	d1,d2, d3,d4			
	Foco		d1,d2	d3,d4	
	Seleção Crenças				d3,d4
2º Ciclo	Perceber Ambiente	d5,d6, d7			d3,d4
	Foco		d7	d5,d6	d3,d4
	Seleção Crenças				d5,d6
3º Ciclo	Perceber Ambiente				d5,d6
	Foco				d5,d6
	Seleção Crenças				

Figura 3.6: Modelo DIT, conjuntos de informações durante a deliberação.

Fonte: Autor

Diante do cenário apresentado, o modelo DIP executou três ações (*act1*, *act2* e *act3*) no ambiente, enquanto os outros dois modelos executaram as duas primeiras (*act1* e *act2*). A quantidade de dados armazenadas consideradas no processo de Foco também é diferente para cada um dos três modelos. Estas diferenças entre os modelos pode influenciar tanto no custo computacional como na quantidade de informação útil para execução de outras tarefas cognitivas (como a revisão de crenças).

3.4 Processo de Esquecimento dos Dados Inativos

Análogo ao comportamento humano, agentes baseados no modelo também podem e (dependendo do cenário) devem esquecer uma informação percebida em um ciclo anterior de sua execução. Para isto, é necessário armazenar o *momento* em que a informação foi percebida e delimitar o período máximo de permanência do dado no conjunto de Dados Inativos. O Processo de Esquecimento pode ser definido como:

1. t^{atual} é o valor do momento atual.
2. t^ϕ é o valor do momento em que ϕ foi percebido.
3. m é o valor do período máximo que as percepções são mantidas na D_I

$$\forall \phi \in D_I, \text{ Se } t^\phi \leq t^{atual} - m \text{ então } D_I - \{\phi\} \quad (3.14)$$

O modelo de esquecimento aqui proposto utiliza a técnica de janelamento temporal, onde m pode ser fixo ou variável, dependendo da necessidade do agente. A unidade de medida de t^{atual} , t^ϕ e m deve ser comum aos três e pode ser uma unidade de tempo (segundos, minutos, horas) ou o valor sequencial do ciclo de raciocínio do agente.

A Tabela 3.2 demonstra o comportamento do algoritmo de esquecimento aplicado nas percepções $d1, d2, d3, d4$ e $d5$ ao longo de seis ciclos de raciocínios do agente utilizando o modelo de armazenamento DIP (conjunto D_I permanente). A coluna Percepções indica o conjunto de dados percebidos no ciclo atual e a coluna $Esquecimento(D_I)$ contém o resultado do processo de Esquecimento aplicado no D_I atual. A unidade de medida utilizada foi o valor sequencial do ciclo de raciocínio. O período de esquecimento m corresponde a dois ciclos.

Tabela 3.2: Instanciação de um processo de Esquecimento.

Ciclo	Percepções	D_I	t^{d1}	t^{d2}	t^{d3}	t^{d4}	t^{d5}	$t^{atual} - m$	Esquecimento(D_I)
1	{ $d1, d2$ }	{ $d1, d2$ }	1	1	-	-	-	$1 - 2 = -1$	{ $d1, d2$ }
2	{ $d3, d4$ }	{ $d1, d2, d3, d4$ }	1	1	2	2	-	$2 - 2 = 0$	{ $d1, d2, d3, d4$ }
3	{ $d5$ }	{ $d1, d2, d3, d4, d5$ }	1	1	2	2	3	$3 - 2 = 1$	{ $d3, d4, d5$ }
4	{ $d3$ }	{ $d3, d4, d5$ }	-	-	4	2	3	$4 - 2 = 2$	{ $d3, d5$ }
5	{}	{ $d3, d5$ }	-	-	4	-	3	$5 - 2 = 3$	{ $d3$ }
6	{}	{ $d3$ }	-	-	4	-	-	$6 - 2 = 4$	{}

Fonte: Autor

No modelo proposto, o processo de Esquecimento é executado antes do Foco, então, por exemplo, no terceiro ciclo da Tabela 3.2, o Foco tem disponíveis os dados $d3, d4$ e $d5$, já que os dados $d1$ e $d2$ foram esquecidos de acordo com a Fórmula 3.14.

Quando um dado é percebido e já existe em D_I , seu t^ϕ é atualizado com o valor de t^{atual} , como demonstrado no quarto ciclo do exemplo acima.

3.5 Conclusão

Este capítulo tratou das principais contribuições desta dissertação. Até aqui, o modelo DBR não tinha uma proposta suficientemente especificada para ser aplicada em agentes BDI. O

escopo da contribuição considerou os itens do modelo DBR que são essenciais para que o filtro de percepções possa ser modelado e aplicado em agentes práticos.

O primeiro item estendido foi o processo de Foco. Na seção, foi formalizado a diferença entre Dados e Crenças dentro do processo epistêmico do agente. Também foi definido o algoritmo do processo de Foco considerando as características do modelo BDI. O limitador l também é uma contribuição do presente trabalho, com ele é possível limitar a quantidade de crenças relevantes que continuam no ciclo epistêmico do agente.

A segunda contribuição está relacionada à propriedade Relevância. Seu conceito original foi estendido, com a formalização dos valores da qualidade e quantidade do dado. Três modelos para o cálculo da propriedade foram propostos, a serem considerados em diferentes cenários de atuação de um agente.

A terceira contribuição é referente ao tipo de armazenamento de Dados Inativos. De acordo com a necessidade do agente um dado deve ser persistido ou eliminado da base de Dados Inativos. Para comportar esses diferentes cenários, os modelos DIP, DIPP e DIT foram definidos e detalhados.

A quarta contribuição presente neste capítulo é a extensão do processo de Esquecimento de Dados Inativos. Foi proposto um algoritmo baseado em janelas temporais para tratar dados que não interessam mais ao agente.

Capítulo 4

Aplicação do filtro de percepções na plataforma Jason

A plataforma Jason, detalhada em [Bordini et al., 2007]), é apenas uma escolha operacional para mostrar a viabilidade de implementação do modelo de filtro de percepções proposto nesse trabalho. Outras plataformas similares baseadas em BDI também podem utilizar o modelo de filtragem de percepções proposto por este trabalho. Fundamentado nas contribuições apresentadas no capítulo anterior, este capítulo apresenta o impacto da inclusão do modelo de filtro de percepções na arquitetura da plataforma Jason, assim como propõe os algoritmos e estruturas para torna-lo viável.

4.1 Impacto do filtro de percepções no Jason

Na Figura 4.1, é possível visualizar todos os componentes do ciclo de interpretação de um agente construído no Jason que são impactados diretamente com a extensão do modelo DBR definida nesse trabalho. Em cinza, os novos componentes criados. Conjuntos são representados como retângulos, funções de seleção são representadas como losangos e círculos representam processos envolvidos na interpretação de programas Jason, como a revisão de crenças e a execução da intenção no ambiente.

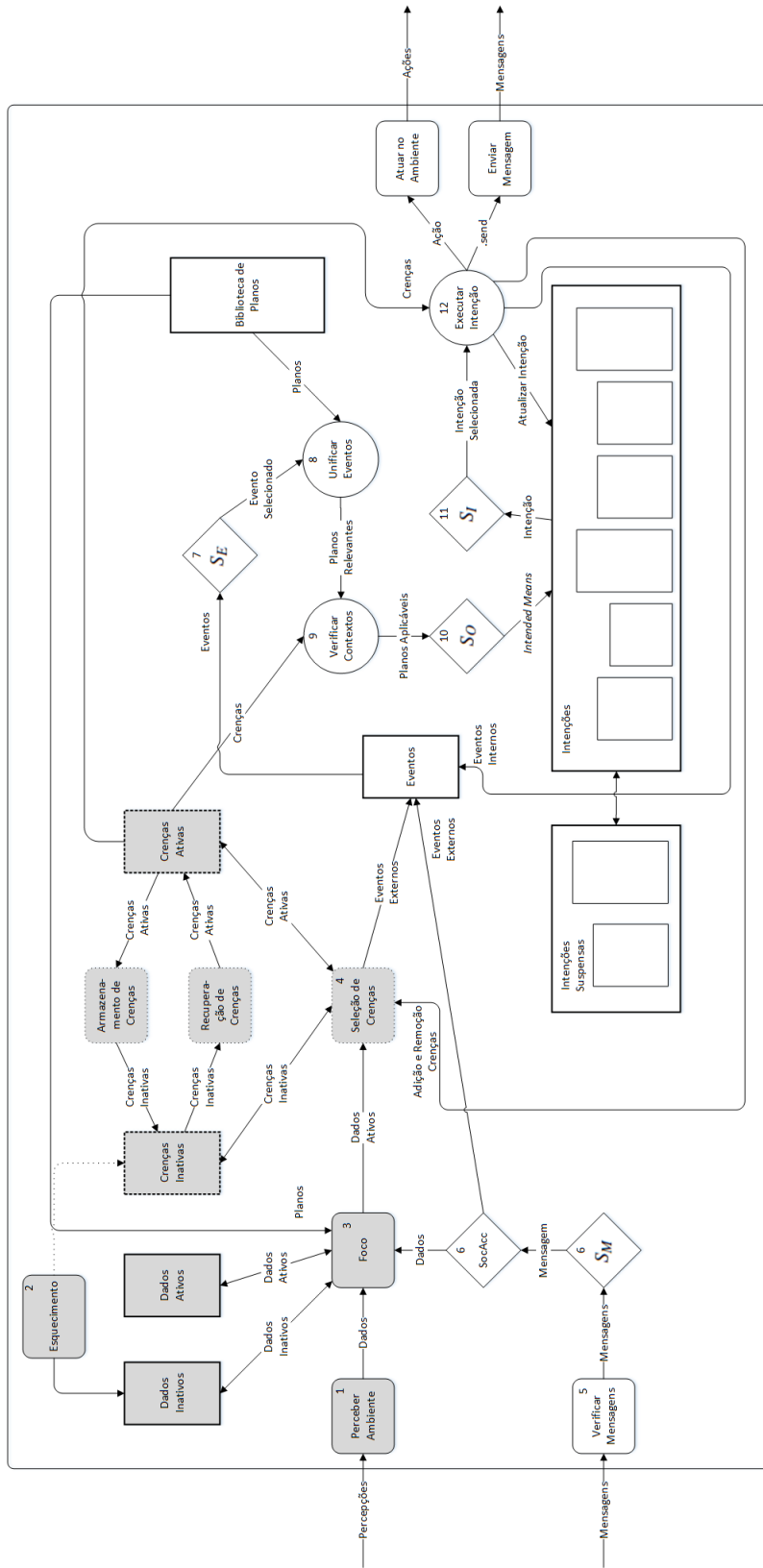


Figura 4.1: Ciclo de interpretação so Jason com a inclusão do DBR.

Fonte: Adaptado de [Bordini et al., 2007]

Com a inserção dos processos de Foco e Esquecimento na arquitetura, o ciclo de raciocínio do agente é alterado e são criadas novas conexões entre os componentes. O novo ciclo de raciocínio tem doze passos¹, dois deles são novos:

1. Perceber o ambiente: A percepção continua inalterada, o agente continua recebendo todas as informações disponibilizadas aos seus sensores.
2. Esquecer informações antigas: O Esquecimento é um novo processo incluído no ciclo de raciocínio do agente, assim como outros processos do Jason, ele tem uma implementação padrão que pode ser customizada pelo desenvolvedor do agente. O Esquecimento atua nos *Dados Inativos* e nas *Crenças Inativas*. Seguindo o escopo, o Esquecimento proposto aqui fará o expurgo da base de *Dados Inativos*.
3. Focar percepções nos objetivos: O novo processo de foco seleciona as percepções *Relevantes* aos planos do agente, chamadas de *Dados Ativos*. Percepções não *Relevantes* são desconsideradas do ciclo de raciocínio atual do agente, mas podem ser armazenadas como *Dados Inativos* e em um ciclo posterior sofrem ação do processo de *Esquecimento*.
4. Atualizar base de crenças: Corresponde à execução do processo de *Seleção de Crenças*, os *Dados Ativos* selecionados pelo processo de Foco são os candidatos a crenças da iteração atual do ciclo de raciocínio. A implementação do processo de *Seleção de Crenças* proposto pelo modelo DBR está fora do escopo desse trabalho, desta forma, os agentes criados pelo modelo aqui definido utilizam a atualização da base de crenças padrão do Jason, significando que todo *Dado Ativo* será considerado uma crença válida.
5. Receber comunicação de outros agentes: A comunicação permanece com o mesmo comportamento: o agente verifica se recebeu mensagens de outros agentes.
6. Selecionar mensagens socialmente aceitáveis: Assim como a comunicação, a seleção das mensagens socialmente aceitáveis não é impactada com o modelo proposto.
7. Selecionar um evento: A seleção de eventos mantém a mesma lógica, porém, utilizando o DBR, somente alterações no ambiente que são *Relevantes* ao agente irão gerar eventos, diminuindo o tamanho do conjunto de eventos nos casos onde o agente tenha percepções não interessantes aos seus planos.
8. Recuperar planos relevantes: Com o evento selecionado, o Jason procura, na biblioteca de planos, por todos os planos relevantes ao evento. Isto é feito considerando todos os gatilhos de eventos que podem ser unificados com o evento selecionado.
9. Determinar planos aplicáveis: Em relação ao custo computacional, este é o processo mais impactado com a introdução do *Foco*. Em cenários com muitas percepções e, consequentemente, com uma base de crenças muito grande, o cálculo da consequência lógica do contexto dos planos pode comprometer o tempo de resposta do ciclo de raciocínio do agente. Com o *Foco*, a base de crenças pode ser drasticamente reduzida, auxiliando o algoritmo de unificação do Jason.

¹O apêndice A detalha a arquitetura padrão do Jason, inclusive os 10 passos originais.

10. Selecionar um plano aplicável: O agente escolhe um dos possíveis planos aplicáveis e se compromete em executá-lo.
11. Selecionar uma intenção para execução: A inclusão do Foco pode diminuir a quantidade de *intended means* concorrentes do agente, direcionando a atenção do agente para os planos relevantes mais importantes do ciclo. Em cada ciclo, uma fórmula de uma intenção do conjunto de intenções é escolhida para execução.
12. Executar um passo de uma intenção: A fórmula escolhida pelo passo anterior é executada.

Os processos de Foco e Esquecimento adicionam novos custos computacionais no ciclo de raciocínio do agente. Porém, com a filtragem de percepções sem Relevância, menos eventos são gerados e o espaço de busca da função *Se* é reduzido. Também, somente crenças relevantes são adicionadas na base de crenças, diminuindo o custo da unificação dos contextos dos planos com a base de crenças.

4.2 Especialização do algoritmo para cálculo da Relevância

O Algoritmo 3 é uma extensão do algoritmo 2 proposto no capítulo anterior. Quando aplicado ao Jason, a unificação do dado ϕ ocorre com o evento gatilho e com o contexto do plano :

Algoritmo 3 relevancia(P, ϕ)

Fonte: Autor

```

1: for all  $p \in P$  do
2:    $p^e \leftarrow$  evento gatilho de  $p$ 
3:    $p^c \leftarrow$  contexto de  $p$ 
4:   if unificar( $\phi, p^e$ ) or unificar( $\phi, p^c$ ) then
5:      $q^\phi \leftarrow q^\phi + 1$ 
6:      $v^\phi \leftarrow$  valor do objetivo
7:      $V_\phi \leftarrow V_\phi \cup v^\phi$ 
8:   end if
9: end for
10: return calcular( $V_\phi, q^\phi$ )

```

Quando aplicado no Jason, o cálculo de q_ϕ é automatizado e independente da ação do projetista do agente. Para identificar o valor do objetivo do plano, é requerida uma anotação chamada *valor_objetivo* com um atributo no *label* que dá nome aos planos no código fonte dos planos do agente codificados na linguagem AgentSpeak (A seção A.2 detalha a sintaxe e o comportamento destes componentes do Jason), sinalizando o valor dos objetivos de cada um dos planos do agente. Este valor não é obrigatório, caso não exista esta informação o Foco assume o valor mínimo, ou seja, zero. A Figura 4.2 demonstra a sintaxe do valor dos objetivos em um plano do Jason, nela o plano tem um valor do objetivo igual a nove.

```
-----  
@nome_do_plano[valor_objetivo(9)]  
+evento_gatilho : contexto1 & contexto2  
  <- acao.
```

Figura 4.2: Sintaxe em AgentSpeak para definição do valor objetivo.

Fonte: Autor

4.3 Extensão da estrutura de dados do Jason

Para que seja possível a inclusão do DBR no código fonte do Jason, é necessário alterar a forma que ϕ é armazenado em seu ciclo de raciocínio, para isto a estrutura de dados *Datum* foi criada. Ela é responsável por armazenar, além do predicado do dado, as novas características da informação: Relevância e o momento em que o dado foi percebido.

A figura 4.3 demonstra, utilizando um diagrama de classes UML, como o dado é representado com as extensões apresentadas nesse trabalho. Originalmente o Jason armazena o predicado de ϕ utilizando a classe *Literal*, a nova classe *Datum* agrega *Literal* e as novas classes *Relevance* (abstrata) e *Moment* (de interface).

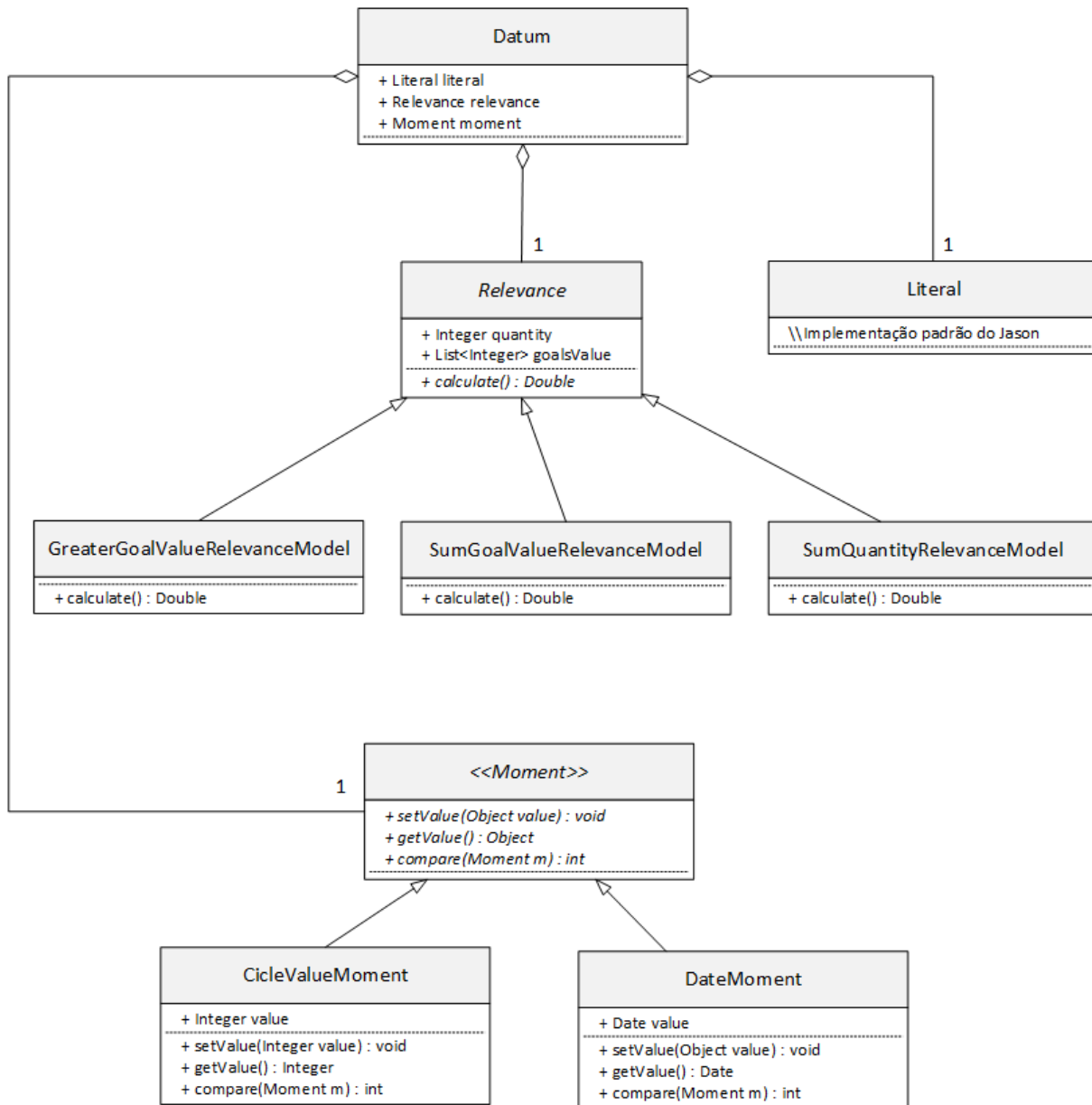


Figura 4.3: Diagrama de classes do *Datum*.

Fonte: Autor

A classe abstrata *Relevance* é estendida por classes que representam os três modelos de cálculo da Relevância propostos. O atributo *quantity* representa q^ϕ e *goalsValue* representa o conjunto V_ϕ . O método *calculate()* contém a implementação das fórmulas 3.11, 3.12 e 3.13.

A interface *Moment* é implementada pelas classes *CycleValueMoment* e *DateMoment*, seus atributos *value* representam t^ϕ , *compare* é um método auxiliar que compara o dois t^ϕ de dois objetos *Moment*.

Os conjuntos D_I e D_A estão representados no diagrama de classes da Figura 4.4, a classe *InactiveData* mantém uma lista de Dados Inativos, enquanto a classe *ActiveData* é responsável por manter a lista de Dados Ativos.

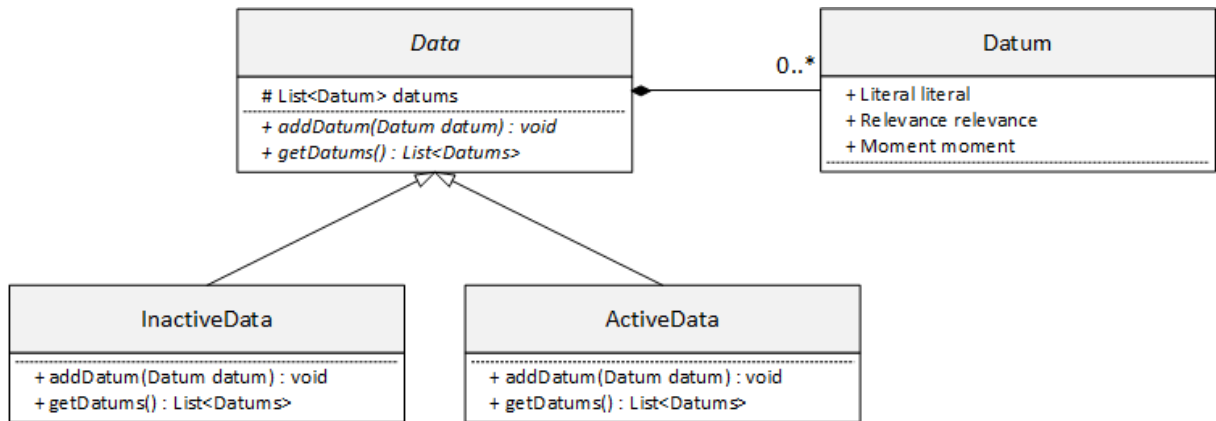


Figura 4.4: Diagrama de classes dos Dados Ativos e Inativos.

Fonte: Autor

Para implementar os processos de Foco e Esquecimento, a classe *Agent* do Jason é estendida pela nova classe *DBRAgent*, seu método *buf(List<Literal> percepts)* é sobrescrito para implementar os algoritmos de Foco, Esquecimento e os modelos de armazenamento de Dados Inativos e de cálculo da Relevância, como apresentado na Figura 4.5.

Na classe *DBRAgent*, o atributo *relevanceThreshold* representa o limite mínimo da Relevância do dado (*l*) e *oblivionThreshold* representa o período máximo de permanência do dado inativo (*m*). A fórmula 3.14 é implementada no método *oblivion()*, o *focusing()* implementa a fórmula 3.10 e realiza as chamadas aos métodos *calcRelevance(Plan p, Datum d)* e *getInactiveDataModel()*, que são responsáveis, respectivamente, pelo cálculo da Relevância e a decisão de qual modelo de Dados Inativos deve ser aplicado.

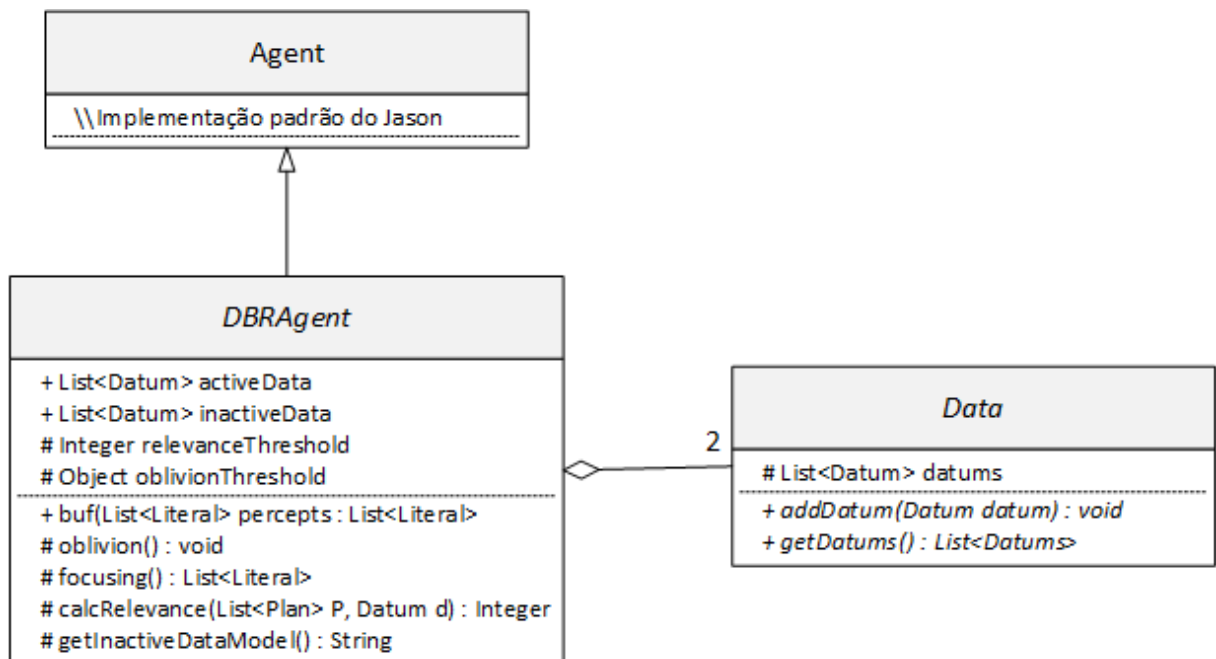


Figura 4.5: Diagrama de classes do agente DBR.

Fonte: Autor

4.4 Trabalhos correlatos aplicados ao Jason

Trabalhos correlatos a esta dissertação que também foram implementados em Jason são apresentados nesta seção. Eles estão divididos em três subseções: Automatização da revisão de crenças, revisão de crenças (aplicação do AGM) e filtro de percepções.

4.4.1 Automatização da revisão de crenças em agentes Jason

Extensões adicionadas ao AgentSpeak (e por consequência, também ao Jason) criaram a necessidade da automação da revisão de crenças dos agentes desenvolvidos nesta linguagem. Sem a automação, o controle de consistência da base seria de responsabilidade do desenvolvedor, o que potencialmente gera um grande esforço de programação por parte do usuário do Jason.

Para [Alechina et al., 2006], a possibilidade de se adicionar notas mentais², é um dos principais motivadores para se automatizar a revisão de crenças no AgentSpeak. Isto porque, a partir da possibilidade de se gerar uma crença do tipo nota mental, tem-se duas fontes distintas de atualização da informação: percepções e o próprio agente. Estas fontes distintas podem gerar contradições entre as crenças e tornar a base de crenças inconsistente. A inconsistência da base, por padrão, não é tratado pelo Jason.

A comunicação é outra extensão que pode implicar na revisão de crenças. Proposta por [Moreira et al., 2004], oferece uma semântica de comunicação que permite aos agentes, através de mensagens, compartilharem seus estados mentais com outros agentes. Para [Ancona et al., 2004], isso não afeta somente as crenças, mas também os planos utilizados pelos agentes, ou seja, agentes podem trocar informações entre si sobre como agir sobre determinado evento.

A incorporação do raciocínio ontológico no AgentSpeak, proposto por [Moreira et al., 2005], estendeu a linguagem para que a base de crenças comportasse operadores das lógicas de Descrição (DL, Description Logics) de [Baader et al., 2003]. Esta extensão, chamada de AgentSpeak-DL, adicionou, além da ABox (fórmulas lógicas que representam os fatos, que no jargão destas lógicas, são as instâncias/indivíduos dos conceitos/classes), o TBox (definições de conceitos e relacionamentos entre eles). Isto resultou em várias mudanças na linguagem, sendo que a mais importante no ponto de vista de revisão de crenças foi o refinamento da atualização de crenças, permitindo que a adição de uma propriedade de um indivíduo somente seja possível se o resultado dessa adição mantenha a base de crenças consistente em relação as descrições do conceito.

A extensão *Belief annotations* é outra importante mudança que motiva o estudo da automação da revisão de crenças no AgentSpeak. Com esta alteração, as fórmulas podem conter anotações. Uma anotação é uma lista de termos dentro de colchetes na sequência de um predicado. Por exemplo, a crença com anotação *greenpath(r1)[doc(0.9)]* poderia ser utilizada por um programador para representar que acreditasse que a pedra r1 tem uma mancha verde e o grau de certeza dessa crença é de 0.9. Tratando-se da base de crenças, a importância das anotações é identificar a origem de uma determinada crença do agente, com isto é possível revisões de crenças dependentes do contexto, ou seja, crenças de uma fonte particular podem ser preferidas em relação a outras fontes quando uma inconsistência é encontrada.

²Notas mentais podem fazer parte do corpo de planos interpretados pelo AgentSpeak. São dados adicionados pelo agente, em tempo de execução, em sua própria base de crenças.

4.4.2 Aplicação do AGM no Jason

O trabalho de [Alechina et al., 2006] foi a primeira tentativa de automatizar a revisão de crenças em uma linguagem prática de desenvolvimento de sistemas multiagentes. Fundamentado no AGM, [Alechina et al., 2006] introduziu a revisão de crenças no AgentSpeak, definindo dois principais objetivos: o primeiro é que o algoritmo deve ser teoricamente bem motivado, isto é, este deve produzir revisões de acordo com um conjunto de postulados comumente aceito, os quais caracterizam a revisão de crenças de uma maneira racional. O segundo é que a linguagem tem que ser realizável, o que significa que o algoritmo de revisão tem que ser eficiente do ponto de vista computacional (executado em tempo polinomial).

O algoritmo para a revisão de crenças proposto por [Alechina et al., 2006] é estruturado como um grafo direcionado, onde os nodos são crenças e justificativas. Uma justificativa consiste em uma crença e uma lista de suporte contendo o contexto do plano usado para derivar esta crença, por exemplo, $(A, [B, C])$ onde A é uma crença derivada que foi declarada por um plano de contexto B e que provocou a adição da crença C . Crenças fundamentais, as que não são derivadas, têm o formato de justificção $(D, [])$. No grafo, cada justificativa tem uma aresta de saída que aponta para a crença justificada, e recebe uma aresta direcionada das crenças pertencentes a sua lista de suporte. É assumido que cada lista de suporte s tem designada uma crença de menor preferência, $w(s)$.

Em [Alechina et al., 2006], a contração de um literal A é definida como a remoção de A e de literais suficientes para que A não seja derivável, como demonstrado no Algoritmo 1 *contract*.

Algoritmo 4 *contract*(A),

Fonte: [Alechina et al., 2006]

```

1: for all  $A$ 's arestas de saída para justificção  $(C, s)$  do
2:   remover( $C, s$ ) do grafo
3: end for
4: for all  $A$ 's arestas de entrada para justificção  $(A, s)$  do
5:   if  $s$  é vazio then
6:     remover( $A, s$ )
7:   else
8:     contrair( $w(s)$ )
9:   end if
10: end for
11: remover( $A$ )

```

Em relação ao algoritmo de revisão, ou seja, a adição de uma crença não pode resultar em uma inconsistência lógica dentro do conjunto de crenças, [Alechina et al., 2006] propôs o Algoritmo 2 *revision*.

O algoritmo de revisão requer a definição de uma ordem de preferência entre os membros a serem contraídos, uma sugestão proposta por [Clark and McCabe, 2007] considera dados percebidos com maior preferência aos dados recebidos por comunicação. No caso de dados de mesma origem, considera-se o mais novo.

Algoritmo 5 revision(A),

Fonte: [Alechina et al., 2006]

-
- 1: $K \leftarrow K + A$
 - 2: aplicar em todos os planos relacionados
 - 3: **while** $\exists(B, \neg B)$ in K **do**
 - 4: *contrair(membro de menor preferência entre o par)*
 - 5: **end while**
-

4.4.3 Filtros de percepção no Jason

O trabalho de [Stabile and Sichman, 2015] construiu uma modificação no interpretador Jason capaz de filtrar percepções, seu objetivo é diminuir o tempo de resposta do ciclo de interpretação do Jason, evitando, assim, uma assincronia entre o Jason e sistemas de simulação como o da RoboCup³ e MAPC⁴.

O filtro é baseado no *middleware* proposto em [van Oijen and Dignum, 2011], que recebe os sinais do ambiente, filtra aqueles que correspondam a itens pré cadastrados em uma tabela de interesses e envia ao agente somente percepções relevantes aos seus objetivos. Outra referência utilizada por [Stabile and Sichman, 2015] é o AGENTlib, criado por [Boulic et al., 1997], que também é um filtro de percepções, selecionando as informações de acordo com características específicas como: intervalo de espaço entre o agente e a percepção, campo de visão do agente e propriedades específicas da informação (cor, tamanho, formato).

O filtro de percepção criado por [Stabile and Sichman, 2015] customiza parte do ciclo de raciocínio de um agente construído no Jason. Antes de transformar uma informação capturada do ambiente em crença, o programa valida em um arquivo de configuração se aquela percepção pode ser incluída na base de crenças.

O arquivo de configuração é o cerne da proposta pois contém todas as regras de filtragem: o predicado da percepção é comparado com os predicados presentes no arquivo de configuração, também podem ser usados operadores relacionais (como maior, menor, igual) para comparar os atributos do predicado. Anotações sobre a origem (comunicação, ambiente ou nota mental) da percepção também podem ser levadas em consideração nas regras. O arquivo de configuração deve ser previamente criado pelo desenvolvedor e também pode ser alterado em tempo de execução utilizando uma nova ação interna disponibilizada pela solução.

4.5 Conclusão

Da forma que foi construído, a arquitetura do Jason permite adaptações, modificações e extensões ao ciclo de raciocínio padrão dos agentes construídos na plataforma, sendo uma ferramenta para o estudo, ampliação e aplicação dos conceitos de agentes racionais, sistemas multiagentes e arquitetura BDI.

A união do DBR (e do filtro de percepções aqui proposto) com uma consolidada plataforma de desenvolvimento de agentes BDI, como é o Jason, permite um aprofundamento no estudo da revisão de crenças dentro da área de agentes racionais e uma forma prática para mensurar os resultados dos programas que utilizam o modelo. Também oferece uma oportunidade

³RoboCup: <http://www.robocup.org/>

⁴MAPC: <https://multiagentcontest.org/>

para contribuir com a atual arquitetura do Jason, incluindo comportamentos epistêmicos que não pertencem a implementação padrão da linguagem.

Também foram tratados dois trabalhos correlatos com aplicação do modelo de percepção de crenças na plataforma Jason. O primeiro é o trabalho da [Alechina et al., 2005], que baseado nos conceitos do AGM estendeu a arquitetura da plataforma Jason implementando uma revisão de crenças automatizada e formal. Apesar de [Alechina et al., 2005] utilizar o modelo AGM como processo revisor de crenças, nada impede sua utilização em conjunto com o modelo de filtro de percepções proposto por esta dissertação. A coexistência dos dois modelos é possível no mesmo agente BDI porque cada um atua em processos diferentes do seu ciclo de raciocínio.

[Stabile and Sichman, 2015] é outro trabalho correlato. Nele é apresentado um filtro de percepções integrado à plataforma Jason, seu cerne é um arquivo de configuração que é consultado a cada percepção recebida pelo agente. A principal diferença entre o modelo de [Stabile and Sichman, 2015] e o proposto por essa dissertação está relacionado à orientação do filtro de percepções. O primeiro está orientado ao arquivo de regras, o segundo é orientado aos objetivos do agente.

Um filtro de percepções que não considera os planos atuais do agente pode comprometer seu correto funcionamento. Isto porque modelos que não estão interligados aos objetivos estão sujeitos a dessincronização: Um agente pode criar, remover ou alterar um plano dinamicamente, em tempo de execução, o que pode criar inconsistências entre o arquivo de configuração e a biblioteca de planos, impactando no ciclo de raciocínio do agente. O cadastro não automatizado também pode gerar inconsistências, já que o projetista do agente pode deixar de criar regras necessárias ou criar regras incorretas.

Capítulo 5

Experimentos

O problema proposto neste capítulo é um cenário sintético, útil para comparações de desempenho e de comportamento de filtros de percepções. Pode, também, ser aplicado em experimentos que utilizam outros modelos de filtragem. A proposta é analisar quando e se o filtro de percepções fundamentado e modelado nos capítulos anteriores pode ajudar no desempenho - reduzindo o tempo de utilização do processador - de agentes que o implementam. Salienta-se que, se o agente for físico (ex. um robô), este ganho de desempenho pode representar ganho de autonomia devido à economia de energia das baterias.

Este capítulo descreve o problema criado, apresenta os resultados das execuções dos experimentos e então compara e analisa seus resultados.

5.1 Descrição do problema sintético

Um problema sintético foi criado para analisar o impacto do filtro de percepções na arquitetura BDI. Os objetivos dos experimentos aqui realizados são:

1. Comprovar que a inclusão do filtro de percepções na arquitetura não altera no modo em que o agente raciocina.
2. Comparar o desempenho de agentes com e sem filtro de percepções implementados no Jason.
3. Comparar o desempenho destes agentes com diferentes quantidades de planos.
4. Comparar o desempenho destes agentes em ambientes com ruídos¹ crescentes.

Uma execução consiste de um agente que percorre um tabuleiro linear de 1.000 posições. O principal objetivo do agente é chegar na última posição do tabuleiro, seus objetivos intermediários são mover-se para próxima posição. Porém, para cada uma das posições, o agente deve completar N planos locais (o código completo dos planos pode ser visualizado no Apêndice B.1) antes de mover-se para próxima posição.

O agente começa na posição 0 do tabuleiro e executa seu raciocínio sobre os N planos disponíveis na sua biblioteca de planos. O agente movimenta-se para a próxima posição do tabuleiro somente após executar os N planos. Repetindo o processo até chegar na última posição.

¹Entende-se como ruído percepções que não têm Relevância aos objetivos do agente.

Ao chegar no fim do tabuleiro, na posição 999 (que corresponde à execução de $999 \times N$ planos), a execução é finalizada.

O código abaixo mostra um exemplo da configuração com 10 planos sintéticos ($N=10$) gerados para a experimentação: para cada nova posição $pos(X)$, o plano @plano10 tem como *achievement goal* o plano plano9, este tem como *achievement goal* o plano plano8 e assim sucessivamente até que o plano @plano1 se torne uma intenção e o agente se mova para a próxima posição do tabuleiro.

```

1 +pos(X) <- !hedgh(X,X) .
2 @plano1[valor_objetivo(3)]+!hdfdd(X,X):abc(f)&dda(h)<-mover.
3 @plano2[valor_objetivo(6)]+!fccgb(X,Y):hnc(b)&fdc(h)<-!hdfdd(X,X) .
4 @plano3[valor_objetivo(6)]+!fbcga(X,Y):cad(c)&fcg(a)<-!fccgb(X,Y) .
5 @plano4[valor_objetivo(2)]+!bdfbb(X,Y):edg(a)&ada(f)<-!fbcga(X,Y) .
6 @plano5[valor_objetivo(9)]+!hbheg(X,Z):cbg(e)&dbh(h)<-!bdfbb(X,Y) .
7 @plano6[valor_objetivo(2)]+!bfeag(X,X):ebe(g)&dba(a)<-!hbheg(X,Z) .
8 @plano7[valor_objetivo(4)]+!egedc(X,Z):bhf(h)&ffg(d)<-!bfeag(X,X) .
9 @plano8[valor_objetivo(8)]+!gcfdd(X,Z):ead(d)&gbg(b)<-!egedc(X,Z) .
10 @plano9[valor_objetivo(2)]+!ghcac(X,X):ehe(e)&bde(c)<-!gcfdd(X,Z) .
11 @plano10[valor_objetivo(6)]+!hedgh(X,X):ehe(h)&gcd(a)<-!ghcac(X,X) .

```

A cada ciclo de percepção, o ambiente fornece percepções suficientes para a unificação dos contextos dos N planos. Isto significa que todos os planos se tornarão uma intenção e executarão o conteúdo de seu corpo. Nos experimentos, o ambiente além de produzir as percepções relevantes, também produz ruídos que devem ser desconsiderados pelo agente com filtro de percepções.

5.1.1 Materiais de experimentação

Para realização dos experimentos, foi utilizado um computador com processador Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz, com 8GB de memória RAM e HD SATA de 120GB. O sistema operacional utilizado foi o Debian 8.0 64bits.

5.1.2 Configuração do filtro

Como demonstrado ao longo do trabalho, existem diferentes combinações de modelos (Cálculo da Relevância, armazenamento da base de Dados Inativos e limitador l) possíveis para a implementação do filtro de percepções em um agente BDI. Porém, para que seja possível a comparação do desempenho de agentes com e sem o filtro de percepções, é necessário que ambas implementações atuem da mesma forma em relação ao tipo de armazenamento do dado e ao limite de percepções que podem se tornar relevantes. Para isto, a implementação do filtro de percepções utilizada neste experimento considerou:

1. Modelo DIP para os Dados Inativos
2. Processo de Foco com $l = 1$.
3. Processo de Esquecimento com $m = 1$.
4. Cálculo da Relevância utilizando o modelo "Quantidade de r^ϕ ".

Com esta configuração, o agente modelado com o filtro tem o mesmo comportamento do agente com a arquitetura original. Isto porque somente os dados percebidos no ciclo atual serão considerados no raciocínio, mantendo a base *DI* sempre com o mesmo tamanho do conjunto de dados percebidos no ciclo. A configuração do Foco com $l = 1$ e o modelo de cálculo de Relevância "Quantidade de r^ϕ " garantem que todos os dados com alguma Relevância se tornem crenças, simulando o comportamento de agentes que não implementam o filtro de percepções.

5.2 Experimentos

A partir do problema, foram realizados diferentes experimentos: para cada um dos N (1, 50, 100 e 200) são executados experimentos em que os ambientes fornecem: 0, 250, 500, 1000, 2000, 4000 e 8000 ruídos. Foram feitas 40 execuções para cada experimento. Os tempos de cada uma das 40 execuções para cada experimento estão disponíveis no Apêndice B.2.

5.2.1 Experimentos com N = 1

O primeiro experimento considera um ambiente sem ruídos, isto é, só existem dados relevantes e que são utilizados no raciocínio. A partir do segundo experimento, o ambiente começa a fornecer ruídos que são considerados no ciclo de raciocínio do agente. Estes ruídos são considerados no ciclo completo (desde a percepção até a o raciocínio) do agente quando o experimento é feito na arquitetura original, e são desconsiderados (pelo processo de Foco) pelo agente que implementa o modelo de filtro de percepções.

O segundo experimento considera o ambiente fornecendo, além das percepções relevantes, 250 percepções como ruído, no quarto, 1.000, no quinto, 2.000, no sexto, 4.000, e no sétimo, 8.000. A Tabela 5.1 e a Figura 5.1 comparam as médias de tempo entre a arquitetura padrão e o modelo de filtros de percepção a medida que o número de ruídos aumenta.

Cada célula das tabelas comparativas abaixo exibem a média de tempo, em segundos, da amostra de 40 execuções para cada experimento. Ao lado, entre parênteses, são exibidos os desvios padrões amostrais (σ) de cada uma das amostras.

Tabela 5.1: Comparação dos tempos com N=1.

Arquit.	Quantidade de percepções ruído.						
	0	250	500	1000	2000	4000	8000
Filtro	0,63(0,01)	3,30(0,05)	5,94(0,27)	13,09(0,09)	40,44(0,21)	127,50(1,63)	498,48(7,68)
Original	0,63(0,01)	4,19(0,08)	9,50(0,31)	35,86(0,81)	122,74(1,95)	410,98(6,26)	1469,08(65,04)

Fonte: Autor

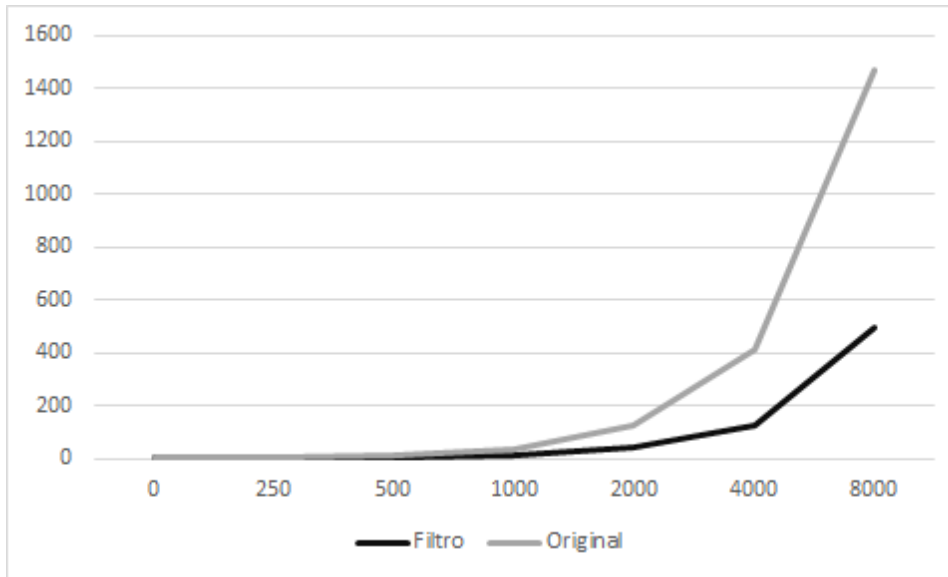


Figura 5.1: Gráfico (ruídos x tempo) comparativo dos tempos com N=1

Fonte: Autor

5.2.2 Experimentos com N = 50

O mesmo método de aplicação dos experimentos com somente um plano local (N=1) é aplicado aqui com 50 planos locais (N=50). A Tabela 5.2 e a figura 5.2 comparam as médias de tempo entre a arquitetura padrão e o modelo de filtros de percepção a medida que o número de ruídos aumenta.

Tabela 5.2: Comparação dos tempos com N=50.

Arquit.	Quantidade de percepções ruído.						
	0	250	500	1000	2000	4000	8000
Filtro	2,66(0,03)	6,38(0,16)	10,31(0,12)	20,84(0,18)	54,12(0,33)	150,44(2,77)	480,93(21,59)
Original	2,05(0,08)	5,63(0,12)	11,35(0,18)	37,30(0,22)	131,05(2,03)	422,77(3,73)	1471,96(47,04)

Fonte: Autor

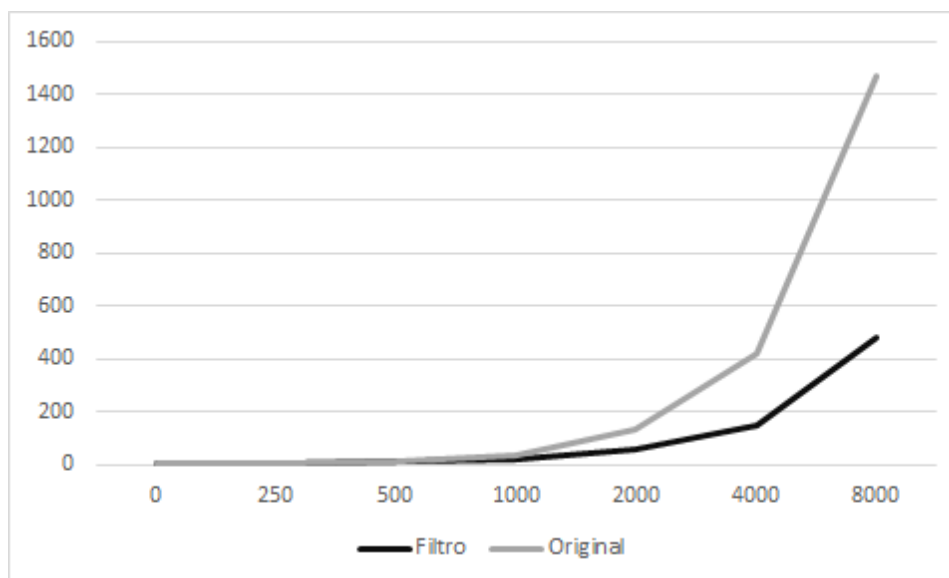


Figura 5.2: Gráfico (ruídos x tempo) comparativo dos tempos com N=50

Fonte: Autor

5.2.3 Experimentos com N = 100

O mesmo método de aplicação dos experimentos com somente um plano local (N=1) é aplicado com 100 planos locais (N=100). A tabela 5.3 compara as médias de tempo entre a arquitetura padrão e o modelo de filtros de percepção a medida que o número de ruídos aumenta.

Tabela 5.3: Comparação dos tempos com N=100.

Arquit.	Quantidade de percepções ruído.						
	0	250	500	1000	2000	4000	8000
Filtro	5,10(0,10)	10,02(0,15)	15,52(0,32)	28,76(0,63)	65,53(0,97)	170,65(0,81)	493,99(13,73)
Original	3,08(0,06)	7,10(0,11)	14,31(0,25)	41,69(0,28)	132,10(1,41)	438,02(8,31)	1440,60(37,12)

Fonte: Autor

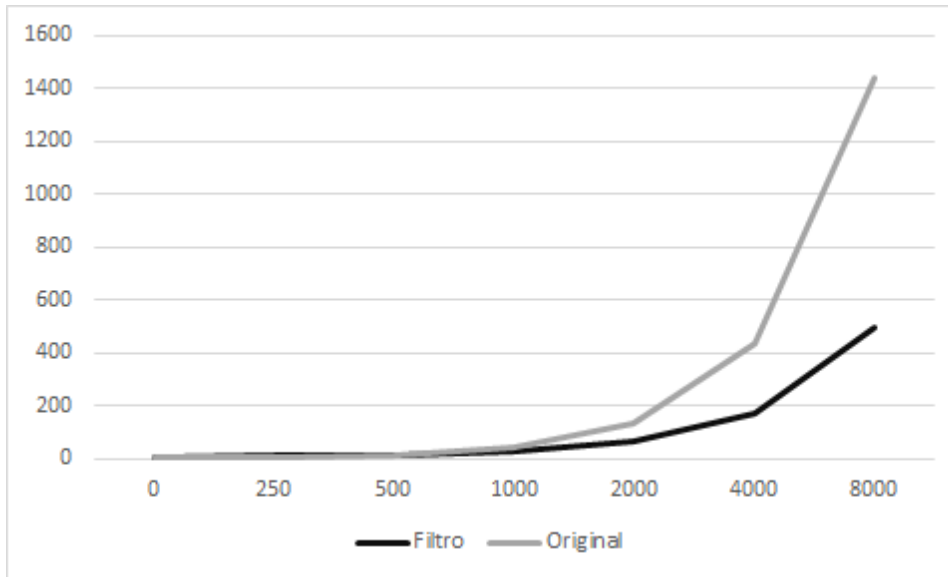


Figura 5.3: Gráfico (ruídos x tempo) comparativo dos tempos com N=100

Fonte: Autor

5.2.4 Experimentos com N = 200

O mesmo método de aplicação dos experimentos com somente um plano local (N=1) é aplicado com 200 planos locais (N=200). A Tabela 5.4 e a figura 5.4 comparam as médias de tempo entre a arquitetura padrão e o modelo de filtros de percepção a medida que o número de ruídos aumenta.

Tabela 5.4: Comparação dos tempos com N=200.

Arquit.	Quantidade de percepções ruído.						
	0	250	500	1000	2000	4000	8000
Filtro	13,99(0,28)	21,92(0,57)	30,25(0,44)	49,73(0,80)	100,00(1,09)	214,49(2,05)	600,94(6,12)
Original	5,82(0,07)	11,13(0,07)	19,97(0,31)	49,41(0,33)	140,47(1,37)	450,59(5,64)	1487,79(33,32)

Fonte: Autor

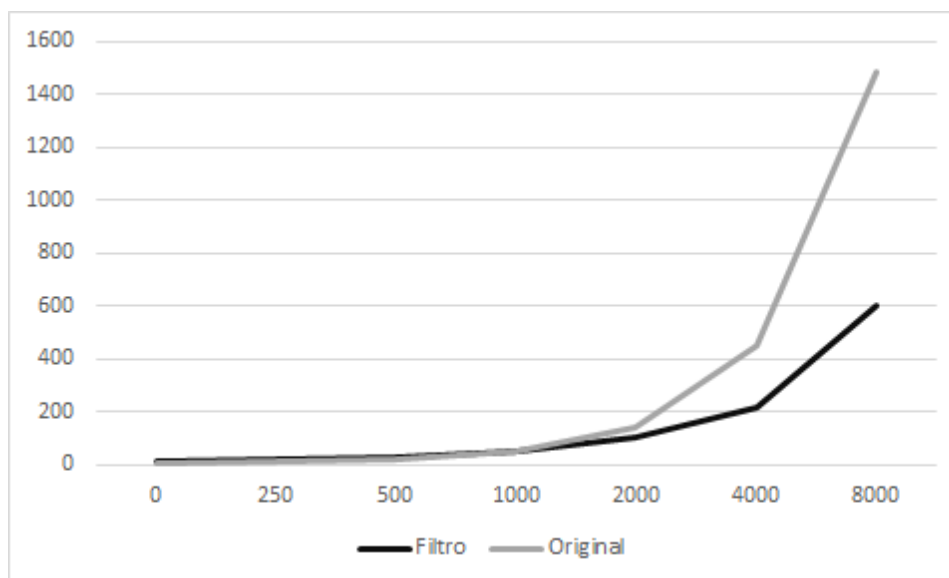


Figura 5.4: Gráfico (ruídos x tempo) comparativo dos tempos com N=200

Fonte: Autor

5.3 Análises dos resultados

As tabelas 5.5 e 5.6 apresentam os experimentos agrupados por arquitetura. Cada célula apresenta o tempo e, entre parênteses, o desvio padrão amostral. Através dos experimentos percebe-se que em ambas arquiteturas, quanto maior o número de ruídos, menor o desempenho do agente, independente da quantidade de planos. Isto comprova que a quantidade de dados na base de crenças do agente tem um impacto significativo no desempenho. Por exemplo, o mesmo agente executa a mesma tarefa em 0,63 segundos ou 1469,08 segundos (agente Jason com N=1), dependendo da quantidade de dados percebidos.

O crescimento do aumento de ruídos tem a mesma consequência em ambas implementações: maior tempo de processamento. Porém, a proporção deste aumento do tempo de processamento é menor no agente que implementa o filtro de percepções. As figuras 5.1, 5.2, 5.3 e 5.4 apresentam esta constatação. Mesmo quando o modelo com filtro tenha um desempenho inferior nos experimentos em que o ambiente tem poucos ruídos (0, 250 ou 500), no final, quando o número de ruídos é alto (8000), seu desempenho é superior.

Tabela 5.5: Experimentos agrupados pela arquitetura com filtro

Planos	Quantidade de percepções ruído.						
	0	250	500	1000	2000	4000	8000
1	0,63(0,01)	3,30(0,05)	5,94(0,27)	13,09(0,09)	40,44(0,21)	127,50(1,63)	498,48(7,68)
50	2,66(0,03)	6,38(0,16)	10,31(0,12)	20,84(0,18)	54,12(0,33)	150,44(2,77)	480,93(21,59)
100	5,10(0,10)	10,02(0,15)	15,52(0,32)	28,76(0,63)	65,53(0,97)	170,65(0,81)	493,99(13,73)
200	13,99(0,28)	21,92(0,57)	30,25(0,44)	49,73(0,80)	100,00(1,09)	214,49(2,05)	600,94(6,12)

Fonte: Autor

Tabela 5.6: Experimentos agrupados pela arquitetura original

Planos	Quantidade de percepções ruído.						
	0	250	500	1000	2000	4000	8000
1	0,63(0,01)	4,19(0,08)	9,50(0,31)	35,86(0,81)	122,74(1,95)	410,98(6,26)	1469,08(65,04)
50	2,05(0,08)	5,63(0,12)	11,35(0,18)	37,30(0,22)	131,05(2,03)	422,77(3,73)	1471,96(47,04)
100	3,08(0,06)	7,10(0,11)	14,31(0,25)	41,69(0,28)	132,10(1,41)	438,02(8,31)	1440,60(37,12)
200	5,82(0,07)	11,13(0,07)	19,97(0,31)	49,41(0,33)	140,47(1,37)	450,59(5,64)	1487,79(33,32)

Fonte: Autor

A Tabela 5.7 apresenta o desempenho do modelo de filtragem em relação à arquitetura original. O valor de cada célula é composto pela divisão da média de tempo das execuções da arquitetura original (MO) pela média de tempo das execuções que implementam o modelo de filtro de percepções (MF): $\frac{MO}{MF}$.

Tabela 5.7: Comparação $\frac{MO}{MF}$ das execuções.

Planos	Quantidade de percepções ruído.						
	0	250	500	1000	2000	4000	8000
1	1,00	1,27	1,60	2,74	3,04	3,22	2,95
50	0,77	0,88	1,10	1,79	2,42	2,81	3,06
100	0,60	0,71	0,92	1,45	2,02	2,57	2,92
200	0,42	0,51	0,66	0,99	1,40	2,10	2,48

Fonte: Autor

Nas células em cinza mais escuro estão as execuções onde a arquitetura original superou o tempo de execução do filtro de percepções. É possível perceber que quanto maior o número de planos do agente menor o desempenho do filtro em relação à arquitetura original. Isto se deve a inclusão do processo de Foco no ciclo de raciocínio. Cada um dos dados percebidos pelo agente é armazenado na base D_I , então o processo de Foco tenta unificar cada um dos dados com os gatilhos e contextos dos planos existentes na biblioteca de planos, aqueles que são unificados e têm relevância suficiente se tornam crenças. Com este procedimento o Foco é responsável pela adição da complexidade $O(|D_I| * |P|)$, onde P é o número planos presentes na biblioteca de planos do agente.

As células restantes da Tabela $\frac{MO}{MF}$ apresentam que o filtro, mesmo adicionando complexidade ao ciclo de raciocínio do agente, se mostrou mais eficiente que a arquitetura padrão.

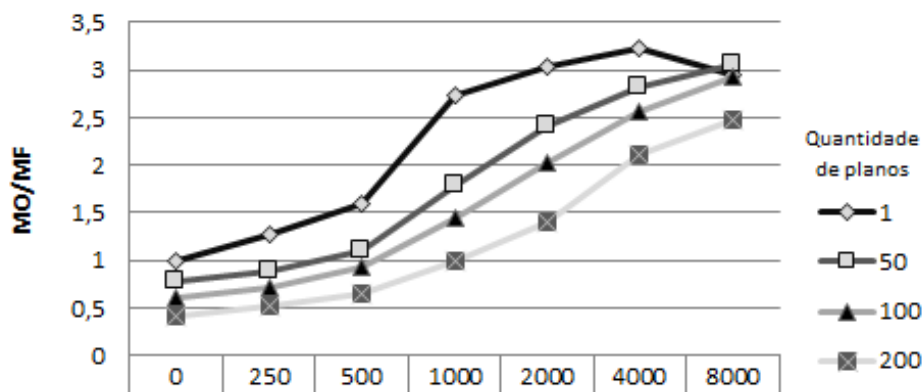


Figura 5.5: Gráfico (ruídos x $\frac{MO}{MF}$) comparativo das execuções.

Fonte: Autor

A Figura 5.5 apresenta a curva de desempenho do filtro em relação à arquitetura original. Todos os pontos maiores que 1 do eixo $\frac{MO}{MF}$ representam o momento em que o filtro supera o desempenho da arquitetura original. Nestes pontos a utilização do filtro é recomendada, pois além de agregar propriedades epistêmicas importantes ao modelo BDI (Relevância do dado, momento da percepção e separação entre dados e crenças), a inclusão dos novos processos reduzem o tempo de processamento do programa. Nestes casos, a diminuição da base de crenças - e a economia que isto acarreta nos processos de geração de opções e filtro de intenções - compensou a adição do processo de Foco na arquitetura, melhorando o desempenho do agente.

Mais especificamente, o melhor desempenho se deve ao impacto do Foco em dois processos do interpretador de AgentSpeak. O primeiro é a recuperação de planos relevantes (etapa 8 da Figura 4.1). Nele é executada a unificação dos eventos gatilhos dos planos com as crenças do agente. O Foco provê a diminuição da base de crenças e com isto o espaço de busca desta etapa é reduzido.

O segundo processo impactado pelo filtro é a determinação dos planos aplicáveis (etapa 9 da Figura 4.1). Neste processo é calculada a satisfabilidade lógica de todos os contextos dos planos. As premissas (contexto do plano) têm que pertencer à base de crenças para que a conclusão (corpo do plano) seja verdadeira. Como no processo de recuperação de planos relevantes, com a diminuição da base de crenças provida pelo processo de Foco, o espaço de busca da determinação dos planos aplicáveis é reduzido.

A figura 5.5 aponta uma similaridade das curvas dos experimentos quando a quantidade de percepções ruído aumenta. Isto indica que a melhora do desempenho do filtro em relação a arquitetura original tende a ter o mesmo comportamento a medida que o número de ruídos e planos aumentam. A curva dos experimentos com um plano apresentam uma inclinação decrescente entre os pontos 4000 e 8000, que pode ser explicada pela variação no tempo destas amostras. As execuções deste experimento foram os que tiveram o maior desvio padrão entre as amostras.

5.4 Conclusão

Quanto ao raciocínio, os dois agentes se comportaram da mesma forma. Ou seja, os objetivos locais, intermediários e final foram cumpridos pelos dois agentes. Isto comprova que a inclusão dos processos de Foco e Esquecimento, além da segregação entre dados e crenças não impactaram na maneira em que agente raciocina.

Em relação ao desempenho, é possível perceber através das análises empíricas que o Foco cria um filtro de percepções eficiente em cenários onde o agente percebe muitos dados não relevantes para seus objetivos. Isto se deve a redução do tamanho da base de crenças e da diminuição do custo da seleção dos planos aplicáveis do agente.

Por outro lado, em cenários com nenhuma ou poucas percepções não relevantes, se a necessidade é desempenho, a implementação do filtro não é recomendada, já que a adição da complexidade do Foco não compensa os ganhos obtidos com os outros processos.

Capítulo 6

Conclusão e trabalhos futuros

6.1 Conclusão

Foi possível, fundamentado em conceitos do DBR, modelar um filtro de percepções aplicável em agentes BDI que os tornam capazes de considerar no ciclo de raciocínio somente percepções relevantes aos seus objetivos. Para que o modelo fosse integrado à arquitetura BDI, além da definição e formalização dos processos de Foco e Esquecimento, da formalização da diferença entre Dados e Crenças, foi necessário criar diferentes modelos de cálculo da Relevância e de armazenamento dos Dados Inativos.

Apesar de se basear no modelo DBR, o modelo de filtro de percepções proposto por este trabalho pode ser utilizado independente do restante do modelo DBR. Isto porque os processos responsáveis pela filtragem das percepções foram segregados e são independentes dos processos de revisão de crenças. Desta forma, é possível a coexistência, dentro do modelo BDI, do filtro de percepções com diferentes modelos de revisão de crenças, como o AGM.

Quanto aos objetivos específicos deste trabalho:

1. Definir e identificar o impacto da inclusão do processo de Foco no ciclo de raciocínio de um agente BDI.

O Foco, que é o principal processo do modelo de filtro de percepções, pôde ser definido ao ponto de se tornar computavelmente aplicável, seu funcionamento foi formalizado na equação 3.10 e no algoritmo 4.

Além do filtro de percepções não relevantes (ruídos), um limitador de percepções relevantes foi proposto por este trabalho. Com ele é possível limitar o número de percepções que se tornarão crenças mesmo que elas tenham alguma Relevância ao agente. Este limitador pode ser aplicado em agentes que necessitam priorizar a execução de determinado plano em detrimento a outro. Porém, este tipo de limitador pode alterar o comportamento do raciocínio do agente (3.2.2).

2. Definir e aplicar modelos de cálculo da Relevância dos dados.

O cerne do processo de Foco é o cálculo da Relevância, que foi identificada como uma propriedade dependente do cenário de atuação do agente. Por isto, o presente trabalho propôs três modelos de cálculo da Relevância (equações 3.11, 3.12 e 3.13), cada qual com a função de atuar em diferentes cenários.

Para que fosse possível a realização do cálculo da Relevância, além de definir qual é o conceito de dependência do dado, foi necessário aprofundar os conceitos de quantidade e valoração do dado (3.2.1). Para isto, este trabalho definiu a quantidade do dado como sendo o número de vezes em que é referenciado pelos planos do agente. Em relação à valoração, o presente trabalho definiu que cada plano tem uma importância definida pelo projetista do agente, quanto maior este valor, maior é a valoração do dado referenciado no plano.

3. Definir e identificar o impacto da inclusão do processo de Esquecimento no ciclo de raciocínio de um agente BDI.

O processo de Esquecimento foi estendido ao ponto de se tornar computacionalmente aplicável, eliminando dados antigos, não relevantes ao agente, controlando o tamanho do conjunto de Dados Inativos. O presente trabalho definiu este controle baseado no tempo ou na quantidade de ciclos que determinado dado está presente na mente do agente. Para permitir o funcionamento do processo, além da formalização do Esquecimento (fórmula 3.14), foi necessário definir e formalizar o atributo *momento*, que armazena a informação de quando o dado foi percebido.

O atributo momento não está definido no modelo BDI ou no modelo DBR, nem mesmo a plataforma de desenvolvimento Jason armazena qualquer informação sobre o momento em que o dado foi percebido. A formalização deste atributo permite que o processo de Esquecimento seja possível. A informação de quando o dado foi percebido também pode ser útil em outros processos cognitivos, como o cálculo da Credibilidade: pode-se utilizar o momento em que o dado foi percebido para selecionar um dado entre vários contrastantes entre si.

4. Criar os novos estados mentais no ciclo de raciocínio de agentes BDI, adequando sua arquitetura com a diferença entre Dados e Crenças.

Foi possível incorporar as novas categorias epistêmicas na modelo BDI, esta incorporação permitiu a classificação dos dados em diferentes categorias. O presente trabalho formalizou, na seção 3.2, as diferenças entre os Dados Ativos, Dados Inativos, Crenças Ativas e Crenças Inativas. Além disto, foi possível identificar que devido aos diferentes modelos de funcionamento de um agente BDI, são necessárias diferentes formas de armazenamento dos dados no conjunto de Dados Inativos.

Para contemplar os cenários destes modelos de funcionamento, foram propostos três diferentes modelos de armazenamento de Dados Inativos: DIP, DIPP e DIT (seção 3.3). Cada um dos modelos altera a maneira que o dado é utilizado no ciclo de raciocínio de um agente. O DIP fornece um armazenamento persistente, uma "memória de dados", com ele o agente pode utilizar em seu ciclo de raciocínio atual dados percebidos em ciclos anteriores. O DIPP também fornece uma memória ao agente, porém, esta memória só pode ser utilizada para consultas, o dado percebido em outros ciclos de raciocínio não é utilizado no ciclo atual do agente. Quando se trata do modelo de armazenamento DIT, o conjunto de Dados Inativos é volátil, dura somente o ciclo de raciocínio atual do agente.

5. Analisar o ganho computacional do modelo em diferentes cenários.

O filtro de percepções adiciona processos que aumentam a complexidade computacional do ciclo de raciocínio. Em ambientes onde inexitem ou existem poucos ruídos, a redução

da base de crenças não compensa o custo do filtro de percepções como é apresentado nas análises em 5.3.

Porém, nos experimentos, foi possível constatar que em cenários onde o ambiente fornece muitos ruídos (percepções não relevantes), o agente que implementa o filtro de percepções apresenta um menor tempo de execução (mais autonomia) quando comparado com o agente que não implementa o modelo. Quanto maior o número de ruídos, maior é o desempenho do filtro quando comparado com o modelo que não o utiliza.

6.1.1 Implementação

A aplicação do modelo de filtragem na plataforma Jason permitiu a análise concreta do modelo. Mais que isso, o presente trabalho pode estender o Jason, fornecendo um controle de percepções bem fundamentado, automatizado, genérico e de fácil utilização. O capítulo 4 mostra quais pontos da arquitetura do Jason foram alterados para comportar o modelo de filtro de percepções proposto por este trabalho. O presente trabalho, também, abriu o caminho para uma completa implementação do modelo DBR em uma plataforma de desenvolvimento de agentes.

6.1.2 Trabalhos futuros

Os itens do DBR não cobertos por esse trabalho dependem de novos trabalhos para que possam se tornar processos práticos, implementados e validados dentro do ciclo de raciocínio de um agente BDI, são eles:

1. Criação e extensão do processo de Seleção de Crenças
2. Sugestões de modelos para determinar a Credibilidade
3. Sugestões de modelos para determinar a Importância
4. Sugestões de modelos para determinar a Atratividade
5. Sugestões de modelos e processos de tratamento de inconsistências na base de crenças
6. Extensão do processo de Esquecimento para atuar nas Crenças Inativas

Fora isto, os itens cobertos no escopo deste trabalho ainda precisam de mais experimentos e/ou refinamentos:

1. Aplicação e análise dos modelos de cálculo de Relevância em outros cenários
2. Utilização dos Dados Inativos a partir da Seleção de Crenças (ou outros processos)
3. Aplicação do processo de Foco em robôs que atuam em ambientes físicos
4. Extensão das estruturas de dados para comportar os outros atributos do dado
5. Comparação do desempenho do modelo com outros modelos de filtro de percepções.

Referências Bibliográficas

- [Alchourron et al., 1985] Alchourron, C., Gardenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, pages 50:510–530.
- [Alechina et al., 2006] Alechina, N., Bordini, R. H., Hubner, J. F., and Logan, B. (2006). Automating belief revision for agentspeak. In *Proceedings of the Fourth International Workshop on Declarative Agent Languages and Technologies*.
- [Alechina et al., 2005] Alechina, N., Jago, M., and Logan, B. (2005). Resource-bounded belief revision and contraction. In *Proceedings of the 3rd International Workshop on Declarative Agent Languages and Technologies*.
- [Alechina and Logan, 2002] Alechina, N. and Logan, B. (2002). Ascribing beliefs to resource bounded agents. In *Proceedings of AAMAS-2002*.
- [Alechina and Logan, 2010] Alechina, N. and Logan, B. (2010). Belief ascription under bounded resources. *Synthese*, 173(2):179–197.
- [Ancona et al., 2004] Ancona, D., Mascardi, V., Hubner, J. F., and Bordini, R. H. (2004). Coagentspeak: Cooperation in agentspeak through plan exchange. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 698–705, New York, NY.
- [Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D., and Patel-Schneider, P. (2003). *Handbook of Description Logics*. Cambridge University Press.
- [Bordini and Hübner, 2005] Bordini, R. H. and Hübner, J. F. (2005). Bdi agent programming in agentspeak using jason. In *IN: PROCEEDINGS OF 6TH INTERNATIONAL WORKSHOP ON COMPUTATIONAL LOGIC IN MULTI-AGENT SYSTEMS (CLIMA VI). VOLUME 3900 OF LNCS*, pages 143–164. Springer.
- [Bordini et al., 2007] Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons.
- [Boulic et al., 1997] Boulic, R., Bécheiraz, P., Emering, L., and Thalmann, D. (1997). Integration of motion control techniques for virtual human and avatar real-time animation. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 111–118, New York, NY, USA. ACM.

- [Boutilier, 1998] Boutilier, C. (1998). A unified model of qualitative belief change: A dynamical systems perspective. *Artificial Intelligence*, 98(1-2):281–316.
- [Bratman, 1999] Bratman, M. E. (1999). *Intention, Plans, and Practical Reason*. Cambridge University Press.
- [Clark and McCabe, 2007] Clark, K. L. and McCabe, F. G. (2007). Ontology schema for an agent belief store. *Int. J. Hum.-Comput. Stud.*, 65(7):640–658.
- [D’Inverno et al., 2004] D’Inverno, M., Luck, M., Georgeff, M., Kinny, D., and Wooldridge, M. (2004). The dmars architecture: A specification of the distributed multi-agent reasoning system. *Autonomous Agents and Multi-Agent Systems*, 9(1-2):5–53.
- [Fermé and Hansson, 2011] Fermé, E. and Hansson, S. (2011). Agm 25 years. *Journal of Philosophical Logic*, 40(2):295–331.
- [Galliers, 1992] Galliers, J. R. (1992). Autonomous belief revision and communication. In Gärdenfors, P., editor, *Belief Revision*, pages 220–246. Cambridge University Press. Cambridge Books Online.
- [Georgeff and Lansky, 1987] Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 2*, pages 677–682, Seattle, Washington. AAAI Press.
- [Moreira et al., 2004] Moreira, A. F., Vieira, R., and Bordini, R. H. (2004). Extending the operational semantics of a bdi agent-oriented programming language for introducing speech-act based communication. In Leite, J., Omicini, A., Sterling, L., and Torroni, P., editors, *Declarative Agent Languages and Technologies*, volume 2990, pages 135–154. Springer Berlin Heidelberg.
- [Moreira et al., 2005] Moreira, A. F., Vieira, R., Bordini, R. H., and Hübner, J. (2005). Agent-oriented programming with underlying ontological reasoning. In *In Baldoni, M., Endriss, U., Omicini, A., and Torroni, P. (Eds.), Proceedings of the Third International Workshop on Declarative Agent Languages and Technologies (DALT-05), held with AAMAS-05, 25th of July*, pages 155–170. Springer-Verlag.
- [Morveli-Espinoza et al., 2015] Morveli-Espinoza, M., Possebom, A., Mendes, G., and Tacla, C. (2015). Using argumentation for cooperative decision making process. In *Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on*, pages 431–436.
- [Paglieri, 2004] Paglieri, F. (2004). Data-oriented belief revision: Towards a unified theory of epistemic processing. In *Proceeding of STAIRS 2004*. IOS Press.
- [Paglieri and Castelfranchi, 2004] Paglieri, F. and Castelfranchi, C. (2004). Revising beliefs through arguments: Bridging the gap between argumentation and belief revision in mas. In *In: Proceedings of the 1st workshop on Argumentation in MAS (ArgMAS)*, pages 78–94. Springer.
- [Pollock and Gillies, 2000] Pollock, J. and Gillies, A. (2000). Belief revision and epistemology. *Synthese*, 122(1-2):69–92.

- [Rao, 1996a] Rao, A. (1996a). Agentspeak(1): Bdi agents speak out in a logical computable language. In Van de Velde, W. and Perram, J., editors, *Agents Breaking Away*, volume 1038 of *Lecture Notes in Computer Science*, pages 42–55. Springer Berlin Heidelberg.
- [Rao, 1996b] Rao, A. (1996b). Decision procedures for prepositional linear-time belief-desire-intention logics. In Wooldridge, M., Müller, J., and Tambe, M., editors, *Intelligent Agents II Agent Theories, Architectures, and Languages*, volume 1037 of *Lecture Notes in Computer Science*, pages 33–48. Springer Berlin Heidelberg.
- [Rao and Georgeff, 1995] Rao, A. S. and Georgeff, M. P. (1995). Bdi agents: From theory to practice. In *IN PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS (ICMAS-95)*, pages 312–319.
- [Segerberg, 1999] Segerberg, K. (1999). *Two Traditions in the Logic of Belief: Bringing them Together*, volume 5 of *Trends in Logic*, pages 135–147. Springer Netherlands.
- [Stabile and Sichman, 2015] Stabile, M. F. and Sichman, J. S. (2015). Incorporando filtros de percepção para aumentar o desempenho de agentes jason. In *Anais do IX Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações*, volume 9, Niterói, RJ.
- [van Harmelen et al., 2007] van Harmelen, F., van Harmelen, F., Lifschitz, V., and Porter, B. (2007). *Handbook of Knowledge Representation*. Elsevier Science, San Diego, USA.
- [van Oijen and Dignum, 2011] van Oijen, J. and Dignum, F. (2011). Scalable perception for bdi-agents embodied in virtual environments. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 02*, pages 46–53, Washington, DC, USA. IEEE Computer Society.
- [Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10:115–152.

Apêndice A

Plataforma de desenvolvimento Jason

A.1 Agentes BDI e o interpretador AgentSpeak

Um agente é uma entidade computacional ativa que percebe e atua em um ambiente. Em resumo, uma informação percebida através dos sensores de um agente é processada e então é escolhida qual a melhor ação a ser tomada para atuar no ambiente com o propósito de atingir um objetivo. Os ambientes que um agente atua podem ser físicos (robôs atuando em um ambiente físico) ou lógicos (software atuando em um sistema operacional). Em geral, um agente tem pouco controle sobre o ambiente, ou seja, o agente pode atuar e modificar uma parte do ambiente, mas não tem o controle das ações aplicadas por outros agentes.

Para [Wooldridge and Jennings, 1995], além de perceber e atuar em um ambiente, agentes racionais precisam ser autônomos, pró-ativos, reativos e ter habilidades social:

1. **Autonomia:** Um agente computacional autônomo tem que ser capaz de realizar operações com independência (sem depender de outros agentes), com o intuito de atingir os objetivos a ele delegados. Estes objetivos são guiados pelos planos fornecidos ao agente, que definem as maneiras que ele pode agir para atingir suas metas.
2. **Proatividade:** Um agente proativo é aquele capaz de exibir um comportamento orientado aos objetivos. Dado um objetivo específico, ele tentará alcançá-lo. Diferente de um agente passivo, que só fará algo caso seja instigado, como por exemplo um *web-service*, nada irá acontecer até que seja chamado um de seus métodos.
3. **Reatividade:** Um agente reativo é aquele que é suscetível às alterações no ambiente e muda suas ações e comportamentos em função delas.
4. **Habilidade Social:** É a capacidade do agente de cooperar e coordenar suas atividades com outros agentes, com o intuito de atingir seus objetivos.

A arquitetura BDI (*belief-desire-intention*) de [Bratman, 1999] é a mais conhecida e a mais estudada arquitetura de agentes cognitivos. A ideia central do modelo BDI é que agentes tenham um estado mental com crenças, desejos e intenções. Crenças são informações que o agente tem do mundo, desejos são objetivos que um agente pode querer realizar e intenções são objetivos que o agente se comprometeu a realizar.

A Figura A.1 mostra o processo de raciocínio prático de um agente BDI abstrato. Abaixo, a função de cada um dos seus componentes:

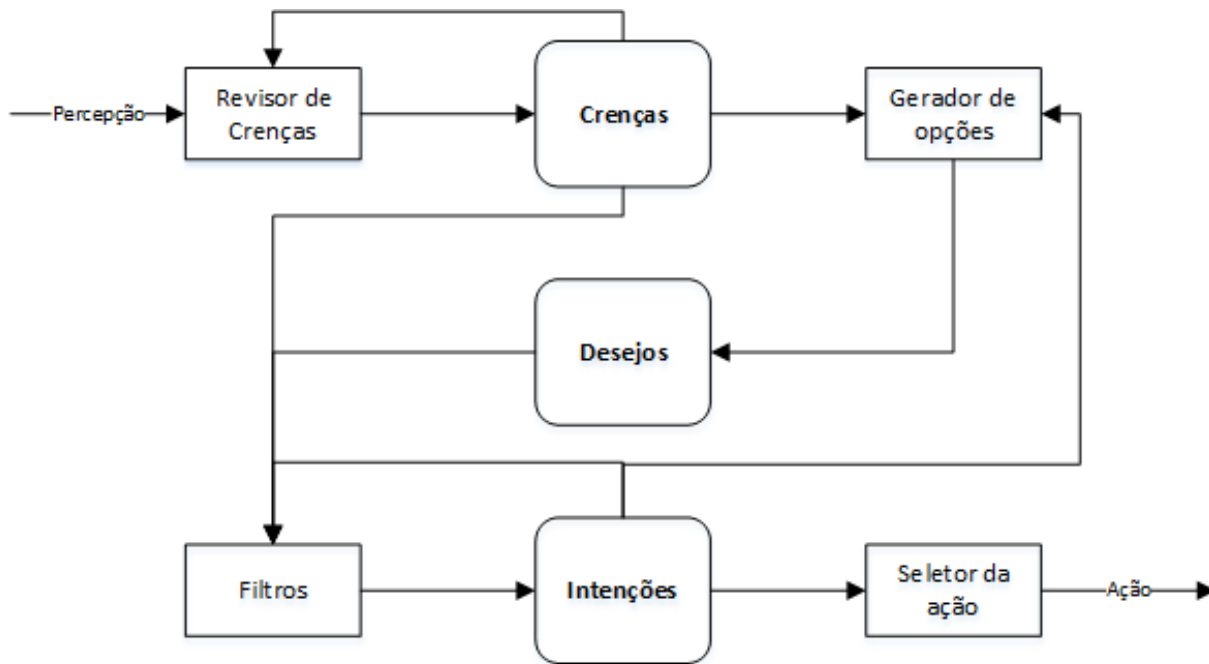


Figura A.1: Componentes BDI.

Fonte: Adaptado de [Bratman, 1999]

1. Revisor de crenças: A partir das percepções do agente e das suas crenças atuais, a revisão de crenças redefine sua base de crenças. O revisor de crenças utiliza lógica não monotônica ¹ para a manutenção da consistência da base.
2. Crenças: O conjunto de informações que representa o ambiente que o agente está atuando.
3. Gerador de opções: Determina os desejos do agente, são criados a partir de suas crenças e intenções atuais. Opções são os planos disponíveis ao agente para realizar seus objetivos.
4. Desejos: As opções de objetivos atuais do agente.
5. Filtros: Representa o processo de deliberação, determina as intenções do agente de acordo com suas crenças, desejos e intenções.
6. Intenções: São os focos de atuação do agente, os objetivos que ele se comprometeu a realizar.
7. Seletor de ação: De acordo com as intenções, determina a ação a ser realizada.

Para [Bordini and Hübner, 2005], enquanto o modelo BDI é a principal arquitetura para agentes cognitivos, o AgentSpeak de [Rao, 1996a] é uma linguagem para a implementação de tais agentes. O AgentSpeak é definido por [Bordini and Hübner, 2005] como uma linguagem de programação orientada a agentes e baseada em programação lógica, inspirada pelos trabalhos

¹Diferente de lógicas monotônicas, onde a base de conhecimento não pode ser reduzida com a chegada de um novo dado, uma lógica não monotônica pode, a partir de um novo dado, reduzir sua base de conhecimento para mantê-la consistente.

sobre arquiteturas BDI de [Rao and Georgeff, 1995], lógicas BDI de [Rao, 1996b], e também sobre implementações práticas de sistemas BDI como o [Georgeff and Lansky, 1987] (PRS) e [D’Inverno et al., 2004] (dMARS). Em sua definição original, o AgentSpeak era simplesmente uma linguagem de programação abstrata. O Jason foi criado utilizando o AgentSpeak como base, acrescido de extensões necessárias para permitir o desenvolvimento de sistemas multiagentes práticos.

A.2 Componentes de um agente no Jason

[Bordini et al., 2007] define que existem três grandes componentes presentes em agentes desenvolvidos no Jason: Crenças, objetivos e planos. Cada um deles com diferentes tipos de estruturas e sintaxes de representação:

1. **Crenças:** Um agente tem uma base de crenças, que representam o conhecimento do agente sobre o ambiente. Como em linguagens de programação de lógica, a informação é representada por predicados e seus termos: *predicado*(*< listadetermos >*) [*listadeanotacoes*], sendo um termo uma constante, uma variável ou uma função. A lista de termos pode ser vazia.

Uma crença pode vir acompanhada de anotações, que são utilizadas para acrescentar informações à crença, como por exemplo: *temperatura*(58)[*sensor*(*esquerdo*)], que em certo contexto significaria que a percepção que a temperatura está em 58° é originada do sensor esquerdo do robô. Por padrão, toda crença tem a anotação *source*, que identifica a origem da informação: percepção do ambiente (*source*(*perception*)), comunicação de outro agente (*source*(*agent*)) ou nota mental adicionada pelo próprio agente (*source*(*self*)).

O motor de inferência sobre a base de crenças utiliza a negação por falha, ou seja, toda fórmula que não puder ser demonstrada verdadeira pelo motor de inferência, é assumida como falsa.

É possível, também, criar regras para uma determinada crença, exemplo:

$$\text{mover}(A,B) : \neg \text{mover}(A,B)[\text{source}(S)] \& (S == \text{percept}).$$

Nesta regra, a crença *mover*(*A,B*) somente será verdadeira quando for obtida por meio de percepção, sendo considerada falsa nos casos em que a fonte da informação seja comunicação ou nota mental.

2. **Objetivos:** Enquanto crenças representam, do ponto de vista do agente, o estado atual do ambiente, objetivos representam o estado em que o agente quer chegar. Quando um agente está comprometido com algum objetivo, ele vai atuar no ambiente até que este objetivo seja verdadeiro.

Existem dois tipos de objetivos: *test goal* e *achievement goal*. O agente utiliza o *test goal* para verificar se a fórmula atômica associada é uma das suas crenças e, portanto, saber que alcançou o objetivo implícito. Um *achievement goal* afirma que o agente quer alcançar um estado do ambiente onde a fórmula atômica associada é verdadeira.

O *test goal* é formado por uma fórmula atômica com o operador '??' como prefixo. O *achievement goal* é prefixado com o operador '!':

- (a) $?posicao(X,Y)$: É verificado se existe uma crença $posicao(X,Y)$ na base de crenças, onde X e Y são variáveis instanciadas em tempo de execução. Caso não exista a crença, é verificado se algum plano pode resolver a fórmula.
 - (b) $!mover(X,Y)$: Quando executado, o agente se compromete a mudar o estado do ambiente até que $!mover(X,Y)$ seja executada.
3. **Planos:** São sequências de ações que o agente executa para atingir seus objetivos, planos são iniciados a partir de alterações nas crenças ou objetivos dos agentes. Um plano no AgentSpeak tem um cabeçalho, que é formado por um evento gatilho; um contexto, definindo as condições que um plano pode ser usado; e um corpo, constituído por uma sequência de ações.

$$gatilho : contexto < -corpo$$

Cada uma das partes do evento tem uma função única importante:

- (a) Evento gatilho: Alterações nas crenças e alterações nos objetivos criam eventos que necessitam uma reação do agente. Um plano se torna candidato a ser executado se um destes eventos corresponder a algum evento gatilho da biblioteca de planos do agente.
- (b) Contexto: Vários planos podem responder a um evento, porém, agentes só irão executar um plano se seu contexto for verdadeiro, ou seja, se o contexto é uma consequência lógica da base de crenças. O Contexto de um plano é usado para verificar a situação atual (estado mental e últimas informações do ambiente) e determinar se um plano, entre vários possíveis, tem as maiores chances de sucesso em tratar o evento disparado.
- (c) Corpo: Contém a sequência de ações a serem executadas pelo plano. O corpo de um plano pode conter seis tipos de fórmulas: atuação no ambiente, *achievement goals*, *test goals*, notas mentais, ações internas e expressões.

É possível acrescentar rótulos (*label*) aos planos. Eles podem simplesmente dar um nome ao plano assim como podem servir como um metadado que, por exemplo, fornecerá informações necessárias ao interpretador para selecionar um determinado plano em detrimento a outro. Rótulos não são obrigatórios, porém são essenciais em agentes complexos que reescrevem funções de seleção ou criam novas tarefas cognitivas (o caso desse projeto) que dependem dos planos do agentes.

Abaixo, no exemplo de um plano, o rótulo *@coletareciclavel* é sua identificação, o evento gatilho é representado pelo predicado binário *posicao* cujos argumentos são duas variáveis (X e Y); o contexto é o predicado unário *lixo* e seu argumento é a constante *reciclavel*; enquanto o corpo tem a ação *aspirar*:

$$@coletareciclavel +!posicao(X,Y) : lixo(reciclavel) <- aspirar.$$

A.3 Ciclo de raciocínio de um agente no Jason

Um programa AgentSpeak é inicialmente determinado por suas crenças e objetivos pré-definidos, um conjunto de eventos e uma biblioteca de planos. As crenças pré definidas são utilizadas para inicializar a base de crenças do agente. Esta inicialização, assim como a inicialização dos objetivos pré definidos, gera a adição de eventos que constituem o conjunto inicial de eventos. O programa também tem planos pré-definidos pelo desenvolvedor que constituem a biblioteca de planos inicial do programa. Quando o agente começa a executar, seu conjunto de intenções está vazio.

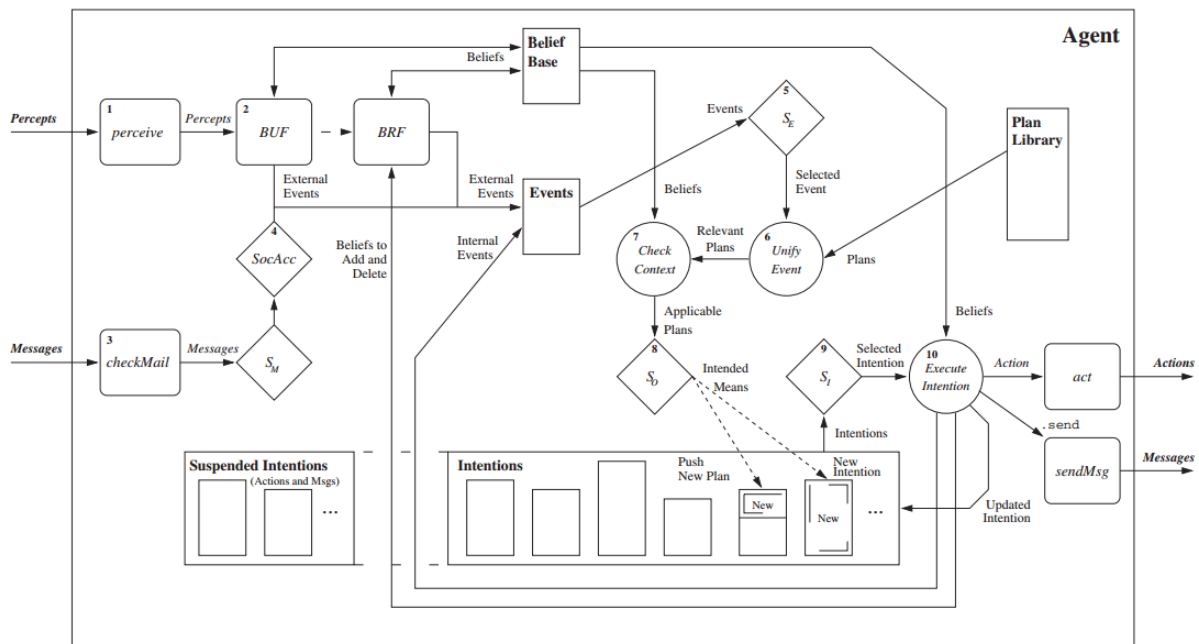


Figura A.2: Ciclo de interpretação do Jason.

Fonte: [Bordini et al., 2007].

[Bordini et al., 2007] resume o ciclo de raciocínio do Jason em dez passos principais, conforme indicados na figura A.2 e listados abaixo:

1. Perceber o ambiente (*perceive*): É a primeira ação do agente dentro do ciclo de raciocínio. A cada ciclo, o agente recebe um conjunto de percepções, cada percepção é representada por um literal que identifica uma propriedade do estado atual do ambiente.
2. Atualizar a base de crenças (*BUF, Belief Update Function e BRF, Belief Revision Function*): O conjunto de percepções do passo anterior é utilizado para atualizar a base de crenças, todas as informações percebidas no ciclo atual são adicionadas na base de crenças. Percepções que não foram recebidas no ciclo atual de raciocínio são removidas da base de crenças. De um modo mais formal, em sua implementação padrão, a atualização da base de crenças funciona da seguinte maneira:

Seja P o conjunto das percepções atuais e B a base de crenças:

$$\forall \phi \in P, \text{ Se } \phi \notin B \text{ então } B \leftarrow B \cup \{\phi\} \quad (\text{A.1})$$

$$\forall \phi \in B, \text{ Se } \phi \notin P \text{ então } B \leftarrow B - \{\phi\} \quad (\text{A.2})$$

Cada alteração na base de crenças, seja uma adição ou uma remoção, gera um novo evento externo que pode disparar um plano.

3. Receber comunicação de outros agentes (*checkMail*): Além de perceber o ambiente, agentes desenvolvidos no Jason podem receber informações de outros agentes. Neste passo do ciclo de raciocínio, o agente verifica se recebeu mensagens de outros agentes. Somente uma mensagem é considerada por ciclo de execução, no caso de recebimento de mais de uma mensagem, a primeira mensagem da lista é considerada.
4. Selecionar mensagens socialmente aceitáveis (*SocAcc*): Antes de processar as mensagens recebidas por outros agentes, este processo executa uma seleção, determinando qual mensagem pode ser considerada como uma nova percepção pelo agente. O comportamento padrão é aceitar qualquer mensagem, cabe aos desenvolvedores dos agentes definir métodos mais sofisticados para indicar a aceitação das mensagens.
5. Selecionar um evento (S_E): Alterações no ambiente e alterações nos objetivos do agente geram eventos. O Jason está continuamente manipulando eventos, em cada ciclo de raciocínio somente um evento é tratado. A função de seleção deste processo é a S_E , na sua implementação padrão, se a iteração do ciclo de raciocínio gerou mais de um evento, a primeira da lista será considerada e as outras entram em uma fila que será considerada nos próximos ciclos. Caso o conjunto de eventos esteja vazio, o interpretador pula para o passo nove do ciclo de interpretação.
6. Recuperar planos relevantes (*Unify Event*): Com o evento selecionado, o Jason procura, na *Plan Library*, por todos os planos relevantes ao evento. Isto é feito considerando todos os gatilhos de eventos que podem ser unificados com o evento selecionado.
7. Determinar planos aplicáveis (*Check Context*): Após identificar os planos relevantes, é necessário selecionar os planos aplicáveis. Para isto é necessário verificar o contexto de cada um dos planos relevantes. Um plano aplicável é aquele com o contexto verdadeiro, ou seja, aquele que o contexto é uma consequência lógica da base de crenças do agente. Mais de um plano pode ser aplicável para um determinado evento.
8. Selecionar um plano aplicável S_O : O agente escolhe apenas um dos possíveis planos aplicáveis e se compromete em executá-lo. Se comprometer com um plano significa que o agente tem a *intenção* de executar as ações propostas nele. A intenção escolhida é chamada de *intended means* e adicionada ao conjunto de intenções do agente. A função de seleção deste processo é o S_O , na sua implementação padrão, no caso em que mais de um plano é aplicável, o que aparecer antes na lista de planos será o escolhido.

Novas intenções podem ser criadas a partir de eventos internos (alterações nos objetivos) ou eventos externos (percepções do ambiente). Cada uma delas é incluída de uma forma diferente no conjunto de intenções:

- (a) Eventos externos: Este tipo de evento é criado a partir de percepções no ambiente, cada *intended means* originado a partir deste tipo de evento cria uma nova intenção no conjunto de intenções. Cada intenção do conjunto significa um foco de atenção diferente. O agente pode dividir sua atenção entre duas ou mais intenções, atendendo-as concorrentemente.
 - (b) Eventos internos: Este tipo de evento é criado quando, durante a execução de uma intenção, o agente recebe um novo objetivo a ser cumprido. Ou seja, um plano "filho" é criado a partir de um plano "pai" e aquele deve ser executado e cumprido antes deste continuar. Neste caso, em vez de se criar uma nova intenção, a *intended means* é adicionada no topo da pilha de *intended means* da intenção atual.
9. Selecionar uma intenção para execução (S_I): Em cada ciclo, uma fórmula de uma intenção do conjunto de intenções é escolhida para execução. A função de seleção deste processo é a S_I , na sua implementação padrão, as intenções do conjunto são alternadamente selecionadas e uma fórmula da intenção escolhida é executada. Isto significa que em um agente com várias intenções, ou seja, com diferentes focos de atenção, os focos são alternadamente selecionados e um é executado a cada ciclo do raciocínio.
10. Executar um passo de uma intenção (*Execute Intention*): A fórmula escolhida pelo passo anterior é executada, ela pertence ao corpo do plano que originou a intenção e pode ser de seis tipos diferentes, tratadas diferentemente pelo interpretador:
- (a) Ação no ambiente: O agente executa uma ação e a intenção fica suspensa até que a atuação no ambiente seja finalizada. Suspendendo a ação, o agente pode utilizar os ciclos de raciocínios posteriores para executar outras intenções do conjunto de intenções, voltando seu foco para a intenção suspensa quando a alteração no ambiente for finalizada. A fórmula só é removida do corpo do plano depois da resposta (sucesso ou falha) da ação.
 - (b) *Achievement Goals*: No caso em que a fórmula executada é um objetivo filho (*sub-goal*), a intenção é suspensa até que o objetivo filho seja executado. Este tipo de fórmula cria um novo evento (derivado do objetivo filho), isto significa que no próximo ciclo o interpretador irá considerar este evento na lista de eventos e ele poderá ser selecionado pela S_E (etapa 5 da Figura A.2). Somente depois da execução do objetivo filho a intenção (objetivo pai) será removida do conjunto de intenções suspensas, voltará para o conjunto de intenções e a fórmula será removida do corpo de seu plano.
 - (c) *Test Goals*: O agente tenta unificar a fórmula com sua base de crenças atual, caso o teste seja positivo, a intenção volta ao conjunto de intenções e pode ser selecionada novamente no próximo ciclo do interpretador. Caso o teste falhe, ou seja, a informação não está no conjunto de crenças, é verificado se existe um plano relevante para o teste, caso exista, um novo evento é criado para ser tratado nos próximos ciclos. Neste caso a intenção ficará suspensa até o fim do *test goal*.
 - (d) Notas mentais: Dentro do ciclo de interpretação atual, a nova informação é adicionada à base de crenças, a fórmula é removida do corpo do plano e a intenção volta ao conjunto de intenções.

- (e) Ações internas: O código da ação é completamente executada dentro do ciclo atual, a fórmula é removida e a intenção volta ao conjunto de intenções, podendo ser utilizada nos próximos ciclos.
- (f) Expressões: São avaliadas no ciclo atual do agente, a fórmula é removida e a intenção volta ao conjunto de intenções.

No final de cada ciclo, o interpretador verifica se existe algum retorno das ações (ações no ambiente, comunicação de outros agentes) das intenções suspensas, caso existam, as intenções são atualizadas e retornam para o conjunto de intenções. Também, no final do ciclo, as intenções finalizadas (aquelas que executaram todas as fórmulas do corpo do plano) são removidas.

Apêndice B

Informações sobre os experimentos

B.1 Planos dos agentes do problema sintético criado

As subseções a seguir contêm os planos utilizados para executar os experimentos do Capítulo 5.

B.1.1 Planos do agente N=1

Para o agente com 1 plano:

```
1 pos(0) .
2 +pos(0) <- mover.
3 +pos(1000) <- .stopMAS.
4 +pos(X) <- !dehec(X,X) .
5 @plano1[valor_objetivo(1)] +!dehec(X,Y) : dfh(b) & fdg(b) <- mover.
```

B.1.2 Planos do agente N=50

Para o agente com 50 planos:

```
1 pos(0) .
2 +pos(0) <- mover.
3 +pos(1000) <- .stopMAS.
4 +pos(X) <- !chcgd(X,X) .
5 @plano1[valor_objetivo(3)] +!hdfdd(X,X) : abc(f) & dda(h) <- mover.
6 @plano2[valor_objetivo(6)] +!fccgb(X,Y) : hhc(b) & fdc(h) <- !hdfdd(X,X) .
7 @plano3[valor_objetivo(6)] +!fbcga(X,Y) : cad(c) & fcg(a) <- !fccgb(X,Y) .
8 @plano4[valor_objetivo(2)] +!bdfbb(X,Y) : edg(a) & ada(f) <- !fbcga(X,Y) .
9 @plano5[valor_objetivo(9)] +!hbheg(X,Z) : cbg(e) & dbh(h) <- !bdfbb(X,Y) .
10 @plano6[valor_objetivo(2)] +!bfeag(X,X) : ebe(g) & dba(a) <- !hbheg(X,Z) .
11 @plano7[valor_objetivo(4)] +!egedc(X,Z) : bhf(h) & ffg(d) <- !bfeag(X,X) .
12 @plano8[valor_objetivo(8)] +!gcfdd(X,Z) : ead(d) & gbg(b) <- !egedc(X,Z) .
13 @plano9[valor_objetivo(2)] +!ghcac(X,X) : ehe(e) & bde(c) <- !gcfdd(X,Z) .
14 @plano10[valor_objetivo(6)] +!hedgh(X,X) : ehe(h) & gcd(a) <- !ghcac(X,X) .
15 @plano11[valor_objetivo(2)] +!dgegh(X,Y) : hbb(c) & dbb(d) <- !hedgh(X,X) .
16 @plano12[valor_objetivo(8)] +!hfged(X,X) : dag(d) & fed(c) <- !dgegh(X,Y) .
17 @plano13[valor_objetivo(3)] +!agcbh(X,Z) : bga(g) & bdg(g) <- !hfged(X,X) .
18 @plano14[valor_objetivo(1)] +!egehg(X,Y) : ahg(f) & bad(a) <- !agcbh(X,Z) .
19 @plano15[valor_objetivo(5)] +!defeb(X,X) : hdh(a) & ceh(c) <- !egehg(X,Y) .
```

```

20 @plano16[valor_objetivo(10)] +!gbgfa(X,X) : edd(h) & dcc(g) <- !defeb(X,X) .
21 @plano17[valor_objetivo(8)] +!ccebd(X,Z) : gab(e) & aba(g) <- !gbgfa(X,X) .
22 @plano18[valor_objetivo(2)] +!bechc(X,X) : hac(c) & acc(f) <- !ccebd(X,Z) .
23 @plano19[valor_objetivo(6)] +!dbdae(X,Y) : fgb(d) & cac(f) <- !bechc(X,X) .
24 @plano20[valor_objetivo(5)] +!chfde(X,Y) : gdg(g) & gdf(g) <- !dbdae(X,Y) .
25 @plano21[valor_objetivo(5)] +!ahcec(X,Y) : fgg(h) & ecf(g) <- !chfde(X,Y) .
26 @plano22[valor_objetivo(2)] +!dehfe(X,Y) : fgg(f) & eac(a) <- !ahcec(X,Y) .
27 @plano23[valor_objetivo(2)] +!heeea(X,Z) : dea(d) & eag(e) <- !dehfe(X,Y) .
28 @plano24[valor_objetivo(3)] +!hfeh(X,X) : bdh(c) & afh(b) <- !heeea(X,Z) .
29 @plano25[valor_objetivo(6)] +!adfdb(X,Y) : bbf(h) & ahf(f) <- !hfeh(X,X) .
30 @plano26[valor_objetivo(7)] +!cdebh(X,X) : egd(e) & cdh(e) <- !adfdb(X,Y) .
31 @plano27[valor_objetivo(5)] +!cbgba(X,Y) : bfd(d) & ehe(g) <- !cdebh(X,X) .
32 @plano28[valor_objetivo(2)] +!hbda(X,X) : cbg(c) & ada(h) <- !cbgba(X,Y) .
33 @plano29[valor_objetivo(4)] +!fdfad(X,X) : bdb(e) & gff(f) <- !hbda(X,X) .
34 @plano30[valor_objetivo(3)] +!ddhaa(X,Y) : dch(h) & ahc(c) <- !fdfad(X,X) .
35 @plano31[valor_objetivo(3)] +!gaaea(X,Y) : dhc(e) & bfh(h) <- !ddhaa(X,Y) .
36 @plano32[valor_objetivo(5)] +!ddbac(X,Y) : afd(h) & dff(a) <- !gaaea(X,Y) .
37 @plano33[valor_objetivo(10)] +!agggg(X,Z) : egh(d) & cbe(f) <- !ddbac(X,Y) .
38 @plano34[valor_objetivo(4)] +!deacd(X,Z) : hbh(c) & agg(g) <- !agggg(X,Z) .
39 @plano35[valor_objetivo(8)] +!dbbef(X,X) : cfg(e) & bee(g) <- !deacd(X,Z) .
40 @plano36[valor_objetivo(8)] +!chgg(X,Z) : bhg(b) & edf(f) <- !dbbef(X,X) .
41 @plano37[valor_objetivo(9)] +!ceaeg(X,Y) : dfh(d) & acc(g) <- !chgg(X,Z) .
42 @plano38[valor_objetivo(7)] +!fagag(X,Z) : abb(d) & dgg(c) <- !ceaeg(X,Y) .
43 @plano39[valor_objetivo(5)] +!eagbh(X,Y) : gga(h) & cab(c) <- !fagag(X,Z) .
44 @plano40[valor_objetivo(3)] +!bffba(X,Y) : fge(d) & cfc(f) <- !eagbh(X,Y) .
45 @plano41[valor_objetivo(8)] +!gbafb(X,X) : ebd(h) & hcf(a) <- !bffba(X,Y) .
46 @plano42[valor_objetivo(1)] +!fddfa(X,Z) : hdg(d) & hbd(a) <- !gbafb(X,X) .
47 @plano43[valor_objetivo(8)] +!afb(X,X) : dhh(f) & cgd(d) <- !fddfa(X,Z) .
48 @plano44[valor_objetivo(8)] +!bfdhg(X,Y) : bce(a) & haf(b) <- !afb(X,X) .
49 @plano45[valor_objetivo(5)] +!haddf(X,Z) : dhh(f) & dcd(c) <- !bfdhg(X,Y) .
50 @plano46[valor_objetivo(4)] +!dhfg(X,Z) : adh(c) & bcf(c) <- !haddf(X,Z) .
51 @plano47[valor_objetivo(4)] +!bgfgd(X,Z) : ghc(c) & cec(e) <- !dhfg(X,Z) .
52 @plano48[valor_objetivo(1)] +!dfdae(X,X) : fgf(h) & bea(b) <- !bgfgd(X,Z) .
53 @plano49[valor_objetivo(5)] +!hhaeh(X,X) : fff(c) & ccg(h) <- !dfdae(X,X) .
54 @plano50[valor_objetivo(5)] +!chcgd(X,Z) : ahe(g) & dgc(b) <- !hhaeh(X,X) .

```

B.1.3 Planos do agente N=100

Para o agente com 100 planos:

```

1 pos(0) .
2 +pos(0) <- mover .
3 +pos(1000) <- .stopMAS .
4 +pos(X) <- !dehec(X,X) .
5 @plano1[valor_objetivo(6)] +!aeaeh(X,Z) : bde(a) & gab(g) <- mover .
6 @plano2[valor_objetivo(1)] +!dcgga(X,Z) : edf(c) & hgh(b) <- !aeaeh(X,Z) .
7 @plano3[valor_objetivo(1)] +!gcgec(X,X) : eef(f) & bgf(h) <- !dcgga(X,Z) .
8 @plano4[valor_objetivo(6)] +!gef(X,Y) : bed(e) & bge(c) <- !gcgec(X,X) .
9 @plano5[valor_objetivo(2)] +!bfegf(X,Y) : hga(f) & dbh(f) <- !gef(X,Y) .
10 @plano6[valor_objetivo(8)] +!becaa(X,X) : dab(g) & cfa(c) <- !bfegf(X,Y) .
11 @plano7[valor_objetivo(3)] +!effdg(X,Z) : dfe(e) & ahf(b) <- !becaa(X,X) .
12 @plano8[valor_objetivo(2)] +!abfcd(X,Z) : gfe(d) & hhh(e) <- !effdg(X,Z) .
13 @plano9[valor_objetivo(8)] +!dfdfb(X,Z) : eed(h) & gdb(e) <- !abfcd(X,Z) .
14 @plano10[valor_objetivo(5)] +!dbcgb(X,X) : cba(a) & bfh(b) <- !dfdfb(X,Z) .
15 @plano11[valor_objetivo(1)] +!cgbea(X,Y) : bah(f) & ggg(e) <- !dbcgb(X,X) .

```

```

16 @plano12[valor_objetivo(2)] +!cbadf(X,X) : hdb(a) & agg(a) <- !cgbea(X,Y) .
17 @plano13[valor_objetivo(9)] +!efbab(X,Y) : fgh(c) & feg(a) <- !cbadf(X,X) .
18 @plano14[valor_objetivo(6)] +!cafah(X,Z) : gch(f) & ccc(g) <- !efbab(X,Y) .
19 @plano15[valor_objetivo(6)] +!ahddg(X,Z) : dab(a) & ebc(b) <- !cafah(X,Z) .
20 @plano16[valor_objetivo(6)] +!dcffh(X,X) : gch(g) & hgf(b) <- !ahddg(X,Z) .
21 @plano17[valor_objetivo(1)] +!edchb(X,X) : edc(b) & chd(c) <- !dcffh(X,X) .
22 @plano18[valor_objetivo(9)] +!afhdc(X,Y) : fae(f) & fdb(f) <- !edchb(X,X) .
23 @plano19[valor_objetivo(6)] +!fhbeh(X,Y) : cge(h) & hgf(g) <- !afhdc(X,Y) .
24 @plano20[valor_objetivo(7)] +!bhgde(X,Z) : fad(h) & bgb(d) <- !fhbeh(X,Y) .
25 @plano21[valor_objetivo(2)] +!gdgcd(X,Y) : caa(f) & cdc(h) <- !bhgde(X,Z) .
26 @plano22[valor_objetivo(6)] +!dbech(X,X) : eee(c) & dda(c) <- !gdgcd(X,Y) .
27 @plano23[valor_objetivo(4)] +!bcdcf(X,Z) : dec(a) & gch(d) <- !dbech(X,X) .
28 @plano24[valor_objetivo(4)] +!efgea(X,Y) : ghg(b) & aag(h) <- !bcdcf(X,Z) .
29 @plano25[valor_objetivo(1)] +!hhchg(X,X) : eha(g) & aab(a) <- !efgea(X,Y) .
30 @plano26[valor_objetivo(5)] +!dggca(X,X) : fae(g) & ggh(f) <- !hhchg(X,X) .
31 @plano27[valor_objetivo(7)] +!gbahc(X,X) : ggb(e) & geh(d) <- !dggca(X,X) .
32 @plano28[valor_objetivo(2)] +!ggfhe(X,X) : ffb(e) & cfc(d) <- !gbahc(X,X) .
33 @plano29[valor_objetivo(7)] +!adfda(X,Y) : edd(f) & bed(h) <- !ggfhe(X,X) .
34 @plano30[valor_objetivo(1)] +!dggfc(X,Z) : fdb(d) & hbf(c) <- !adfda(X,Y) .
35 @plano31[valor_objetivo(4)] +!fhhfe(X,X) : hcg(e) & hgf(c) <- !dggfc(X,Z) .
36 @plano32[valor_objetivo(4)] +!hgege(X,Y) : ech(b) & gdb(h) <- !fhhfe(X,X) .
37 @plano33[valor_objetivo(9)] +!hgcha(X,X) : ehb(b) & ach(h) <- !hgege(X,Y) .
38 @plano34[valor_objetivo(8)] +!aeae(X,Z) : edf(b) & dae(d) <- !hgcha(X,X) .
39 @plano35[valor_objetivo(7)] +!dgagf(X,Y) : fce(e) & gba(d) <- !aeae(X,Z) .
40 @plano36[valor_objetivo(3)] +!chhhf(X,Y) : fhb(d) & dca(a) <- !dgagf(X,Y) .
41 @plano37[valor_objetivo(8)] +!aaabg(X,X) : adb(h) & ebh(c) <- !chhhf(X,Y) .
42 @plano38[valor_objetivo(6)] +!acebg(X,Y) : dcc(f) & fhd(a) <- !aaabg(X,X) .
43 @plano39[valor_objetivo(4)] +!egbfe(X,Z) : bdh(a) & cdb(b) <- !acebg(X,Y) .
44 @plano40[valor_objetivo(2)] +!bgebd(X,Z) : hcc(e) & fgf(g) <- !egbfe(X,Z) .
45 @plano41[valor_objetivo(9)] +!hedga(X,Z) : eed(c) & dhc(e) <- !bgebd(X,Z) .
46 @plano42[valor_objetivo(3)] +!fbgce(X,X) : gea(d) & gfa(c) <- !hedga(X,Z) .
47 @plano43[valor_objetivo(9)] +!dhbea(X,Z) : bdc(e) & chh(a) <- !fbgce(X,X) .
48 @plano44[valor_objetivo(3)] +!hfgcc(X,X) : fge(h) & dbb(a) <- !dhbea(X,Z) .
49 @plano45[valor_objetivo(2)] +!efhfb(X,Y) : hda(e) & aba(h) <- !hfgcc(X,X) .
50 @plano46[valor_objetivo(6)] +!ahhdc(X,Y) : bad(h) & dee(e) <- !efhfb(X,Y) .
51 @plano47[valor_objetivo(4)] +!ecgca(X,Y) : cee(h) & aeb(g) <- !ahhdc(X,Y) .
52 @plano48[valor_objetivo(9)] +!cdffd(X,X) : ebb(d) & hff(h) <- !ecgca(X,Y) .
53 @plano49[valor_objetivo(9)] +!bbbbbb(X,X) : haa(f) & bdg(f) <- !cdffd(X,X) .
54 @plano50[valor_objetivo(8)] +!fdddd(X,X) : hdg(h) & gff(g) <- !bbbbbb(X,X) .
55 @plano51[valor_objetivo(2)] +!ddgdb(X,X) : haf(e) & cff(d) <- !fdddd(X,X) .
56 @plano52[valor_objetivo(3)] +!beaad(X,Y) : hef(h) & ghg(a) <- !ddgdb(X,X) .
57 @plano53[valor_objetivo(8)] +!dccgg(X,Z) : hhf(e) & hfa(b) <- !beaad(X,Y) .
58 @plano54[valor_objetivo(4)] +!gadec(X,Z) : efc(c) & dca(f) <- !dccgg(X,Z) .
59 @plano55[valor_objetivo(1)] +!dbaaa(X,Z) : fge(a) & fcb(e) <- !gadec(X,Z) .
60 @plano56[valor_objetivo(3)] +!dggdb(X,Y) : cgg(d) & dfd(f) <- !dbaaa(X,Z) .
61 @plano57[valor_objetivo(3)] +!chdeh(X,Z) : caa(a) & efb(h) <- !dggdb(X,Y) .
62 @plano58[valor_objetivo(3)] +!fbbdg(X,X) : cfb(b) & dce(e) <- !chdeh(X,Z) .
63 @plano59[valor_objetivo(10)] +!dfcdd(X,Z) : bad(f) & ecb(b) <- !fbbdg(X,X) .
64 @plano60[valor_objetivo(1)] +!eahcc(X,Z) : bhb(b) & acd(e) <- !dfcdd(X,Z) .
65 @plano61[valor_objetivo(5)] +!ecfcc(X,Z) : egg(h) & bfg(g) <- !eahcc(X,Z) .
66 @plano62[valor_objetivo(9)] +!bebga(X,X) : bee(f) & gbf(f) <- !ecfcc(X,Z) .
67 @plano63[valor_objetivo(10)] +!bbgbc(X,Z) : dcf(e) & gcg(h) <- !bebga(X,X) .
68 @plano64[valor_objetivo(2)] +!eeehb(X,Y) : dfg(e) & ccf(e) <- !bbgbc(X,Z) .
69 @plano65[valor_objetivo(6)] +!cbhde(X,Z) : hbg(h) & ebb(f) <- !eeehb(X,Y) .
70 @plano66[valor_objetivo(2)] +!hgcfg(X,Y) : dcc(b) & ceb(e) <- !cbhde(X,Z) .
71 @plano67[valor_objetivo(7)] +!cgaga(X,Z) : dgg(a) & cfg(c) <- !hgcfg(X,Y) .

```

```

72 @plano68[valor_objetivo(10)] +!ffgdf(X,X) : ddh(b) & fgf(f) <- !cgaga(X,Z) .
73 @plano69[valor_objetivo(8)] +!ghdgd(X,Y) : dfb(d) & gef(c) <- !ffgdf(X,X) .
74 @plano70[valor_objetivo(1)] +!dbghg(X,Z) : hcb(b) & ede(b) <- !ghdgd(X,Y) .
75 @plano71[valor_objetivo(7)] +!dgece(X,Z) : faa(e) & heg(c) <- !dbghg(X,Z) .
76 @plano72[valor_objetivo(6)] +!gaefa(X,Y) : bgf(f) & hgd(h) <- !dgece(X,Z) .
77 @plano73[valor_objetivo(3)] +!bgfha(X,X) : cfe(e) & afh(c) <- !gaefa(X,Y) .
78 @plano74[valor_objetivo(5)] +!dgfba(X,Y) : hcb(g) & ace(b) <- !bgfha(X,X) .
79 @plano75[valor_objetivo(2)] +!hdghh(X,X) : fca(e) & afg(e) <- !dgfba(X,Y) .
80 @plano76[valor_objetivo(8)] +!decdc(X,Y) : adg(f) & ecb(h) <- !hdghh(X,X) .
81 @plano77[valor_objetivo(4)] +!bdccd(X,X) : aec(d) & bbb(f) <- !decdc(X,Y) .
82 @plano78[valor_objetivo(9)] +!fcggb(X,X) : chc(b) & bfh(e) <- !bdccd(X,X) .
83 @plano79[valor_objetivo(1)] +!dadac(X,Y) : gfd(e) & dfh(h) <- !fcggb(X,X) .
84 @plano80[valor_objetivo(4)] +!ahcfb(X,Z) : ggh(f) & bfa(f) <- !dadac(X,Y) .
85 @plano81[valor_objetivo(6)] +!aafcd(X,Y) : ach(a) & fcf(b) <- !ahcfb(X,Z) .
86 @plano82[valor_objetivo(1)] +!cegfh(X,Z) : bfd(f) & bfb(d) <- !aafcd(X,Y) .
87 @plano83[valor_objetivo(2)] +!edcab(X,Z) : dca(b) & add(a) <- !cegfh(X,Z) .
88 @plano84[valor_objetivo(10)] +!abgfh(X,Z) : gdf(e) & bec(h) <- !edcab(X,Z) .
89 @plano85[valor_objetivo(10)] +!eaega(X,Z) : gbe(d) & fbh(e) <- !abgfh(X,Z) .
90 @plano86[valor_objetivo(1)] +!cgfha(X,X) : gch(g) & aee(d) <- !eaega(X,Z) .
91 @plano87[valor_objetivo(6)] +!feced(X,Z) : gde(b) & dab(g) <- !cgfha(X,X) .
92 @plano88[valor_objetivo(3)] +!hggbe(X,Z) : ecf(d) & gee(h) <- !feced(X,Z) .
93 @plano89[valor_objetivo(1)] +!ehdfc(X,Z) : cge(h) & hcf(a) <- !hggbe(X,Z) .
94 @plano90[valor_objetivo(2)] +!gaebe(X,Z) : aaf(a) & bbg(c) <- !ehdfc(X,Z) .
95 @plano91[valor_objetivo(1)] +!bhdee(X,X) : gef(b) & cfc(d) <- !gaebe(X,Z) .
96 @plano92[valor_objetivo(6)] +!hefed(X,Y) : fba(g) & bge(b) <- !bhdee(X,X) .
97 @plano93[valor_objetivo(1)] +!caueb(X,Z) : dbb(d) & cac(d) <- !hefed(X,Y) .
98 @plano94[valor_objetivo(2)] +!cggdc(X,Z) : fbc(g) & had(c) <- !caueb(X,Z) .
99 @plano95[valor_objetivo(4)] +!bbbfb(X,Z) : dfd(d) & ehc(f) <- !cggdc(X,Z) .
100 @plano96[valor_objetivo(3)] +!chedf(X,Z) : acb(e) & che(c) <- !bbbfb(X,Z) .
101 @plano97[valor_objetivo(2)] +!dgecc(X,Z) : fdf(g) & hhc(e) <- !chedf(X,Z) .
102 @plano98[valor_objetivo(10)] +!gddgh(X,X) : aag(c) & dhf(c) <- !dgecc(X,Z) .
103 @plano99[valor_objetivo(4)] +!dadgb(X,Y) : bca(h) & dfg(c) <- !gddgh(X,X) .
104 @plano100[valor_objetivo(1)] +!dehec(X,Y) : dfh(b) & fdg(b) <- !dadgb(X,Y) .

```

B.1.4 Plano do agente N=200

Para o agente com 200 planos:

```

1 pos(0) .
2 +pos(0) <- mover .
3 +pos(1000) <- .stopMAS .
4 +pos(X) <- !afdbb(X,X) .
5 @plano1[valor_objetivo(1)] +!ebccf(X,Y) : fab(c) & bgh(c) <- mover .
6 @plano2[valor_objetivo(2)] +!fedgf(X,Y) : edf(g) & aeg(g) <- !ebccf(X,Y) .
7 @plano3[valor_objetivo(3)] +!chfae(X,X) : hdd(f) & def(d) <- !fedgf(X,Y) .
8 @plano4[valor_objetivo(4)] +!fcdac(X,X) : aab(d) & cde(f) <- !chfae(X,X) .
9 @plano5[valor_objetivo(5)] +!babcd(X,X) : bef(a) & gbb(d) <- !fcdac(X,X) .
10 @plano6[valor_objetivo(6)] +!hhbgc(X,X) : hfh(d) & fdb(h) <- !babcd(X,X) .
11 @plano7[valor_objetivo(7)] +!bafdg(X,X) : fch(c) & dgd(e) <- !hhbgc(X,X) .
12 @plano8[valor_objetivo(8)] +!eefaf(X,Z) : fhh(f) & fcg(f) <- !bafdg(X,X) .
13 @plano9[valor_objetivo(9)] +!bdfcc(X,Z) : ebg(a) & gfh(a) <- !eefaf(X,Z) .
14 @plano10[valor_objetivo(10)] +!gfghc(X,Y) : cdb(d) & ggb(f) <- !bdfcc(X,Z) .
15 @plano11[valor_objetivo(1)] +!dehhd(X,Y) : dee(c) & ddd(a) <- !gfghc(X,Y) .
16 @plano12[valor_objetivo(2)] +!cegfb(X,X) : hch(b) & abc(d) <- !dehhd(X,Y) .
17 @plano13[valor_objetivo(3)] +!hcafe(X,Y) : gbh(e) & cbb(f) <- !cegfb(X,X) .

```



```

18 @plano14[valor_objetivo(4)] +!cecbe(X,Z) : hfg(c) & cbb(a) <- !hcafe(X,Y) .
19 @plano15[valor_objetivo(5)] +!cfcfg(X,Y) : bdc(b) & bdh(g) <- !cecbe(X,Z) .
20 @plano16[valor_objetivo(6)] +!abfef(X,Y) : bfh(h) & dec(d) <- !cfcfg(X,Y) .
21 @plano17[valor_objetivo(7)] +!hbcbe(X,X) : hce(e) & bde(f) <- !abfef(X,Y) .
22 @plano18[valor_objetivo(8)] +!eachg(X,Z) : ffa(a) & hfg(d) <- !hbcbe(X,X) .
23 @plano19[valor_objetivo(9)] +!ceaed(X,Y) : cae(f) & ach(d) <- !eachg(X,Z) .
24 @plano20[valor_objetivo(10)] +!bafbh(X,Y) : dbd(d) & cac(a) <- !ceaed(X,Y) .
25 @plano21[valor_objetivo(1)] +!fedba(X,Y) : dbh(c) & hga(e) <- !bafbh(X,Y) .
26 @plano22[valor_objetivo(2)] +!gcghd(X,Y) : ggd(g) & fgc(g) <- !fedba(X,Y) .
27 @plano23[valor_objetivo(3)] +!fdabb(X,Y) : bag(c) & ghc(f) <- !gcghd(X,Y) .
28 @plano24[valor_objetivo(4)] +!becfh(X,Z) : afa(c) & cfb(a) <- !fdabb(X,Y) .
29 @plano25[valor_objetivo(5)] +!acggc(X,Y) : hhe(d) & gbb(h) <- !becfh(X,Z) .
30 @plano26[valor_objetivo(6)] +!edabg(X,X) : bbd(a) & eac(d) <- !acggc(X,Y) .
31 @plano27[valor_objetivo(7)] +!fffge(X,Z) : dhc(a) & bab(h) <- !edabg(X,X) .
32 @plano28[valor_objetivo(8)] +!habah(X,Y) : dfb(e) & aac(c) <- !fffge(X,Z) .
33 @plano29[valor_objetivo(9)] +!beghe(X,Z) : fbd(g) & cfc(e) <- !habah(X,Y) .
34 @plano30[valor_objetivo(10)] +!chhaa(X,Y) : dbh(d) & hca(c) <- !beghe(X,Z) .
35 @plano31[valor_objetivo(1)] +!cgaha(X,X) : aah(b) & fcd(e) <- !chhaa(X,Y) .
36 @plano32[valor_objetivo(2)] +!gdheh(X,Y) : hhe(f) & bgg(b) <- !cgaha(X,X) .
37 @plano33[valor_objetivo(3)] +!bbadd(X,Y) : fah(g) & cdb(f) <- !gdheh(X,Y) .
38 @plano34[valor_objetivo(4)] +!hcgda(X,Z) : fhb(c) & ffc(h) <- !bbadd(X,Y) .
39 @plano35[valor_objetivo(5)] +!ffahf(X,Y) : edc(a) & fcf(f) <- !hcgda(X,Z) .
40 @plano36[valor_objetivo(6)] +!fhffe(X,X) : fdc(e) & aec(a) <- !ffahf(X,Y) .
41 @plano37[valor_objetivo(7)] +!dghfd(X,Y) : bfh(b) & bbh(h) <- !fhffe(X,X) .
42 @plano38[valor_objetivo(8)] +!edfhc(X,Z) : ddc(g) & ghf(b) <- !dghfd(X,Y) .
43 @plano39[valor_objetivo(9)] +!fdghf(X,Z) : eac(b) & hcc(h) <- !edfhc(X,Z) .
44 @plano40[valor_objetivo(10)] +!hgccd(X,Y) : hcd(g) & bad(a) <- !fdghf(X,Z) .
45 @plano41[valor_objetivo(1)] +!ceecd(X,Z) : gbf(c) & aee(e) <- !hgccd(X,Y) .
46 @plano42[valor_objetivo(2)] +!ehhbf(X,Y) : fge(d) & fcc(a) <- !ceecd(X,Z) .
47 @plano43[valor_objetivo(3)] +!eadad(X,Z) : gcc(h) & faa(c) <- !ehhbf(X,Y) .
48 @plano44[valor_objetivo(4)] +!fdcaa(X,Y) : hde(f) & gbb(b) <- !eadad(X,Z) .
49 @plano45[valor_objetivo(5)] +!fghee(X,Y) : dda(f) & ghb(d) <- !fdcaa(X,Y) .
50 @plano46[valor_objetivo(6)] +!hedeh(X,Z) : hee(d) & eec(f) <- !fghee(X,Y) .
51 @plano47[valor_objetivo(7)] +!cdace(X,Y) : fbc(d) & bfa(g) <- !hedeh(X,Z) .
52 @plano48[valor_objetivo(8)] +!gfggf(X,Z) : ffb(f) & bdb(g) <- !cdace(X,Y) .
53 @plano49[valor_objetivo(9)] +!dbhde(X,Z) : bdd(d) & gha(a) <- !gfggf(X,Z) .
54 @plano50[valor_objetivo(10)] +!edacc(X,Y) : cgf(e) & edf(c) <- !dbhde(X,Z) .
55 @plano51[valor_objetivo(1)] +!efagd(X,Z) : hbc(e) & eca(g) <- !edacc(X,Y) .
56 @plano52[valor_objetivo(2)] +!bhgeh(X,X) : aac(d) & bcb(e) <- !efagd(X,Z) .
57 @plano53[valor_objetivo(3)] +!bccbh(X,X) : bed(c) & bbg(b) <- !bhgeh(X,X) .
58 @plano54[valor_objetivo(4)] +!ffbea(X,Y) : dgd(a) & fdg(f) <- !bccbh(X,X) .
59 @plano55[valor_objetivo(5)] +!ghffc(X,Z) : bfe(c) & dgh(f) <- !ffbea(X,Y) .
60 @plano56[valor_objetivo(6)] +!hchdf(X,Z) : bbf(g) & dcd(h) <- !ghffc(X,Z) .
61 @plano57[valor_objetivo(7)] +!acbff(X,X) : gdd(d) & acf(f) <- !hchdf(X,Z) .
62 @plano58[valor_objetivo(8)] +!abfdh(X,Y) : bfh(b) & ccd(e) <- !acbff(X,X) .
63 @plano59[valor_objetivo(9)] +!bafed(X,X) : afb(e) & egh(h) <- !abfdh(X,Y) .
64 @plano60[valor_objetivo(10)] +!dgceg(X,Y) : abg(b) & fda(b) <- !bafed(X,X) .
65 @plano61[valor_objetivo(1)] +!eccae(X,Z) : gaf(f) & heb(d) <- !dgceg(X,Y) .
66 @plano62[valor_objetivo(2)] +!hdafc(X,Z) : ahf(h) & fag(f) <- !eccae(X,Z) .
67 @plano63[valor_objetivo(3)] +!hgdfc(X,Y) : cbb(c) & hdh(b) <- !hdafc(X,Z) .
68 @plano64[valor_objetivo(4)] +!facfa(X,Y) : eab(b) & eee(h) <- !hgdfc(X,Y) .
69 @plano65[valor_objetivo(5)] +!badff(X,Z) : gbb(a) & bhh(h) <- !facfa(X,Y) .
70 @plano66[valor_objetivo(6)] +!fadba(X,X) : fbc(e) & heh(h) <- !badff(X,Z) .
71 @plano67[valor_objetivo(7)] +!abcdg(X,Z) : hag(c) & bhh(a) <- !fadba(X,X) .
72 @plano68[valor_objetivo(8)] +!haggg(X,Z) : fdd(h) & gfh(d) <- !abcdg(X,Z) .
73 @plano69[valor_objetivo(9)] +!abbca(X,Z) : hhh(b) & ccb(a) <- !haggg(X,Z) .

```

```

74 @plano70[valor_objetivo(10)] +!bagdc(X,Z) : ccf(h) & cbe(b) <- !abbca(X,Z) .
75 @plano71[valor_objetivo(1)] +!ccfec(X,X) : bad(g) & ggc(d) <- !bagdc(X,Z) .
76 @plano72[valor_objetivo(2)] +!ggbaf(X,Y) : eeg(c) & hfd(g) <- !ccfec(X,X) .
77 @plano73[valor_objetivo(3)] +!aeede(X,Y) : acg(d) & ccg(d) <- !ggbaf(X,Y) .
78 @plano74[valor_objetivo(4)] +!ehhgg(X,X) : bcf(e) & cfe(b) <- !aeede(X,Y) .
79 @plano75[valor_objetivo(5)] +!hbffg(X,Y) : bec(g) & bef(a) <- !ehhgg(X,X) .
80 @plano76[valor_objetivo(6)] +!adaac(X,Z) : fac(d) & fga(a) <- !hbffg(X,Y) .
81 @plano77[valor_objetivo(7)] +!hegdf(X,Y) : bee(g) & cab(c) <- !adaac(X,Z) .
82 @plano78[valor_objetivo(8)] +!gggdc(X,Y) : bgh(h) & fdb(d) <- !hegdf(X,Y) .
83 @plano79[valor_objetivo(9)] +!ebbbh(X,Z) : gcb(e) & afg(e) <- !gggdc(X,Y) .
84 @plano80[valor_objetivo(10)] +!afdce(X,X) : aac(c) & eea(g) <- !ebbbh(X,Z) .
85 @plano81[valor_objetivo(1)] +!aacbf(X,X) : dbg(e) & hdd(d) <- !afdce(X,X) .
86 @plano82[valor_objetivo(2)] +!bcebf(X,Z) : fhe(c) & bab(c) <- !aacbf(X,X) .
87 @plano83[valor_objetivo(3)] +!hhebf(X,X) : fee(d) & hbd(a) <- !bcebf(X,Z) .
88 @plano84[valor_objetivo(4)] +!gcaae(X,X) : adf(g) & abb(f) <- !hhebf(X,X) .
89 @plano85[valor_objetivo(5)] +!agfhf(X,X) : hbd(g) & hdg(e) <- !gcaae(X,X) .
90 @plano86[valor_objetivo(6)] +!fhgbd(X,Z) : hed(c) & eha(h) <- !agfhf(X,X) .
91 @plano87[valor_objetivo(7)] +!fhgha(X,X) : cac(g) & dbb(f) <- !fhgbd(X,Z) .
92 @plano88[valor_objetivo(8)] +!dheff(X,Z) : hhg(c) & dbe(c) <- !fhgha(X,X) .
93 @plano89[valor_objetivo(9)] +!eacac(X,X) : deb(c) & gbh(g) <- !dheff(X,Z) .
94 @plano90[valor_objetivo(10)] +!fffhg(X,Y) : bhb(g) & aed(b) <- !eacac(X,X) .
95 @plano91[valor_objetivo(1)] +!hhafe(X,Y) : fag(a) & bdf(h) <- !fffhg(X,Y) .
96 @plano92[valor_objetivo(2)] +!gbefg(X,Y) : hcd(a) & ccd(a) <- !hhafe(X,Y) .
97 @plano93[valor_objetivo(3)] +!cbadb(X,X) : ceg(c) & gab(b) <- !gbefg(X,Y) .
98 @plano94[valor_objetivo(4)] +!ggche(X,Y) : dhb(e) & heb(f) <- !cbadb(X,X) .
99 @plano95[valor_objetivo(5)] +!bcgef(X,X) : fgc(d) & eff(f) <- !ggche(X,Y) .
100 @plano96[valor_objetivo(6)] +!aebaa(X,Z) : eec(d) & hde(g) <- !bcgef(X,X) .
101 @plano97[valor_objetivo(7)] +!hddeg(X,Y) : ghf(a) & cad(g) <- !aebaa(X,Z) .
102 @plano98[valor_objetivo(8)] +!ffccb(X,Y) : dhb(b) & hbf(g) <- !hddeg(X,Y) .
103 @plano99[valor_objetivo(9)] +!dadde(X,X) : cdb(e) & fha(e) <- !ffccb(X,Y) .
104 @plano100[valor_objetivo(10)] +!eghga(X,Z) : cag(b) & afg(f) <- !dadde(X,X) .
105 @plano101[valor_objetivo(1)] +!bbbcf(X,X) : ffg(f) & ded(f) <- !eghga(X,Z) .
106 @plano102[valor_objetivo(2)] +!adhbh(X,Z) : hfe(g) & fdd(e) <- !bbbcf(X,X) .
107 @plano103[valor_objetivo(3)] +!gfbae(X,Z) : feb(b) & ecb(d) <- !adhbh(X,Z) .
108 @plano104[valor_objetivo(4)] +!fbhbg(X,X) : dab(d) & dfh(g) <- !gfbae(X,Z) .
109 @plano105[valor_objetivo(5)] +!edgab(X,Z) : bee(b) & cec(d) <- !fbhbg(X,X) .
110 @plano106[valor_objetivo(6)] +!egdeh(X,X) : bed(c) & bah(c) <- !edgab(X,Z) .
111 @plano107[valor_objetivo(7)] +!edbcc(X,X) : gcg(h) & ccc(h) <- !egdeh(X,X) .
112 @plano108[valor_objetivo(8)] +!hbggb(X,Z) : hea(a) & ehd(e) <- !edbcc(X,X) .
113 @plano109[valor_objetivo(9)] +!hbbag(X,Y) : bgd(e) & agh(d) <- !hbggb(X,Z) .
114 @plano110[valor_objetivo(10)] +!fghgb(X,X) : bch(e) & ade(c) <- !hbbag(X,Y) .
115 @plano111[valor_objetivo(1)] +!eghge(X,X) : hfh(f) & eda(d) <- !fghgb(X,X) .
116 @plano112[valor_objetivo(2)] +!dhcgh(X,Y) : aba(d) & cec(h) <- !eghge(X,X) .
117 @plano113[valor_objetivo(3)] +!bgffg(X,Z) : dbd(c) & daa(f) <- !dhcgh(X,Y) .
118 @plano114[valor_objetivo(4)] +!fgdgg(X,X) : fbe(a) & cdb(f) <- !bgffg(X,Z) .
119 @plano115[valor_objetivo(5)] +!agahb(X,X) : aef(b) & gfb(b) <- !fgdgg(X,X) .
120 @plano116[valor_objetivo(6)] +!hahae(X,X) : hhd(d) & bfb(d) <- !agahb(X,X) .
121 @plano117[valor_objetivo(7)] +!dhcga(X,Y) : dch(e) & dgh(h) <- !hahae(X,X) .
122 @plano118[valor_objetivo(8)] +!hahbb(X,X) : eed(a) & geh(h) <- !dhcga(X,Y) .
123 @plano119[valor_objetivo(9)] +!gehga(X,X) : ehd(d) & ccd(d) <- !hahbb(X,X) .
124 @plano120[valor_objetivo(10)] +!eddaf(X,Y) : gac(g) & hba(e) <- !gehga(X,X) .
125 @plano121[valor_objetivo(1)] +!ahbac(X,X) : cbg(e) & fgd(d) <- !eddaf(X,Y) .
126 @plano122[valor_objetivo(2)] +!caghb(X,X) : hhb(h) & fea(a) <- !ahbac(X,X) .
127 @plano123[valor_objetivo(3)] +!hehaf(X,X) : cfc(g) & dba(f) <- !caghb(X,X) .
128 @plano124[valor_objetivo(4)] +!bcbbd(X,Y) : baf(b) & hab(g) <- !hehaf(X,X) .
129 @plano125[valor_objetivo(5)] +!hhbbb(X,Z) : hed(b) & hfb(f) <- !bcbbd(X,Y) .

```

```

130 @plano126[valor_objetivo (6)] +!agedf (X, Y) : eae (a) & heb (f) <- !hhbbb (X, Z) .
131 @plano127[valor_objetivo (7)] +!ghdah (X, Y) : gad (g) & dcg (g) <- !agedf (X, Y) .
132 @plano128[valor_objetivo (8)] +!bfdbg (X, X) : eaf (d) & cba (e) <- !ghdah (X, Y) .
133 @plano129[valor_objetivo (9)] +!adgab (X, Z) : eee (e) & add (e) <- !bfdbg (X, X) .
134 @plano130[valor_objetivo (10)] +!ggdce (X, X) : bgf (d) & fed (c) <- !adgab (X, Z) .
135 @plano131[valor_objetivo (1)] +!edfge (X, X) : cbh (d) & dbc (c) <- !ggdce (X, X) .
136 @plano132[valor_objetivo (2)] +!edhdb (X, Y) : gbc (e) & cdc (e) <- !edfge (X, X) .
137 @plano133[valor_objetivo (3)] +!efhfh (X, X) : fgf (b) & ega (c) <- !edhdb (X, Y) .
138 @plano134[valor_objetivo (4)] +!aafef (X, X) : aaa (b) & aba (b) <- !efhfh (X, X) .
139 @plano135[valor_objetivo (5)] +!fbggd (X, Z) : cef (g) & aea (h) <- !aafef (X, X) .
140 @plano136[valor_objetivo (6)] +!ebdgh (X, X) : caf (e) & fhf (c) <- !fbggd (X, Z) .
141 @plano137[valor_objetivo (7)] +!fbcah (X, Z) : cac (g) & gfb (a) <- !ebdgh (X, X) .
142 @plano138[valor_objetivo (8)] +!edddc (X, Y) : gcc (c) & ded (d) <- !fbcah (X, Z) .
143 @plano139[valor_objetivo (9)] +!fgbah (X, Y) : dha (a) & gba (d) <- !edddc (X, Y) .
144 @plano140[valor_objetivo (10)] +!ehbea (X, X) : cdc (h) & daf (d) <- !fgbah (X, Y) .
145 @plano141[valor_objetivo (1)] +!aeegf (X, X) : ega (h) & agc (f) <- !ehbea (X, X) .
146 @plano142[valor_objetivo (2)] +!dgheh (X, Y) : fdf (e) & ccb (h) <- !aeegf (X, X) .
147 @plano143[valor_objetivo (3)] +!hfcfc (X, Z) : gdg (a) & dbg (e) <- !dgheh (X, Y) .
148 @plano144[valor_objetivo (4)] +!fccad (X, Z) : ebg (c) & gcf (e) <- !hfcfc (X, Z) .
149 @plano145[valor_objetivo (5)] +!cggd (X, X) : ccf (a) & gee (e) <- !fccad (X, Z) .
150 @plano146[valor_objetivo (6)] +!hgdcb (X, X) : bab (f) & bbc (c) <- !cggd (X, X) .
151 @plano147[valor_objetivo (7)] +!edceh (X, X) : dfe (d) & ece (a) <- !hgdcb (X, X) .
152 @plano148[valor_objetivo (8)] +!acfgc (X, Y) : gdf (h) & dhf (d) <- !edceh (X, X) .
153 @plano149[valor_objetivo (9)] +!dedge (X, Y) : gde (d) & dhd (d) <- !acfgc (X, Y) .
154 @plano150[valor_objetivo (10)] +!haafb (X, Y) : bad (h) & acf (g) <- !dedge (X, Y) .
155 @plano151[valor_objetivo (1)] +!caafg (X, Z) : bfg (h) & beh (b) <- !haafb (X, Y) .
156 @plano152[valor_objetivo (2)] +!fdheh (X, X) : age (f) & aga (e) <- !caafg (X, Z) .
157 @plano153[valor_objetivo (3)] +!hhfhg (X, X) : ahc (d) & cgd (g) <- !fdheh (X, X) .
158 @plano154[valor_objetivo (4)] +!dabhe (X, X) : fgf (d) & aab (c) <- !hhfhg (X, X) .
159 @plano155[valor_objetivo (5)] +!geegg (X, Y) : ceb (e) & gef (e) <- !dabhe (X, X) .
160 @plano156[valor_objetivo (6)] +!fhada (X, X) : agg (d) & daa (c) <- !geegg (X, Y) .
161 @plano157[valor_objetivo (7)] +!fbbfe (X, Y) : gdg (f) & hhc (c) <- !fhada (X, X) .
162 @plano158[valor_objetivo (8)] +!daefb (X, Z) : ccf (e) & egf (b) <- !fbbfe (X, Y) .
163 @plano159[valor_objetivo (9)] +!edfcd (X, Z) : ehf (b) & geh (d) <- !daefb (X, Z) .
164 @plano160[valor_objetivo (10)] +!dbfbb (X, X) : dcg (d) & fag (g) <- !edfcd (X, Z) .
165 @plano161[valor_objetivo (1)] +!ebfch (X, X) : aaf (a) & aef (a) <- !dbfbb (X, X) .
166 @plano162[valor_objetivo (2)] +!ggfhh (X, Z) : hbc (b) & aeg (e) <- !ebfch (X, X) .
167 @plano163[valor_objetivo (3)] +!gdadc (X, Z) : dbe (g) & fea (h) <- !ggfhh (X, Z) .
168 @plano164[valor_objetivo (4)] +!bhacg (X, X) : ceh (e) & abg (d) <- !gdadc (X, Z) .
169 @plano165[valor_objetivo (5)] +!ghgdd (X, Z) : faf (b) & gfc (a) <- !bhacg (X, X) .
170 @plano166[valor_objetivo (6)] +!gdgbe (X, Y) : dag (c) & ghf (g) <- !ghgdd (X, Z) .
171 @plano167[valor_objetivo (7)] +!gagdh (X, X) : ecc (f) & bed (a) <- !gdgbe (X, Y) .
172 @plano168[valor_objetivo (8)] +!gbdhd (X, Y) : aac (e) & hhc (g) <- !gagdh (X, X) .
173 @plano169[valor_objetivo (9)] +!acggh (X, X) : aaf (a) & ahh (f) <- !gbdhd (X, Y) .
174 @plano170[valor_objetivo (10)] +!adaaf (X, Z) : bcd (f) & chc (g) <- !acggh (X, X) .
175 @plano171[valor_objetivo (1)] +!hgbgc (X, Y) : aca (a) & fcd (b) <- !adaaf (X, Z) .
176 @plano172[valor_objetivo (2)] +!hbddd (X, X) : ffa (c) & feg (f) <- !hgbgc (X, Y) .
177 @plano173[valor_objetivo (3)] +!ehbeb (X, X) : aag (c) & cgg (c) <- !hbddd (X, X) .
178 @plano174[valor_objetivo (4)] +!hhefe (X, Z) : eba (c) & adb (h) <- !ehbeb (X, X) .
179 @plano175[valor_objetivo (5)] +!gbdge (X, Y) : dff (f) & bga (c) <- !hhefe (X, Z) .
180 @plano176[valor_objetivo (6)] +!ahffe (X, Z) : bhd (a) & gad (a) <- !gbdge (X, Y) .
181 @plano177[valor_objetivo (7)] +!dhagg (X, X) : bea (g) & dgf (h) <- !ahffe (X, Z) .
182 @plano178[valor_objetivo (8)] +!eahgc (X, X) : dgf (d) & ahh (g) <- !dhagg (X, X) .
183 @plano179[valor_objetivo (9)] +!agbba (X, Y) : cab (d) & afd (f) <- !eahgc (X, X) .
184 @plano180[valor_objetivo (10)] +!ahfgg (X, Z) : hfa (c) & hec (d) <- !agbba (X, Y) .
185 @plano181[valor_objetivo (1)] +!beaed (X, Z) : afa (a) & fhb (h) <- !ahfgg (X, Z) .

```

```

186 @plano182[valor_objetivo(2)] +!dhcfh(X,Y) : gad(h) & bga(f) <- !beaed(X,Z) .
187 @plano183[valor_objetivo(3)] +!agaac(X,Z) : gcb(b) & fdc(d) <- !dhcfh(X,Y) .
188 @plano184[valor_objetivo(4)] +!cahdg(X,Z) : had(g) & fbe(h) <- !agaac(X,Z) .
189 @plano185[valor_objetivo(5)] +!cbhah(X,Z) : acg(g) & dbc(b) <- !cahdg(X,Z) .
190 @plano186[valor_objetivo(6)] +!beehd(X,Y) : hfh(f) & dfh(e) <- !cbhah(X,Z) .
191 @plano187[valor_objetivo(7)] +!ebhbb(X,X) : gbb(b) & ecf(a) <- !beehd(X,Y) .
192 @plano188[valor_objetivo(8)] +!egecf(X,X) : fac(b) & ecg(e) <- !ebhbb(X,X) .
193 @plano189[valor_objetivo(9)] +!dhbba(X,Y) : bhg(c) & fef(h) <- !egecf(X,X) .
194 @plano190[valor_objetivo(10)] +!fefda(X,Y) : bbc(e) & gde(c) <- !dhbba(X,Y) .
195 @plano191[valor_objetivo(1)] +!eddfc(X,Y) : cea(b) & bfc(b) <- !fefda(X,Y) .
196 @plano192[valor_objetivo(2)] +!fhgeh(X,Y) : chg(c) & cbb(e) <- !eddfc(X,Y) .
197 @plano193[valor_objetivo(3)] +!gaaba(X,X) : edh(a) & bbg(d) <- !fhgeh(X,Y) .
198 @plano194[valor_objetivo(4)] +!cgdce(X,Y) : fdh(h) & ehf(d) <- !gaaba(X,X) .
199 @plano195[valor_objetivo(5)] +!afbhf(X,Z) : dgf(h) & cdc(f) <- !cgdce(X,Y) .
200 @plano196[valor_objetivo(6)] +!ehahg(X,Y) : ffa(a) & beh(d) <- !afbhf(X,Z) .
201 @plano197[valor_objetivo(7)] +!acfad(X,Z) : efe(h) & hfb(b) <- !ehahg(X,Y) .
202 @plano198[valor_objetivo(8)] +!baeee(X,Z) : hge(a) & ffh(e) <- !acfad(X,Z) .
203 @plano199[valor_objetivo(9)] +!fffdh(X,X) : ege(b) & bbc(d) <- !baeee(X,Z) .
204 @plano200[valor_objetivo(10)] +!afdabb(X,Y) : gca(d) & gce(a) <- !fffdh(X,X) .

```

B.2 Tempo das execuções dos experimentos

As subseções a seguir apresentam os tempos (sem segundos) das 40 execuções para cada configuração dos experimentos realizados no Capítulo 5:

B.2.1 Modelo filtro de percepções com N=1

Tabela B.1: Lista dos tempos das execuções dos experimentos N=1 com filtro

Execução	0	250	500	1000	2000	4000	8000
1	0.615	3.297	6.168	13.003	40.433	128.287	496.209
2	0.631	3.252	6.066	12.96	40.355	128.659	501.54
3	0.632	3.328	5.966	13.272	40.51	127.911	513.458
4	0.623	3.25	5.991	13.245	40.485	127.977	507.341
5	0.638	3.28	6.123	13.219	41.153	127.384	482.016
6	0.612	3.264	6.006	13.094	40.34	126.619	496.735
7	0.629	3.281	6.005	12.98	40.381	126.556	506.335
8	0.608	3.284	6.006	13.123	40.642	126.468	501.628
9	0.622	3.31	5.996	13.11	40.353	126.527	491.14
10	0.658	3.394	6.026	13.058	40.878	126.244	506.457
11	0.636	3.508	5.988	13.078	40.134	126.634	507.025
12	0.612	3.247	6.002	12.996	40.548	127.682	481.747
13	0.621	3.4	4.955	13.056	40.154	128.253	497.168
14	0.656	3.335	5.977	13.129	40.393	134.242	484.956
15	0.621	3.336	6.048	13.132	40.375	126.838	499.864
16	0.649	3.365	6.009	13.102	40.566	127.269	500.628
17	0.633	3.264	5.997	12.951	40.601	126.963	489.529

18	0.636	3.256	6.004	13.218	40.364	127.189	503.039
19	0.629	3.281	6.007	12.982	40.384	128.245	494.447
20	0.625	3.309	6.062	13.097	40.276	127.687	500.983
21	0.646	3.3	5.964	12.983	40.024	133.277	500.355
22	0.634	3.303	5.989	13.115	40.141	127.31	500.965
23	0.635	3.282	5.978	13.04	40.698	126.767	492.934
24	0.612	3.357	6.0	13.022	40.145	127.474	498.724
25	0.624	3.255	6.065	13.046	40.348	128.047	505.814
26	0.636	3.294	5.997	13.083	40.449	126.084	504.239
27	0.622	3.279	5.993	13.002	40.137	127.34	497.122
28	0.635	3.265	6.0	13.196	40.281	127.206	504.924
29	0.648	3.3	5.941	13.045	40.629	127.237	491.332
30	0.654	3.28	4.872	13.267	40.552	127.793	504.57
31	0.623	3.25	5.969	13.057	40.277	128.08	501.292
32	0.642	3.265	5.987	13.004	40.575	126.193	497.375
33	0.624	3.276	6.03	13.183	40.634	126.004	503.425
34	0.622	3.283	5.265	13.002	40.48	127.831	486.541
35	0.635	3.284	6.006	13.117	40.449	126.706	493.71
36	0.613	3.261	5.994	13.152	40.453	126.145	502.783
37	0.61	3.297	5.971	13.118	40.338	126.057	496.293
38	0.636	3.303	5.98	13.039	40.4	127.698	513.335
39	0.633	3.295	6.014	13.017	40.453	126.727	494.592
40	0.643	3.301	6.0	13.116	40.688	126.489	486.753

B.2.2 Modelo filtro de percepções com N=50

Tabela B.2: Lista dos tempos das execuções dos experimentos N=50 com filtro

Execução	0	250	500	1000	2000	4000	8000
1	2.636	6.504	10.34	21.15	54.578	149.17	442.395
2	2.755	6.459	10.343	20.62	53.918	150.652	441.003
3	2.652	6.678	10.26	21.595	54.48	161.129	479.661
4	2.64	6.684	10.207	20.796	53.616	162.931	453.888
5	2.694	6.417	10.339	20.743	54.253	150.918	512.003
6	2.68	6.449	10.305	20.858	54.944	149.606	515.417
7	2.666	6.291	10.284	20.737	54.025	149.631	468.439
8	2.64	6.326	10.259	20.894	53.897	150.49	469.819
9	2.647	6.279	10.166	20.958	53.805	149.632	470.561
10	2.632	6.58	10.551	20.754	54.271	150.105	501.242
11	2.648	6.264	10.098	20.998	54.191	150.343	463.862
12	2.694	6.268	10.434	20.729	53.94	149.114	505.027
13	2.656	6.258	10.281	20.971	53.931	149.273	511.967
14	2.655	6.26	10.394	20.942	53.848	150.307	473.213
15	2.671	6.397	10.153	20.899	53.975	149.968	461.501
16	2.642	6.203	10.231	20.838	54.161	149.547	509.841

17	2.644	6.489	10.142	20.572	53.852	150.533	506.539
18	2.661	6.247	10.488	20.868	54.419	149.134	504.542
19	2.671	6.244	10.526	20.653	53.926	149.596	462.737
20	2.643	6.197	10.319	20.845	54.063	149.074	465.637
21	2.65	6.251	10.346	20.863	54.205	149.253	483.868
22	2.654	6.255	10.382	21.067	54.039	149.791	510.109
23	2.685	6.416	10.289	20.85	54.407	149.645	479.575
24	2.674	6.283	10.295	20.719	54.143	150.491	469.402
25	2.652	6.603	10.271	20.678	54.226	149.817	483.681
26	2.706	6.364	10.069	20.936	54.077	149.145	500.985
27	2.604	6.339	10.314	20.648	53.781	150.601	459.59
28	2.695	6.553	10.242	20.828	54.223	149.815	515.819
29	2.625	6.298	10.44	20.738	53.981	150.239	468.855
30	2.645	6.375	10.481	20.95	53.884	150.664	464.171
31	2.673	6.61	10.269	20.807	53.969	149.584	477.948
32	2.664	6.27	10.225	20.653	53.898	149.203	466.793
33	2.679	6.356	10.247	20.857	53.94	150.522	513.769
34	2.665	6.729	10.305	20.803	53.683	150.148	469.406
35	2.666	6.212	10.337	20.942	54.319	149.495	509.457
36	2.616	6.412	10.101	20.788	55.279	148.619	472.797
37	2.64	6.205	10.45	20.711	53.763	149.5	465.063
38	2.636	6.282	10.411	20.646	54.063	149.798	468.786
39	2.679	6.232	10.459	20.729	54.555	149.014	465.061
40	2.667	6.681	10.224	20.806	54.306	151.204	472.583

B.2.3 Modelo filtro de percepções com N=100

Tabela B.3: Lista dos tempos das execuções dos experimentos N=100 com filtro

Execução	0	250	500	1000	2000	4000	8000
1	4.998	10.154	16.713	28.306	70.345	169.481	499.09
2	5.062	10.084	15.777	29.314	64.223	169.574	489.479
3	4.922	9.945	15.067	29.346	65.246	171.999	507.303
4	5.185	10.2	15.179	29.339	64.515	171.309	485.331
5	5.132	10.231	15.351	29.598	64.87	169.855	509.428
6	5.215	10.086	15.342	28.208	64.667	171.037	487.089
7	5.081	10.162	15.545	28.381	64.463	170.264	495.648
8	5.238	10.131	15.671	27.867	64.946	170.426	489.897
9	5.001	9.918	15.868	29.071	64.796	170.458	487.656
10	5.142	10.095	15.125	29.918	65.107	171.439	490.887
11	5.024	9.873	15.192	29.209	64.624	173.093	512.171
12	4.96	10.07	15.448	29.273	64.689	169.776	483.372
13	5.017	10.011	15.571	28.549	64.444	171.316	486.536
14	5.261	9.704	15.363	28.487	65.663	169.922	499.005
15	5.019	9.733	15.968	28.888	65.603	170.828	502.758

16	5.231	9.681	15.559	29.287	65.701	170.101	484.498
17	5.079	9.943	15.194	29.518	65.548	170.717	481.49
18	5.011	10.075	15.671	27.968	65.224	171.384	508.381
19	5.145	9.853	15.513	29.225	66.147	170.769	500.207
20	5.186	10.084	15.284	29.251	65.559	170.209	484.242
21	5.15	10.052	14.754	28.448	65.104	170.955	481.576
22	5.214	10.197	15.296	29.588	65.911	170.516	487.15
23	5.053	10.178	15.746	29.128	65.522	170.6	480.795
24	5.055	10.165	15.684	27.57	65.635	169.881	481.076
25	5.197	10.116	15.621	28.231	65.383	170.202	489.666
26	5.203	10.146	15.769	29.087	65.75	170.353	506.349
27	5.021	10.235	15.885	28.184	65.697	170.9	486.471
28	5.079	10.073	15.29	28.578	65.27	170.764	500.044
29	5.0	10.019	14.985	27.756	65.469	170.177	479.234
30	5.178	10.084	15.689	28.569	66.035	171.376	482.116
31	5.211	9.851	15.447	28.266	66.239	172.622	483.745
32	5.036	10.009	15.572	28.278	65.416	170.49	504.761
33	5.212	9.761	15.72	29.134	66.033	171.311	480.6
34	5.046	10.132	15.339	29.031	65.358	170.439	534.327
35	4.959	10.012	15.558	27.694	66.245	170.84	536.094
36	5.04	9.854	15.59	29.47	66.148	169.087	485.699
37	4.998	9.735	15.674	29.22	66.516	170.8	484.142
38	5.044	10.014	15.515	28.387	66.253	170.679	512.301
39	5.231	10.146	15.52	29.215	65.454	169.328	488.723
40	5.216	10.069	15.658	27.546	65.447	170.66	490.127

B.2.4 Modelo filtro de percepções com N=200

Tabela B.4: Lista dos tempos das execuções dos experimentos N=200 com filtro

Execução	0	250	500	1000	2000	4000	8000
1	13.992	23.311	29.626	48.845	99.839	216.072	602.786
2	13.724	24.658	29.636	49.176	98.118	217.084	594.985
3	13.825	21.418	30.802	50.506	100.712	211.281	601.438
4	14.484	21.688	30.04	48.679	99.968	216.655	598.936
5	13.561	21.461	30.399	49.296	101.481	212.453	602.442
6	13.724	21.808	29.767	49.811	100.301	215.542	602.409
7	14.174	21.65	30.052	50.364	101.379	213.597	597.844
8	14.126	22.309	30.724	48.994	100.876	212.952	602.797
9	14.146	21.9	30.995	49.444	99.979	210.232	597.656
10	13.651	22.184	30.201	49.048	100.494	214.021	593.074
11	14.326	21.792	30.502	48.749	100.467	213.839	590.885
12	13.711	21.834	30.03	49.532	97.002	214.066	605.663
13	14.019	22.237	30.584	49.116	99.275	214.508	598.115
14	14.168	21.955	30.139	49.534	100.162	213.357	598.609

15	13.706	21.957	30.421	48.928	99.219	216.713	599.507
16	13.922	21.929	30.485	48.794	100.1	213.955	604.197
17	14.559	22.424	31.009	50.494	100.037	215.197	597.643
18	13.716	21.481	30.639	50.261	98.031	216.798	601.035
19	13.898	21.896	29.913	49.766	99.851	214.25	595.306
20	13.629	21.783	29.841	49.941	100.415	214.552	599.502
21	14.3	21.945	29.922	49.757	98.457	214.895	599.331
22	14.044	22.171	30.232	49.409	99.767	212.361	598.056
23	13.752	22.093	30.293	48.912	102.656	212.29	608.834
24	13.982	21.857	29.281	49.438	99.171	218.05	603.046
25	13.681	21.649	30.274	50.657	101.214	212.264	597.638
26	14.527	21.404	30.945	49.418	100.485	214.801	605.279
27	14.395	21.982	30.106	48.244	100.551	213.666	599.618
28	13.895	21.322	30.75	48.793	99.694	215.223	614.269
29	14.135	22.112	29.111	49.931	100.28	218.497	624.059
30	14.017	21.525	30.632	51.095	100.976	214.825	607.541
31	14.206	21.898	29.957	50.824	100.239	212.528	603.687
32	13.934	21.746	30.594	51.217	100.501	214.561	602.159
33	13.815	21.766	30.52	50.738	98.613	214.867	611.402
34	14.169	21.885	30.294	49.545	99.625	215.171	596.324
35	14.146	21.616	30.276	50.301	99.009	219.102	599.323
36	14.038	21.441	30.586	51.194	98.546	210.658	598.85
37	13.478	21.635	30.012	50.482	101.336	216.907	596.748
38	13.738	21.903	29.863	48.815	100.289	213.199	598.453
39	14.171	21.827	30.687	50.052	99.684	212.457	595.307
40	14.297	21.509	29.925	50.922	101.249	216.239	592.819

B.2.5 Modelo original com N=1

Tabela B.5: Lista dos tempos das execuções dos experimentos N=1 sem filtro

Execução	0	250	500	1000	2000	4000	8000
1	0.644	4.248	9.943	35.927	121.638	415.026	1486.319
2	0.626	4.244	9.884	35.254	125.173	415.401	1443.305
3	0.636	4.247	9.957	35.747	122.681	406.942	1751.927
4	0.644	4.254	9.718	35.656	117.585	412.049	1514.842
5	0.615	4.223	10.797	35.522	117.591	404.449	1526.239
6	0.652	4.271	9.154	35.552	122.647	409.989	1542.579
7	0.614	4.061	9.322	35.36	121.416	402.801	1465.061
8	0.626	4.175	9.978	36.095	120.914	409.128	1576.269
9	0.634	4.239	9.328	36.24	122.306	437.14	1514.052
10	0.643	4.159	9.449	35.297	120.672	414.051	1509.589
11	0.609	4.333	8.987	37.708	122.039	404.386	1478.242
12	0.632	4.156	9.333	35.427	122.083	408.35	1419.781
13	0.594	4.312	9.453	35.09	122.062	412.025	1447.11

14	0.617	4.147	9.323	35.142	120.575	420.138	1431.763
15	0.624	4.04	9.545	35.252	122.692	410.713	1387.648
16	0.617	4.295	9.423	36.411	121.401	419.583	1469.383
17	0.625	4.184	9.6	37.615	120.581	409.039	1436.67
18	0.62	4.203	9.477	35.979	125.452	402.26	1461.408
19	0.638	4.305	9.851	35.112	124.924	405.357	1421.92
20	0.625	4.074	9.278	35.321	122.497	415.214	1421.643
21	0.64	4.232	9.424	35.277	122.909	410.43	1397.466
22	0.613	4.065	9.341	35.359	122.413	407.979	1486.463
23	0.626	4.178	9.281	35.572	123.9	422.025	1433.818
24	0.624	4.261	9.576	37.683	126.274	410.772	1490.106
25	0.625	4.189	9.487	35.617	121.757	406.84	1450.581
26	0.615	4.066	9.399	35.109	124.319	406.972	1439.165
27	0.646	4.29	9.264	36.767	124.878	415.335	1403.94
28	0.647	4.075	9.476	35.582	124.771	410.254	1388.811
29	0.638	4.178	9.321	35.384	124.026	408.755	1486.52
30	0.611	4.234	9.397	35.27	122.325	405.344	1468.159
31	0.63	4.067	9.448	35.802	122.519	411.614	1470.241
32	0.624	4.207	9.406	35.306	122.539	406.841	1504.696
33	0.623	4.23	9.45	37.602	122.538	405.728	1423.105
34	0.624	4.24	9.318	37.805	124.963	410.388	1478.827
35	0.635	4.28	9.776	35.359	125.668	405.891	1419.992
36	0.643	4.205	9.327	35.312	125.605	407.652	1398.836
37	0.619	4.146	9.525	36.625	122.706	411.18	1520.966
38	0.635	4.161	9.293	36.397	121.162	413.921	1457.487
39	0.615	3.992	9.494	35.821	123.472	413.584	1413.105
40	0.611	4.162	9.247	35.187	123.819	413.803	1452.323

B.2.6 Modelo original com N=50

Tabela B.6: Lista dos tempos das execuções dos experimentos N=50 sem filtro

Execução	0	250	500	1000	2000	4000	8000
1	1.926	5.615	11.505	37.246	132.163	423.484	1565.487
2	1.947	5.58	11.322	37.423	131.185	418.512	1571.774
3	1.999	5.802	11.323	37.355	132.576	424.897	1520.03
4	1.983	5.525	11.276	37.303	133.038	417.39	1561.813
5	2.102	5.527	11.267	37.26	130.604	423.438	1540.803
6	2.37	5.53	11.323	37.157	132.846	418.441	1506.62
7	2.003	5.714	11.201	37.289	131.16	417.07	1481.97
8	2.086	5.674	11.248	37.936	130.234	415.79	1532.919
9	2.008	5.446	11.219	37.077	132.528	426.94	1532.991
10	2.128	5.492	11.221	37.166	130.57	422.601	1496.228
11	1.983	5.574	11.33	37.189	134.314	419.811	1476.68
12	2.165	5.759	11.202	37.151	133.872	423.223	1431.083

13	2.049	5.715	11.534	36.943	129.452	426.855	1450.253
14	2.014	5.518	11.266	37.275	130.881	428.354	1459.712
15	1.975	5.783	11.25	37.428	131.781	424.792	1400.172
16	2.003	5.823	11.189	37.491	131.917	423.913	1448.64
17	2.067	5.578	11.3	37.162	130.181	419.658	1485.279
18	2.072	5.945	11.23	37.293	131.364	425.842	1446.814
19	2.024	5.535	11.324	37.198	133.424	422.654	1499.438
20	2.029	5.555	11.311	36.973	132.603	425.576	1428.111
21	2.064	5.515	11.254	37.125	130.425	427.28	1379.733
22	2.048	5.762	11.271	37.185	132.091	428.156	1446.137
23	2.034	5.693	11.271	37.332	130.854	430.101	1443.183
24	2.053	5.605	11.274	37.301	131.747	423.012	1445.374
25	1.996	5.522	11.207	37.399	130.599	423.68	1441.236
26	2.03	5.568	11.203	37.641	132.347	422.782	1499.85
27	2.073	5.669	11.346	37.229	133.418	419.97	1443.959
28	2.024	5.805	11.338	37.694	129.672	428.468	1427.934
29	2.067	5.517	11.256	37.439	128.46	414.383	1471.324
30	2.157	5.514	11.525	37.296	128.886	421.157	1436.088
31	1.979	5.612	11.305	37.655	131.08	421.105	1437.852
32	2.117	5.502	11.3	37.002	132.045	425.105	1441.008
33	2.139	5.815	12.138	37.482	130.893	419.613	1415.203
34	1.946	5.683	11.637	37.546	131.398	422.946	1494.57
35	2.011	5.486	11.441	36.944	131.659	423.439	1404.836
36	1.988	5.595	11.734	36.97	133.475	421.578	1463.202
37	1.997	5.783	11.207	37.145	125.612	423.909	1445.785
38	1.961	5.692	11.409	37.173	126.527	426.074	1471.197
39	2.066	5.591	11.348	37.489	128.042	421.637	1523.802
40	2.178	5.472	11.496	37.667	125.905	417.141	1509.21

B.2.7 Modelo original com N=100

Tabela B.7: Lista dos tempos das execuções dos experimentos N=100 sem filtro

Execução	0	250	500	1000	2000	4000	8000
1	3.153	7.173	14.629	42.217	132.866	439.264	1470.766
2	3.119	7.309	15.332	41.395	133.55	441.13	1385.644
3	3.119	7.185	14.403	41.967	133.275	436.325	1378.081
4	3.168	6.997	13.858	41.403	132.497	435.556	1421.44
5	3.122	7.089	14.304	41.94	133.866	453.994	1392.924
6	3.089	6.991	14.108	42.263	133.053	433.541	1445.091
7	3.153	7.102	14.245	41.583	131.208	438.336	1474.317
8	3.137	7.046	14.427	41.527	132.516	476.124	1513.363
9	3.143	7.018	14.335	41.835	131.897	449.089	1404.423
10	3.312	6.966	14.283	41.933	132.183	437.081	1416.757
11	3.029	7.186	14.417	41.953	133.238	440.385	1461.146

12	3.031	7.245	14.328	42.012	131.682	434.808	1458.639
13	3.085	7.297	14.296	41.36	138.566	446.484	1524.297
14	3.024	7.13	14.307	41.436	130.561	432.224	1415.424
15	3.038	7.477	14.313	41.701	132.481	437.771	1453.83
16	3.135	7.039	14.301	42.016	131.14	437.62	1445.38
17	3.049	7.11	14.356	41.567	131.231	428.689	1402.728
18	3.073	7.171	14.329	42.03	131.437	440.657	1375.598
19	3.015	7.162	14.316	41.377	132.764	431.904	1500.615
20	3.034	7.197	14.375	41.628	131.268	431.818	1417.209
21	3.024	7.083	14.323	41.768	130.099	440.402	1448.241
22	3.154	7.057	14.109	41.62	131.669	439.979	1426.879
23	3.067	7.107	14.354	41.996	131.862	433.137	1398.164
24	3.068	7.093	14.245	41.729	132.533	436.447	1449.914
25	3.068	7.086	14.476	41.206	131.716	432.403	1447.662
26	3.079	6.99	14.255	41.991	133.39	437.69	1439.116
27	3.056	7.092	14.283	42.1	130.773	440.817	1526.813
28	3.155	6.988	14.369	41.817	131.032	446.217	1415.439
29	3.128	7.337	14.398	41.536	132.505	438.622	1453.609
30	3.015	7.052	14.213	41.478	131.066	422.265	1478.084
31	3.067	7.129	14.345	41.371	132.873	437.576	1429.843
32	3.033	7.002	14.333	41.315	131.909	430.774	1448.105
33	3.022	7.035	14.58	41.573	131.113	435.698	1402.022
34	3.104	7.032	14.399	41.694	130.958	433.025	1463.976
35	3.041	7.072	14.31	41.34	132.009	434.497	1442.386
36	3.02	7.045	14.309	41.452	132.289	435.748	1431.19
37	3.124	6.967	13.624	41.939	130.652	432.804	1466.054
38	3.015	6.958	14.327	41.581	132.137	435.484	1436.811
39	3.065	7.018	13.804	41.406	132.078	436.139	1421.264
40	3.056	7.082	14.205	41.712	130.161	438.302	1451.409

B.2.8 Modelo original com N=200

Tabela B.8: Lista dos tempos das execuções dos experimentos N=200 sem filtro

Execução	0	250	500	1000	2000	4000	8000
1	5.796	11.21	20.767	50.552	138.822	459.399	1536.084
2	5.762	11.017	20.33	49.499	140.057	458.883	1507.415
3	5.848	11.156	20.26	49.613	144.084	440.77	1484.798
4	5.995	11.204	20.105	49.175	139.985	451.83	1484.45
5	5.756	11.166	19.834	49.17	140.508	450.638	1529.027
6	5.999	11.122	19.938	49.444	139.751	449.779	1512.12
7	5.836	11.15	19.943	49.048	139.794	453.382	1440.905
8	5.817	11.223	19.852	49.535	139.43	459.298	1465.917
9	5.81	11.074	19.962	49.529	140.592	450.974	1476.757
10	5.926	11.191	20.253	49.613	140.77	451.692	1459.338

11	5.828	11.127	20.103	49.394	138.661	450.693	1493.218
12	5.766	11.022	19.839	49.211	140.617	449.189	1447.274
13	5.779	11.059	19.821	49.231	140.004	448.318	1461.816
14	5.74	11.096	19.798	49.301	139.682	446.202	1487.48
15	5.799	11.074	19.905	48.958	140.611	454.176	1469.988
16	5.756	11.032	19.712	49.084	140.274	447.107	1456.286
17	5.909	11.131	19.747	49.642	140.219	470.752	1418.902
18	5.837	11.088	19.817	49.79	140.973	452.254	1579.983
19	5.881	11.061	19.795	49.333	145.003	449.504	1514.252
20	5.933	11.3	20.097	49.571	141.366	445.796	1491.308
21	5.867	11.076	20.058	50.002	140.173	448.058	1477.93
22	5.889	11.171	19.964	49.463	142.759	456.918	1485.279
23	5.771	11.06	19.792	48.958	139.141	450.485	1485.108
24	5.778	11.033	19.779	49.006	140.205	452.166	1469.074
25	5.759	11.134	19.862	49.591	140.304	451.855	1562.244
26	5.736	11.102	20.17	49.909	140.468	450.756	1461.288
27	5.772	11.059	19.933	49.272	140.119	445.233	1476.205
28	5.776	11.137	19.906	49.432	140.015	449.251	1541.248
29	5.814	11.13	19.781	48.992	139.727	436.868	1495.478
30	5.72	11.204	19.886	49.06	140.353	449.377	1473.189
31	5.755	11.243	19.865	49.558	144.345	445.381	1463.588
32	5.844	11.108	20.046	49.087	140.479	442.833	1528.694
33	5.715	11.149	21.342	49.506	140.06	448.554	1483.32
34	5.862	11.096	20.09	49.066	140.764	457.243	1490.7
35	5.829	11.094	19.913	48.998	139.688	448.381	1468.148
36	5.843	11.195	19.661	49.491	139.619	448.968	1497.136
37	5.709	11.295	19.74	49.663	140.089	453.026	1465.332
38	5.829	11.25	19.84	49.648	138.436	448.521	1460.304
39	5.747	11.049	19.653	49.254	140.942	450.034	1540.122
40	5.866	11.094	19.747	49.613	139.989	448.88	1470.011
