

**UNIVERSIDADE FEDERAL DE PELOTAS**  
Centro de Desenvolvimento Tecnológico  
Programa de Pós-Graduação em Computação



Dissertação

**GGMA: Gramática de Grafos para Sistemas Multiagentes baseados no PopOrg**

**André Moura de Mello**

Pelotas, 2014

**André Moura de Mello**

**GGMA: Gramática de Grafos para Sistemas Multiagentes baseados no PopOrg**

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Luciana Foss  
Co-orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Simone André da Costa Cavalheiro

Pelotas, 2014

Universidade Federal de Pelotas / Sistema de Bibliotecas  
Catalogação na Publicação

M527g Mello, André Moura de

GGMA: Gramática de Grafos para Sistemas Multiagentes baseados no PopOrg / André Moura de Mello; Luciana Foss, orientadora; Simone André da Costa Cavalheiro, coorientadora. – Pelotas, 2014.

64 f. : il.

Dissertação (Mestrado) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2014.

1. Gramática de Grafos. 2. Sistemas Multiagentes. 3. PopOrg. 4. Gramática de Grafos para Sistemas Multiagentes. I. Foss, Luciana, orient. II. Cavalheiro, Simone André da Costa, coorient. III. Título.

CDD: 005

**Insira AQUI a folha de aprovação  
(será entregue após a banca)**

**Dedico este trabalho a toda a minha família, pai e mãe, esposa, irmãos e a meus amigos, pelo apoio incondicional, força, incentivo, amizade e amor. Sem eles nada disto seria possível.**

## **AGRADECIMENTOS**

Agradeço a Deus por guiar os meus caminhos e permitir chegar onde cheguei dando provas de Sua imensa benevolência.

Agradeço a minha família, por acreditar em mim, me ensinando a importância do estudo, pois o conhecimento é o único bem que não pode ser tirado do ser humano.

A minha esposa, que está comigo ao longo desta caminhada, sempre me dando apoio nos momentos mais difíceis, mesmo quando roubava-lhe o tempo precioso da sua convivência.

Agradeço em especial à Professora Dr<sup>a</sup> Luciana Foss, pela paciência, amizade, dedicação, sua generosidade, carinho me conduzindo no caminho certo e também pela sua capacidade de me incentivar nos momentos difíceis. Sua orientação contribuiu muito para o meu desenvolvimento acadêmico e pessoal.

Agradeço à Professora Dr<sup>a</sup> Simone André da Costa Cavaleiro, pelos puxões de orelha, pelas ajudas, pela dedicação, paciência, por me incentivar a concluir de forma positiva todas as atividades que me são dadas, muito obrigado por tudo.

Aos meus amigos de fé, que muito me ajudaram e responderam pacientemente as minhas dúvidas, angústias e questionamentos na minha pesquisa e que contribuíram para este trabalho.

**Vencedor é o que se vence**  
— VITOR RAMIL

## RESUMO

MELLO, André Moura de. **GGMA: Gramática de Grafos para Sistemas Multiagentes baseados no PopOrg**. 2014. 64 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2014.

A cada dia os sistemas estão se tornando mais complexos e sofisticados. A tarefa de especificar um *software* não é algo natural. A abordagem de agentes se mostra adequada para o desenvolvimento de sistemas complexos. Sabe-se que para provar propriedades de um sistema este deve ser especificado através de uma linguagem que ofereça métodos de análise. Neste caso, o sistema é especificado formalmente através de um modelo matemático. Gramática de Grafos (GG) é uma linguagem formal bastante adequada para especificar sistemas distribuídos e reativos, que tenham uma topologia complexa (vários tipos de elementos e vários tipos de relações) e o comportamento orientado a dados (eventos são disparados por configurações particulares do estado). Em uma GG os estados do sistema são modelados por grafos e a mudança entre os estados como regras. O uso de GG torna-se interessante pelo fato de existirem diferentes técnicas e ferramentas para especificação e verificação de sistemas descritos nesta linguagem. Além disso, as GGs possuem um layout gráfico, que é bastante intuitivo até para não teóricos, além de permitir a descrição modular de sistemas. O objetivo deste trabalho é propor um *framework* de especificação para modelos de sistemas multiagentes baseados no modelo PopOrg usando gramática de grafos.

**Palavras-chave:** Gramática de Grafos, Sistemas Multiagentes, PopOrg, Gramática de Grafos para Sistemas Multiagentes.

## ABSTRACT

MELLO, André Moura de. **MAGG: Multi-Agent Graph Grammar based on PopOrg**. 2014. 64 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2014.

Everyday systems are becoming more complex and sophisticated. The task of specifying a software is not something natural. The agents approach is adequate for complex systems development. It is known that for proving properties of a system it should be specified by a language that provides analysis methods. In this case, the system is formally specified by a mathematical model. Graph Grammar (GG) is a formal language well suited for specifying distributed and reactive systems, which have a complex topology (various kinds of elements and various types of relations among them) and are behavior-oriented data (events are triggered by particular configurations of state). In a GG, the system states are modeled by graphs and the state changes are described by rules. The use of GGs becomes interesting because there are different techniques and tools for specification and verification of systems described in this language. In addition, the GGs have a graphical layout that is intuitive enough even for non-theoreticians, and allow the description of modular systems. The aim of this work is to propose a specification framework to model multi-agent systems based on PopOrg model using graph grammars.

**Keywords:** Graph Grammar, Multi-Agent Systems, PopOrg, Multi-Agent Graph Grammar.

## LISTA DE FIGURAS

|           |   |    |
|-----------|---|----|
| Figura 1  | Grafo Tipado $G$ . . . . .  | 23 |
| Figura 2  | Uma regra do grafo . . . . .  | 24 |
| Figura 3  | Exemplo de uma GG . . . . .   | 25 |
| Figura 4  | Derivação direta de $G$ para $H$ usando a regra <i>Accept</i> baseada em $m$ . . . . .                  | 27 |
| Figura 5  | PopOrg com os níveis micro-macro da estrutura organizacional . . . . .                                  | 30 |
| Figura 6  | Relação de implementação entre os níveis micro-organizacional e populacional do modelo PopOrg . . . . . | 39 |
| Figura 7  | Grafo esquema para GGMA's. . . . .  | 40 |
| Figura 8  | Esquema de regras para GGMA's. . . . .  | 41 |
| Figura 9  | AgGG para os agentes $Formiga_1$ e $Formiga_2$ . . . . .  | 44 |
| Figura 10 | Conjunto de regras para o agente $Formiga_1$ . . . . .  | 44 |
| Figura 11 | Conjunto de regras para o agente $Formiga_2$ . . . . .  | 44 |
| Figura 12 | AgGG para os agentes $Formiga_3$ e $Formiga_4$ . . . . .  | 44 |
| Figura 13 | Conjunto de regras para o agente $Formiga_3$ . . . . .  | 45 |
| Figura 14 | Conjunto de regras para o agente $Formiga_4$ . . . . .  | 45 |
| Figura 15 | Grafos inicial e tipo da gramática de grafos populacional $PGG$ . . . . .                               | 46 |
| Figura 16 | RGG para os papéis <i>Reprodutora</i> , <i>Operária</i> e <i>Soldado</i> . . . . .                      | 49 |
| Figura 17 | Conjunto de regras para o papel <i>Reprodutora</i> . . . . .  | 49 |
| Figura 18 | Conjunto de regras para o papel <i>Operária</i> . . . . .   | 49 |
| Figura 19 | Conjunto de regras para o papel <i>Soldado</i> . . . . .  | 50 |
| Figura 20 | Grafos inicial e tipo da gramática de grafos organizacional $OGG$ . . . . .                             | 51 |
| Figura 21 | Diagrama de composição de regras . . . . .  | 55 |
| Figura 22 | Composição de duas regras da $AgGG$ . . . . .   | 56 |
| Figura 23 | Grafos tipo e inicial da $OGG$ . . . . .  | 58 |

## **LISTA DE TABELAS**

|          |                                     |    |
|----------|-------------------------------------|----|
| Tabela 1 | Comparação das abordagens . . . . . | 20 |
|----------|-------------------------------------|----|

## LISTA DE ABREVIATURAS E SIGLAS

|        |   |
|--------|---|
| SMA    | Sistema Multiagente                               |
| PopOrg | Modelo População-Organização                      |
| GG     | Gramática de Grafos                               |
| GH     | Gramática de Hipergrafos                          |
| EGG    | Gramática de Grafos Embarcada                     |
| AGG    | Gramática de Grafos Ativa                         |
| CGG    | Gramática de Grafos Controlada                    |
| AgGG   | Gramática de Grafos para Agentes                  |
| RGG    | Gramática de Grafos para Papéis                   |
| IMP    | Relação de Implementação                          |
| PGG    | Gramática de Grafos Populacional                  |
| OGG    | Gramática de Grafos Organizacional                |
| GGMA   | Gramática de Grafos para Sistemas Multiagentes    |
| AOSE   | Engenharia de <i>Software</i> Orientada a Agentes |
| CSP    | <i>Communicating Sequential Processes</i>         |
| RSL    | <i>Raise Specification Language</i>               |

## SUMÁRIO

|            |  |           |
|------------|--|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>  | <b>14</b> |
| <b>1.1</b> | <b>Objetivo</b>  | <b>15</b> |
| 1.1.1      | Objetivos específicos  | 15        |
| <b>2</b>   | <b>REFERENCIAL TEÓRICO</b>   | <b>17</b> |
| <b>3</b>   | <b>GRAMÁTICA DE GRAFOS</b>   | <b>22</b> |
| <b>4</b>   | <b>MODELO POPORG</b>   | <b>29</b> |
| 4.1        | Estrutura Populacional (POP)   | 31        |
| 4.2        | Estrutura Organizacional (ORG)   | 34        |
| 4.3        | Relações de Implementação  | 36        |
| <b>5</b>   | <b>GRAMÁTICA DE GRAFOS PARA SISTEMAS MULTIAGENTES BASEADOS NO POPORG</b>             | <b>40</b> |
| 5.1        | Gramática de Grafos para População   | 40        |
| 5.2        | Gramática de Grafos para Organização   | 47        |
| 5.3        | Relação de Implementação   | 51        |
| <b>6</b>   | <b>SEMÂNTICA DA GRAMÁTICA DE GRAFOS PARA SISTEMAS MULTIAGENTES BASEADO NO POPORG</b> | <b>54</b> |
| <b>7</b>   | <b>CONCLUSÃO</b>   | <b>59</b> |
|            | <b>REFERÊNCIAS</b>   | <b>61</b> |

# 1 INTRODUÇÃO

Um dos principais desafios da computação moderna é a construção de sistemas para resolver os diferentes problemas que surgem todos os dias. Esses sistemas possuem vários aspectos que aumentam a sua complexidade. De um modo geral, os sistemas modernos são abertos, distribuídos e reativos, executando em ambientes dinâmicos e complexos. Muitos estudos foram feitos sobre resolução de problemas complexos nos mais diversos ramos da computação. A abordagem de agentes pode ser utilizada para tratar de sistemas complexos, devido à sua grande capacidade de representar comportamentos, papéis e interações entre agentes autônomos. Métodos, ferramentas e modelos para apoiar esta abordagem se fazem necessários, dando origem à área de pesquisa de Engenharia de Software Orientada a Agentes (AOSE).

De acordo com Demazeu e Costa (1996), aspectos sociais e organizacionais, como o relacionamento entre agentes e papéis que esses agentes podem realizar, vêm ganhando grande importância na área de Sistemas Multiagentes (SMA). Tipicamente os agentes são entendidos como um modelo de três atributos: crenças, desejos e intenções. Podem ainda ser observadas, nos SMAs, características reguladoras do sistema como normas, compromissos e acordos. Um dos mais importantes recursos dos sistemas multiagentes é a autonomia dos agentes (WOOLDRIDGE, 2001). Entretanto, a autonomia do sistema global pode gerar comportamentos indesejados. Uma das maneiras de resolver esse problema é através da utilização de organizações, que podem ser vistas como um conjunto de restrições de comportamentos que devem ser satisfeitas pelos agentes do sistema. O modelo Populacional-Organizacional (PopOrg) (DEMAZEAU; COSTA, 1996) pode ser visto como um *framework* de modelagem para representar organizações sociais de um SMA. Este modelo considera os níveis populacional e organizacional de um SMA e as relações que podem ocorrer entre esses níveis. Além disso, papéis e comportamentos que os agentes e grupos de agentes podem realizar nos sistemas podem ser descritos usando o modelo PopOrg.

A modelagem de sistemas complexos, mesmo usando abordagens de SMAs que permitem um determinado controle desta complexidade, é uma tarefa bastante difícil. Em geral, não é trivial garantir que o sistema apresente determinadas proprieda-

des. Verificação formal é um método de análise que permite garantir que o sistema satisfaça certas propriedades. Para verificar formalmente um sistema, este deve ser especificado através de uma linguagem que ofereça métodos de análise. Neste caso, o sistema é especificado formalmente através de um modelo matemático, isto é, através de uma linguagem formal que possui sintaxe e semântica bem definidas. Gramática de Grafos (GGs) (ROZENBERG, 1997) é uma linguagem formal bastante adequada para especificar sistemas distribuídos e reativos, que tenham uma topologia complexa (vários tipos de elementos e vários tipos de relações entre eles) e o comportamento orientado a dados (eventos são disparados por configurações particulares do estado). Em uma GG os estados do sistema são modelados por grafos e as mudanças de estados são descritas por regras de transformação de grafos. O uso de GGs torna-se interessante pelo fato de existirem diferentes técnicas e ferramentas para especificação e verificação de sistemas (MELLO et al., 2011). Além disso, GGs possuem um layout gráfico, que é bastante intuitivo até para não teóricos e permitem a descrição modular de sistemas (EHRIG et al., 1999).

## 1.1 Objetivo

O objetivo deste trabalho é permitir a especificação de sistemas multiagentes considerando o modelo PopOrg e utilizando a linguagem formal gramática de grafos (MELLO; FOSS; COSTA CAVALHEIRO, 2013a,b, 2012).

### 1.1.1 Objetivos específicos

Para atingir o objetivo geral deste trabalho, alguns objetivos específicos foram necessários:

- Definir a sintaxe e a semântica de uma gramática de grafos para sistemas multiagentes (GGMA);
- Especificar os níveis populacional e organizacional, bem como a relação de implementação entre os níveis do modelo PopOrg;
- Incluir mecanismos de controle nas gramática de grafos a fim de regular a ordem de aplicação das regras e a interação entre os agentes e papéis;
- Definir método de composição de gramáticas que permitam compor sistemas multiagentes a partir da especificação de agentes individuais.

Com esses objetivos, a abordagem oportunizará aos projetistas de *software* que utilizam o modelo de SMAs especificarem seu projeto utilizando todos aspectos da GG, levando em conta a relação entre a população e organização destes sistemas. A

abordagem apresentada proporcionará ao modelo PopOrg uma representação formal, visual e intuitiva. Existe atualmente abordagens e ferramentas de análise para verificação formal de sistemas utilizando a técnica de prova de teoremas (COSTA, 2010). A escolha do modelo PopOrg, se deve ao fato, principalmente, por ser um modelo que agrupa as características essenciais dos SMAs, acreditando assim, que a abordagem proposta possa ser aplicada a outros modelos de SMAs.

O restante deste trabalho está organizado como descrito a seguir. No Capítulo 2 serão apresentados os trabalhos relacionados, apresentando abordagens existentes de SMAs que utilizam GGs. No Capítulo 3 é apresentada a linguagem de especificação gramática de grafos e no Capítulo 4 serão descritos os conceitos básicos do modelo PopOrg. No Capítulo 5 serão apresentadas as definições de gramática de grafos para sistemas multiagentes baseadas no modelo PopOrg e no Capítulo 6 é apresentada a semântica desta gramática. Finalmente, as conclusões, as contribuições e os trabalhos futuros serão apresentados no Capítulo 7.

## 2 REFERENCIAL TEÓRICO

Neste Capítulo serão apresentados alguns trabalhos relacionados com sistemas multiagentes que utilizam gramática de grafos como linguagem formal. Para realizar essa pesquisa, foram levados em consideração trabalhos que utilizam GG para especificar todo ou partes de um sistema multiagente.

Um modelo formal de transformação de grafos baseado em sistemas multiagentes foi proposto por KNIRSCH; KREOWSKI (1999). Neste trabalho, os autores dão algumas orientações sobre como especificar os agentes por meio de sistemas de transformação de grafos. Os agentes e os sistemas de agentes são representados por grafos, e as ações desempenhadas por esses agentes são descritas como regras de transformação de grafos. Os agentes são entidades reativas que interagem e se comunicam com o ambiente, pela mudança dele. Nessa abordagem, um agente é definido por um conjunto de regras, que descrevem as possíveis ações, e a condição de capacidade, restringindo a ordem em que essas regras podem ser aplicadas. Essa restrição da ordem de aplicação das regras é feita através de expressões regulares. Além disso, os autores propuseram uma semântica sequencial para descrever o comportamento de um sistema multiagente, onde a aplicação da regra deve levar em conta a sua condição de capacidade.

Em DEPKE; HECKEL (2000), um sistema multiagente é representado por uma rede de agentes autônomos e reativos que agem e interagem concorrentemente, acessando recursos compartilhados e trocando mensagens. Esta rede pode ser reconfigurada pela criação ou deleção de objetos, agentes e links. Nessa abordagem, um sistema multiagente pode ser visto em três visões diferentes: a visão funcional que descreve a computação concorrente do sistema, onde agentes e objetos são representados por grafos e regras de transformação de grafos; na visão estrutural, descrita por um grafo tipo, que define os diferentes tipos de agentes e objetos e as possíveis conexões entre eles; e, na visão dinâmica, onde diagramas de estado são utilizados para controlar a aplicação das regras. Essa relação é formalizada por meio de processos de grafos que extraem a relação causal da aplicação das regras para uma execução global do sistema.

Em WANG; LIANG (2006a,b) e WANG; LIANG; ZHAO (2006), os autores apresentam um modelo de transformação de grafos para descrever a evolução estrutural em SMAs com estrutura organizacional, onde agentes devem ser coordenados para alcançar os objetivos da organização. São considerados os seguintes aspectos das organizações: papéis e suas inter-relações; coordenação entre agentes; e as ações dos agentes nos papéis. Os autores definem um meta-modelo para representar a estrutura organizacional. Este meta-modelo é baseado em grafos multi-níveis que permitem representar os três aspectos da organização. O grafo de papéis descreve o nível mais alto e é composto por papéis e as dependências entre eles. Se o objetivo de um papel é um conjunto de sub-objetivos de outros papéis, ele é considerado um grupo. O grafo de agentes descreve o nível mais baixo e consiste de agentes e a coordenação entre eles. O grafo de conexão descreve o nível intermediário, onde os vértices representam papéis e agentes, e as arestas representam as atividades dos agentes nos papéis. Estas arestas conectam o nível mais alto com o mais baixo. Mudanças na estrutura organizacional, bem como, nos papéis desempenhados pelos agentes são descritos por regras de transformação de grafos multiníveis.

Em LASKER et al. (2004) foi definido um modelo categorial para sistemas multiagentes. Os objetos da categoria representam agentes de diferentes tipos, os morfismos representam todos os tipos de relações entre os agentes (comunicação e cooperação geral). Cada sistema multiagente tem um diagrama base associado, representando as estruturas de cooperação e comunicação. Este diagrama é interpretado como uma categoria *Path* (morfismos são sequências de setas), definindo uma semântica categorial de estruturas relacionais. O trabalho de LASKER et al. (2004) foi estendido por PFALZGRAF; SOBOLL (2008) com a introdução da noção de categorias tipadas que permitem descrever sistemas multiagentes de forma mais detalhada. Os objetos e morfismos da categoria têm um tipo associado, onde os tipos de objetos descrevem agentes com propriedades diferentes e os tipos de morfismos descrevem as relações entre diferentes agentes. O foco do trabalho está na estrutura global dos sistemas e não para modelagem de agentes específicos. Ainda neste trabalho, é definida uma noção de funtor que permite relacionar sistemas multiagentes. Como as relações entre os agentes podem mudar, os autores definem um sistema de transformação de grafos para sistemas multiagentes, usando a abordagem *double-pushout*. Nessa abordagem, cada categoria de sistemas multiagentes pode ser vista como um grafo e os funtores como morfismos de grafos. Assim, regras de transformação são definidas para representar mudanças na estrutura de sistemas multiagentes. Essas regras são aplicadas na categoria (ou grafos) e podem descrever a adição ou remoção de agentes, bem como, mudanças nas relações entre eles.

Um sistema multiagente baseado em gramática foi introduzido por ŚLUSARCZYK (2008) para especificar sistemas modulares e concorrentes. Um sistema é definido por

um número de agentes, um ambiente e um *buffer* de mensagens, através das quais os agentes se comunicam com o ambiente. Para modelar o seu conhecimento local, cada agente possui uma gramática de grafos associada. Hipergrafos são usados para representar as estruturas dos objetos construídos (usados) pelos agentes. Estes grafos possuem dois tipos de hiperarestas: uma para representar objetos hierárquicos; e, outra para representar relações entre os objetos (sem hierarquia). As regras descrevem a construção do objeto e diagramas de controle são usados para restringir a ordem em que as regras são aplicadas.

Em SMITH et al. (2009) foi apresentado um *framework* para especificações de tarefas descentralizadas para sistemas multi-robô. Para especificar as tarefas multi-robôs, três elementos devem estar presentes: leis de controle, que são definidas com relação à dinâmica dos agentes e dos mapeamentos dos sensores de entrada para as ações de controle; regras de dependência da topologia de rede, que determinam quais leis de controle são usadas sob quais condições da rede; e, condições geométricas, onde as mudanças de leis de controle podem ocorrer. Os autores apresentam a implementação de uma Gramática de Grafos Embarcada (EGG) para resolver o problema de cobertura de sensores para três robôs alcançarem uma configuração triangular. A EGG ilustra a especificação de leis de controle descentralizadas para redes de sistemas de robôs. Neste trabalho é apresentado um estudo de caso, onde é definido um sistema que produz uma família de trajetórias de robôs através das derivações de uma gramática de grafos. Neste sistema, cada vértice é um robô, o rótulo de cada vértice indica o modo de operação (qual lei de controle está obedecendo). As arestas representam a comunicação e percepção entre robôs, bem como os relacionamentos que ocorrem nas leis de controle. As regras determinam as mudanças de estados do sistema, isto é, as derivações da gramática a partir de um grafo inicial. As mudanças de estados originados pela aplicação das regras da gramática definem uma família de trajetórias dos robôs. Além disso, um grafo embarcado é composto por um grafo e por uma função de realização, a qual mapeia cada vértice em um espaço de estados de um robô (um grafo embarcado modela não somente o estado discreto, mas também estados contínuos do robô). EGGs estendem a noção de GGs por levar em conta restrições de geometria e dinâmica.

A plataforma GRADIS (KOTULSKI; SDZIWIY, 2010; KOTULSKI, 2008) é um *framework* para um ambiente multiagente que utiliza gramática de grafos como apoio. Nessa abordagem, grafos complementares (ou parciais) são usados para representar o conhecimento de cada agente, enquanto o grafo centralizado é usado para representar o sistema como um todo. O grafo centralizado deve ser dividido em subgrafos distribuídos que representam as porções de informação conhecidas por cada agente. Neste trabalho, os autores fazem uma proposta de análise de propriedades através de simulações. Porém, antes de iniciar uma simulação, um processo de distribuição

inicial (ou decomposição) deve ser realizado. Nesse processo, os grafos centralizados são decompostos em subgrafos parciais (que representam o conhecimento de cada agente) e cada subgrafo é associado a um agente. A cooperação entre agentes é descrita por um conjunto de regras de transformação de grafos, que é herdado a partir do modelo centralizado, isto é, cada agente tem acesso a todas as regras da gramática, apesar de ter conhecimento parcial das informações do sistema.

Em SCHATTEN (2012), estendido em SCHATTEN (2013), foi introduzida uma abordagem de gramática de grafos ativa (AGG) para reorganização em arquiteturas multiagentes. A reescrita de grafos ativa é um formalismo inspirado pela teoria de banco de dados ativos (PATON; DÍAZ, 1999). O objetivo deste trabalho é prover um *framework* holístico para modelar aspectos da reorganização. Segundo os autores, a arquitetura de uma organização multiagente é vista sobre cinco perspectivas: unidade organizacional ou de agentes, processos (atividades de um agente), estratégia organizacional (objetivos), cultura organizacional (conhecimento, normas, processos) e a estrutura organizacional. A estrutura organizacional é formalizada utilizando o princípio fractal e aplicada a organizações de multiagentes. O trabalho foi formalizado utilizando teoria dos grafos introduzindo gramática de grafos ativa para organizações de multiagentes. Agentes são modelados como vértices e os seus objetivos e a interação entre eles como arestas rotuladas. As mudanças de estados do sistema são modelados por eventos em um certo tempo.

Em BARBOSA (2011), foi proposta uma abordagem para especificação formal de sistemas multiagentes baseada no modelo PopOrg, utilizando as linguagens CSP e RSL. A linguagem CSP foi utilizada para descrever comportamentos dos papéis no nível micro-organizacional e processos de troca que podem ser realizados entre estes papéis. Por outro lado, a linguagem RSL foi usada para especificar ambos os aspectos, estrutural e operacional, dos sistemas PopOrg.

Tabela 1: Comparação das abordagens

|     | <b>Abordagem</b>              | <b>População</b> | <b>Organização</b> | <b>Implementação</b> |
|-----|-------------------------------|------------------|--------------------|----------------------|
| (1) | Knirsch e Kreowski(1999)      | SIM              | NÃO                | NÃO                  |
| (2) | Depke e Heckel(2000)          | SIM              | NÃO                | NÃO                  |
| (3) | Wang e Liang(2006a,b)         | SIM              | SIM                | SIM                  |
| (4) | Lasker e Dubois(2004)         | NÃO              | NÃO                | NÃO                  |
| (5) | Ślusarczyk(2008)              | SIM              | NÃO                | NÃO                  |
| (6) | Smith et al.(2009)            | SIM              | NÃO                | NÃO                  |
| (7) | Kotulski e Sdziwy(2010)(2008) | SIM              | NÃO                | NÃO                  |
| (8) | Schatten(2012)(2013)          | SIM              | SIM                | NÃO                  |
| (9) | Barbosa(2011)                 | SIM              | SIM                | NÃO                  |

A Tabela 1 mostra um comparativo entre os trabalhos encontrados na literatura, relacionando quais dos aspectos dos SMAs pode ser modelado pela abordagem. Por essa Tabela, pode ser visto que a maioria das abordagens apresentadas não considera a organização dos sistemas multiagentes. As únicas que permitem modelar este aspecto são aquelas propostas nos trabalhos (3), (8) e (9) respectivamente.

Os dois níveis (agentes e organização) são descritos em um mesmo modelo. Isso faz com que as mudanças realizadas em um nível tenham que ser refletidas também no outro. Na abordagem (3), o enfoque do trabalho está na reorganização e adaptabilidade do sistema como um todo, baseado na evolução dos agentes do sistema. Outro aspecto considerado neste trabalho foi uma relação (implementação) entre os níveis operacionais de um sistema multiagente. A abordagem (8), por outro lado, mostra a organização de um sistema multiagente como um conjunto de agentes que interagem para alcançar os objetivos da organização. Não existem definições da organização e da população do ambiente, já que os dois níveis são representados por uma única especificação.

O modelo PopOrg dá enfoque na relação entre uma organização e seus agentes (população), isto é, como deve ser a estrutura da população para implementar uma organização. Deste modo, cada nível é especificado separadamente e uma relação define como a implementação de cada papel é realizada pelos agentes. Entretanto, essa relação de implementação entre os níveis de um sistema multiagente não é apresentada na abordagem (9).

O próximo Capítulo apresenta algumas definições importantes sobre a linguagem formal gramática de grafos.

### 3 GRAMÁTICA DE GRAFOS

Neste Capítulo, serão revisados os principais conceitos de GGs de acordo com a abordagem algébrica (EHRIG; PFENDER; SCHNEIDER, 1973; ROZENBERG, 1997; EHRIG et al., 1999). Um grafo consiste em vértices e arestas que conectam esses vértices. Dois grafos podem ser relacionados se eles têm estruturas compatíveis. A relação é dada por um morfismo de grafos que preserva origem e destino de cada aresta.

**Definição 1. (Grafos e Morfismos de Grafos)** *Um grafo é definido por uma tupla  $H = (V_H, E_H, o^H, d^H)$ , onde  $V_H$  é um conjunto de vértices,  $E_H$  é um conjunto de arestas,  $o^H : E_H \rightarrow V_H^*$  e  $d^H : E_H \rightarrow V_H^*$  são funções totais que associam cada aresta a uma lista de vértices de origem e de destino, respectivamente. Um morfismo de grafos  $g : G \rightarrow H$  é dado por uma tupla  $g = (g_V, g_E)$ , consistindo de duas funções  $g_V : V_G \rightarrow V_H$  e  $g_E : E_G \rightarrow E_H$ , tal que,  $o^H \circ g_E = g_V \circ o^G$  e  $d^H \circ g_E = g_V \circ d^G$ . Um morfismo de grafos é total ou injetor se ambas as funções que o compõem são totais ou injetoras, respectivamente.*

Elementos de um grafo podem ser rotulados. Esses rótulos são determinados por um mapeamento a partir de elementos de um grafo em um grafo tipo. Um grafo tipo é o que define todos os tipos de vértices e arestas em um sistema. A compatibilidade entre grafos tipados é determinado por um morfismo de grafos tipados que respeitam o tipo de cada item do grafo.

**Definição 2. (Grafos Tipados e Morfismos de Grafos Tipados)** *Dado um grafo  $T$ , chamado grafo tipo, um grafo tipado sobre  $T$  é uma tupla  $H^T = (H, t^H)$ , onde  $H$  é um grafo e  $t^H : H \rightarrow T$  é um morfismo de grafos. Se o tipo pode ser entendido a partir do contexto, escreve-se  $H$  em vez de  $H^T$ . Um morfismo de grafos tipados sobre  $T$ , denotado por  $g^T : H^T \rightarrow G^T$ , é um morfismo de grafos  $g : H \rightarrow G$ , tal que  $t^H \circ g = t^G$ . Um morfismo de grafos tipados  $g^T$  é total ou injetor se o morfismo de grafos  $g$  é total ou injetor, respectivamente.*

**Exemplo 1. (Grafos e morfismos de grafos (tipados))** *A Figura 1 mostra dois grafos  $T$  e  $G$  e o morfismo  $t^G$  entre eles. Os vértices são representados por quadrados com*

cantos arredondados, por exemplo, *Operator* e *Pump*; e as arestas são representadas por setas com nomes (inscritos ou não em polígonos ou círculos), por exemplo, *Free*, *op* e *Start*. Quando origem e destino são o mesmo vértice, omite-se a origem na imagem, por exemplo as arestas *Busy* e *Suply* que têm o vértice *Pump* como origem e destino. O morfismo  $t^G$  mapeia os vértices e arestas de  $G$  para vértices e arestas de  $T$ , respectivamente. O mapeamento é definido pela numeração. O Grafo  $G$  e o morfismo  $t^G$  compõem um grafo tipado  $G^T = (G, t^G)$ .

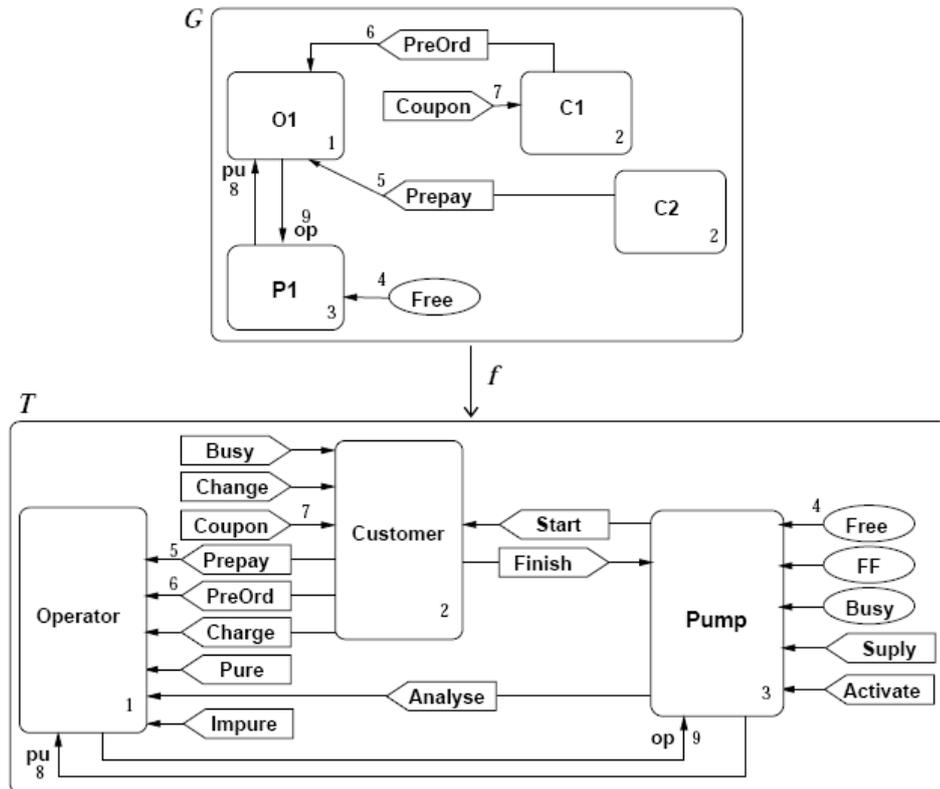


Figura 1: Grafo Tipado  $G$

O comportamento de uma gramática de grafos é determinado por regras de re-escrita. Seguindo a abordagem *Double Pushout* (DPO) (ROZENBERG, 1997), uma regra é composta por três grafos: o lado esquerdo  $L$ , o lado direito  $R$ , e a interface  $K$  representando os elementos comuns nos lados  $L$  e  $R$ . Isso especifica que, se uma ocorrência do grafo  $L$  é encontrada no estado corrente do sistema, ela pode ser substituída pelo grafo  $R$ , preservando  $K$ .

**Definição 3. (Regras)** Uma regra tipada sobre  $T$  é uma tupla  $p = (L_p, K_p, R_p, l_p, r_p)$ , denotado por  $p : L_p \xleftrightarrow{l_p} K_p \xrightarrow{r_p} R_p$ , onde  $L_p$ ,  $K_p$  e  $R_p$  são grafos tipados sobre  $T$ , chamados lado esquerdo, interface e lado direito, respectivamente.  $l_p : K_p \rightarrow L_p$  e  $r_p : K_p \rightarrow R_p$  são morfismos totais de grafos tipados, onde  $l_p$  é uma inclusão e  $r_p$  é injetor. O conjunto de todas as regras com relação a um grafo tipo  $T$  é denotado por  $Regras(T)$ .

**Exemplo 2. (Regras)** A Figura 2 mostra uma regra  $q$  cujo o lado esquerdo, o lado direito e a interface são os grafos  $L_q$ ,  $R_q$  e  $K_q$ , respectivamente. O grafo  $K_q$  é mapeado para  $L_q$  e  $R_q$  por dois morfismos:  $l$  e  $r$ . Esta regra pode ser aplicada no estado que contém o grafo  $L_q$  (contendo vértices tipados por Customer e Operator e a aresta PrePay do Customer para Operator), resultando em um grafo onde PrePay é deletada; Customer e Operator são preservados; e Coupon e PreOrd são criadas.

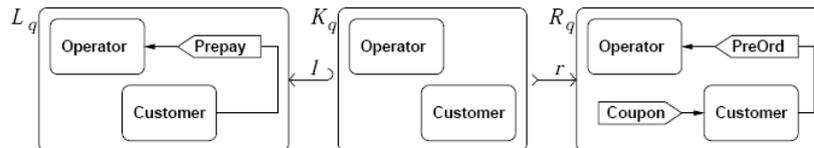


Figura 2: Uma regra do grafo

Em geral, uma gramática de grafos que descreve um sistema é constituída de um grafo tipo, que caracteriza os tipos de vértices e arestas permitidos no sistema, um grafo inicial representando o estado inicial deste sistema, e um conjunto de regras as quais descrevem as possíveis mudanças do sistema. Além disso, as regras podem ter nomes associados.

**Definição 4. (Gramática de Grafos)** Uma gramática de grafos (tipados) é uma tupla  $GG = (T, H, P, \pi)$ , onde  $T$  é o grafo tipo,  $H$  é um grafo tipado sobre  $T$ , chamado grafo inicial,  $P$  é o conjunto de nomes de regras e  $\pi : P \rightarrow \text{Regras}(T)$  é uma função total, que associa cada nome de regra a uma regra tipada sobre  $T$ .

**Exemplo 3. (Gramática de Grafos)** Nesse exemplo é especificado o sistema de um posto de combustíveis usando gramática de grafos. Um cliente (Customer) pré-paga (Prepay) uma certa quantidade de dinheiro ao operador (Operator) do posto de combustível pelo combustível que será retirado da bomba (Pump). Se a bomba estiver livre, ela fornece o combustível e o operador devolve o troco ao cliente, baseado na quantidade real de dinheiro gasto pelo cliente. Caso contrário, o cliente recebe uma mensagem avisando para tentar novamente. Outra atividade realizada no posto de combustível é a análise da bomba para verificar a pureza (Purity) ou impureza (Impurity) do combustível.

Na Figura 3, a GG que modela o sistema do operador da bomba do posto de combustível é mostrada. O grafo tipo do sistema foi representado na Figura 1. Nesse sistema três entidades são modeladas por vértices: Customer, Operator e Pump. Cada entidade pode ter atributos (por exemplo FF ou Free) e recebe mensagens (por exemplo Prepay ou Supply). Cada entidade pode ter referências para outras, que também representam atributos. Ambos, atributos e mensagens, são modelados por arestas. Assim, pelo grafo tipo pode-se ver a entidade Pump com seus atributos: operator (op)

que a opera em três flags que indicam se ela está livre (Free), ou ativada (FF), ou ocupada (Busy). A entidade Customer não possui atributos; e a entidade Operator tem um atributo indicando a bomba que ele opera (pu). O grafo  $G$  (mostrado na Figura 1), mostra o estado inicial do sistema.

O comportamento do sistema é descrito pela regra no topo da Figura 3 - o grafo de interface de cada regra é omitido mas ele pode ser construído pela intersecção do lado esquerdo e do lado direito de cada regra. A entidade Operator controla o acesso à entidade Pump quando o cliente tenta usá-la. Se a bomba não estiver sendo usada, o operador aceita (gerando uma mensagem de PreOrd) o pré-pagamento, entrega ao cliente um cupom (regra Accept) e prepara a bomba para ser ativada (regra Serve). A bomba pode ser ativada, mudando o flag de livre (Free) para ativado (FF) (regra Activate) e pode iniciar o abastecimento de combustível mudando seu flag de FF para Busy (regra Start). Eventualmente, a bomba finaliza o abastecimento de combustível - mudando seu flag de Busy para Free novamente - e indica para o operador (gerando a mensagem Charge) o valor real que o cliente gastou (regra Stop). Finalmente, o operador pode devolver o troco para o cliente (regra Finish). Se a bomba estiver sendo usada, o operador avisa o cliente (enviando a mensagem Busy) para tentar novamente (regra Reject). Além disso, a entidade operador (Operator) pode desempenhar a atividade de análise da bomba, verificando se o combustível distribuído está puro ou não (regras PureAnalysis ou ImpureAnalysis, respectivamente).

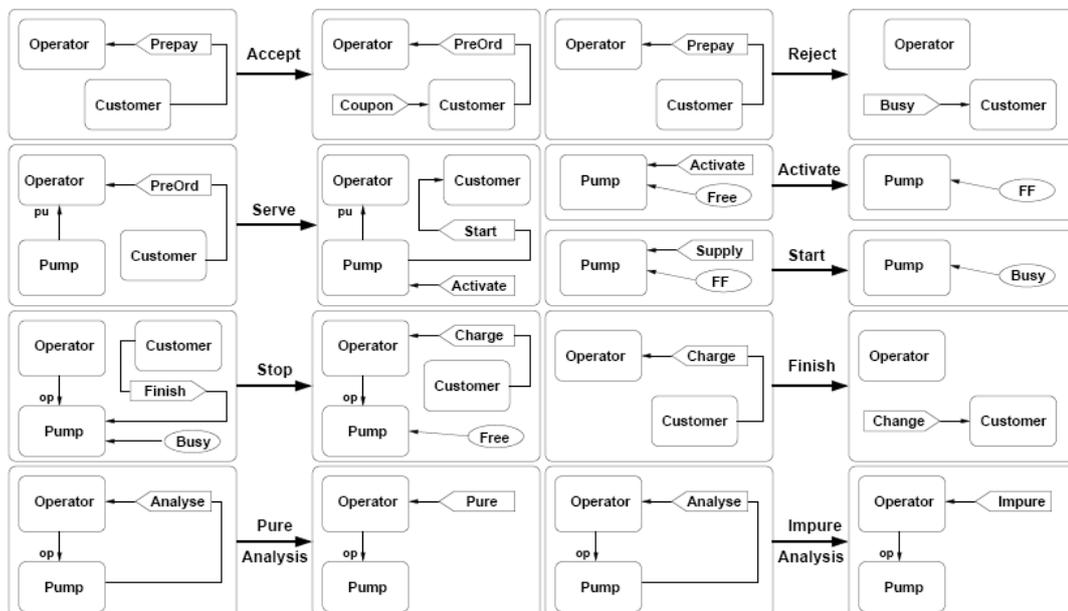
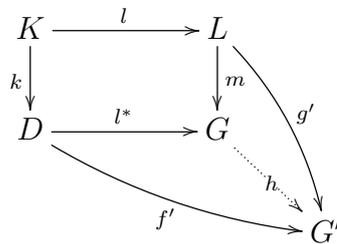


Figura 3: Exemplo de uma GG

A semântica operacional de uma gramática de grafos é dada por derivações. A derivação descreve a aplicação de uma regra em um grafo representando o estado do sistema. A interpretação operacional da aplicação de uma regra é: itens no lado

esquerdo que não estão na interface, são removidos; itens no lado esquerdo da regra que estão na interface são preservados; e, itens no lado direito da regra que não estão na imagem de  $r_p$  são criados. Na abordagem algébrica da gramática de grafos, uma derivação é dada em termos de pushouts em  $(THGrafo(T))$ . Intuitivamente, um pushout de dois grafos com relação a outro, chamado grafo interface, é dada pela colagem (*gluing*) destes dois grafos, identificando os itens da interface.

**Definição 5. (Pushout e Complemento de Pushout)** Dada uma categoria  $C$  e duas setas  $l : K \rightarrow L$  e  $k : K \rightarrow D$  de  $C$ , uma tripla  $(G, l^* : D \rightarrow G, m : L \rightarrow G)$  como no diagrama abaixo, é chamada de pushout de  $(l, k)$  se as seguintes restrições são satisfeitas:



- **Comutatividade:**  $m \circ l = l^* \circ k$ , e
- **Propriedade Universal:** para todos os objetos  $G'$  e setas  $g' : L \rightarrow G'$  e  $f' : D \rightarrow G'$ , com  $g' \circ l = f' \circ k$ , existe uma única seta  $h : G \rightarrow G'$  tal que,  $h \circ m = g'$  e  $h \circ l^* = f'$ .

Neste caso,  $G$  é chamado de pushout de  $(l, k)$ . Dadas as setas  $l : K \rightarrow L$  e  $m : L \rightarrow G$ , o complemento de pushout de  $(l, m)$  é a tripla  $(D, k : K \rightarrow D, l^* : D \rightarrow G)$  tal que  $\langle G, m, l^* \rangle$  é o pushout de  $\langle l, k \rangle$ . Nesse caso  $D$  é chamado de complemento de pushout de  $(l, m)$ .

Na abordagem seguida neste trabalho (DPO), uma derivação direta é definida por dois *pushouts* na categoria de grafos tipados e morfismos de grafos tipados  $(THGrafo(T))$ : o primeiro deleta todos os itens que devem ser consumidos e o segundo inclui todos os itens que devem ser criados. Uma derivação é uma sequência de derivações diretas onde o grafo final de uma é o grafo inicial da outra. A aplicação de uma regra a um grafo só é possível se houver um *match*, isto é, uma ocorrência do lado esquerdo da regra no grafo do estado corrente do sistema. Um *match* (ocorrência) é definido por um morfismo total de grafos tipados.

**Definição 6. (Derivação direta e derivação)** Dado um grafo  $H$  tipado em  $T$ , uma regra  $p : L_p \xleftarrow{l_p} K_p \xrightarrow{r_p} R_p$  tipada sobre  $T$  e um morfismo total e injetor de grafos tipados  $m : L_p \hookrightarrow H$ , denominado *match*, uma derivação direta de  $H$  para  $H'$  usando  $p$  (baseado em  $m$ ) existe, se e somente se, o diagrama abaixo pode ser construído,

onde ambos os quadrados são *pushouts* em  $THGrafo(T)$ . Nesse caso a derivação direta é denotada por  $\delta : H \xrightarrow{p,m} H'$  ou  $\delta : H \xrightarrow{p} H'$  se  $m$  não precisar ser explícito. Dada uma  $GG = (T, H, P, \pi)$ , uma derivação  $\rho : H_0 \xrightarrow{p^1,m^1} H_1 \xrightarrow{p^2,m^2} H_2 \dots$  de  $GG$  é uma sequência de derivações diretas  $\delta : H_i \xrightarrow{p_i,m_i} H_{i+1}$ , onde  $H_{i+1} = H'_i, i \geq 0$  e  $p_i \in p$ . A semântica de uma gramática de grafos  $GG$  é definida pelo conjunto de todas as derivações em  $GG$ , denotado por  $Der(GG)$ .

$$\begin{array}{ccccc} L_p & \xleftarrow{l_p} & K_p & \xrightarrow{r_p} & R_p \\ m \downarrow & & \downarrow & & \downarrow \\ H & \xleftarrow{\quad} & D & \xrightarrow{\quad} & H' \end{array}$$

A construção do diagrama da Definição 6 depende da existência de  $D$ , denominado complemento do *pushout*. Para garantir esta existência, o *match*  $m$  deve satisfazer a condição de colagem com relação à  $l_p$ . Esta condição é dividida em duas partes: a condição de arestas pendentes, isto é, se um vértice é deletado, não podem existir arestas que chegam ou partem deste vértice; e, a condição de identificação, isto é, dois vértices podem ser identificados por  $m$  somente se ambos são preservados. A Figura 4 mostra uma derivação direta usando a regra *Accept*. No grafo  $G$  é encontrada uma ocorrência do lado esquerdo da regra *Accept*, o indica que a regra pode ser aplicada. O grafo  $H$  é obtido pela aplicação desta regra no grafo  $G$ , onde os vértices *Customer* e *Operator* são preservados, as arestas *PreOrd* e *Coupon* são criadas e a aresta *Prepay* é deletada.

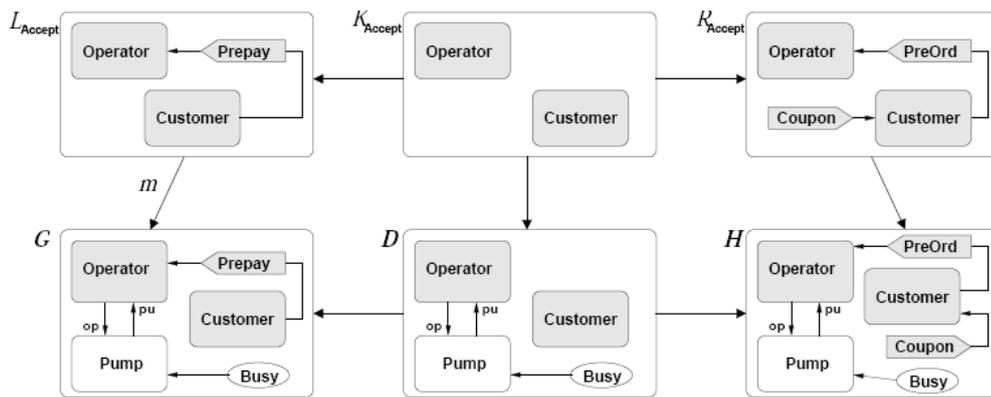


Figura 4: Derivação direta de  $G$  para  $H$  usando a regra *Accept* baseada em  $m$

Em gramática de grafos regras podem ser aplicadas em paralelo. Se existir uma ocorrência no grafo inicial e existir mais de uma regra que possa ser aplicada, elas serão aplicadas em paralelo. Essa característica de não-determinismo da gramática de grafos pode levar a estados não desejados no sistema. Para garantir que uma regra somente possa ser realizada antes de outra, uma regra seja pré-condição de outra ou até mesmo colocar uma ordem de aplicação de regras, é necessário que a gramática

tenha algum tipo de controle na aplicação de regras.

Condições de controle podem ser usadas para restringir o não determinismo na aplicação das regras de uma gramática de grafos. Um típico exemplo é permitir somente derivações onde a sequência de aplicação de regras permitem um controle particular da linguagem. Expressões regulares são utilizadas para especificar essas condições. Aqui, considera-se a classe de expressões regulares sobre o vocabulário  $\Sigma$ , denotado por  $EXP^\Sigma$ , que é definida por  $\varepsilon, \emptyset \in EXP^\Sigma$ ,  $\Sigma \subseteq EXP^\Sigma$  e  $(E_1; E_2), (E_1 + E_2), (E_1)^* \in EXP^\Sigma$ , se  $E_1, E_2 \in EXP^\Sigma$ . A fim de omitir parênteses, assume-se a seguinte precedência: primeiro  $*$ , então  $;$  e por fim  $+$ . Intuitivamente, a expressão  $\varepsilon$  descreve a aplicação de nenhuma regra;  $\emptyset$  representa o conjunto vazio;  $p \in \Sigma$  especifica a aplicação da regra  $p$ , exatamente uma vez;  $(E_1; E_2)$  descreve a aplicação das regras especificadas em  $E_1$  seguido da aplicação das regras especificadas em  $E_2$ ;  $(E_1 + E_2)$  especifica a escolha não-determinística entre  $E_1$  e  $E_2$ ; e  $(E_1)^*$  representa a interação de frequência arbitrária de  $E_1$ . A linguagem descrita por uma expressão regular  $E$  é denotada por  $\mathcal{L}(E)$ . O caminho de rótulos de uma derivação  $\rho : H_0 \xrightarrow{p_1, m_1} H_1 \xrightarrow{p_2, m_2} \dots \xrightarrow{p_n, m_n} H_n$ , denotado por  $path(\rho)$ , é dado por  $p_1; p_2; \dots; p_n$ .

**Definição 7. (Gramática de Grafos Controlada)** *Seja  $GG = (T, H, P, \pi)$  uma gramática de grafos e  $exp = \{E | E \in EXP^P\}$  um conjunto de expressões regulares sobre  $P$ . Uma gramática de grafos controlada é dada por um par  $CGG = (GG, exp)$ . O conjunto de todas derivações da  $CGG$ , denotado por  $Der(CG)$ , é definido pelo conjunto de todas as derivações  $\rho \in Der(GG)$ , tal que  $path(\rho) \in \mathcal{L}(E)$ , para alguma  $E \in exp$ .*

**Exemplo 4. (GG Controlada)** *Este exemplo mostra uma gramática de grafos controlada. Considerando a gramática definida no Exemplo 3, pode-se construir uma gramática controlada adicionando-se o seguinte conjunto de expressões regulares sobre as regras da gramática  $exp = \{(PureAnalysis + ImpureAnalysis), (Accept + Reject; Serve; Activate; Start; Stop; Finish), \varepsilon\}$ . Por esse conjunto de expressões regulares a aplicação das regras está restrita a três possíveis comportamentos: (1) uma das duas regras,  $PureAnalysis$  ou  $ImpureAnalysis$ , poderá ser aplicada (a escolha é feita não-deterministicamente); (2) é feita uma escolha não determinística entre aplicação da regra  $Reject$  e a aplicação de uma sequência de regras começando por  $Accept$ , seguindo por  $Serve$ ,  $Activate$ ,  $Start$ ,  $Stop$  e finalizando por  $Finish$ ; ou ainda, (3) nenhuma regra é aplicada  $\varepsilon$ .*

O próximo Capítulo apresenta-se o modelo PopOrg com as definições dos níveis populacional, organizacional e das relações de implementação entre os níveis.

## 4 MODELO POPORG

O Modelo Populacional-Organizacional (PopOrg) foi introduzido por (DEMAZEAU; COSTA, 1996), como um modelo mínimo e formal para sistemas multiagentes (SMA) com organizações dinâmicas. Este modelo considera dois dos aspectos mais importantes dos SMAs: a população e a organização. A população é composta por um conjunto de agentes (WOOLDRIDGE, 2001), um conjunto de comportamentos que os agentes são capazes de realizar e um conjunto de todas as interações que podem ser estabelecidas entre agentes.

A organização, por sua vez, é composta de dois níveis: micro e macro. No nível micro, pode-se identificar funcionalidades como papéis, ações, comportamentos e ligações organizacionais, onde um papel pode ser definido em relação a alguns processos sociais ou por uma interação da qual ele participe. Os papéis identificam as principais funções dos SMAs, podendo ter algumas características como permissões, protocolos e responsabilidades. Ligações organizacionais são influências mútuas entre papéis envolvidos em um certo processo global. O nível macro, também chamado de nível de grupos, é composto por papéis, ligações e comportamentos.

Existem relações de implementação entre os níveis populacional e micro-organizacional e ainda entre os níveis micro e macro-organizacional. A implementação do nível micro é dada por uma relação que associa cada papel (micro-organizacional) a um conjunto de agentes (populacional) que o implementam. Além disso, cada ligação entre dois papéis também é associada a um conjunto de processos de troca que são realizados pelos agentes que implementam estes papéis. A relação de implementação de nível macro é definida de modo análogo, associando grupos a conjuntos de papéis.

**Definição 8. (Modelo PopOrg)** *O Modelo PopOrg de um Sistema Multi-Agente é a estrutura  $PopOrg = (POP, ORG, IMP)$  constituída por uma estrutura populacional  $POP$ , uma estrutura organizacional  $ORG$  e uma relação de implementação  $IMP$  estabelecida entre essas estruturas.*

Agentes, papéis ou grupos em um SMA podem realizar ações. As ações e a ordem em que essas ações podem ser realizadas são determinadas pelos possíveis

comportamentos do sistema. Além disso, os agentes (papéis ou grupos) podem cooperar, realizando uma sequência de ações em conjunto. Essa cooperação é modelada por um processo de troca, que especifica para cada instante de tempo, dois conjuntos de ações, determinando as possíveis trocas entre um par de agentes (papéis ou grupos), os quais executam suas respectivas ações juntos ou de forma intercalada. Nos níveis organizacionais existem as chamadas micro e macroligações, que são influências mútuas entre dois papéis. A Figura 5 mostra um esquema do modelo PopOrg com todos os seus níveis.



Figura 5: PopOrg com os níveis micro-macro da estrutura organizacional

No modelo PopOrg os comportamentos que os agentes podem realizar dão a dinâmica do modelo. Cada agente (papéis ou grupos) tem uma série de ações que pode realizar. A sequência de execução das ações em um determinado instante é denominado comportamento. Além disso, é possível descrever no modelo a interação entre agentes (papéis ou grupos). Essa interação é chamada de processos de troca. Um processo de troca é dado por uma função que especifica para cada instante de tempo um conjunto de ações que podem ser realizadas concomitantemente por um par de agentes (papéis ou grupos).

**Definição 9. (Comportamento e Processo de Troca)** Dado um universo de ações  $Act \neq \emptyset$  e uma sequência discreta de instantes de tempo  $T$ , em que  $t \in T$ . Um comportamento é definido por uma função  $bh : T \rightarrow \wp(Act)$ , que especifica, para cada tempo  $t$ , um subconjunto de ações que podem ser realizadas por um agente, papel ou grupo. Conjunto de todos os comportamentos é denotado por  $Bh = [T \rightarrow \wp(Act)]$ . Um processo de troca é definido por uma função  $e : T \rightarrow \wp(Act) \times \wp(Act)$  que especifica, para cada tempo  $t$ , um par de subconjuntos de ações. O conjunto de todos os processos de troca é denotado por  $Ep = [T \rightarrow \wp(Act) \times \wp(Act)]$ .

**Notação 1.** O universo de ações é particionado em  $Act = Act_{Ag} \cup Act_{Ro} \cup Act_{Grp}$ , onde  $Act_{Ag}$ ,  $Act_{Ro}$  e  $Act_{Grp}$  são os conjuntos de ações dos agentes, papéis e grupos, respectivamente. O universo de comportamentos é dividido em  $Bh = Bh_{Ag} \cup Bh_{Ro} \cup Bh_{Grp}$ ,

onde  $Bh_{Ag}$ ,  $Bh_{Ro}$  e  $Bh_{Grp}$  são os conjuntos de comportamentos dos agentes, papéis e grupos, respectivamente. O universo de todos os processos de troca também é particionado em  $Ep = Ep_{Ag} \cup Ep_{Ro} \cup Ep_{Grp}$ , onde  $Ep_{Ag}$ ,  $Ep_{Ro}$  e  $Ep_{Grp}$  são os conjuntos de processos de troca dos agentes, papéis e grupos, respectivamente.

#### 4.1 Estrutura Populacional (POP)

A estrutura populacional é o nível mais baixo do modelo PopOrg - o nível de agentes. Essa estrutura é composta por um conjunto de todas as ações dos agentes, um conjunto de comportamentos que esses agentes podem ter, os possíveis processos de troca que podem ocorrer no sistema a cada instante, bem como funções que definem as capacidades de comportamento e de troca de cada agente. A capacidade de troca entre dois agentes é restrita pela capacidade de comportamento de ambos os agentes, ou seja, se dois agentes não possuem em seus comportamentos as ações necessárias para realizar uma determinada troca, esta não pode ser definida.

**Definição 10. (Estrutura Populacional de um SMA)** Dado um universo de agentes  $Ag \neq \emptyset$  e uma sequência de instantes de tempo  $T$ . A estrutura populacional de um sistema multiagente é um conjunto tempo-indexado POP de estados da estrutura populacional, com  $POP^t = (AG^t, ACT^t, BH^t, EP_{Ag}^t, Bc^t, Ec^t)$ , onde para todo  $t \in T$ :

- $AG^t \in \wp(Ag)$  é a população do sistema no tempo  $t$ ;
- $ACT^t \in \wp(Act_{Ag})$  é o conjunto de todas as ações dos agentes no tempo  $t$ ;
- $BH^t \in \wp(Bh_{Ag})$ , com  $BH^t \subseteq [T \rightarrow \wp(ACT^t)]$ , é o conjunto dos possíveis comportamentos dos agentes no tempo  $t$ ;
- $EP_{Ag}^t \in \wp(Ep_{Ag})$ , com  $EP_{Ag}^t \subseteq [T \rightarrow \wp(ACT^t) \times \wp(ACT^t)]$ , é o conjunto de possíveis processos de troca entre agentes no tempo  $t$ ;
- $Bc^t : AG^t \rightarrow \wp(BH^t)$  é a função de capacidade comportamental no tempo  $t$ ;
- $Ec^t : AG^t \times AG^t \rightarrow \wp(EP_{Ag}^t)$  é a função de capacidade de troca no tempo  $t$ , tal que, capacidade de troca de um par de agentes é restrita por sua capacidade comportamental, isto é:  $\forall a_1, a_2 \in AG^t, \forall e \in Ec^t(a_1, a_2), \forall t' \in T$ :

$$Prj_1(e(t')) \subseteq \bigcup \{b(t') | b \in Bc^t(a_1)\} \quad (1)$$

$$Prj_2(e(t')) \subseteq \bigcup \{b(t') | b \in Bc^t(a_2)\} \quad (2)$$

onde  $Prj_1$  e  $Prj_2$  são funções de projeção.

O exemplo a seguir ilustra a definição de uma população, onde são definidos os agentes, seus possíveis comportamentos e as possíveis interações entre eles. Para os exemplos deste trabalho, considera-se um cenário de insetos sociais (formigas) (FERREIRA JR., 2008; WILSON, 1980), problema bastante estudado em SMAs e que aqui será utilizado como exemplo deste trabalho. Alguns problemas são comumente encontrados na literatura sobre o uso de insetos sociais para resolução de problemas complexos, como a divisão de trabalho, resolução de problemas NP-Completo, sistemas de telecomunicação, paralelismo, controle de robôs, entre outros. As formigas são insetos sociais muito bem organizados. Nessa sociedade, cada inseto age independente de um controle central, cada indivíduo realiza suas tarefas e a sociedade funciona perfeitamente. Cada *casta* tem uma função específica na sociedade. Além disso, existem ações importantes para a organização que são desempenhadas pelas *castas*.

**Exemplo 5. (População)** Neste cenário, os agentes representam as formigas de uma sociedade, as quais podem realizar algumas ações e interagir entre si. O sistema  $Pop^t$  a seguir define esse cenário. Dado um universo de agentes  $Ag = \{Formiga_1, Formiga_2, Formiga_3, Formiga_4\}$ , considera-se a seguinte população, em um instante  $t$ :  $POP^t = (AG^t, ACT^t, BH^t, EP^t, Bc^t, Ec^t)$ . O conjunto de agentes é definido por  $AG^t = \{Formiga_1, Formiga_2, Formiga_3, Formiga_4\}$ . As ações que estes agentes podem realizar estão definidas em  $ACT^t = \{Fecundar, PorOvos, RecAlimento, Reproduzir, CuiOvos, CuiLarvas, CuiPupas, ColAlimento, DistAlimento, ConsNinho, ManuNinho, CuiEntrada, CuiProle, AtacPresa\}$ . Os comportamentos que podem ocorrer neste instante da população são definidos por  $Bh_{Ag}^t = \{C_1, C_2, C_3, C_4, C_5, C_6\}$ , onde:

$$C_1 = \{0 \mapsto \{Reproduzir, Fecundar\}, 1 \mapsto \{PorOvos\}, 2 \mapsto \{RecAlimento\}\}$$

$$C_2 = \{0 \mapsto \{CuiOvos\}, 1 \mapsto \{CuiLarvas\}, 2 \mapsto \{CuiPupas\}\}$$

$$C_3 = \{0 \mapsto \{ConsNinho\}, 1 \mapsto \{ColAlimento\}, 2 \mapsto \{DistAlimento\}\}$$

$$C_4 = \{0 \mapsto \{CuiEntrada\}, 1 \mapsto \{CuiProle\}, 2 \mapsto \{ManuNinho\}\}$$

$$C_5 = \{0 \mapsto \{AtacPresa\}, 1 \mapsto \{CuiEntrada\}, 2 \mapsto \{CuiProle, ManuNinho\}\}$$

$$C_6 = \{0 \mapsto \{Reproduzir\}, 1 \mapsto \{RecAlimento\}, 2 \mapsto \{RecAlimento\}\}$$

Pela definição de  $BH^t$ , pode-se verificar que existem seis diferentes comportamentos, os quais podem ser realizados pelos agentes desta população. Pelo comportamento  $C_1$ , por exemplo, um agente que apresente este comportamento pode realizar, no instante 0, as ações *Reproduzir* e/ou *Fecundar*, no instante 1, a ação *PorOvos* e no instante 2, a ação *RecAlimento*. No exemplo, tem-se o seguinte conjunto de possíveis interações  $EP_{Ag}^t = \{T_1, T_2, T_3, T_4\}$ , onde:

$$T_1 = \{0 \mapsto (\{Reproduzir, Fecundar\}, \{CuiOvos\}), 1 \mapsto (\{PorOvos\}, \{CuiLarvas\}), 2 \mapsto (\{RecAlimento\}, \{CuiPupas\})\}$$

$$T_2 = \{0 \mapsto (\{Reproduzir, Fecundar\}, \{Reproduzir\}),$$

$$\begin{aligned}
& 1 \mapsto (\{PorOvos\}, \{RecAlimento\}), 2 \mapsto (\{RecAlimento\}, \{RecAlimento\}) \\
T_3 = & \{0 \mapsto (\{Reproduzir, Fecundar\}, \{AtacPresas\}), 1 \mapsto (\{PorOvos\}, \{CuiEntrada\}), \\
& 2 \mapsto (\{RecAlimento\}, \{CuiProle, ManuNinho\})\} \\
T_4 = & \{0 \mapsto (\{ConsNinho\}, \{CuiEntrada\}), 1 \mapsto (\{ColAlimento\}, \{CuiProle\}), \\
& 2 \mapsto (\{DistAlimento\}, \{ManuNinho\})\}
\end{aligned}$$

Pelo processo de troca  $T_1$ , dois agentes podem interagir da seguinte forma: no instante 0, o primeiro agente pode realizar a ação *Reproduzir* e/ou *Fecundar*, enquanto o segundo agente pode realizar a ação *CuiOvos*; já no instante 1, o primeiro agente realiza a ação *PorOvos* e o segundo realiza a ação *CuiLarvas*; por fim, no instante 2 o primeiro agente realiza a ação *RecAlimento* e o segundo realiza a ação *CuiPupas*. Os demais processos de troca definem outras possíveis interações entre os agentes.

Além de definir as ações e interações possíveis no sistema, deve-se também definir quais os comportamentos que cada agente pode apresentar e de quais interações eles podem participar. Assim a função  $Bc^t$  associa a cada agente um conjunto de comportamentos. No exemplo, define-se esta função como segue:

$$Bc^t = \{Formiga_1 \mapsto \{C_1\}, Formiga_2 \mapsto \{C_2, C_3\}, Formiga_3 \mapsto \{C_4, C_5\}, Formiga_4 \mapsto \{C_6\}\}.$$

Por essa definição pode-se observar que o agente  $Formiga_1$  apresenta o comportamento  $C_1$ ; já o agente  $Formiga_2$  possui dois comportamentos  $C_2$  e  $C_3$ ; o agente  $Formiga_3$  também possui dois comportamentos identificados por  $C_4$  e  $C_5$ ; e por fim o agente  $Formiga_4$  possui o comportamento  $C_6$ .

A função  $Ec^t$  descreve quais agentes podem interagir, realizando determinados processos de troca. No cenário apresentado, tem-se as seguintes capacidades de troca

$$\begin{aligned}
Ec = & \{(Formiga_1, Formiga_2) \mapsto \{T_1\}, (Formiga_1, Formiga_4) \mapsto \{T_2\}, \\
& (Formiga_1, Formiga_3) \mapsto \{T_3\}, (Formiga_2, Formiga_3) \mapsto \{T_4\}\}.
\end{aligned}$$

Por esta definição pode-se observar que as formigas  $Formiga_1$  e  $Formiga_2$  podem interagir de acordo com o processo de troca  $T_1$ , as formigas  $Formiga_1$  e  $Formiga_4$  podem interagir pelo processo de troca  $T_2$ . O processo de troca  $T_3$  é a possível interação entre as formigas  $Formiga_1$  e  $Formiga_3$ , e por fim, a interação entre as formigas  $Formiga_2$  e  $Formiga_3$  é dado pelo processo de troca  $T_4$ .

No modelo PopOrg, o nível organizacional é o nível mais alto do modelo. É possível identificar na organização dois subníveis: micro e macro organizacionais. Em cada nível é possível representar certas características pertinentes a papéis e grupos dessa organização. Na próxima Seção serão abordados os conceitos básicos da estrutura organizacional do modelo PopOrg.

## 4.2 Estrutura Organizacional (ORG)

A estrutura organizacional é composta de um nível micro-organizacional e um nível macro-organizacional. O nível micro-organizacional é aquele em que se encontram os papéis, os comportamentos dos papéis e as ligações da organização. No nível macro-organizacional, também chamado de nível de grupos, encontram-se os papéis dos grupos, as ligações e os comportamentos dos grupos.

**Definição 11. (Estrutura Organizacional de um SMA)** *Uma estrutura  $\omega\Omega$ -organizacional é um par  $ORG_{\omega\Omega} = (ORG_{\omega}, ORG_{\Omega})$ , onde  $ORG_{\omega}$  e  $ORG_{\Omega}$  são os níveis micro e macro da estrutura organizacional, respectivamente.*

Neste trabalho será modelado apenas a estrutura micro-organizacional. Por isso, as definições da macro-organização serão omitidas.

O nível micro da estrutura organizacional é composto por papéis, ligações e a função de capacidade de ligação dos papéis. Os papéis são muito importantes no nível micro da estrutura organizacional. Eles possuem ações que são realizadas em determinados instantes e dão origem ao comportamento do nível. As ligações, por sua vez, são interações entre pares de papéis, cada qual executando suas ações em conjunto ou alternadamente. A capacidade de ligação é uma função que descreve quais papéis podem realizar cada ligação.

**Definição 12. (Micropapel)** *Um micropapel  $r$  é um subconjunto não vazio de comportamentos de papéis em  $Bh_{Ro}$ , ou seja,  $r \in \wp(Bh_{Ro})$ , e  $r \neq \emptyset$ . O conjunto de todos os micropapéis é denotado  $Ro_{\omega}$ .*

As interações entre os papéis são determinadas por meio de ligações. Essas ligações são descritas por dois papéis e um processo de troca que determina como essa ligação pode ser realizada.

**Definição 13. (Microligação)** *Uma microligação  $l$  entre um par de micropapéis  $r_1, r_2 \in Ro_{\omega}$  é definida por uma tupla  $l = (r_1, r_2, e)$ , especificando um processo de troca  $e \in Ep_{Ro}$ , que os micropapéis ligados  $r_1$  e  $r_2$  podem realizar. O conjunto de todas as microligações é denotado por  $Li_{\omega}$ .*

A estrutura micro-organizacional é composta por um conjunto de micropapéis, um conjunto de microligações organizacionais e uma função de capacidade de ligação. Também podem ser descritas, nessa estrutura, as interações que podem ocorrer entre os papéis. Uma ligação pode ser estabelecida se e somente se os papéis envolvidos têm a capacidade de realizá-la. Essa restrição é dada pela função de capacidade de ligação.

**Definição 14. (Estrutura de nível Micro-Organizacional)** Dado um universo de papéis  $Ro_\omega \neq \emptyset$  e uma sequência de instantes de tempo  $T$ . Uma estrutura micro-organizacional de um SMA é um conjunto tempo-indexado  $ORG_\omega$ , com  $ORG_\omega^t = (RO_\omega^t, LI_\omega^t, Lc_\omega^t)$ , para todo  $t \in T$ :

- $RO_\omega^t \in \wp(Ro_\omega)$ , é o conjunto de micropapéis existentes na organização no tempo  $t$ .
- $LI_\omega^t \in \wp(Li_\omega)$ , é o conjunto de microligações existentes na organização no tempo  $t$ .
- $Lc_\omega^t : RO_\omega^t \times RO_\omega^t \rightarrow \wp(LI_\omega^t)$  é a função de capacidade de microligação no tempo  $t$ , especificando um conjunto de microligações que pode ocorrer entre dois papéis.
- $\forall l = (r_1, r_2, e) \in LI_\omega^t : l \in Lc_\omega^t(r_1, r_2)$ , isto é, cada possível microligação tem que estar na capacidade de ligação dos dois micropapéis que ela liga no tempo  $t$ .
- $\forall r_1, r_2 \in RO_\omega^t, \forall e \in Ep_{Ro}, \forall t \in T : (r_1, r_2, e) \in LI_\omega^t \Rightarrow \forall t' \in T :$

$$Prj_1(e(t')) \subseteq \bigcup \{b(t') \mid b \in r_1\} \quad (3)$$

$$Prj_2(e(t')) \subseteq \bigcup \{b(t') \mid b \in r_2\} \quad (4)$$

onde  $Prj_3, Prj_4$  são funções de projeção, de modo que as capacidades de troca dos micropapéis são restringidas por suas respectivas capacidades de comportamento.

**Exemplo 6. (Organização)** Neste exemplo, os papéis representam as castas Reprodutora, Operária e Soldado, os quais podem realizar algumas ações e interagir entre si. Considera-se a seguinte organização, em um instante  $t$ :  $ORG_\omega^t = (RO_\omega^t, LI_\omega^t, Lc_\omega^t)$ . O conjunto de papéis é definido por  $RO_\omega^t = \{Reprodutora, Operária, Soldado\}$ , onde:

$Reprodutora = \{0 \mapsto \{VooNupcial\}, 1 \mapsto \{Reprodução\}, 2 \mapsto \{Alimentação\}\}$

$Operária = \{0 \mapsto \{Construção\}, 1 \mapsto \{CuiProle, ColAlimento\}, 2 \mapsto \{Manutenção\}\}$

$Soldado = \{0 \mapsto \{Defesa\}, 1 \mapsto \{Ataque\}, 2 \mapsto \{Manutenção\}\}$

Pode-se observar que esta organização tem três micropapéis: Reprodutora, Operária e Soldado. Cada um deles é definido por diferentes comportamentos. Por exemplo, o papel Reprodutora requer as seguintes ações: primeiramente deve ser realizada a ação  $VooNupcial$ , depois a ação  $Reprodução$  e, por fim, a ação  $Alimentação$ .

As ligações entre os três papéis do nível micro é dada por  $LI_{RO}^t = \{l_1, l_2\}$ , onde a ligação  $l_1$  entre os papéis Reprodutora e Operária, é dada pelo processo de troca  $T_1$ :

$l_1 = (Reprodutora, Operária, T_5)$ , onde:

$$T_5 = \{0 \mapsto \{(\{VooNupcial\}, \{Construção\})\}\},$$

$$1 \mapsto \{(\{Reprodução\}, \{CuiProle, ColAlimento\})\},$$

$$2 \mapsto \{(\{Alimentação\}, \{Manutenção\})\}.$$

$l_2 = (Operária, Soldado, T_6)$ , onde:

$$T_6 = \{0 \mapsto \{(\{CuiProle_O\}, \{Defesa_S\})\},$$

$$1 \mapsto \{(\{ColAlimento_O\}, \{Defesa_S\})\},$$

$$2 \mapsto \{(\{Manutenção_O\}, \{\})\}.$$

A capacidade de ligação da organização é dada por:

$$Lc_\omega^t = \{(Reprodutora, Operária) \mapsto \{l_1\}, (Operária, Soldado) \mapsto \{l_2\}\}.$$

Pela capacidade de ligação, a microligação  $l_1$  é realizada entre os micropapéis *Reprodutora* e *Operária*. Já os micropapéis *Operária* e *Soldado* realizam a microligação  $l_2$ . As trocas associadas a cada microligação são restritas pelo comportamento dos papéis envolvidos, ou seja, só pode ocorrer uma troca entre papéis, se ambos tiverem um comportamento associado e, ainda, esse comportamento deve ser respeitado pelo processo de troca.

### 4.3 Relações de Implementação

As relações de implementação definem como é feito o vínculo entre as estruturas populacional e organizacional do modelo PopOrg. A relação de implementação do nível micro mostra como uma população implementa a micro-organização.

Para permitir que as ações dos papéis possam ser compostas por uma ou mais ações dos agentes, é definida a função *Ras*, a qual associa cada ação de um papel a um conjunto de ações de agentes. A função *Ras* pode ser naturalmente estendida para conjuntos de ações de papéis, ambas chamadas de *Ras*. Se  $Ras(\rho) = Ac_{ag}$ ; diz-se que a ação do papel  $\rho$  é propriamente implementada pelo conjunto de ações de agentes  $Ac_{ag}$ .

A relação de implementação de nível micro define as relações entre o nível populacional e o nível micro-organizacional de acordo com a função *Ras*.

**Definição 15. (Relação de implementação de nível micro)** Uma relação de implementação de nível micro  $IMP_\omega$ , para  $ORG_\omega = (RO_\omega, LI_\omega, Lc_\omega)$ , sobre  $POP = (AG, ACT, BH, EP_{Ag}, Bc, Ec)$  de acordo com a função  $Ras : Act_{Ro} \rightarrow \wp(Act_{Ag})$  é um conjunto tempo-indexado das relações de implementação de nível micro  $IMP_\omega^{Ras}$ , com  $(IMP_\omega^{Ras})^t \subseteq (RO^t \times AG^t) \cup (LI^t \times EP_{Ag}^t)$ , para cada  $t \in T$ :

- $(RO^t \times AG^t)$  é o conjunto de todas as possíveis implementações de micropapéis, isto é, o conjunto de todas as formas de atribuir micropapéis a agentes no tempo  $t$ , de modo que, se  $(r, a) \in (IMP_\omega^{Ras})^t$ , então o micropapel  $r$  é implementado pelo agente  $a$  no tempo  $t$  (possivelmente em um modo compartilhado e não exclusivo);

- $(LI^t \times EP_{Ag}^t)$  é conjunto de todas possíveis implementações de microligações, isto é, o conjunto de todas as formas de atribuir microligações a processos de troca de agentes, de modo que se  $(l, e) \in (IMP_{\omega}^{Ras})^t$ , então a microligação  $l$  é implementada (possivelmente em um modo compartilhado, e não exclusivo) por um processo de troca  $e$  no tempo  $t$ ;

Uma implementação de micropapel é definida de acordo com os papéis que os agentes podem desempenhar no modelo. Para que a relação de implementação seja própria, o conjunto de comportamentos exigidos pelo papel deve ser desempenhado pelo conjunto de agentes associados a este papel.

**Definição 16. (Implementação própria de micropapel)** Um micropapel  $r \in RO_{\omega}^t$  em  $ORG_{\omega}^t$  é propriamente implementado por  $(IMP_{\omega}^{Ras})^t$  sobre  $POP^t$ , se e somente se, existir um conjunto  $A \subseteq AG^t$  de agentes, tal que as seguintes condições sejam satisfeitas:

- $\forall a \in A : (r, a) \in (IMP_{\omega}^{Ras})^t$ , todos os agentes em  $A$  participam da implementação do micropapel  $r$  no tempo  $t$ ;
- $\forall t' \in T : \bigcup \{ \bigcup Ras(b_r(t')) \mid b_r \in r \} \subseteq \bigcup \{ b_a(t') \mid b_a \in Bc^t(a), a \in A \}$ , o conjunto de comportamentos exigidos pelo micropapel  $r$  pode ser desempenhado pelos agentes em  $A$  (possivelmente em um modo compartilhado e não exclusivo) de acordo com a função  $Ras$ ;

Da mesma forma que os agentes podem implementar um papel de forma própria as ligações também podem ser implementadas pelos processos de troca de agentes. Uma implementação própria de uma microligação é uma relação entre o nível populacional e o nível micro-organizacional, onde uma ligação entre dois papéis é implementada por um conjunto de processos de troca dos agentes.

**Definição 17. (Implementação própria de uma microligação)** Uma microligação  $l = (r_1, r_2, e) \in LI_{\omega}^t$  em  $ORG_{\omega}^t$  é propriamente implementada por  $(IMP_{\omega}^{Ras})^t$  sobre  $POP^t$ , se e somente se, existe um subconjunto  $E \subseteq \bigcup \{ Ec^t(a_1, a_2) \mid (r_1, a_1), (r_2, a_2) \in IMP_{\omega} \}$  de um conjunto de processos de troca pertencentes à capacidade de troca dos agentes que implementam os papéis  $r_1$  e  $r_2$  no tempo  $t \in T$ , tal que as seguintes condições sejam satisfeitas:

- $\forall e' \in E : (l, e') \in (IMP_{\omega}^{Ras})^t$ , cada processo de troca em  $E$  participa da implementação da microligação  $l$  no tempo  $t \in T$ ;
- $\forall t' \in T : \bigcup Ras(e(t')) \subseteq \bigcup \{ e'(t') \mid e' \in E \}$ , isto é, toda projeção  $Prj_i$  ( $i = 1, 2$ ) do processo de troca exigido por  $l$  pode ser realizada pelo conjunto de projeções correspondentes do processos de troca  $E$  (possivelmente em um modo compartilhado e não exclusivo).

**Definição 18. (Relação de implementação própria de nível micro)** *Uma relação de implementação de nível micro  $IMP_{\omega}^{Ras}$  para  $ORG_{\omega}$  sobre  $POP$ , é própria se e somente se, a cada tempo  $t$ , cada micropapel  $r \in RO^t$  em  $ORG_{\omega}^t$  e cada microligação  $l = (r_1, r_2, e) \in LI^t$  em  $ORG_{\omega}^t$  é propriamente implementada por  $(IMP_{\omega}^{Ras})^t$  sobre  $POP^t$ .*

**Exemplo 7. (Relação de Implementação)** *Neste exemplo é mostrada uma relação de implementação para o modelo PopOrg descrito nos exemplos anteriores. Utiliza-se aqui o conjunto de agentes presentes no Exemplo 5, o conjunto de papéis do Exemplo 6. A Figura 6 ilustra essa relação.*

*A função  $Ras$ , considerada nesta implementação, associa cada ação de papel a um conjunto de ações de agentes como segue:  $Ras(VooNupcial, Reprodução, Alimentação, Construção, Manutenção, CuiProle, ColAlimento, Defesa, Ataque) = \{Reproduzir, Fecundar, PorOvos, RecAlimento, ConsNinho, Limpeza, ManuNinho, CuiOvos, CuiLarvas, CuiPupas, CuiProle, ColAlimento, DistAlimento, CuiProle, CuiEntrada, Ataque\}$ . Por essa definição pode-se observar que a ação do papel  $VooNupcial$  possui as ações associadas  $Reproduzir$  e  $Fecundar$ .*

*A relação de implementação  $(IMP_{\omega}^{Ras}) = \{(Reprodutora, Formiga_1), (Reprodutora, Formiga_2), (Reprodutora, Formiga_4), (Operária, Formiga_1), (Operária, Formiga_2), (Operária, Formiga_3), (Operária, Formiga_4), (Soldado, Formiga_2), (Soldado, Formiga_3)\} \cup \{(l_1, T_1), (l_1, T_3), (l_2, T_1), (l_2, T_3), (l_2, T_4)\}$  é uma relação de implementação própria de  $ORG_{\omega}$  sobre  $POP$ , quando  $Ras$  é uma injeção unitária, ou seja,  $Ras(\alpha) = \{\alpha\}$ .*

*A Figura 6 mostra um esquema do modelo PopOrg com as estruturas populacional e micro-organizacional e todas as suas ligações. Este esquema mostra a relação de implementação do nível de micro-organizacional pelo nível populacional com relação a função  $Ras$ .*

No próximo Capítulo serão apresentadas as definições da gramática de grafos para sistemas multiagentes (GGMA).

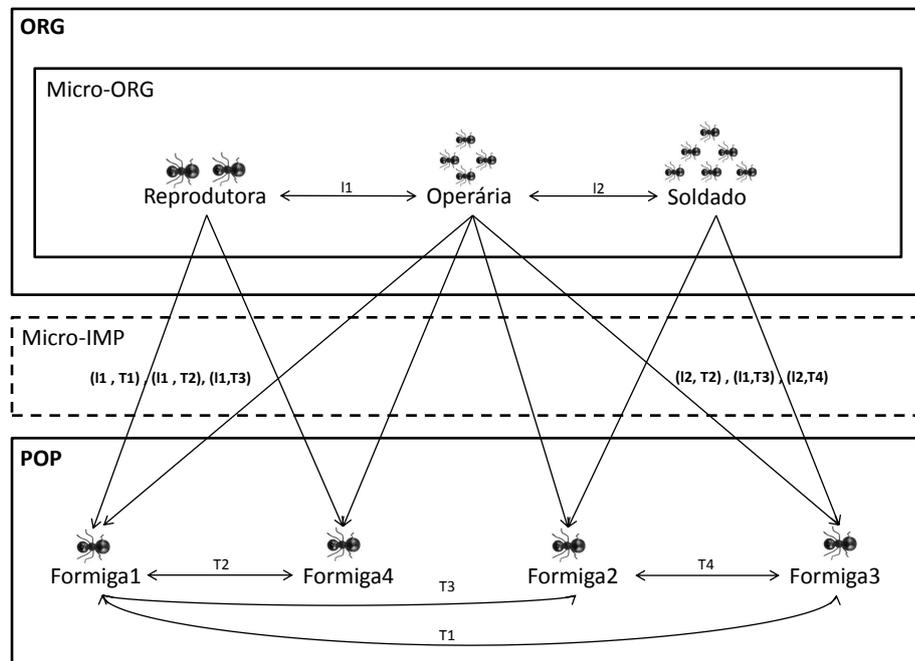


Figura 6: Relação de implementação entre os níveis micro-organizacional e populacional do modelo PopOrg

## 5 GRAMÁTICA DE GRAFOS PARA SISTEMAS MULTIAGENTES BASEADOS NO POPORG

Para descrever um sistema multiagente usando gramática de grafos, deve-se descrever os agentes e suas ações como grafos que obedecem ao grafo esquema mostrado na Figura 7. Agentes e papéis são modelados por vértices e ações são modeladas como arestas, onde o destino de cada aresta indica que o agente ou papel pode realizar aquela ação. Note que as arestas não têm origem, somente destino. Uma Gramática de Grafos para Sistemas Multiagentes (GGMA) é composta por duas gramáticas, a gramática de grafos para população e a gramática de grafos para organização (cada uma descrevendo um nível da organização) e uma relação de implementação que relaciona os dois níveis.

**Definição 19. (Gramática de Grafos para Sistemas Multiagentes)** *Uma Gramática de Grafos para Sistemas Multiagentes é dada por uma tupla  $GGMA = (PGG, OGG, IMP)$ , onde  $PGG$  é uma gramática de grafos populacional,  $OGG$  é uma gramática de grafos organizacional e  $IMP$  é a relação de implementação entre os níveis populacional e organizacional.*

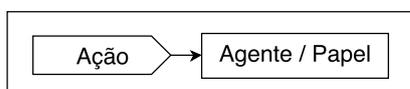


Figura 7: Grafo esquema para GGMA.

### 5.1 Gramática de Grafos para População

Nessa abordagem, o nível populacional de um sistema multiagente é modelado por um conjunto de gramática de grafos para agentes (AgGG). Cada AgGG é formada por uma GG e uma função comportamental. Nessa gramática são definidos: os tipos de agentes e todas as possíveis ações que eles podem realizar (grafo tipo); o estado inicial desses agentes (grafo inicial); e a execução de cada tipo de ação (regras).

A função comportamental define os comportamentos válidos de um agente, restringindo as derivações da gramática. Essas restrições são definidas por uma função que associa um conjunto de comportamentos a cada vértice do grafo inicial. Um comportamento define um conjunto de ações que podem ser realizadas a cada instante de tempo. A função comportamental é usada para definir um conjunto de expressões regulares que permitem controlar a aplicação das regras para cada agente, ou seja, somente derivações que respeitam a ordem das ações impostas por um comportamento do agente são derivações válidas. Todas as regras devem obedecer ao esquema mostrado na Figura 8, onde a aresta que representa a ação deve ser deletada e criada novamente e o vértice que representa o agente (ou papel) deve ser preservado.

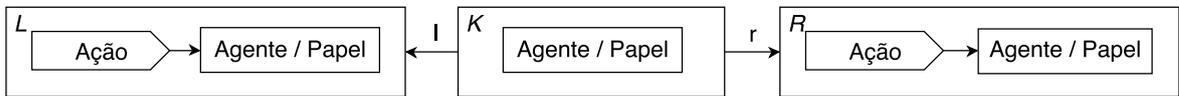


Figura 8: Esquema de regras para GGMA.

**Definição 20. (Comportamentos e Capacidade Comportamental)** Seja  $GG = (T, H, P, \pi)$  uma GG tipada. Um **comportamento** associado a uma GG é definido por uma função  $bh : I \rightarrow 2^P$ , onde  $I$  é um conjunto de instantes de tempo. O conjunto de todos os comportamentos associados a uma GG é denotado por  $Bh(GG)$ . A capacidade comportamental é dada por  $bc : V_H \rightarrow 2^{Bh(GG)}$  que é uma função total que associa um vértice a um conjunto de comportamentos da gramática.

**Definição 21. (Gramática de Grafos para Agentes)** Seja  $GG = (T, H, P, \pi)$  uma GG tipada. Uma gramática de grafos para agentes é dada por  $AgGG = (GG, bc)$ , onde:

1.  $V_T$  e  $V_H$  são conjuntos unitários;
2.  $\forall r \in P, E_{L_r}, V_{L_r}, E_{R_r}$  e  $V_{R_r}$  são conjuntos unitários,  $E_{K_r} = \emptyset$  e  $V_{K_r} = V_{L_r}$ ;
3.  $bc : V_H \rightarrow 2^{Bh(GG)}$  é uma função total, tal que:  $\forall v \in V_H, \forall c \in bc(v), \forall r \in rng(c), \exists e \in E_H, \exists el \in E_{L_r} : d^H(e) = v \wedge t^H(e) = t^{L_r}(el)$ .

As restrições impostas a uma  $AgGG$  garantem que (1) cada gramática descreve o comportamento de um único agente, (2) cada regra representa a execução de uma única ação, que é consumida e criada novamente. Além disso, a restrição (3) garante que existe um conjunto de comportamentos associados a cada agente e existe uma regra para cada ação, permitindo sua execução.

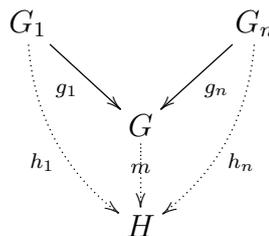
No modelo PopOrg, a interação entre pares de agentes é dada por um processo de troca. Um processo de troca define, para cada instante, as ações que são realizadas

por cada agente, de modo que os agentes desempenhem suas ações simultaneamente. Além disso, é definida a capacidade de troca dos agentes, associando um par de agentes a um conjunto de processos de troca. Na GG populacional, capacidade de troca que é dada por uma função que associa a cada par de vértices um conjunto de processos de troca.

**Definição 22. (Processos de Troca)** *Seja  $GG = (T, H, P, \pi)$  uma GG tipada. Um processo de troca associado a uma GG é definido por uma função  $ep : I \rightarrow 2^P \times 2^P$ , onde  $I$  é o conjunto de instantes. O conjunto de todos os processos de troca associados à gramática GG é denominado  $Ep(GG)$ .*

Para definir a união de GGs, é necessário usar uma operação que permita fazer a união de grafos. O coproduto é uma operação definida na teoria das categorias e generaliza o conceito de união disjunta da teoria de conjuntos. Intuitivamente, o resultado do coproduto de  $n$  grafos é um novo grafo onde os  $n$  grafos são colocados juntos com nenhum elemento em comum. Formalmente, esta operação é definida como segue.

**Definição 23. (Coproducto de Grafos)** *Dada a categoria dos grafos e morfismos de grafos e uma família indexada de grafos  $G_i$ , com  $i \in \{1, \dots, n\}$ . O coproduto desta família é um grafo  $G$  junto com uma coleção de morfismos de grafos  $g_i : G_i \rightarrow G$  que satisfazem a propriedade universal: para qualquer outro grafo  $H$  e qualquer coleção de morfismos de grafos  $h_i : G_i \rightarrow H$ , existe um único morfismo de grafos  $m : G \rightarrow H$  tal que  $m \circ g_i = h_i$ . Isto é, o diagrama abaixo comuta:*

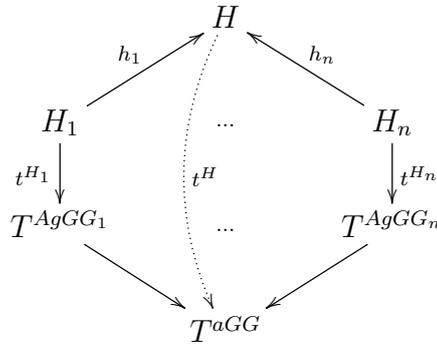


Uma gramática de grafos para população é dada pela união de um conjunto de gramáticas de grafos para agentes e uma função descrevendo a capacidade de troca dos agentes. A capacidade de troca de cada par de agentes é restrita por seus comportamentos.

**Definição 24. (Gramática de Grafos Populacional)** *Seja  $GG = (T, H, P, \pi)$  uma GG tipada. Uma Gramática de Grafos Populacional, com relação a um conjunto de gramática de grafos para agentes  $SA = \{AgGG_1, \dots, AgGG_n\}$ , é definida por  $PGG^{SA} = (aGG, ec)$ , onde:*

- $aGG = (T^{aGG}, H^{aGG}, P^{aGG}, \pi^{aGG})$  é uma gramática de grafos obtida como segue, para todo  $AgGG_i = ((T^{AgGG_i}, H^{AgGG_i}, P^{AgGG_i}, \pi^{AgGG_i}), bc_i) \in SA$ :

- $T^{aGG}$  é o coproduto dos grafos tipo  $T^{AgGG_i}$ ;
- $H^{aGG} = (H, t_H)$ , onde
  - \*  $H$  é o coproduto dos grafos  $H_i$ , onde  $H^{AgGG_i} = (H_i, t^{H_i})$
  - \*  $t_H$  é o morfismo universal do coproduto do diagrama abaixo:



- $P^{aGG}$  é a união disjunta de conjuntos  $P^{AgGG_i}$ ;
- $\pi^{aGG}$  é a união disjunta de funções  $r^{AgGG_i}$ .
- $ec : V_{H^{aGG}} \times V_{H^{aGG}} \rightarrow 2^{Ep(aGG)}$  é a função que associa um par de vértices em  $H^{aGG}$  a um conjunto de processos de troca, tal que:
 
$$\forall v_1, v_2 \in V_{H^{aGG}}, \forall ep \in ec(v_1, v_2), \forall t \in I:$$

$$Prj_1(ep(t)) \subseteq \bigcup \{bh(t) | bh \in bc_k(v_1)\},$$

$$Prj_2(ep(t)) \subseteq \bigcup \{bh(t) | bh \in bc_l(v_2)\}$$

onde  $v_1 \in V_{H^{AgGG_k}}, v_2 \in V_{H^{AgGG_l}}$  e  $Prj_1$  e  $Prj_2$  são funções de projeção.

Quando não é necessário referenciar explicitamente todas as gramáticas de grafos de agentes que compõem uma gramática de grafos populacional  $PGG^{SA}$ , pode-se omitir  $SA$  da notação, utilizando somente  $PGG$ .

Um exemplo será utilizado para ilustrar o nível populacional de uma gramática de grafos para sistemas multiagentes. Nesse exemplo, será utilizado o mesmo problema especificado no Capítulo poporg, utilizando  $GG$ .

**Exemplo 8. (Gramática de Grafos Populacional)** Na Figura 9, estão ilustrados os grafos tipo e iniciais das  $AgGGs$  dos agentes  $Formiga_1$  e  $Formiga_2$ . As regras para estes agentes são mostradas nas Figuras 10 e 11 respectivamente.

Para o agente  $Formiga_3$  e  $Formiga_4$  os grafos inicial e tipo são vistos na Figura 12. As regras para estes agentes são mostradas nas Figuras 13 e 14 respectivamente.

Essas gramáticas definem que o agente  $Formiga_1$  pode realizar as ações Reproduzir, Fecundar, PorOvos e RecAlimento; o agente  $Formiga_2$ , por sua vez, pode realizar as ações ConsNinho, CuiOvos, CuiLarvas, CuiPupas, ColAlimento,

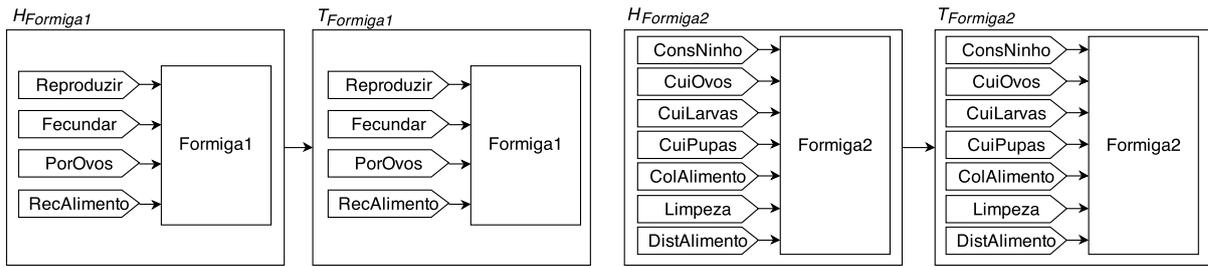


Figura 9: AgGG para os agentes  $Formiga_1$  e  $Formiga_2$

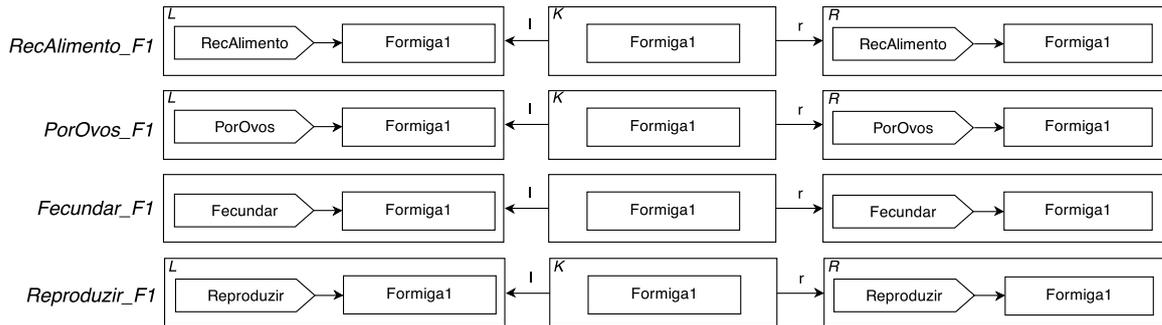


Figura 10: Conjunto de regras para o agente  $Formiga_1$

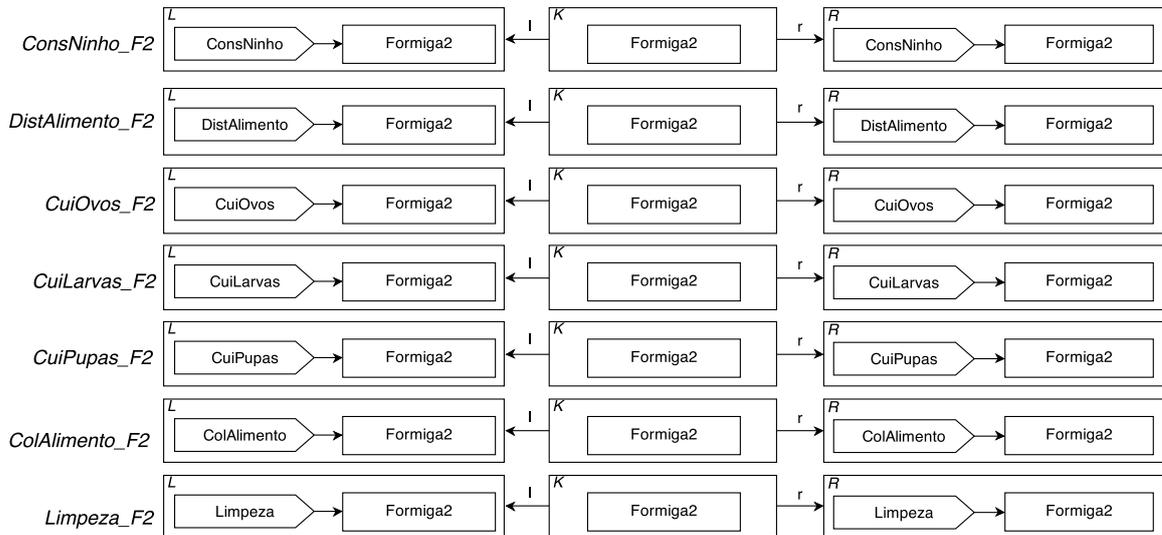


Figura 11: Conjunto de regras para o agente  $Formiga_2$

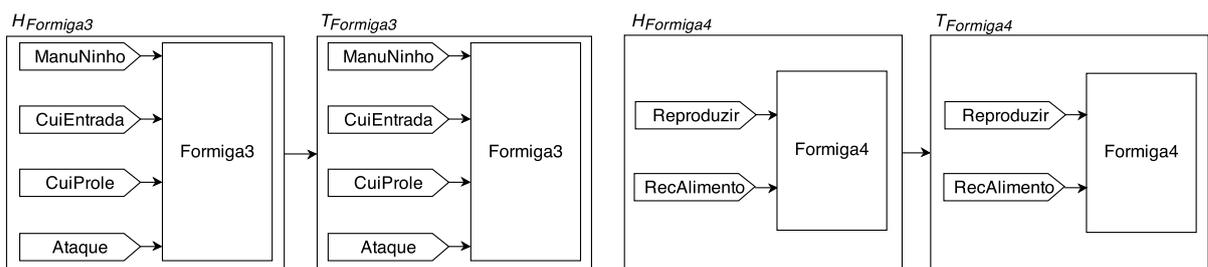


Figura 12: AgGG para os agentes  $Formiga_3$  e  $Formiga_4$

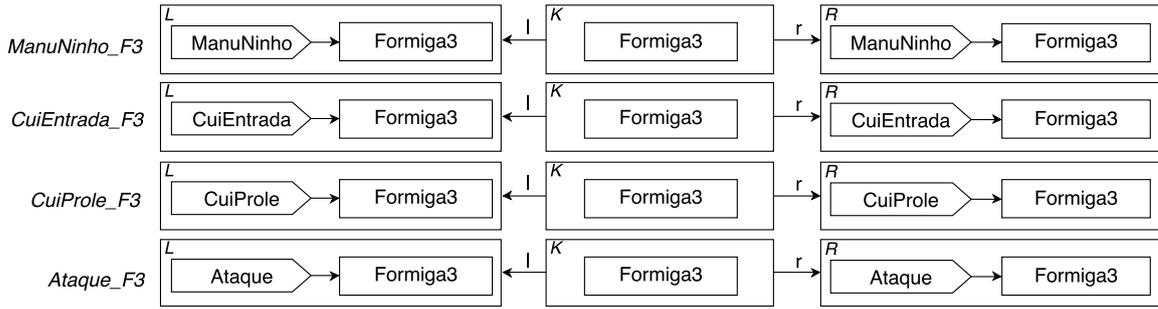


Figura 13: Conjunto de regras para o agente  $Formiga_3$

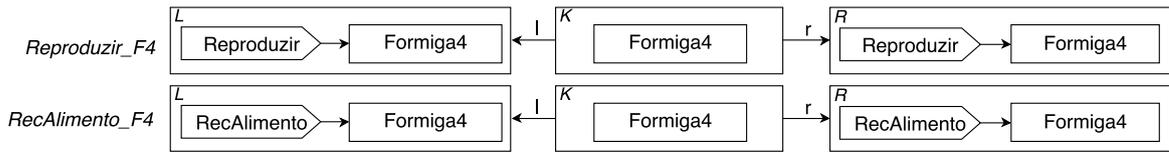


Figura 14: Conjunto de regras para o agente  $Formiga_4$

*DistAlimento e Limpeza; já o agente  $Formiga_3$  pode realizar as ações ManuNinho, CuiEntrada, CuiProle e Ataque; e por fim, o agente  $Formiga_4$  pode realizar as ações Reproduzir e RecAlimento. Note que as regras obedecem ao esquema de regras (mostrado na Figura 8) e descrevem a execução de cada ação dos agentes.*

*As capacidades comportamentais dos agentes são definidas por*

$$bc_{F_1} = \{C_1\}$$

$$bc_{F_2} = \{C_2, C_3\}$$

$$bc_{F_3} = \{C_4, C_5\}$$

$$bc_{F_4} = \{C_6\}$$

*Os comportamentos que podem ocorrer no exemplo são definidos por*

$$Bh^{AgGG} = \{C_1, C_2, C_3, C_4, C_5, C_6\}, \text{ onde:}$$

$$C_1 = \{0 \mapsto \{Fecundar_{F_1}, Reproduzir_{F_1}\}, 1 \mapsto \{PorOvos_{F_1}\}, 2 \mapsto \{RecAlimento_{F_1}\}\};$$

$$C_2 = \{0 \mapsto \{CuiOvos_{F_2}\}, 1 \mapsto \{CuiLarvas_{F_2}\}, 2 \mapsto \{CuiPupas_{F_2}\}\};$$

$$C_3 = \{0 \mapsto \{ConsNinho_{F_2}\}, 1 \mapsto \{ColAlimento_{F_2}\}, 2 \mapsto \{DistAlimento_{F_2}\}\};$$

$$C_4 = \{0 \mapsto \{CuiEntrada_{F_3}\}, 1 \mapsto \{CuiProle_{F_3}\}, 2 \mapsto \{ManuNinho_{F_3}\}\};$$

$$C_5 = \{0 \mapsto \{Ataque_{F_3}\}, 1 \mapsto \{CuiEntrada_{F_3}\}, 2 \mapsto \{CuiProle_{F_3}, ManuNinho_{F_3}\}\};$$

$$C_6 = \{0 \mapsto \{Reproduzir_{F_4}\}, 1 \mapsto \{RecAlimento_{F_4}\}, 2 \mapsto \{\}\}.$$

*Por estas definições, pode-se observar, por exemplo, que o agente  $F_1$  pode realizar as ações Reproduzir e/ou Fecundar (regras Reproduzir $_{F_1}$  e Fecundar $_{F_1}$ ), depois o agente pode realizar a ação PorOvos (regra PorOvos $_{F_1}$ ), e finalmente no último instante o agente realiza a ação RecAlimento (regra RecAlimento $_{F_1}$ ).*

*No exemplo, tem-se o seguinte conjunto de possíveis interações  $Ep^{aGG} = \{T_1, T_2, T_3, T_4\}$ , onde:*

$$\begin{aligned}
T_1 &= \{0 \mapsto (\{Fecundar_{F_1}, Reproduzir_{F_1}\}, \{CuiOvos_{F_2}\}), \\
&\quad 1 \mapsto (\{PorOvos_{F_1}\}, \{CuiLarvas_{F_2}\}), \\
&\quad 2 \mapsto (\{RecAlimento_{F_1}\}, \{CuiPupas_{F_2}\})\}; \\
T_2 &= \{0 \mapsto (\{Fecundar_{F_1}, Reproduzir_{F_1}\}, \{ConsNinho_{F_2}\}), \\
&\quad 1 \mapsto (\{PorOvos_{F_1}\}, \{ColAlimento_{F_2}\}), \\
&\quad 2 \mapsto (\{RecAlimento_{F_1}\}, \{DistAlimento_{F_2}\})\}; \\
T_3 &= \{0 \mapsto (\{Fecundar_{F_1}, Reproduzir_{F_1}\}, \{Ataque_{F_3}\}), \\
&\quad 1 \mapsto (\{PorOvos_{F_1}\}, \{CuiEntrada_{F_3}\}), \\
&\quad 2 \mapsto (\{RecAlimento_{F_1}\}, \{CuiProle_{F_3}, ManuNinho_{F_3}\})\}; \\
T_4 &= \{0 \mapsto (\{ConsNinho_{F_2}\}, \{ManuNinho_{F_3}\}), \\
&\quad 1 \mapsto (\{ColAlimento_{F_2}\}, \{CuiEntrada_{F_3}\}), \\
&\quad 2 \mapsto (\{DistAlimento_{F_2}\}, \{ManuNinho_{F_3}\})\}.
\end{aligned}$$

O grafo inicial, o grafo tipo e as regras da gramática de grafos populacional deste sistema são obtidas pela união disjunta de cada componente das AgGGs da *Formiga<sub>1</sub>*, *Formiga<sub>2</sub>*, *Formiga<sub>3</sub>* e *Formiga<sub>4</sub>* apresentados na Figura 20. A capacidade de troca é dada por:

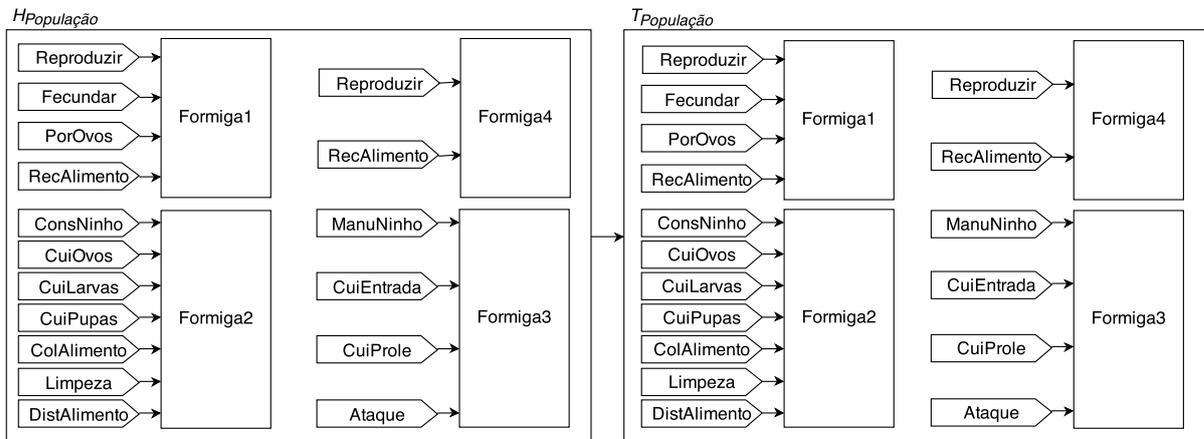


Figura 15: Grafos inicial e tipo da gramática de grafos populacional PGG

$$\begin{aligned}
Ec &= \{(Formiga_1, Formiga_2) \mapsto \{T_1\}, (Formiga_1, Formiga_4) \mapsto \{T_2\}, \\
&\quad (Formiga_1, Formiga_2) \mapsto \{T_3\}, (Formiga_2, Formiga_3) \mapsto \{T_4\}\}.
\end{aligned}$$

Por essa função, os agentes *Formiga<sub>1</sub>* e *Formiga<sub>2</sub>* podem interagir como segue: primeiramente, *Formiga<sub>1</sub>* pode realizar as ações *Reproduzir* e/ou *Fecundar* (regras  $Reproduzir_{F_1}$  e  $Fecundar_{F_1}$ ) enquanto o agente *Formiga<sub>2</sub>* pode realizar as ações *CuiOvos* (regras  $CuiOvos_{F_2}$ ); depois, *Formiga<sub>1</sub>* pode realizar a ação *PorOvos* (regra  $PorOvos_{F_1}$ ) enquanto *Formiga<sub>2</sub>* pode realizar as ações *CuiLarvas* (regras  $CuiLarvas_{F_2}$ ); e finalmente, *Formiga<sub>1</sub>* pode realizar a ação *RecAlimento* (regra  $RecAlimento_{F_1}$ ) enquanto *Formiga<sub>2</sub>* pode realizar as ações *CuiPupas* (regras  $CuiPupas_{F_2}$ ).

## 5.2 Gramática de Grafos para Organização

A modelagem do nível organizacional do modelo PopOrg é apresentada nessa Seção. Assim, como as PGGs são compostas por uma coleção de AgGGs, uma gramática de grafos para organização é composta de uma coleção de Gramáticas de Grafos para Papéis (RGG). Cada  $RGG$  é formada por uma  $GG$  e uma função comportamental  $bc$ . Essa definição é análoga à definição da gramática de grafos para agentes. Nessa gramática são definidos: o tipo do papel e todas as possíveis ações que ele pode realizar (grafo tipo); o estado inicial deste papel (grafo inicial); e a execução de cada tipo de ação (regras). No nível organizacional, existe uma função de ligação, que define as interações válidas entre os papéis, restringindo as derivações da gramática. Uma ligação define um conjunto de comportamentos que podem ser realizados por um par de papéis a cada instante de tempo. As ligações do sistema são definidas por uma função que associa um conjunto de comportamentos a cada par de vértices do grafo inicial. A função de ligação é usada para controlar a aplicação das regras de cada papel, ou seja, somente derivações que respeitam a ordem em que as ações podem ocorrer de acordo com uma ligação são derivações válidas. Todas as regras de uma RGG também devem obedecer ao esquema mostrado na Figura 8. Além disso, as restrições impostas a uma  $RGG$  são as mesmas impostas a uma  $AgGG$  e, por isso, a definição a seguir é análoga à Definição 21.

**Definição 25. (Gramática de Grafos para Papéis)** *Seja  $GG = (T, H, P, \pi)$  uma  $GG$  tipada. Uma gramática de grafos para papéis é dada por  $RGG = (GG, bc)$ , onde:*

1.  $V_T$  e  $V_H$  são conjuntos unitários;
2.  $\forall r \in P, E_{L_r}, V_{L_r}, E_{R_r}$  e  $V_{R_r}$  são conjuntos unitários,  $E_{K_r} = \emptyset$  e  $V_{K_r} = V_{L_r}$ ;
3.  $bc : V_H \rightarrow 2^{Bh(GG)}$  é uma função total, tal que:

$$\forall v \in V_H, \forall c \in bc(v), \forall r \in rng(c), \exists e \in E_H, \exists el \in E_{L_r} : d^H(e) = v \wedge t^H(e) = t^{L_r}(el).$$

No nível organizacional do modelo PopOrg, os papéis são capazes de realizar interações com outros papéis. Para isso é necessário definir as ligações e a capacidade de ligação. Uma ligação entre papéis é a interação que pode ocorrer no nível organizacional, de modo que dois papéis podem realizar suas ações em conjunto, desde que respeitem sua capacidade de ligação. A ligação é uma função que associa a um instante de tempo um par de conjunto de regras de papéis. A capacidade de ligação é uma função que associa um par de vértices do grafo inicial (papéis) a um conjunto de ligações possíveis entre esses papéis.

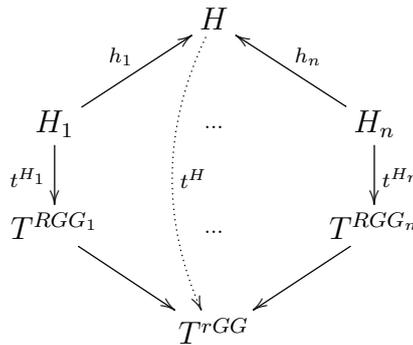
**Definição 26. (Ligações Organizacionais)** *Seja  $GG = (T, H, P, \pi)$  uma gramática de grafos tipada. Uma ligação é definida por uma função  $li : I \rightarrow 2^P \times 2^P$  que associa*

um par de conjuntos de ações dos papéis a cada instante de tempo  $I$ . O conjunto de todas as ligações associados a uma gramática  $GG$  é denotado por  $Li(GG)$ .

A definição abaixo é análoga à Definição 24.

**Definição 27. (Gramática de Grafos Organizacional)** Uma gramática de grafos organizacional, com relação a um conjunto de gramática de grafos para papéis  $SR = \{RGG_1, \dots, RGG_n\}$ , é definida por  $OGG^{SR} = (rGG, lc)$ , onde:

- $rGG = (TrGG, HrGG, PrGG, \pi^{rGG})$  é uma gramática de grafos obtida como segue, para todo  $RGG_i = ((T^{RGG_i}, H^{RGG_i}, P^{RGG_i}, \pi^{RGG_i}), bc_i) \in SR$ :
  - $TrGG$  é o coproduto de grafos tipos  $T^{RGG_i}$ ;
  - $HrGG = (H, t_H)$ , onde
    - \*  $H$  é o coproduto dos grafos  $H_i$ , onde  $H^{RGG_i} = (H_i, t^{H_i})$
    - \*  $t_H$  é o morfismo universal do coproduto do diagrama abaixo:



- $PrGG$  é a união disjunta de conjuntos  $P^{RGG_i}$ ;
- $\pi^{rGG}$  é a união disjunta de funções  $r^{RGG_i}$ .
- e  $lc : V_{HrGG} \times V_{HrGG} \rightarrow 2^{Li(rGG)}$  é a função que associa um par de vértices em  $HrGG$  a um conjunto de ligações, tal que:
 
$$\forall v_1, v_2 \in V_{HrGG}, \forall li \in lc(v_1, v_2), \forall t \in I:$$

$$Prj_1(li(t)) \subseteq \bigcup \{bh(t) | bh \in bc_k(v_1)\},$$

$$Prj_2(li(t)) \subseteq \bigcup \{bh(t) | bh \in bc_l(v_2)\},$$

onde  $v_1 \in V_{HrGG_k}, v_2 \in V_{HrGG_l}, Prj_1$  e  $Prj_2$  são funções de projeção.

Quando não é necessário referenciar explicitamente todas as gramáticas de grafos de agentes que compõem uma gramática de grafos populacional  $OGG^{SR}$ , pode-se omitir  $SR$  da notação, utilizando somente  $OGG$ .

Para ilustrar o nível organizacional de uma GGMA, será apresentado um exemplo. No exemplo, é possível identificar três tipos de papéis: *Reprodutora*, *Operária* e *Soldado*.

**Exemplo 9. (Gramática de Grafos Organizacional)** Na Figura 16, estão ilustrados os grafos tipo e iniciais das RGGs dos papéis *Reprodutora*, *Operária* e *Soldado*. As regras para estes papéis são mostradas nas Figuras 17, 18 e 19 respectivamente.

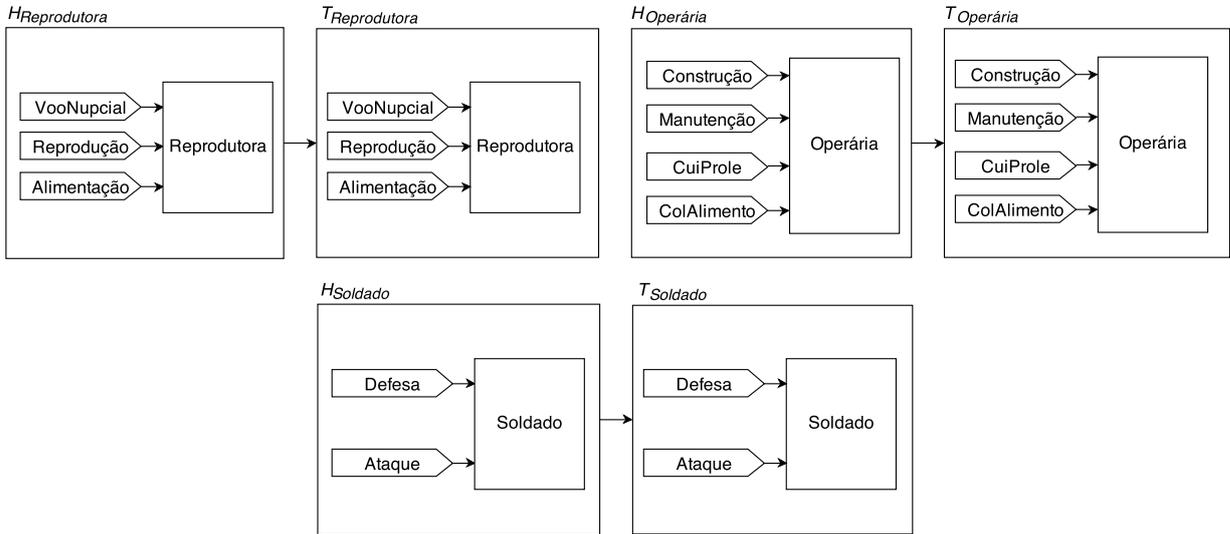


Figura 16: RGG para os papéis *Reprodutora*, *Operária* e *Soldado*

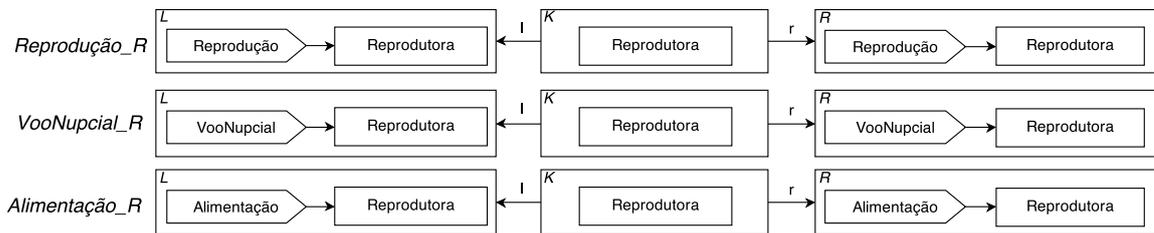


Figura 17: Conjunto de regras para o papel *Reprodutora*

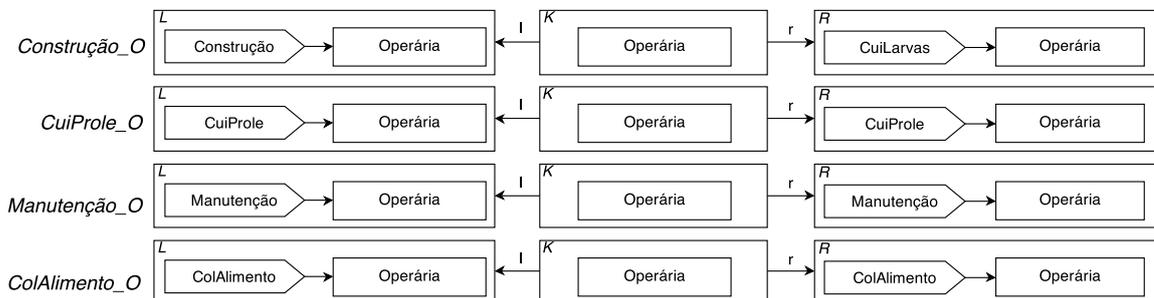


Figura 18: Conjunto de regras para o papel *Operária*

Essas gramáticas mostram, por exemplo, que o papel *Reprodutora* pode realizar as ações *VooNupcial*, *Ovular* *Alimentar*; o papel *Operária* por sua vez, pode realizar as ações *Construção*, *CuiProle*, *ColAlimento* e *Manutenção*; para o papel *Soldado* as ações que podem ser realizadas são *Ataque* e *Defesa*. As regras obedecem ao

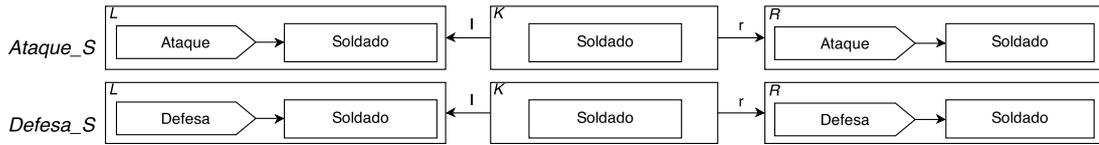


Figura 19: Conjunto de regras para o papel *Soldado*

esquema de regras (mostrado na Figura 8) e descrevem a execução de cada tipo de ação dos papéis.

As capacidades comportamentais dos papéis são definidas por

$$bc_R = \{C_1\}$$

$$bc_O = \{C_2\}$$

$$bc_S = \{C_3\}$$

Os comportamentos que podem ocorrer no exemplo são definidos por  $Bh^{RGG} = \{C_1, C_2, C_3\}$ , onde:

Os comportamentos dos papéis dão dados por

$$C_1 = \{0 \mapsto \{VooNupcial_R\}, 1 \mapsto \{Reprodução_R\}, 2 \mapsto \{Alimentação_R\}\}$$

$$C_2 = \{0 \mapsto \{Construção_O\}, 1 \mapsto \{CuiProle_O, ColAlimento_O\}, \\ 2 \mapsto \{Manutenção_O\}\}$$

$$C_3 = \{0 \mapsto \{Defesa_S\}, 1 \mapsto \{Ataque_S\}, 2 \mapsto \{\}\}$$

Cada papel possui diferentes comportamentos. Por exemplo, o papel *Reprodutora* pode realizar as seguintes ações em um instante de tempo: primeiramente pode realizar a ação *VooNupcial* (regra  $VooNupcial_R$ ), depois o papel pode realizar a ação *Reprodução* (regra  $Reprodução_R$ ) e por fim o papel pode realizar a ação *Alimentação* (regra  $Alimentação_R$ ).

As ligações entre os três papéis do nível micro são dadas por

O grafo inicial, o grafo tipo e as regras da gramática de grafos organizacional deste sistema são obtidas pela união disjunta de cada componente das RGGs da *Reprodutora*, *Operária* e *Soldado* apresentados na Figura ???. A capacidade de troca é dada por:

$Li^{RGG} = \{l_1, l_2\}$ , onde a ligação entre os papéis *Reprodutora* e *Operária*  $l_1$  é dada pelo processo de troca  $T_5$  e a ligação entre os papéis *Soldado* e *Operária*  $l_2$ , já a ligação entre os papéis *Operária* e *soldado*, é dada pelo processo de troca  $T_6$ :

$l_1 = (Reprodutora, Operária, T_5)$ , onde:

$$T_5 = \{0 \mapsto \{(\{VooNupcial_R\}, \{Construção_O\})\}, \\ 1 \mapsto \{(\{Reprodução_R\}, \{CuiProle_O, ColAlimento_O\})\}, \\ 2 \mapsto \{(\{Alimentação_R\}, \{Manutenção_O\})\}\}.$$

$l_2 = (Soldado, Operária, T_6)$ , onde:

$$T_6 = \{0 \mapsto \{(\{CuiProle_O\}, \{Defesa_S\})\}, \\ 1 \mapsto \{(\{ColAlimento_O\}, \{Defesa_S\})\}, 2 \mapsto \{(\{Manutenção_O\}, \{\})\}\}.$$

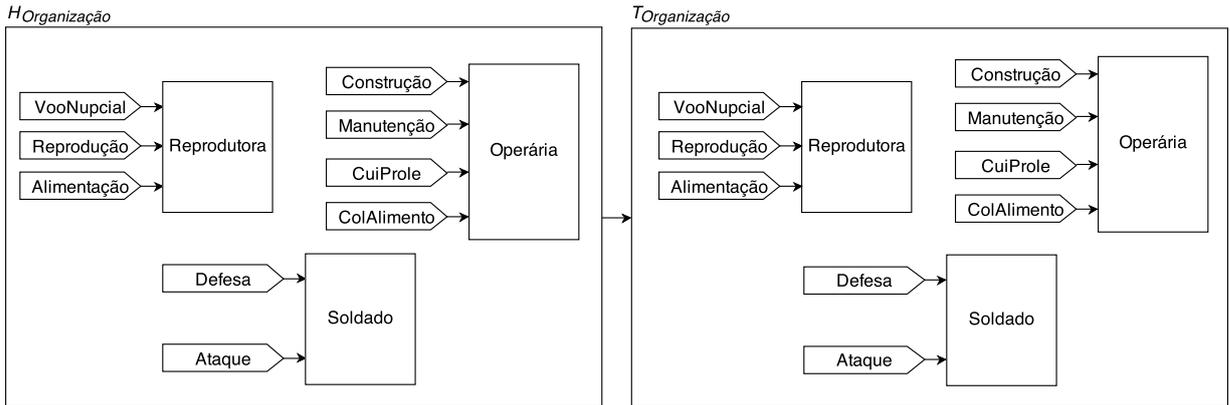


Figura 20: Grafos inicial e tipo da gramática de grafos organizacional  $OGG$

A capacidade de ligação da organização é dada por:  
 $lc = \{(Reprodutora, Operária) \mapsto \{l_1\}, (Operária, Soldado) \mapsto \{l_2\}\}.$

Pela capacidade de ligação, a microligação  $l_1$  é realizada entre os micropapéis *Reprodutora* e *Operária*. Já os micropapéis *Operária* e *Soldado* realizam a microligação  $l_2$ . As trocas associadas a cada microligação são restritas pelo comportamento dos papéis envolvidos, ou seja, só pode ocorrer uma troca entre papéis, se ambos tiverem um comportamento associado e, ainda, esse comportamento deve ser respeitado pelo processo de troca.

### 5.3 Relação de Implementação

Em muitos SMAs a fronteira entre os níveis organizacionais não é bem clara. Ainda assim, os agentes (populacional) têm a capacidade de realizar tarefas que são do nível organizacional. No modelo PopOrg, a possibilidade de que as ações dos papéis (organizacional) sejam realizadas pelos agentes se dá por meio de relações de implementação. Uma relação de implementação descreve como uma ação de um papel pode ser executada por um conjunto de ações dos agentes. Assim uma ação da organização do SMA é realizada pelo nível mais baixo do modelo, o populacional.

Dada uma GG populacional e uma GG organizacional, uma relação de implementação é dada por duas funções, que fazem a relação entre os papéis e os agentes e as ligações com os processos de troca. A função de estruturação das ações de papel é que define a relação entre uma ação de um papel sobre um conjunto de ações de agentes.

**Definição 28. (Função de Estruturação das Ações de Papel)** *Sejam  $OGG^{SR} = ((T_1, G_1, P_1, \pi_1), lc)$  e  $PGG^{SA} = ((T_2, G_2, P_2, \pi_2), ec)$  gramáticas organizacional e populacional, respectivamente. A função de estruturação da ação do papel  $Ras : E_{G_1} \rightarrow 2^{E_{G_2}}$  associa cada ação de um papel em  $OGG^{SR}$  a um conjunto de ações de agentes em  $PGG^{SA}$ . A função  $\overline{Ras} : 2^{E_{G_1}} \rightarrow 2^{E_{G_2}}$ , que*

estende a função  $Ras$  para um conjunto de ações de papéis, é definida por  $\overline{Ras}(Act) = Ras(\alpha_1) \cup \dots \cup Ras(\alpha_n)$ , onde  $Act \subseteq E_{G_1}$  e  $\alpha_i \in Act$ .

A relação de implementação de uma gramática de grafos organizacional sobre a gramática de grafos populacional é realizada por meio de mapeamentos. O primeiro associa um papel a um conjunto de agentes, e o outro associa uma ligação a um conjunto de processos de troca. Este mapeamento acontece com relação a  $Ras$ , isto é, um agente pode estar relacionado com um papel se ele puder realizar a ação necessária para implementar uma ação daquele papel.

**Definição 29. (Relação de Implementação)** Uma relação de implementação de  $OGG^{SR} = ((T_1, G_1, P_1, \pi_1), lc)$  sobre  $PGG^{SA} = ((T_2, G_2, P_2, \pi_2), ec)$  com relação a  $Ras$  é definida por  $IMP^{Ras} = (ra, le)$ , onde  $ra : V_{G_1} \rightarrow 2^{V_{G_2}}$  é uma função que associa um papel a um conjunto de agentes e  $le : Li(G_1) \rightarrow 2^{Ep(G_2)}$  é uma função que associa uma ligação a um conjunto de processos de troca de um agente, tal que:

$$1. \forall vr \in V_{G_1}, \forall i \in I : \bigcup \{ \overline{Ras}(bh(i)) \mid bh \in bc_r(vr) \} \subseteq \bigcup \{ bh(i) \mid bh \in bc_a(va), va \in A \},$$

$$\text{onde } A = ra(vr), vr \in V_{G_r}, ((T_r, G_r, P_r, \pi_r), bc_r) \in SR,$$

$$va \in V_{G_a}, ((T_a, G_a, P_a, \pi_a), bc_a) \in SA$$

$$2. \forall li \in dom(le), \forall i \in I : \bigcup \overline{Ras}(Prj_k(li(i))) \subseteq \bigcup \{ Prj_k(ep(i)) \mid ep \in E \},$$

$$\text{onde } li \in lc(r_1, r_2),$$

$$E = \{ ep \in ec(a_1, a_2) \mid a_1 \in ra(r_1), a_2 \in ra(r_2), ep \in le(li) \},$$

$$K \in \{1, 2\}$$

Pela restrição (1) é garantido que todas as ações requeridas por cada papel em cada instante de tempo estão nos comportamentos dos agentes associados àquele papel, isto é, considerando o conjunto de agentes associados a um papel, cada agente pode executar algumas dessas ações requeridas pelo papel, mas de forma conjunta, todos os agentes devem realizar no mínimo todas as ações requeridas pelo papel. As ações requeridas por um papel são determinadas pela função  $Ras$ ; de forma análoga a restrição (2) garante que os participantes de todos os processos de troca associados a um link apresentam, de forma conjunta, todos os comportamentos requeridos para cada participante do link.

**Exemplo 10. (Relação de Implementação)** Considerando as gramáticas apresentadas nos Exemplos 8 e 9, podemos definir uma relação de implementação da gramática  $OGG^{SR}$  sobre a gramática  $PPG^{SA}$  com relação à função  $Ras$  como segue:

$$\begin{aligned}
Ras = \{ & CuiProle_O \mapsto \{CuiOvos_{F_2}, CuiLarvas_{F_2}, CuiPupas_{F_2}, CuiProle_{F_3}\}, \\
& Constru\c{c}ao_O \mapsto \{ConsNinho_{F_2}\}, \\
& Manuten\c{c}ao_O \mapsto \{ConsNinho_{F_2}, Limpeza_{F_2}, ManuNinho_{F_3}\}, \\
& ColAlimento_O \mapsto \{ColAlimento_{F_2}, DistAlimento_{F_2}, RecAlimento_{F_4}\}, \\
& Defesa_S \mapsto \{CuiEntrada_{F_3}, CuiProle_{F_3}, ManuNinho_{F_3}\}, \\
& Ataque_S \mapsto \{RecAlimento_{F_4}, ColAlimento_{F_2}\}, \\
& Reprodu\c{c}a_R \mapsto \{Fecundar_{F_1}, PoOvos_{F_1}\}, \\
& VooNupcial_R \mapsto \{Reproduzir_{F_1}, Reproduzir_{F_4}, Fecundar_{F_1}\}, \\
& Alimenta\c{c}a_R \mapsto \{RecAlimento_{F_1}\}\}.
\end{aligned}$$

$IMP^{Ras} = (ra, le)$ , onde

$$\begin{aligned}
ra = \{ & Oper\acute{a}ria \mapsto \{Formiga_2, Formiga_3\}, \\
& Reprodutora \mapsto \{Formiga_1, Formiga_2, Formiga_4\}, \\
& Soldado \mapsto \{Formiga_2, Formiga_3\}\}.
\end{aligned}$$

$$le = \{l_1 \mapsto \{T_1, T_2, T_3, T_4\}, l_2 \mapsto \{T_3, T_4\}\}.$$

*Pela função ra o papel Operária é relacionado com os agentes Formiga<sub>2</sub> e Formiga<sub>3</sub>. Pela função le a ligação l<sub>1</sub> é relacionada com os processos de troca T<sub>1</sub> (Formiga<sub>1</sub> e Formiga<sub>2</sub>), T<sub>2</sub> (Formiga<sub>1</sub> e Formiga<sub>4</sub>), T<sub>3</sub> (Formiga<sub>1</sub> e Formiga<sub>3</sub>) e T<sub>4</sub> (Formiga<sub>2</sub> e Formiga<sub>3</sub>). Pela função Ras as ações exigidas pela ação do papel CuiProle são Ras(r2<sub>O</sub>) = {CuiOvos<sub>F<sub>2</sub></sub>, CuiLarvas<sub>F<sub>2</sub></sub>, CuiPupas<sub>F<sub>2</sub></sub>, CuiProle<sub>F<sub>3</sub></sub>}.*

## 6 SEMÂNTICA DA GRAMÁTICA DE GRAFOS PARA SISTEMAS MULTIAGENTES BASEADO NO POPORG

Neste Capítulo define-se o comportamento da gramática de grafos para sistemas multiagentes. O comportamento é dado por um conjunto de derivações que respeitam a ordem e as interações entre agentes ou papéis.

A fim de descrever a interação entre dois agentes ou papéis, precisa-se de um mecanismo que permita aplicar mais de uma regra ao mesmo tempo. Portanto, a seguir é definida uma composição de regras, permitindo a aplicação de diferentes regras em uma única derivação direta. Essa composição é definida como uma união de regras, identificando os elementos do mesmo tipo e, neste caso, a regra composta contém todas as ações que os agentes ou papéis podem executar em paralelo.

**Definição 30. (Regras de Ações Paralelas)** Dadas duas regras  $p_1 = L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1$  e  $p_2 = L_2 \xleftarrow{l_2} K_2 \xrightarrow{r_2} R_2$ , tipadas sobre  $T$ , a regra de ações paralelas associada a  $p_1$  e  $p_2$  é definida por  $p_1 \oplus p_2 : L_{12} \xleftarrow{l_{12}} K_{12} \xrightarrow{r_{12}} R_{12}$ , conforme diagrama da Figura 21, onde:

- $L_{12}$  é obtido pela construção do pullback (2), seguido do pushout (3) e  $t_{L_{12}}$  é o morfismo universal dado pela propriedade universal do pushout (3);
- $K_{12}$  é obtido pela construção do pullback (1), seguido do pushout (4) e  $t_{K_{12}}$  é o morfismo universal dado pela propriedade universal do pushout (4);
- $R_{12}$  é obtido pela construção do pullback (7), seguido do pushout (8) e  $t_{R_{12}}$  é o morfismo universal dado pela propriedade universal do pushout (8);
- $l_{12}$  é o morfismo universal do pushout (4), uma vez que:
  1.  $t_{K_1} \circ c = t_{K_2} \circ d$  - pelo pullback (1);
  2.  $t_{L_1} \circ l_1 \circ c = t_{L_2} \circ l_2 \circ d$  - pela comutatividade de (5) e (6) (definição dos morfismos de grafos tipados  $l_1$  e  $l_2$ );
  3.  $t_{L_{12}} \circ e \circ l_1 \circ c = t_{L_{12}} \circ f \circ l_2 \circ d$  - pela propriedade universal do pushout (3);
  4.  $e \circ l_1 \circ c = f \circ l_2 \circ d$  - por serem morfismos totais e injetores.

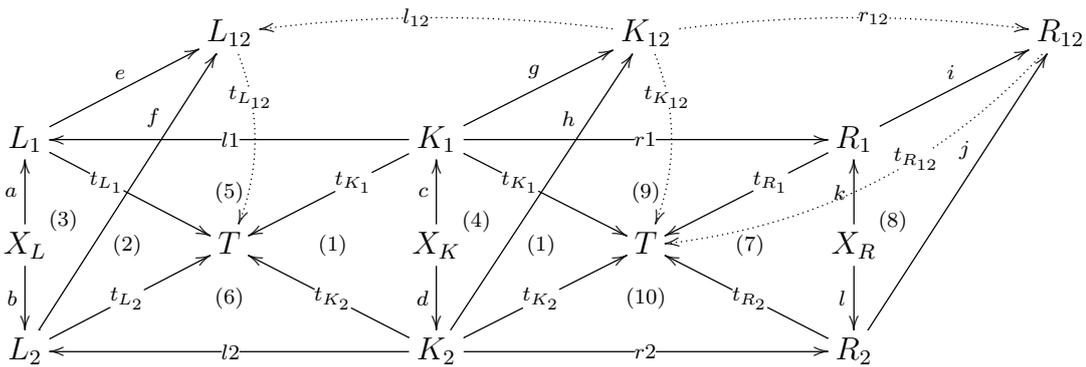


Figura 21: Diagrama de composição de regras

- $r_{12}$  é o morfismo universal do pushout (9), uma vez que:
  1.  $t_{K_1} \circ c = t_{K_2} \circ d$  - pela comutatividade do pullback (1);
  2.  $t_{R_1} \circ r_1 \circ c = t_{R_2} \circ r_2 \circ d$  - pela comutatividade de (9) e (10) (definição dos morfismos de grafos tipados  $r_1$  e  $r_2$ );
  3.  $t_{R_{12}} \circ i \circ r_1 \circ c = t_{R_{12}} \circ j \circ r_2 \circ d$  - pela propriedade universal do pushout (5);
  4.  $j \circ r_2 \circ d = i \circ r_1 \circ c$  - por serem morfismos totais e injetores.

O diagrama da Figura 21 mostra como é feita a composição de regras.

A composição de regras definida por uma regra de ações paralelas é comutativa, associativa e tem um elemento identidade ( $\varepsilon$ ), isto é:

- comutatividade:  $p_1 \oplus p_2 = p_2 \oplus p_1$ ;
- associatividade:  $(p_1 \oplus p_2) \oplus p_3 = p_1 \oplus (p_2 \oplus p_3)$ ; e,
- identidade:  $p_1 \oplus \varepsilon = \varepsilon \oplus p_1 = p_1$

onde  $\varepsilon$  é a regra vazia. Os grafos  $L, K, R$  de uma regra vazia têm os conjuntos de vértices e arestas vazios.

**Intuitivamente**  $\oplus$  define a **união** das regras. A Figura 22 mostra a composição de duas regras da gramática de grafos para agentes. É possível intuir que a união de duas regras é a criação de uma nova regra que contém as duas ações do agente na nova regra composta.

O comportamento de uma GGMA é dado por um conjunto de derivações de uma GGC que respeita um determinado conjunto de expressões regulares. O conjunto de regras de uma GGC associada a uma GGMA é constituído por todas as regras de ações paralelas permitidas pelos processos de troca. Por exemplo, se existe um processo de troca entre dois agentes, onde o primeiro pode realizar as ações  $p_1$  e  $p_2$ ,

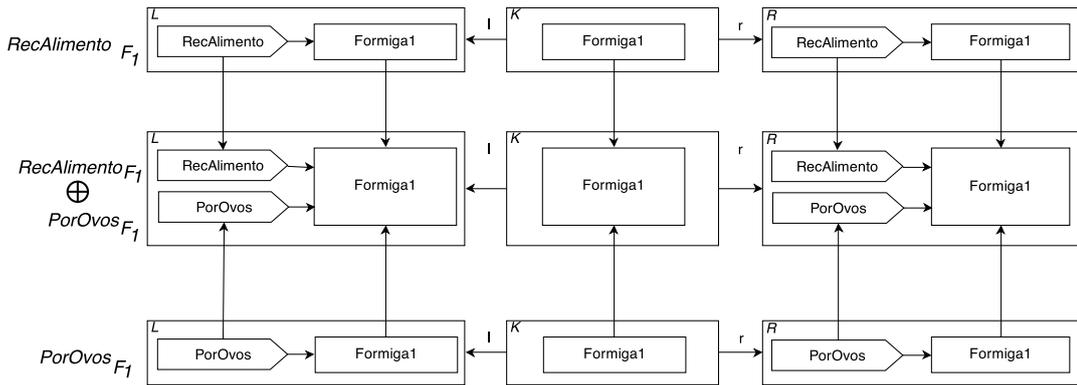


Figura 22: Composição de duas regras da  $AgGG$

e o segundo agente pode realizar as ações  $p_3$  e  $p_4$ ; então o conjunto de regras da  $GGC$  conterá as regras  $p_1 \oplus p_3$ ,  $p_1 \oplus \varepsilon$ ,  $p_1 \oplus p_4$ ,  $p_2 \oplus p_3$ ,  $p_2 \oplus \varepsilon$ ,  $p_2 \oplus p_4$ ,  $p_3 \oplus \varepsilon$ ,  $p_4 \oplus \varepsilon$ , e  $\varepsilon \oplus \varepsilon$ . Deste modo, o conjunto de expressões regulares da  $GGC$  será composto de uma expressão para cada processo de troca associado a um par de vértices (agentes ou papéis). Cada expressão regular, sobre o conjunto de regras, é definida por uma concatenação de somatórios de regras. A concatenação determina a ordem (instante de tempo) em que as regras podem ser aplicadas; e o somatório permite a escolha entre diferentes ações que podem ocorrer a cada instante, incluindo a possibilidade de não fazer nada.

Dado um conjunto  $A$  de regras de um mesmo agente (ou papel), é possível obter um conjunto contendo todas as possíveis regras de ações paralelas para as regras do conjunto  $A$ , compondo todas as regras em cada possível subconjunto de  $A$ . Este conjunto é obtido pela função definida a seguir.

**Definição 31. (Combinação de regras)** *Seja um conjunto de regras  $A$ ,  $P_A^U = \{f(X) | X \in 2^A\}$ , onde  $f$  é definido como segue:*

$$f(X) = \begin{cases} \varepsilon & \text{se } X = \emptyset \\ p_i \oplus \dots \oplus p_n, & \text{caso contrário,} \end{cases}$$

onde  $p_i \in X, i \in \{1, \dots, \#X\}$

**Definição 32. (Semântica Populacional de uma PGG)** *Seja  $PGG = (aGG, ec)$  uma gramática de grafos populacional, com  $aGG = (T, H, P, \pi)$ . A semântica de PGG é dada por um conjunto de derivações  $DerC(GGC)$  da gramática de grafos controlada  $GGC = ((T, H, P', \pi'), exp)$ , obtida como segue:*

- $P' = \{p_1 \oplus p_2 \mid a_1, a_2 \in V_H, ep \in ec(a_1, a_2), t \in I, ep(t) = (A_1, A_2), p_1 \in P_{A_1}^U, p_2 \in P_{A_2}^U\}$ ;
- $\pi'$  associa cada nome de regra  $p_1 \oplus p_2$  uma regra de ação paralela associada a  $p_1$  e  $p_2$ ;

- $exp$  é o conjunto de expressões regulares definido por:

$$exp = \{sum(t_1); \dots; sum(t_n) | a_1, a_2 \in V_H,$$

$$ep \in ec(a_1, a_2), t_1, \dots, t_n \in I, ep(t_i) = (A_1, A_2) :$$

$$sum(t_i) = \sum_{p_1 \in P_{A_1}^U, p_2 \in P_{A_2}^U} p_1 \oplus p_2 \}$$

A semântica de uma  $OGG$  é análoga à semântica de uma  $PGG$ . O que diferencia uma da outra é a capacidade de ligação presente no nível organizacional. A definição da semântica organizacional será apresentada abaixo.

**Definição 33. (Semântica Organizacional de uma  $OGG$ )** Seja  $OGG = (oGG, lc)$  uma gramática de grafos organizacional, em que  $oGG = (T, H, P, \pi)$ . A semântica de uma  $OGG$  é dada por um conjunto de derivações  $DerC(GGC)$  de uma gramática de grafos controlada  $GGC = ((T, H, P', \pi'), exp)$ , onde:

- $P' = \{p_1 \oplus p_2 \mid r_1, r_2 \in V_H, Li \in lc(r_1, r_2), t \in I, li(t) = (R_1, R_2), p_1 \in P_{R_1}^U, p_2 \in P_{R_2}^U\}$ ;
- $\pi'$  associa cada nome de regra  $p_1 \oplus p_2$  uma regra de ação paralela associada a  $p_1$  e  $p_2$ ;
- $exp$  é o conjunto de expressões regulares definido por:

$$exp = \{sum(t_1); \dots; sum(t_n) | r_1, r_2 \in V_H,$$

$$li \in lc(r_1, r_2), t_1, \dots, t_n \in I, li(t_i) = (R_1, R_2) :$$

$$sum(t_i) = \sum_{p_1 \in P_{R_1}^U, p_2 \in P_{R_2}^U} p_1 \oplus p_2 \}$$

**Exemplo 11. (Semântica de uma  $OGG$ )** Considerando a  $OGG$  descrita na Figura 23, a gramática de grafos controlada ( $CGG$ ) associada a esta gramática é composta por:

- Grafos inicial e tipo da  $OGG$  (Figura 23)
- Um conjunto de regras

$$P' = \{V_{oo}Nupcial_R, Reproduçãõ_R, Alimentaçãõ_R, CuiProle_O, Manutençãõ_O, ColAlimento_O, Construçãõ_O, Ataques_S, Defesas_S, Construçãõ_O \oplus CuiProle_O, ColAlimento_O \oplus CuiProle_O, \varepsilon\};$$

- *E um conjunto de expressões regulares*

$$exp = \{(VooNupcial_R + Constru\c{c}ao_O \oplus CuiProle_O + Defesa_S + \varepsilon);$$

$$(Reprodu\c{c}ao_R + CuiProle_O \oplus ColAlimento_O + Ataques_S + \varepsilon);$$

$$(Alimenta\c{c}ao_R + Manuten\c{c}ao_O + \varepsilon)\}.$$

As expressões regulares descritas no exemplo devem obedecer a capacidade de ligação dos papéis. As capacidades de ligação da organização são dadas pelos papéis *Reprodutora*, *Operária* e a ligação  $l_1$ , e os papéis *Operária*, *Soldado* e a ligação  $l_2$ .

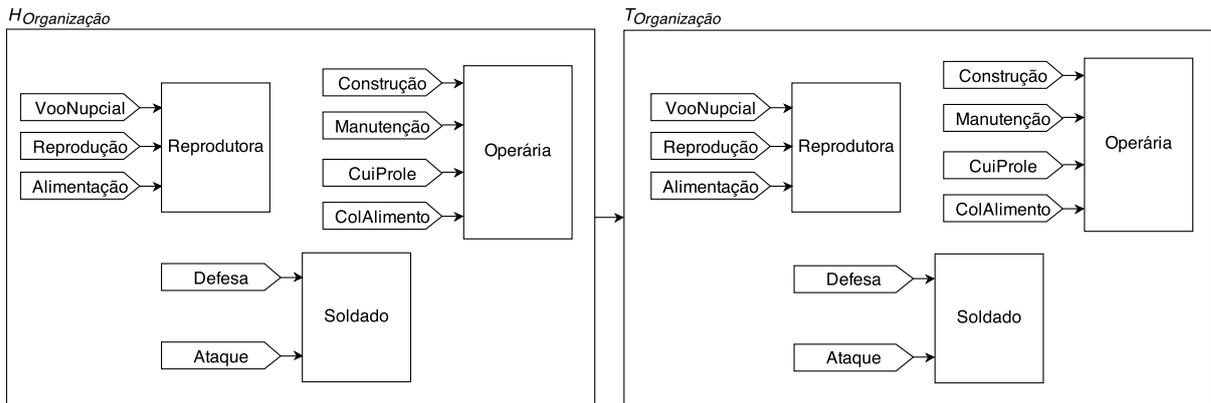


Figura 23: Grafos tipo e inicial da OGG

## 7 CONCLUSÃO

Como a tarefa de especificar *software* complexos está cada vez mais difícil, apresentou-se aqui um *framework* para modelagem de sistemas multiagentes baseado no modelo PopOrg. Este trabalho é um passo importante para a verificação formal de SMAs, usando gramática de grafos como linguagem. A gramática de grafos foi utilizada por ser uma linguagem formal, intuitiva, visual e ainda possuir ferramentas para análise para essa linguagem. A especificação apresentada é uma implementação do modelo para SMAs PopOrg, por ser um modelo mínimo, formal e que leva em conta diferentes níveis das organizações de sistemas multiagentes.

Foi definida, neste trabalho, uma gramática de grafos para os diferentes níveis de um sistema multiagente. Uma gramática de grafos populacional, onde é possível especificar diferentes tipos de agentes, determinar as ações que os agentes podem realizar, o comportamento que estes podem ter e as possíveis interações que podem ocorrer entre dois agentes. Também, definiu-se uma gramática de grafos organizacional, na qual, de forma análoga a populacional, é possível modelar os papéis e as interações entre os papéis da organização. Além disso, foi estabelecida uma relação de implementação entre os níveis populacional e organizacional do PopOrg, a qual demonstra como uma população pode implementar de forma mais concreta os comportamentos requeridos pelos papéis da organização. Finalmente, neste trabalho, a semântica das GGMA foi apresentada, a qual inclui todas as derivações que descrevem as possíveis interações entre cada par de agentes ou papéis no sistema.

A abordagem GGMA proposta restringe as ações dos agentes (ou papéis) àquelas descritas no grafo tipo cada agente (ou papel). Assim, cada agente (ou papel) tem suas próprias ações, não podendo compartilhá-las com os demais. Esta restrição poderia ser afrouxada caso fossem permitidos mais de um agente (ou papel) no grafo inicial de cada AgGG (ou RGG). Como trabalhos futuros, pretende-se estender esta abordagem para eliminar esta restrição.

A principais contribuições deste trabalho é propor um *framework* formal baseado no formalismo de gramática de grafos para especificação de sistemas multiagentes descritos com o modelo PopOrg. Com isso, projetistas de sistemas podem fazer uso

das diferentes técnicas e ferramentas disponíveis para GGs a fim de analisar tais sistemas. Além disso, a abordagem apresentada proporcionará ao modelo PopOrg uma representação visual e intuitiva, que possui ferramentas de análise para verificação formal de sistemas utilizando a técnica de prova de teoremas. Por fim, podemos destacar a definição de uma semântica PopOrg em Gramática de Grafos Controlada, a representação em GG do modelo PopOrg (estrutural e operacional), possibilitando a aplicação de ferramentas de verificação formal aos sistemas especificados de acordo com esse modelo. Mecanismos de controle para gramática de grafos foram definidos a fim de regular a ordem de aplicação das regras da gramática, a união de regras e a interação entre os agentes e papéis.

Dentre os próximos passos dentro desta abordagem, pode-se destacar: proposta de um método de análise para SMAs descritos no modelo PopOrg, utilizando uma abordagem de verificação de propriedades de GGs usando provador de teoremas (em andamento); estender esta abordagem para permitir que mais de um agente/papel possa estar presente no grafo inicial da AgGG ou RGG, com isso, seja possível compartilhar ações com mais de um tipo de agente/papel; extensão da abordagem para considerar o nível macro da organização; demonstração formal de que a relação de implementação própria preserva as derivações das OGGs; definição de um método para descrever a evolução de um SMA, usando transformação de GGs.

## REFERÊNCIAS

BARBOSA, R. d. M. **Especificação formal de organizações de sistemas multiagentes**. 2011. Tese (Doutorado em Ciência da Computação) — INF, UFRGS, Brazil.

COSTA, S. A. da. **Relational Approach of Graph Grammars**. 2010. Tese (Doutorado em Ciência da Computação) — INF, UFRGS, Brazil.

DEMAZEAU, Y.; COSTA, A. C. R. Populations and Organizations in Open Multi-agent Systems. In: NATIONAL SYMPOSIUM ON PARALLEL AND DISTRIBUTED ARTIFICIAL INTELLIGENCE (PDAI'96), 1996, Hyderabad, India. **Proceedings...** [S.l.: s.n.], 1996.

DEPKE, R.; HECKEL, R. Formalizing the Development of Agent-Based Systems Using Graph Processes. In: ICALP SATELLITE WORKSHOPS, 2000. **Anais...** [S.l.: s.n.], 2000. p.419–426.

EHRIG, H.; ENGELS, G.; KREOWSKI, H.-J.; ROZENBERG, G. (Ed.). **Handbook of graph grammars and computing by graph transformation**: vol. 2: applications, languages, and tools. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1999.

EHRIG, H.; PFENDER, M.; SCHNEIDER, H. J. Graph-grammars: An algebraic approach. In: ANNUAL SYMPOSIUM ON SWITCHING AND AUTOMATA THEORY, 14., 1973, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 1973. p.167–180. (SWAT '73).

FERREIRA JR., P. R. **Coordenação de sistemas multiagente atuando em cenários complexos**: uma abordagem baseada na divisão de trabalho dos insetos sociais. 2008. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

KNIRSCH, P.; KREOWSKI, H.-J. A Note on Modeling Agent Systems by Graph Transformation. In: AGTIVE, 1999. **Anais...** [S.l.: s.n.], 1999. p.79–86.

KOTULSKI, L. Distributed Graphs Transformed by Multiagent System. In: ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING, 9., 2008, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2008. p.1234–1242. (ICAISC '08).

KOTULSKI, L.; SDZIWIY, A. GRADIS - The multiagent environment supported by graph transformations. **Simulation Modelling Practice and Theory**, [S.l.], v.18, n.10, p.1515 – 1525, 2010.

LASKER, G.; DUBOIS, D.; SYSTEMS RESEARCH, I. I. for Advanced Studies in; CYBERNETICS. **Anticipative and Predictive Models in Systems Science**. [S.l.]: International Inst. for Advanced Studies in Systems Research and Cybernetics, 2004. n.v. 1.

MELLO, A. M. de; FOSS, L.; COSTA CAVALHEIRO, S. A. da. Gramática de Grafos para modelagem de Sistemas Multi-Agentes baseados no PopOrg. In: JORNADA DE PÓS-GRADUAÇÃO E PESQUISA - CONGREGA URCAMP, 10., 2012. **Proceedings...** [S.l.: s.n.], 2012.

MELLO, A. M. de; FOSS, L.; COSTA CAVALHEIRO, S. A. da. Towards the use of Graph Grammars for specification of Multi-Agent System Organizations. In: XVI SIMPÓSIO BRASILEIRO DE MÉTODOS FORMAIS (SBMF), 2013. **Proceedings...** [S.l.: s.n.], 2013.

MELLO, A. M. de; FOSS, L.; COSTA CAVALHEIRO, S. A. da. Graph Grammar Specification for the Populational Level of Multi-agent System Organizations. In: WORKSHOP-SCHOOL ON THEORETICAL COMPUTER SCIENCE (WEIT), 2., 2013. **Proceedings...** IEEE Computer Society, 2013. p.97–104.

MELLO, A. M. de; JUNIOR, L. C. L.; FOSS, L.; COSTA CAVALHEIRO, S. A. da. Graph Grammars: A Comparison between Verification Methods. In: WORKSHOP-SCHOOL ON THEORETICAL COMPUTER SCIENCE (WEIT), 2011. **Proceedings...** IEEE Computer Society, 2011. p.88–94.

PATON, N. W.; DÍAZ, O. Active Database Systems. **ACM Comput. Surv.**, New York, NY, USA, v.31, n.1, p.63–103, Mar. 1999.

PFALZGRAF, J.; SOBOLL, T. On a General Notion of Transformation for Multiagent Systems and its Implementation. **ECEASST**, [S.l.], v.12, 2008.

ROZENBERG, G. (Ed.). **Handbook of graph grammars and computing by graph transformation**: volume I. foundations. River Edge, USA: W.S.Publishing Co., Inc., 1997.

SCHATTEN, M. Active Graph Rewriting Rules for Modeling Multi-Agent Organizational Dynamics. In: IBC 2012: 1ST INTERNATIONAL INTERNET & BUSINESS CONFERENCE, ROVINJ, CROATIA, JUNE 27-28, 2012, 2012. **Proceedings...** BIT Society, 2012. p.180–185. (International Internet & Business Conference).

SCHATTEN, M. Reorganization in Multi-Agent Architectures: An Active Graph Grammar Approach. **Business Systems Research**, [S.I.], v.4, n.1, p.14–20, 2013.

ŚLUSARCZYK, G. A grammar-based multiagent system in dynamic design. **Artif. Intell. Eng. Des. Anal. Manuf.**, New York, NY, USA, v.22, p.129–145, January 2008.

SMITH, B.; HOWARD, A.; MCNEW, J.-M.; WANG, J.; EGERSTEDT, M. Multi-robot deployment and coordination with Embedded Graph Grammars. **Auton. Robots**, Hingham, MA, USA, v.26, p.79–98, January 2009.

WANG, Z.-G.; LIANG, X.-H. A Graph Transformation Based Meta-Model for Structural Evolution in Multi-Agent Systems. In: IEEE COMPUTER SOCIETY, 2006, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2006. p.548 – 553.

WANG, Z.-G.; LIANG, X.-H. A Graph Based Simulation of Reorganization in Multi-agent Systems. In: IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON INTELLIGENT AGENT TECHNOLOGY, 2006, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2006. p.129–132. (IAT '06).

WANG, Z.-G.; LIANG, X.-H.; ZHAO, Q.-P. A graph transformation system model of dynamic reorganization in multi-agent systems. In: INTELLIGENT DATA ENGINEERING AND AUTOMATED LEARNING, 7., 2006, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2006. p.1182–1190. (IDEAL'06).

WILSON, E. O. **Sociobiology**: the abridged edition. [S.I.]: Cambridge, MA: The Belknap Press of Harvard University Press, 1980. n.v. 70.

WOOLDRIDGE, M. J. **Introduction to Multiagent Systems**. New York, NY, USA: John Wiley & Sons, Inc., 2001.

**GGMA: Gramática de Grafos para Sistemas Multiagentes baseados no PopOrg – André Moura de Mello**



**UNIVERSIDADE FEDERAL DE PELOTAS**  
Centro de Desenvolvimento Tecnológico  
Programa de Pós-Graduação em Computação



Dissertação

**GGMA: Gramática de Grafos para Sistemas Multiagentes baseados no PopOrg**

**ANDRÉ MOURA DE MELLO**

Pelotas, 2014