

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL - UFRGS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

SIRLEI INÊS SULZBACH

**Definição e Especificação Formal
do Jogo Diferencial “Lobos e Cordeiro”**

Dissertação apresentada como requisito
parcial para a obtenção do grau de
Mestre em Ciência da Computação

Prof^a. Dr^a. Laira Vieira Toscani
Orientadora
Prof^a. Dr^a. Leila Ribeiro
Co-Orientadora

Porto Alegre, maio de 2005

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Sulzbach, Sirlei Inês

Definição e Especificação Formal do Jogo Diferencial “Lobos e Cordeiro” / Sirlei Inês Sulzbach – Porto Alegre: Programa de Pós-Graduação em Computação, 2005.

97 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2005. Orientadora: Laira Vieira Toscani; Co-orientadora: Leila Ribeiro.

1. Teoria dos Jogos. 2. Jogos Diferenciais. 3. Redes de Petri. 4. Especificação Formal. I. Toscani, Laira Vieira. II. Ribeiro, Leila. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer aos meus pais João e Claudência e aos meus irmãos Marilei e Júnior, que estiveram sempre do meu lado me incentivando e me apoiando durante todas as minhas dificuldades, nunca me deixando desistir.

Ao meu amigo e companheiro Eduardo, pelos estímulos e paciência para a conclusão deste trabalho.

À minha orientadora Laira Vieira Toscani, por me auxiliar em todos os passos deste trabalho, me incentivando e resolvendo pacientemente todas as minhas dúvidas. Igualmente agradeço à minha co-orientadora Leila Ribeiro.

Aos meus amigos que, mesmo longe, sempre estiveram interessados no andamento deste trabalho.

A todas as pessoas que contribuíram, direta ou indiretamente, para a realização deste trabalho, e a Deus, por dar-me forças para continuar.

SUMÁRIO

| | |
|---|----|
| LISTA DE ABREVIATURAS | 06 |
| LISTA DE FIGURAS | 08 |
| LISTA DE TABELAS | 10 |
| RESUMO | 11 |
| ABSTRACT | 12 |
| 1 INTRODUÇÃO | 13 |
| 1.1 Motivação | 13 |
| 1.2 Objetivos | 14 |
| 1.3 Estrutura do texto | 14 |
| 2 TEORIA DOS JOGOS | 16 |
| 2.1 A Classificação dos Jogos | 16 |
| 2.2 Os Jogos Diferenciais | 18 |
| 3 REDES DE PETRI | 19 |
| 3.1 As Características das Redes de Petri | 19 |
| 3.2 Redes de Petri Condição/Evento | 20 |
| 4 O JOGO DIFERENCIAL “LOBOS E CORDEIRO” | 22 |
| 4.1 A Definição do Jogo | 23 |
| 4.2 As Regras do Jogo | 25 |
| 4.3 Exemplos de Partidas | 27 |
| 4.4 As Estratégias do Jogo | 35 |
| 4.5 A Especificação Formal do Jogo | 39 |
| 4.5.1 Especificação Inicial | 40 |
| 4.5.2 Características da Especificação | 48 |
| 4.5.3 Especificação Ampliada | 54 |
| 4.5.4 Especificação para Estratégias do Jogo | 60 |
| 4.6 Situações de Jogo | 62 |

| | | |
|------------|--|-----------|
| 4.6.1 | <i>Liveness</i> | 63 |
| 4.6.2 | Alcançabilidade | 64 |
| 4.6.3 | <i>Deadlock</i> | 66 |
| 4.7 | Conclusões | 68 |
| 4.7.1 | Características de Jogos em Geral | 69 |
| 5 | PROBLEMAS RESOLVIDOS DO JOGO DIFERENCIAL “LOBOS E CORDEIRO” | 71 |
| 5.1 | Problemas Simples (não dependentes de estratégias) | 72 |
| 5.2 | Problemas Não Simples (dependentes de estratégias) | 79 |
| 5.3 | Algoritmos de Estratégias | 86 |
| 5.4 | Conclusões | 87 |
| 5.4.1 | Questões Respondidas | 88 |
| 6 | CONCLUSÃO | 90 |
| | REFERÊNCIAS | 94 |

LISTA DE ABREVIATURAS

| | |
|------|-------------------------|
| C/E | Condição/Evento. |
| EF | Esquerda Frente. |
| EA | Esquerda Atrás. |
| DF | Direita Frente. |
| DA | Direita Atrás. |
| E | Esquerda. |
| D | Direita. |
| C | Cordeiro. |
| L | Lobos. |
| RPJ | Rede de Petri do Jogo. |
| cmc | Condição m do cordeiro. |
| cnc | Condição n do cordeiro. |
| cml | Condição m dos lobos. |
| cnc | Condição n dos lobos. |
| ex | Evento x. |
| ey | Evento y. |
| ez | Evento z. |
| jl | Joga lobo. |
| jc | Joga cordeiro. |
| cm | Condição m. |
| cn | Condição n. |
| ejl1 | Evento joga lobo 1. |

| | |
|------|---------------------|
| ejl2 | Evento joga lobo 2. |
| ejl3 | Evento joga lobo 3. |
| ejl4 | Evento joga lobo 4. |
| jl1 | Joga lobo 1. |
| jl2 | Joga lobo 2. |
| jl3 | Joga lobo 3. |
| jl4 | Joga lobo 4. |
| L1 | Lobo 1. |
| L2 | Lobo 2. |
| L3 | Lobo 3. |
| L4 | Lobo 4. |
| ezl1 | Evento z lobo 1. |
| ezl2 | Evento z lobo 2. |
| ezl3 | Evento z lobo 3. |
| ezl4 | Evento z lobo 4. |
| cm11 | Condição m lobo 1. |
| cm12 | Condição m lobo 2. |
| cm13 | Condição m lobo 3. |
| cm14 | Condição m lobo 4. |
| cn11 | Condição n lobo 1. |
| cn12 | Condição n lobo 2. |
| cn13 | Condição n lobo 3. |
| cn14 | Condição n lobo 4. |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 3.1: As quatro estações e suas mudanças | 21 |
| Figura 4.1: As peças que representam os lobos e o cordeiro | 23 |
| Figura 4.2: Paredes consideradas para o jogo | 23 |
| Figura 4.3: Direção dos movimentos direita e esquerda..... | 24 |
| Figura 4.4: Representação das posições dos jogadores no tabuleiro | 35 |
| Figura 4.5: Rede de Petri representando as condições, os eventos e o <i>token</i> do cordeiro | 45 |
| Figura 4.6: Rede de Petri representando as condições, os eventos e os <i>tokens</i> dos lobos..... | 46 |
| Figura 4.7: Rede de Petri representando as condições, os eventos e os <i>tokens</i> da vez para jogar e das condições livres..... | 47 |
| Figura 4.8: Rede <i>RPJ</i> para a característica de alternância entre jogadores | 48 |
| Figura 4.9: Exemplo para a característica de alternância entre jogadores | 49 |
| Figura 4.10: Ampliação da rede <i>RPJ</i> para o controle de condições livres | 50 |
| Figura 4.11: Exemplo para o controle de condições livres..... | 51 |
| Figura 4.12: Distribuição do <i>token</i> vez para um dos lobos..... | 51 |
| Figura 4.13: <i>Token</i> vez atribuído para um dos lobos | 52 |
| Figura 4.14: A ampliação da condição <i>jl</i> na rede <i>RPJ</i> | 53 |
| Figura 4.15: Rede <i>RPJ</i> | 56 |
| Figura 4.16: Exemplo de vitória do cordeiro..... | 58 |
| Figura 4.17: Exemplo de vitória dos lobos | 59 |
| Figura 4.18: Situação de jogo para uma estratégia dos lobos..... | 60 |
| Figura 4.19: Parte da rede <i>RPJ</i> para uma estratégia de jogo dos lobos | 61 |
| Figura 4.20: Situação de jogo para uma estratégia do cordeiro..... | 61 |
| Figura 4.21: Parte da rede <i>RPJ</i> para uma estratégia de jogo do cordeiro | 62 |
| Figura 4.22: Rede <i>RPJ</i> para velocidades diferentes para os jogadores..... | 70 |

| | |
|--|----|
| Figura 4.23: Rede <i>RPJ</i> para obstáculos existentes no espaço do jogo | 70 |
| Figura 5.1: Tabuleiro do jogo “lobos e cordeiro”, de dimensão 8x8..... | 71 |
| Figura 5.2: Situação inicial do jogo | 73 |
| Figura 5.3: Final de jogo: número de passos mínimos para o cordeiro vencer o jogo | 74 |
| Figura 5.4: Final de jogo: número de passos mínimos para os lobos vencerem o jogo | 74 |
| Figura 5.5: Final de jogo: número de passos máximos para o cordeiro vencer o jogo | 75 |
| Figura 5.6: Final de jogo: número de passos máximos para os lobos vencerem o jogo | 75 |
| Figura 5.7: Situação da figura 5.3 aplicada à rede <i>RPJ</i> | 76 |
| Figura 5.8: Situação da figura 5.4 aplicada à rede <i>RPJ</i> | 77 |
| Figura 5.9: Situação da figura 5.5 aplicada à rede <i>RPJ</i> | 78 |
| Figura 5.10: Situação da figura 5.6 aplicada à rede <i>RPJ</i> | 79 |
| Figura 5.11: Final de jogo: o cordeiro alcança a parede de chegada, após 9 jogadas... | 80 |
| Figura 5.12: O cordeiro passa os lobos..... | 80 |
| Figura 5.13: Parte da rede <i>RPJ</i> para chegar à situação de jogo da figura 5.12..... | 81 |
| Figura 5.14: Final de jogo: os lobos prendem o cordeiro, após 6 jogadas..... | 81 |
| Figura 5.15: Os lobos formam uma barreira, após 4 jogadas | 82 |
| Figura 5.16: Parte da rede <i>RPJ</i> para chegar à situação de jogo da figura 5.15..... | 82 |
| Figura 5.17: Final de jogo: o cordeiro alcança a parede de chegada, após 28 jogadas | 83 |
| Figura 5.18: O cordeiro passa os lobos, mas não pode alcançar a parede de chegada | 83 |
| Figura 5.19: Parte da rede <i>RPJ</i> para chegar à situação de jogo da figura 5.18..... | 84 |
| Figura 5.20: Final de jogo: os lobos prendem o cordeiro, após 26 jogadas..... | 84 |
| Figura 5.21: Os lobos formam uma barreira, após 24 jogadas | 85 |
| Figura 5.22: Parte da rede <i>RPJ</i> para chegar à situação de jogo da figura 5.21 | 85 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 2.1: Anos de prisão para cada indivíduo, dependendo de sua atitude..... | 17 |
| Tabela 4.1: Possibilidades de jogadas para os jogadores, no jogo A..... | 36 |
| Tabela 4.2: Possibilidades de jogadas para os jogadores, no jogo B..... | 37 |
| Tabela 4.3: Estratégias adotadas por cada um dos jogadores..... | 38 |

RESUMO

No presente trabalho serão apresentadas questões usuais em jogos diferenciais, nos quais os jogadores envolvidos têm objetivos diferentes; ou seja, enquanto um dos jogadores tenta “fugir”, o outro tenta “pegar”.

Além disso, será definido um modelo de especificação para o jogo diferencial “lobos e cordeiro”. As Redes de Petri foram escolhidas como forma de especificação para o jogo proposto.

Assim, o objetivo será estabelecer estratégias eficientes para o jogo lobos e cordeiro para que se possa realizar um estudo da complexidade das questões apresentadas para este jogo, levando-se em consideração a especificação formal apresentada para tal jogo.

Palavras-chave: teoria dos jogos, jogos diferenciais, redes de Petri, especificação formal.

Definition and Formal Specification of the Differential Game “Wolfs and Lamb”

ABSTRACT

In this work usual questions in differential games will be presented, in which the involved players have different objectives; that is, while one of the players tries "to run away", the other tries "to catch".

Moreover, a specification for the differential game "wolves and lamb" will be defined. The Petri Nets had been chosen as specification formalism for the considered game.

Thus, the objective is to establish efficient strategies for the game wolves and lamb so that we can carry out a study of the complexity of the presented questions, taking into consideration the presented formal specification for the game.

Keywords: game theory, differential games, Petri nets, formal specification.

1 INTRODUÇÃO

O paradigma da teoria dos jogos está naturalmente incorporado à Ciência da Computação, uma vez que, vários problemas desta ciência podem ser formulados em função daquela teoria. Isto significa que tais problemas apresentam muitos incentivos para o desenvolvimento de algoritmos através da teoria dos jogos, os quais podem seguramente, ser empregados como base para determinadas soluções de problemas da Ciência da Computação e vice-versa.

A motivação da teoria dos jogos leva em consideração colocações de alguns autores da literatura, onde se tem que, pela natureza de determinados jogos já se sabe de antemão que muitos de seus problemas são considerados intratáveis computacionalmente.

Os jogos estão relacionados aos modelos de computação, principalmente no que diz respeito às linguagens e às máquinas; onde, por exemplo, segundo [PET 2000], um jogo pode ser visto como uma máquina.

Assim, pode-se utilizar as contribuições da Teoria da Computação, mais especificamente da Complexidade de Algoritmos, e realizar-se um estudo da complexidade para o desenvolvimento de algoritmos da teoria dos jogos; uma vez que, a análise da complexidade é de grande importância na determinação do desempenho dos algoritmos.

Com o crescimento do tamanho e complexidade dos sistemas de software, surgiu a necessidade de construir especificações mais precisas para descrever o comportamento desses sistemas, bem como possibilitar a verificação de propriedades destes sistemas e da correção de suas implementações [DÉH 2000].

Deste modo, não se pode deixar de salientar que uma especificação formal possibilita entender situações a serem representadas para um jogo, por exemplo. Por isso, devido ao fato de décadas de estudo envolvendo as redes de Petri, elas se mostram grandes aliadas no estudo das situações que ocorrem e/ou que podem vir a ocorrer, quando se fala em representar os jogos.

1.1 Motivação

Este trabalho é motivado pela teoria dos jogos que sempre se mostrou promissora na resolução de problemas de interesse geral. Isso pois, quando se fala em jogos de uma forma generalizada, logo se pensa em vitória e uma vez que se constata esta característica, não há como deixar de considerar as estratégias para obter-se a referida vitória.

O ponto de partida para este estudo são as afirmações de autores da área de teoria dos jogos, os quais afirmam que existem problemas desta teoria que são considerados intratáveis computacionalmente.

Ainda há o fato de que os conceitos da teoria dos jogos, bem como os resultados obtidos na mesma teoria, são utilizados em várias áreas do estudo acadêmico e do estudo profissional.

1.2 Objetivos

O objetivo inicial deste trabalho é definir um jogo diferencial e para isto foi escolhido o jogo “lobos e cordeiro”. Para este jogo diferencial será apresentada uma especificação formal usando redes de Petri. Esta especificação formal deve ajudar a entender o comportamento do jogo, ao ser transposto em termos computacionais.

Com a especificação formal há uma facilidade no desenvolvimento de algoritmos eficientes para os jogos, uma vez que se tem o conhecimento do problema a ser resolvido.

Há algumas questões em jogos diferenciais que precisam ser respondidas – estas questões são descritas na seção 2.2. Com base nas questões levantadas para o jogo diferencial lobos e cordeiro, a especificação formal deste jogo possibilita a compreensão e o esclarecimento destas questões.

Para a teoria dos jogos estas questões são relevantes, pois uma vez que elas são resolvidas, pode-se definir se determinadas situações são possíveis de serem alcançadas no contexto de um jogo. Já no que diz respeito à Ciência da Computação, a resolução destas questões significa o aprimoramento dos algoritmos desenvolvidos para os jogos.

1.3 Estrutura do texto

O texto aborda temas que abrangem: a teoria dos jogos, as redes de Petri, o jogo diferencial lobos e cordeiro e alguns problemas que giram em torno do mesmo jogo.

Após este capítulo introdutório, esta dissertação está organizada em cinco capítulos intitulados: Teoria dos Jogos, Redes de Petri, O Jogo Diferencial “Lobos e Cordeiro”, Problemas Resolvidos do Jogo Diferencial “Lobos e Cordeiro” e Conclusão.

O capítulo 2 apresenta um estudo introdutório da teoria dos jogos, sendo feita uma breve descrição desta teoria. Aborda-se a classificação dos jogos e os jogos diferenciais, cujos conceitos são a base do presente trabalho.

No capítulo 3 relembram-se as definições para as redes de Petri, extraídas de [REI 85]. Para tanto, descreve-se as características das redes de Petri condição/evento, as quais são utilizadas na formulação de um modelo de especificação formal para o jogo escolhido neste trabalho.

O capítulo 4 traz o coração deste estudo, ou seja, é onde o jogo diferencial lobos e cordeiro é explicado. Inicialmente, define-se o jogo diferencial propriamente dito; parte-se então para as regras do jogo e em seguida, são mostrados dois exemplos de partidas. Fala-se também das estratégias encontradas para este jogo e apresenta-se finalmente, a especificação formal deste jogo. Além disso, algumas situações de jogo são descritas e

alguns resultados encontrados na literatura para as redes de Petri e para os jogos são apresentados.

No capítulo 5 serão identificados os problemas resolvidos para o jogo diferencial lobos e cordeiro, tendo por base o número mínimo e máximo de jogadas para cada jogador – lobos e cordeiro. Chega-se com isso, aos problemas simples, ou aqueles que não dependem de estratégia e aos problemas não simples, ou aqueles que dependem de estratégia. Mostra-se também, a idéia dos algoritmos de estratégias utilizados para o jogo lobos e cordeiro.

O capítulo 6 apresenta as conclusões deste trabalho, bem como os aspectos que ficaram em aberto como sugestões a trabalhos futuros.

Com a finalidade de esclarecer a terminologia utilizada ao longo do trabalho, tem-se que ao referir-se ao “jogo diferencial lobos e cordeiro” ou ao “jogo lobos e cordeiro”, está-se considerando igualmente as duas formas. Ou seja, as duas expressões têm o mesmo sentido (a mesma idéia).

2 TEORIA DOS JOGOS

A teoria dos jogos foi originalmente formulada por John Von Neumann e Oskar Morgenstern, *apud in* [PET 2000], em 1944, com a publicação de *Games and Economic Behavior* por Von Neumann e Morgenstern. Logo após, em 1950, John Nash amplia os estudos de Von Neumann e de Morgenstern para solucionar problemas da teoria dos jogos até então em aberto, com a sua revolucionária teoria do equilíbrio em situações de jogo, conhecida como “teoria do equilíbrio de Nash”. Atualmente, as descobertas da teoria dos jogos vêm sendo aplicadas em diversos campos, desde a área acadêmica até a área profissional.

2.1 A Classificação dos Jogos

Segundo [BAZ 2001], a teoria dos jogos pode ser definida como o modelo teórico de conflito que ocorre entre jogadores (aqueles que tomam decisão), com situações onde:

- há várias pessoas envolvidas;
- a solução coletiva depende da decisão individual destas pessoas envolvidas;
- há uma relação de dependência entre tais pessoas envolvidas (ou melhor, existe um conhecimento comum);
- há a avaliação de todos os fatores no processo decisório.

O objetivo de tal teoria é que os jogadores maximizem suas utilidades em todas as situações do jogo, para assim, chegarem à solução; portanto, nenhum jogador desvia deste propósito. Há a análise das regras, onde um jogador racional, através do seu conhecimento, sabe que os demais jogadores também são racionais.

Com a definição de equilíbrio de Nash, *apud in* [BAZ 2001], em 1950, vieram os conceitos de jogo cooperativo - onde existe a comunicação e a realização de acordos entre os jogadores - e também, de jogo não-cooperativo - no qual não há acordos entre os jogadores e estes maximizam seus ganhos independentemente. Há ainda a classificação quanto à soma dos ganhos (jogo de soma zero e jogo de soma não-zero) e quanto ao número de jogadores (com dois jogadores ou com n jogadores, onde $n > 2$)¹.

¹ Existem jogos que envolvem somente um jogador, mas tais jogos não consideram a definição do equilíbrio de Nash, devido as suas características.

Outra classificação para um jogo pode ser dada pelo jogo de informação completa, no qual se sabe o conjunto de estratégias e conhece-se a matriz de jogo e ainda, o jogo de informação incompleta, onde nem todos os jogadores conhecem o conjunto de informações do jogo. Seguindo os conceitos, tem-se o jogo com informação perfeita, ou seja, o conjunto informacional de um jogador está associado a um único nodo de decisão (diz-se que há a atuação de um jogador por vez, e não jogadores simultâneos) e o jogo com informação imperfeita, no qual há um conjunto informacional não único para cada jogador [BAZ 2001].

Considerando-se estas abordagens, o equilíbrio de Nash é interpretado como um conjunto de estratégias, um para cada jogador e cada um sendo a melhor resposta para as jogadas dos demais jogadores. Assim, o conceito de equilíbrio leva ao conceito de solução, que consiste da melhor ação para cada jogador do jogo, ou melhor, as estratégias dos jogadores para maximizar seus ganhos individuais.

Um exemplo clássico encontrado na literatura é o “dilema do prisioneiro”, o qual foi apresentado pela primeira vez na Universidade de Princeton em 1950, como um exemplo da teoria dos jogos, e consiste do seguinte: a polícia prende dois indivíduos – indivíduo *A* e indivíduo *B* – suspeitos de cometerem um crime leve (roubo de carro, por exemplo) e os coloca em duas celas separadas, sem possibilidade de comunicação entre eles. O detetive suspeita que um deles cometeu o crime e faz uma proposta. Quem denunciar o outro e der as pistas para a condenação fica livre, enquanto o outro pega cinco anos de pena. Se os dois se acusarem mutuamente, os dois pegam três anos. Se os dois ficarem calados, eles pegam um ano de cadeia cada um.

A tabela 2.1 abaixo mostra os anos de prisão para cada um dos indivíduos, dependendo das atitudes de cada um deles para a situação do dilema do prisioneiro.

Tabela 2.1: Anos de prisão para cada indivíduo, dependendo de sua atitude.

| | Indivíduo <i>B</i> fica calado | Indivíduo <i>B</i> confessa |
|--------------------------------|--|--|
| Indivíduo <i>A</i> fica calado | 1 ano para o indivíduo <i>A</i> 1 ano para o indivíduo <i>B</i> | 5 anos para o indivíduo <i>A</i> 0 anos para o indivíduo <i>B</i> |
| Indivíduo <i>A</i> confessa | 0 anos para o indivíduo <i>A</i> 5 anos para o indivíduo <i>B</i> | 3 anos para o indivíduo <i>A</i> 3 anos para o indivíduo <i>B</i> |

O dilema do prisioneiro, na sua versão clássica ou em sua versão modificada (onde há possibilidade de interação), tem sido usado para estudar o problema da cooperação entre indivíduos, grupos e nações em diversos tipos de problemas.

A classificação que inclui os jogos diferenciais, os quais são apresentados na secção 2.2 a seguir, é dada pelo aspecto do não-cooperativismo, ou seja, cada jogador envolvido no jogo – geralmente, um buscador e um evasor – atua de forma independente, com base em estratégias que venham a maximizar os seus ganhos individuais. As características dos jogos diferenciais, base deste trabalho, são apresentadas e ampliadas em [REI 93] e em [SZU 96].

2.2 Os Jogos Diferenciais

O foco deste estudo está voltado para os "jogos diferenciais", encontrados e analisados em [REI 93] e em [SZU 96]. Estes jogos serão abordados em um "domínio contínuo", por poderem capturar aspectos particulares das situações de jogo - como, por exemplo, a próxima jogada de um jogador *A* pode basear-se em uma estratégia que o favoreça, considerando a jogada realizada anteriormente por um jogador *B*. Ou melhor, diz-se que um domínio contínuo é aquele no qual, a partir de um início consegue-se chegar a um final, por meio de sucessivas jogadas finitas (jogadas estas, também ditas passos discretos). Isso ocorre com a "alternação" dessas jogadas entre os jogadores, podendo-se a partir disso chegar ao conceito de "alternação contínua".

Os jogos diferenciais são assim denominados, porque os jogadores envolvidos têm objetivos diferentes, ou seja, enquanto um dos jogadores tenta "fugir", o outro tenta "pegar" (por isso esse tipo de jogo é dito também "jogo de busca" e assim sendo, os objetivos dos jogadores envolvidos são conflitantes). Emprega-se a terminologia "buscador" e "evasor", onde o buscador tenta pegar o evasor e, o evasor tenta fugir do buscador.

Um exemplo típico de um jogo diferencial é conhecido como o "chofer homicida" (descrito por Lewin em 1994, *apud in* [SZU 96]). Neste jogo, o motorista de um carro em movimento atua com o objetivo de derrubar um pedestre, o qual, é claro, não deseja ser atingido. O carro pode mover-se mais rapidamente do que o pedestre, mas este por sua vez, pode manobrar de forma melhor (tem maior mobilidade). Assim, a melhor estratégia ao buscador (o carro) e ao evasor (o pedestre) para seguir de maneira a alcançar seus objetivos é conflitante.

Há alguns problemas que são encontrados em jogos diferenciais que dizem respeito à forma de como proceder para se alcançar uma situação de vitória. De fato, nestes jogos é interessante, a partir de um estado para um jogador, questionar:

- Existe uma estratégia segura, isto é, uma estratégia que leve de um ponto seguro (ponto de um caminho vitorioso) para outro ponto seguro?
- Existe uma estratégia vitoriosa, isto é, uma estratégia que chegará ao ponto de vitória, independente das jogadas do adversário?
- Existe uma estratégia (segura / vitoriosa) ótima (por exemplo, aquela que é feita em um período de tempo menor)?
- Existe uma estratégia, que talvez não seja ótima, mas sob certas condições conhecidas ela é segura / vencedora?
- Como se encontram tais estratégias?

Para o caso dos jogos diferenciais, cada jogador (buscador/evasor) procurará por uma estratégia vencedora conforme seus objetivos, que são naturalmente conflitantes. E a melhor estratégia portanto, será também a que impor mais dificuldades ao adversário.

3 REDES DE PETRI

Um modelo é uma representação, geralmente em termos matemáticos, das principais características de um objeto ou sistema. Através da análise do modelo, um sistema real pode ser estudado sem o perigo, o custo ou a inconveniência da manipulação de seus elementos. Dentre as técnicas formais para modelar e analisar sistemas tem-se, por exemplo, as Redes de Petri [JON 77].

As redes de Petri são uma ferramenta gráfica e matemática de modelagem que pode ser aplicada em diversos tipos de sistemas. Em especial, podem ser usadas para modelagem de sistemas síncronos e com alto índice de concorrência e paralelismo [ESP 98a]. Em redes de Petri existem propriedades, cuja verificação é indecidível, portanto não são verificáveis automaticamente. Outras são NP-Difícil ou NP-Completo, o que significa que não se espera encontrar um algoritmo eficiente para verificá-las e ainda, outras têm um algoritmo de verificação eficiente, isto é, polinomial.

3.1 As Características das Redes de Petri

O conceito de redes de Petri foi introduzido por Carl Adam Petri, em sua tese de doutorado intitulada *Kommunikation mit Automaten*, em 1962 na faculdade de Matemática e Física da Universidade de Darmstadt, na Alemanha Ocidental. O trabalho foi introduzido nos Estados Unidos por A. W. Holt, que liderava o projeto de teoria de sistemas de informação do *Applied Data Research, Inc.* Nos anos seguintes, à medida que seu trabalho foi sendo introduzido em outros países, esse conceito difundiu-se e vem sendo aperfeiçoado através da promoção de *workshops*, conferências internacionais e publicações de novos trabalhos, os quais vêm formando mestres e doutores nessa área [MAC 96].

Como dito inicialmente, as redes de Petri são uma ferramenta gráfica e matemática de modelagem (descrição/especificação), que podem ser aplicadas em diversos tipos de sistemas, apresentando um bom nível de abstração em comparação a outros modelos gráficos. Além disso, as redes de Petri possibilitam a verificação da correção do sistema especificado. Usando-se as mesmas, pode-se modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos. Tais sistemas englobam: redes de computadores e protocolos de comunicação, sistemas operacionais, programação paralela, bancos de dados distribuídos e sistemas flexíveis de manufatura, entre muitas outras áreas. A análise das redes de Petri pode revelar informações importantes sobre a estrutura e o comportamento dinâmico do sistema modelado.

Quaisquer das áreas citadas no parágrafo anterior, em que a concorrência seja um fator preponderante, são passíveis de ter vantajosamente aplicadas as redes de Petri.

Outra característica importante, é que elas permitem a análise da estrutura e do comportamento dinâmico do sistema modelado, e tais informações são obtidas da mesma representação [MAC 96].

Todo o formalismo matemático das redes de Petri possibilita a análise precisa do modelo; verificações de propriedades inerentes aos sistemas concorrentes, tais como relações de precedência entre eventos, sincronização e (in)existência de bloqueio, são feitas formalmente. Além de todas estas vantagens, elas ainda permitem a visualização dos processos e a comunicação entre eles.

A principal razão para o sucesso das redes de Petri deve-se ao fato destas possuírem conceitos simples de lugares e transições e porque fornecem uma representação gráfica do sistema. Além da grande aceitação na indústria, existe uma teoria sólida de concorrência por detrás das redes de Petri. Grande parte da teoria de redes é desenvolvida por diversos modelos básicos de redes como redes de condição/evento e lugar/transição, as quais enfocam a descrição do resultado das informações (*tokens*) no sistema, e não a representação da informação onde a mesma deve ser codificada graficamente usando lugares e/ou transições, originando redes grandes e complexas.

Nas redes lugar/transição os lugares podem acumular vários *tokens*, assim como os arcos podem ser valorados [MAC 96]. Já nas redes condição/evento os lugares acumulam somente um *token*, dependendo se uma condição é satisfeita ou não através da ocorrência de um evento [REI 85].

3.2 Redes de Petri Condição/Evento

As definições que seguem sobre redes de Petri condição/evento são extraídas de [REI 85]. É através deste tipo de rede de Petri condição/evento que o jogo diferencial lobos e cordeiro será especificado. Isto porque, a rede precisa ter a característica de poder acumular um único *token* por lugar (condição) a cada instante que uma transição (evento) ocorre.

Uma rede condição/evento (C/E), como o próprio nome define, é uma rede que consiste de condições e eventos. Nesta rede as condições são representadas por elementos de um conjunto S e os eventos são representados por elementos de um conjunto T .

Se os elementos de S e de T são interpretados como condições e eventos respectivamente, pode-se obter então a tripla $(B, E; F)$. Esta tripla é originalmente definida por [REI 85] como a tripla $(S, T; F)$, onde S é o conjunto de lugares, T é o conjunto de transições e $F \subseteq (S \times T) \cup (T \times S)$ é o conjunto de pré e pós-condições da rede, onde a pré-condição é uma condição que é satisfeita antes de um evento ocorrer e a pós-condição é uma condição que é satisfeita depois de um evento ocorrer.

Definição 3.1: Rede de Petri Condição/Evento. Formalmente, uma rede condição/evento é uma tripla $N = (B, E; F)$ [REI 85], onde já se sabe que B é um conjunto de condições, E é um conjunto de eventos e $F \subseteq (B \times E) \cup (E \times B)$ é um conjunto de pré e pós-condições da rede e portanto:

- i) um subconjunto $c \subseteq B$ é denominado um *caso*;

ii) seja $e \in E$ e $c \subseteq B$, e possui concessão em c (e está habilitado para c), se e somente se $\bullet e \subseteq c \wedge e^\bullet \cap c = \emptyset$;

iii) seja $e \in E$, seja $c \subseteq B$ e seja e habilitado para c . $c' = (c \setminus \bullet e) \cup e^\bullet$ é denominado o caso seguinte de c sob e (c' resulta da ocorrência de e no caso c) e pode-se obter então $c [e > c'$.

Esclarece-se que “ $\bullet e$ ” e “ e^\bullet ” representam respectivamente, a pré e a pós-condições, ou seja, $pre(e)$ e $pos(e)$. Esta notação $pre(e)$ e $pos(e)$ será utilizada na especificação do jogos diferencial lobos e cordeiro. Da mesma forma, a palavra “caso” empregada por [REI 85] será substituída por “estado”.

Nas redes de Petri condição/evento as condições podem ser satisfeitas ou não e a ocorrência de eventos altera a satisfação destas condições. Em cada uma das configurações desta rede algumas condições são satisfeitas enquanto as demais não são satisfeitas. O conjunto de condições que são satisfeitas em uma configuração é chamado de estado.

Assim sendo, um evento “ e ” pode ocorrer em um estado “ c ”, se e somente se as pré-condições de e pertencem a c (significando que as condições estão marcadas por um token) e as pós-condições de e não pertencem a c (significando que as condições não estão marcadas por um token). Quando e ocorre, as pré-condições de e param de ser satisfeitas e as pós-condições de e começam a ser satisfeitas.

Para representar graficamente um estado c , coloca-se uma marca (um token) em cada círculo que pertence a c . A figura 3.1 mostra uma rede condição/evento; esta rede também é vista como um estado.

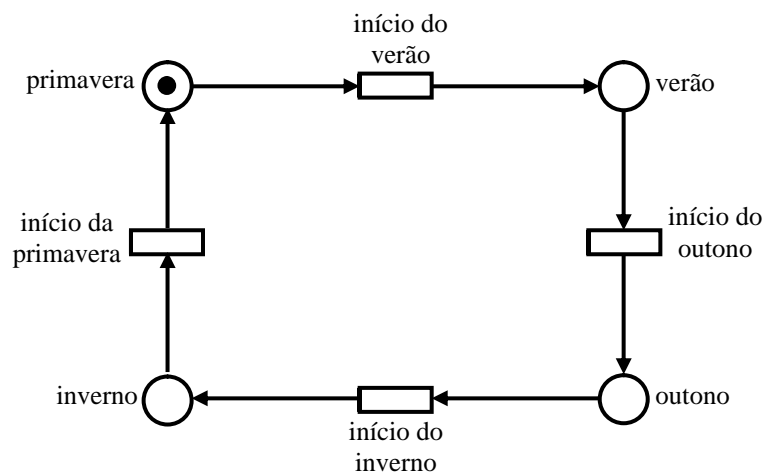


Figura 3.1: As quatro estações e suas mudanças.

Nesta figura 3.1 percebe-se que os círculos representam as condições, os retângulos representam os eventos e as setas representam as pré e pós-condições da rede.

4 O JOGO DIFERENCIAL “LOBOS E CORDEIRO”

Para especificar-se formalmente qualquer jogo, necessita-se de um formalismo que permita tal especificação. A especificação formal de um sistema qualquer possibilita o seu entendimento, tornando mais clara a realidade imposta pelo sistema em questão. As características das redes de Petri já foram apresentadas anteriormente na secção 3.1.

Primeiramente, é necessário definir-se o jogo propriamente dito para após, formalizá-lo. Optou-se pelo jogo lobos e cordeiro, porque ele possui uma definição simples e de fácil compreensão, vindo ao encontro dos objetivos estipulados para este trabalho, que foram abordados em seu início.

Desta maneira, as redes de Petri foram escolhidas para que se realizasse a especificação formal do jogo lobos e cordeiro, pelo fato de permitirem uma generalização das definições deste jogo – como exemplo disto, tem-se a ampliação ou a diminuição das dimensões do espaço de jogo (o tabuleiro) ou ainda, o acréscimo ou a retirada de lobos e cordeiro (os jogadores).

Existem outros formalismos, além das redes de Petri, para especificar um jogo, dentre eles cita-se a Gramática de Grafos. Sabe-se que a gramática de grafos tem como forte característica a modelagem do paralelismo de sistemas. Já as redes de Petri, entre outras características, possibilitam modelar a alternância que é a principal definição encontrada no jogo lobos e cordeiro. Ou seja, esta alternância diz respeito às jogadas realizadas pelos jogadores, onde os mesmos jogadores alternam estas jogadas entre si (ora lobos, ora cordeiro).

Primeiramente, pensou-se na possibilidade da utilização da gramática de grafos para especificar o jogo diferencial lobos e cordeiro, dadas as suas características e os estudos que envolviam tal especificação. Pela aparência promissora da gramática de grafos iniciou-se um protótipo que pode ser visto em [SUL 2004], incorporando o jogo proposto e o ideal da especificação através da gramática de grafos.

Assim, o desenvolvimento do jogo foi sendo realizado normalmente pela especificação até então proposta. Devido ao fato da especificação por gramática de grafos ser feita através de regras, a sua utilização parecia ser apropriada para a especificação de um jogo.

A dificuldade surgiu pelo motivo de apresentar a característica do paralelismo, a qual não correspondia à definição do jogo que diz respeito à alternância entre jogadores – ou seja, eles atuam alternadamente e não paralelamente – e levou ao estudo de uma especificação que não envolvesse a gramática de grafos.

Após esta ocorrência, percebeu-se que a melhor maneira de especificar o jogo diferencial lobos e cordeiro poderia ser pelas redes de Petri, já que o ambiente do jogo, bem como seus jogadores poderiam ser facilmente transpostos para esta especificação.

4.1 A Definição do Jogo

O jogo lobos e cordeiro possui características da teoria dos jogos diferenciais, onde, por sua vez, tem-se a representação do buscador e a do evasor que participam do processo de jogo.

Especificamente, para este jogo lobos e cordeiro, há 4 “lobos” e 1 “cordeiro”. A evolução do jogo se dá em um tabuleiro de “damas” – que é quadriculado, com “casas” de duas cores alternadas, tendo 8 linha e 8 colunas e os jogadores realizam movimentos diagonais – sendo que os lobos movimentam-se em diagonal de forma independente, somente para frente; enquanto que o cordeiro pode movimentar-se em diagonal tanto para frente quanto para trás – conforme parecer mais adequado – no decorrer das jogadas. Neste jogo os lobos são ditos os buscadores e o cordeiro, o evasor.

A figura 4.1 mostra que os lobos são representados pelas “peças pretas” e o cordeiro, pela “peça branca”.

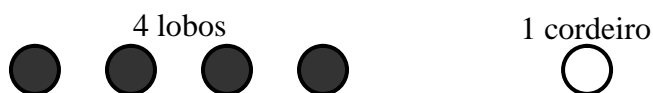


Figura 4.1: As peças que representam os lobos e o cordeiro.

Pelo fato de estar-se considerando o tabuleiro de damas – ou seja, tabuleiro de dimensão 8x8 (8 linhas e 8 colunas) – como sendo o espaço do jogo, serão identificadas a parede de saída, a parede de chegada e as paredes laterais (esquerda e direita). Estas paredes podem ser visualizadas na figura 4.2.

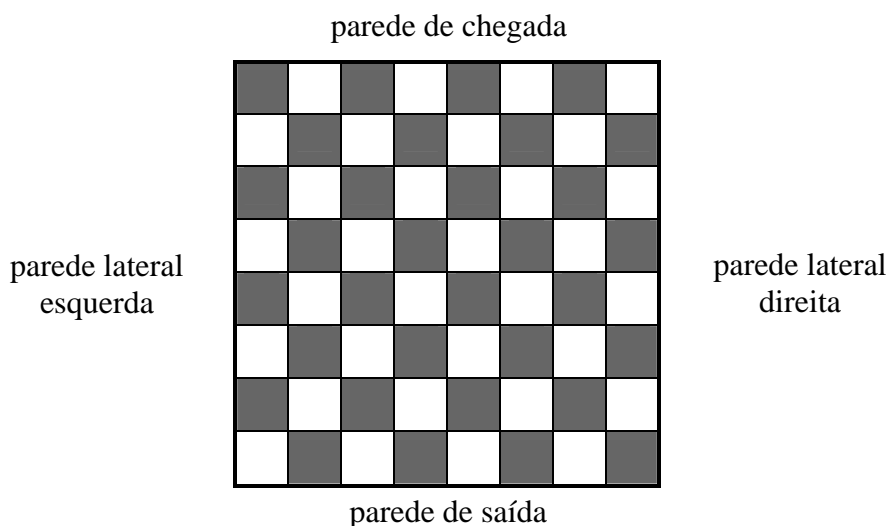


Figura 4.2: Paredes consideradas para o jogo.

Salienta-se que esta disposição de paredes conforme a figura 4.2 é atribuída ao jogador evasor (cordeiro), que parte do extremo indicado pela parede de saída. Seu

objetivo então é, percorrendo o tabuleiro, alcançar a parede de chegada sem ficar “preso” pelo jogador buscador (lobos). Se acontecer esta situação, então o evasor é o vitorioso.

Já para o caso do jogador buscador, a sua saída fica relacionada ao extremo oposto do extremo de saída do jogador evasor (segundo a situação descrita no parágrafo anterior). Este buscador assim, tem o objetivo de percorrer o tabuleiro, tentando fazer com que o evasor fique preso em alguma circunstância, que o torne um buscador vitorioso. Assim, o buscador não necessita chegar ao extremo de chegada, uma vez que ele tem por objetivo prender o evasor em alguma situação.

Para ambos jogadores alcançarem seus objetivos (independentes), eles dispõem de estratégias que são seguidas de acordo com algum critério para chegar à vitória e tais estratégias são estabelecidas pelos próprios jogadores, ou seja, conforme lhe parecer melhor efetuar as seqüências das jogadas.

Segundo as características do jogo lobos e cordeiro, os jogadores atuam de maneira independente entre si. O que ocorre é que um determinado jogador pode utilizar estratégias que são baseadas em movimentos/jogadas realizados pelo seu adversário. Por isso, o procedimento do jogo depende dessas estratégias que são formuladas por cada jogador, para aumentar suas chances de vitória.

Observa-se que a direção do movimento – direita ou esquerda é relativa às paredes direita e esquerda – ocorre em um único sentido, ou melhor, considera-se apenas a direita e a esquerda de um dos jogadores. Desta forma, para as situações descritas a partir de agora, está-se tratando da direção de movimento do cordeiro, ou seja, da parede de saída para a parede de chegada. Como base, ilustra-se o tabuleiro da figura 4.3, com a finalidade de esclarecer esta observação.

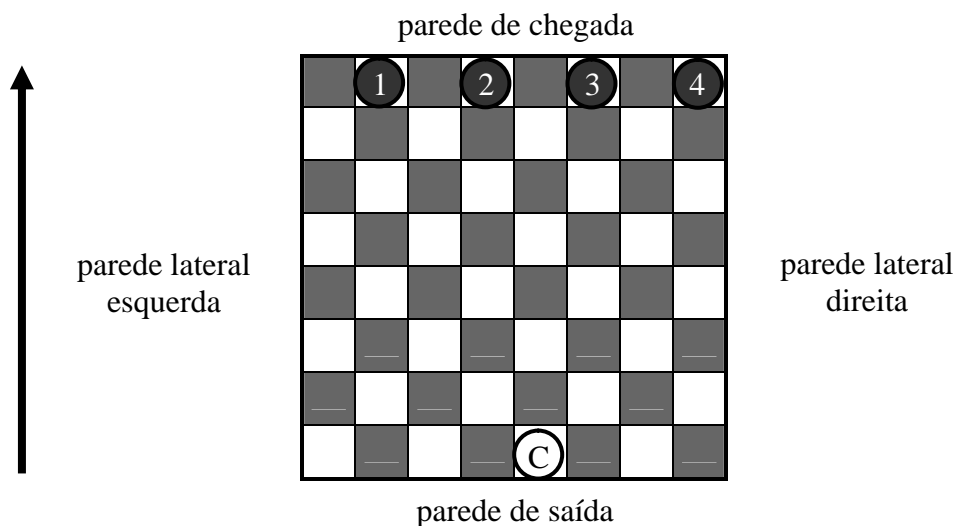


Figura 4.3: Direção dos movimentos direita e esquerda.

4.2 As Regras do Jogo

Para qualquer jogo precisa haver regras a serem seguidas, possibilitando desta maneira a fluência e o alcance do objetivo do mesmo jogo. Por isso, a seguir, são descritas as regras a serem seguidas para o jogo lobos e cordeiro.

Regras para o jogo lobos e cordeiro:

1 - O jogo lobos e cordeiro é praticado num tabuleiro de 64 “casas” (8 linhas e 8 colunas), alternativamente brancas e pretas. Porém, destas 64 casas são utilizadas somente 32 casas, correspondentes às casas brancas, as quais são as posições que podem ser ocupadas pelos jogadores ao longo do jogo.

2 - O jogo é efetuado nas casas brancas do tabuleiro.

3 - O jogo é disputado entre 2 jogadores, onde um deles representa os lobos e o outro representa o cordeiro. Os lobos são em um número de quatro (quatro peças), enquanto que o cordeiro é somente um (uma peça).

4 - O objetivo do jogo:

a) o cordeiro precisa passar pelos lobos, através do tabuleiro, saindo da parede de saída e chegando na parede de chegada, evitando ser capturado (preso, impossibilitado de se movimentar) pelos lobos ao longo de sua trajetória;

b) os lobos precisam, de alguma forma, capturar (prender, impossibilitar de se movimentar) o cordeiro, impedindo que ele chegue até a parede de chegada.

5 - Movimento dos lobos:

a) cada um dos lobos desloca-se para frente pelas diagonais esquerda ou direita. Podem somente avançar uma casa de cada vez, desde que a casa alvo (contínua) esteja livre;

b) não é permitido movimento retrógrado;

c) os lobos não podem saltar por cima do cordeiro, uma vez que o objetivo dos lobos é prender de alguma forma, pelas diagonais, o cordeiro;

d) a cada jogada, um único lobo se movimenta.

6 - Movimento do cordeiro:

a) o cordeiro desloca-se, tanto para frente quanto para trás, pelas diagonais esquerda ou direita. Ele pode somente, igualmente aos lobos, avançar uma casa de cada vez, desde que a casa alvo esteja livre;

b) é permitido movimento retrógrado;

c) o cordeiro não pode saltar por cima dos lobos, uma vez que o objetivo

do cordeiro é escapar de alguma forma, pelas diagonais, dos lobos.

7 - Captura do cordeiro:

O cordeiro estará capturado quando estiver em uma situação em que não consiga mais realizar movimentos pelo tabuleiro, pelo fato de estar cercado por paredes e/ou por lobos. Esta situação indica a vitória dos lobos.

Convém destacar que este caso das paredes é válido somente para as paredes denominadas: parede de saída, parede lateral esquerda e parede lateral direita; há a exceção da parede de chegada, que é o objetivo do cordeiro.

8 - Escape do cordeiro:

a) o cordeiro terá condições de escapar dos lobos quando puder realizar algum movimento, seja para frente ou pra trás, para a esquerda ou para a direita;

b) o cordeiro igualmente, poderá escapar quando já tiver ultrapassado os quatro lobos, no tabuleiro. Esta situação indica a vitória do cordeiro!

9 - A derrota:

O jogo é perdido pelo jogador (lobos ou cordeiro) que não conseguiu alcançar os objetivos do jogo (definido no *item 4*). Para os lobos representa o *item 8.b* e para o cordeiro representa o *item 7*.

10 - A vitória:

O jogo é vencido pelo jogador (lobos ou cordeiro) que conseguiu alcançar os objetivos do jogo (definido no *item 4*). Para o cordeiro representa o *item 8.b* e para os lobos representa o *item 7*.

11 - A situação inicial:

O início do jogo acontece com o cordeiro estando posicionado em alguma das quatro posições (ficando a sua escolha) que correspondem à parede de saída, e os lobos por sua vez, estando posicionados no extremo oposto ao cordeiro, ou seja, na parede que corresponde à chegada (nas quatro posições desta parede). Cada jogador posiciona-se nas “casas” mais externas do tabuleiro, ou melhor, exatamente na parede indicada para cada jogador.

12 - As jogadas permitidas:

a) para os lobos são permitidos somente movimentos para frente, para a esquerda ou para a direita, na diagonal; reforçando as idéias do *item 5.a* e do *item 5.b*;

b) para o cordeiro são permitidos movimentos tanto para frente quanto para trás, para a esquerda ou para a direita, na diagonal; reforçando as idéias do *item 6.a* e do *item 6.b*;

c) ambos jogadores podem somente percorrer uma “casa” de cada vez, seguindo a alternância descrita no *item 13*.

13 - A alternância de jogadas:

Os movimentos do jogo são identificados pela alternância de jogadas entre os jogadores, ou seja, cada jogador executa exatamente um movimento a cada vez que este jogador tiver a oportunidade de jogar. Para melhor entender, quando o cordeiro realiza um movimento, os lobos esperam; quando um dos lobos realiza um movimento, o cordeiro espera. É importante observar que somente um dos lobos se movimenta a cada vez, havendo uma escolha sobre qual irá se movimentar, conforme parecer melhor.

14 - Quem inicia jogando:

O critério para definir o jogador que iniciará jogando fica ao encargo dos jogadores envolvidos no jogo; desde que após o início haja a alternância das jogadas, como definido no *item 13*.

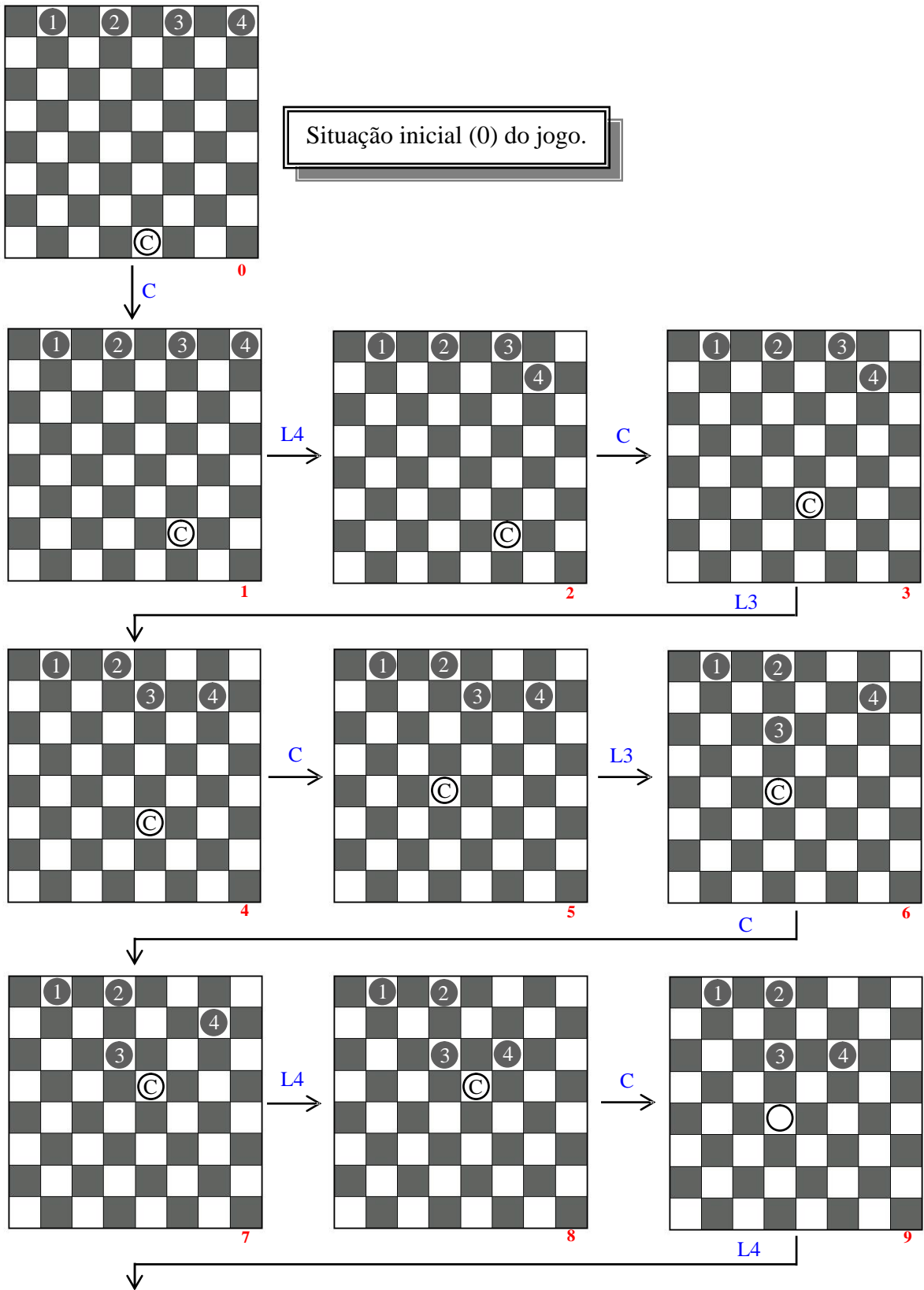
15 - A situação final:

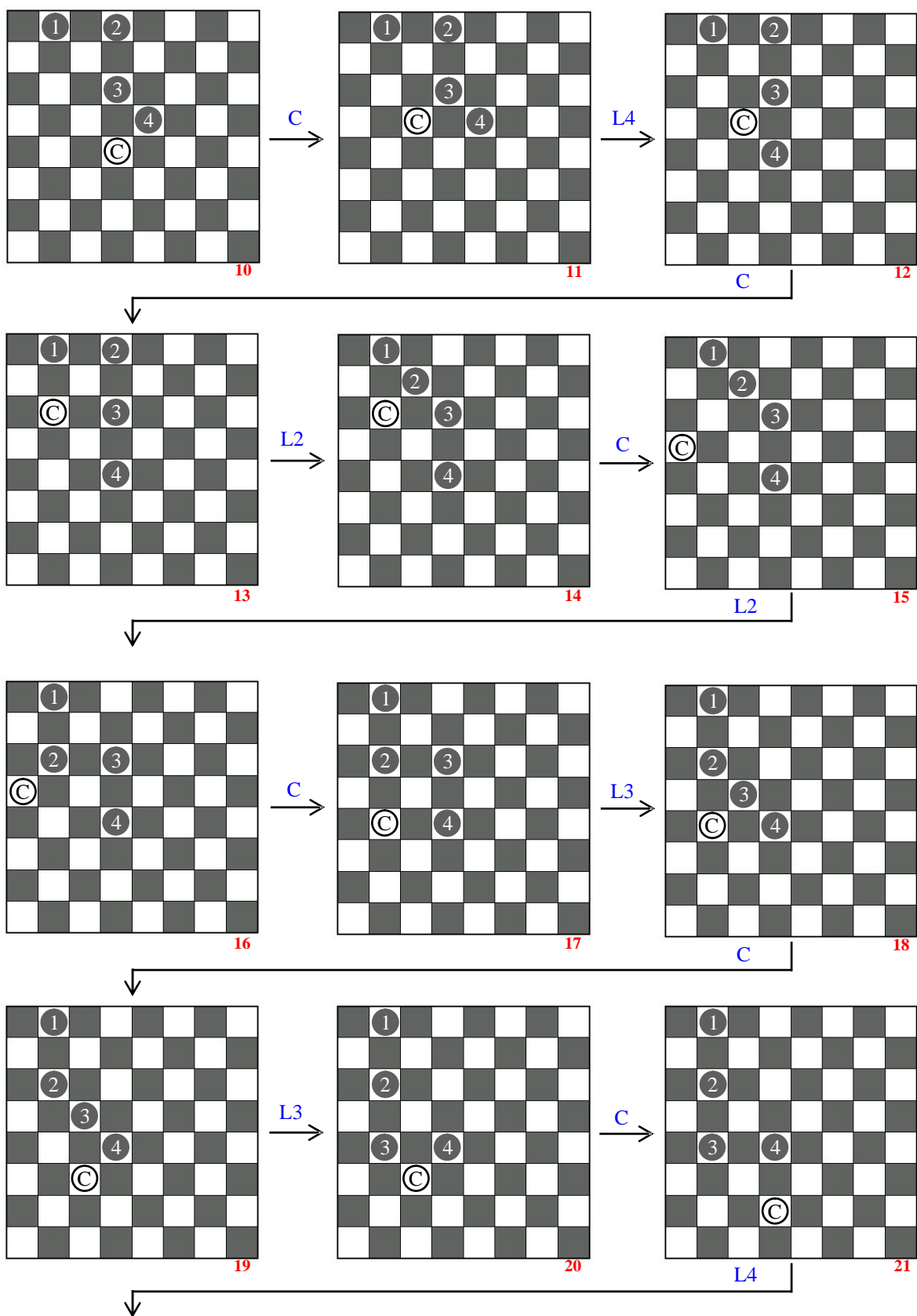
O final do jogo acontece quando ou os lobos atingiram seu objetivo ou o cordeiro atingiu seu objetivo, como no *item 4.a* ou no *item 4.b*. Ou seja, se os lobos conseguiram prender o cordeiro, de alguma forma, então eles são os vitoriosos; ou se o cordeiro conseguiu chegar à parede de chegada, então ele é o vitorioso. Obviamente, a vitória de um jogador significa a derrota do outro jogador. O empate é impossível!

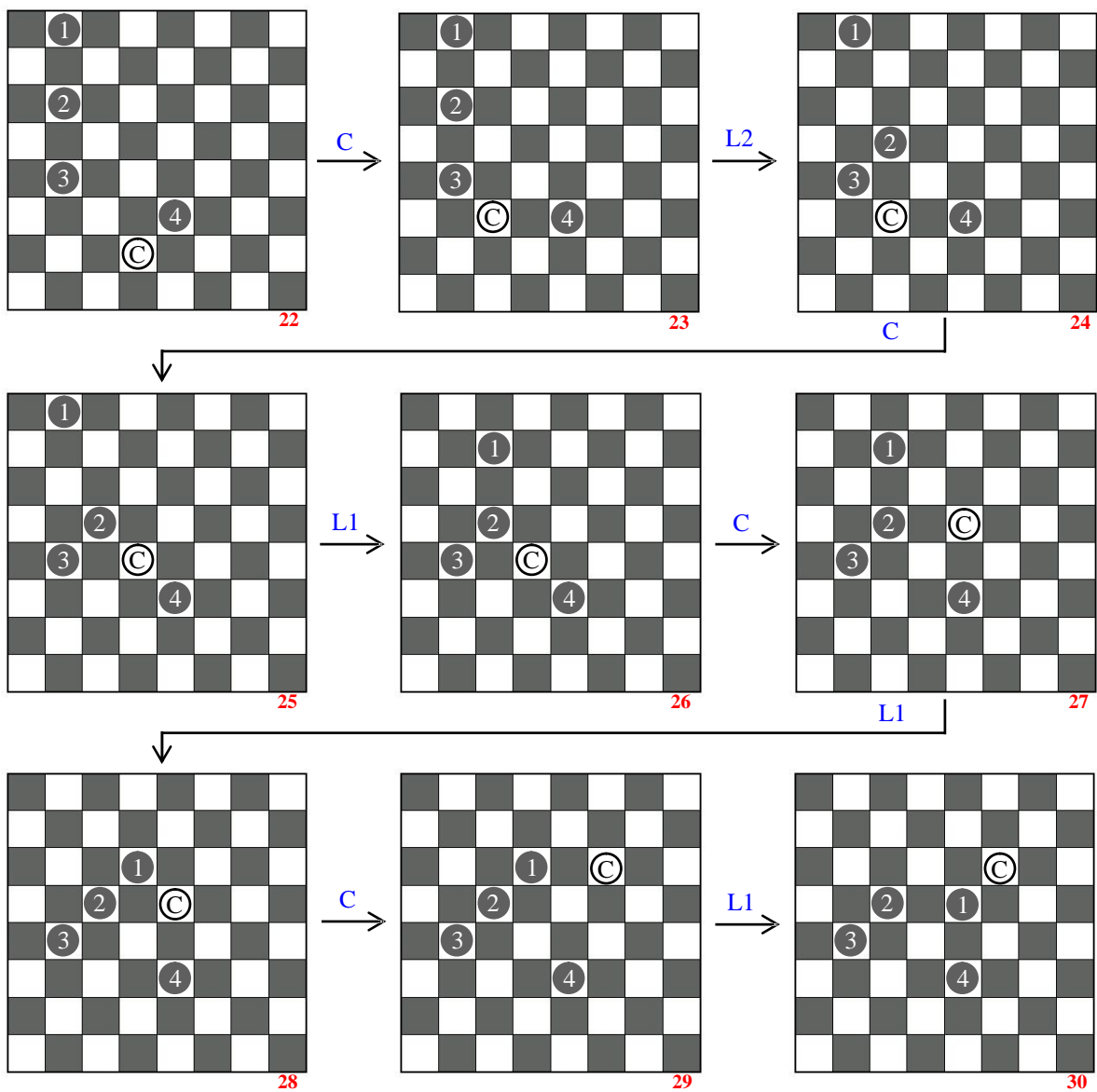
4.3 Exemplos de Partidas

Nos exemplos de partidas mostrados a seguir, utilizou-se a simbologia C, L1, L2, L3 e L4 que representa respectivamente, a jogada do cordeiro, a jogada do lobo 1, a jogada do lobo 2, a jogada do lobo 3 e a jogada do lobo 4.

Jogo A – o cordeiro é o vencedor.



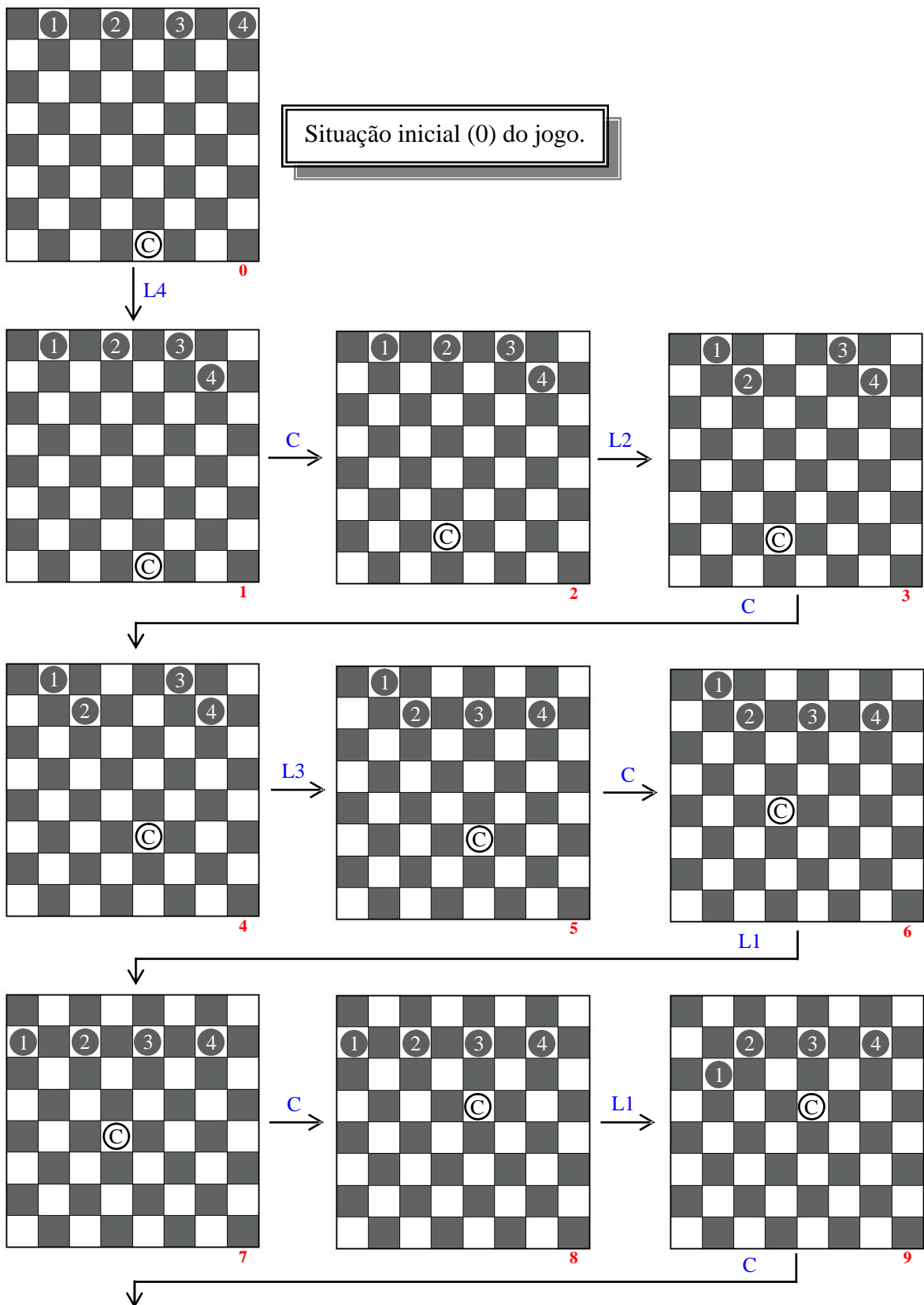


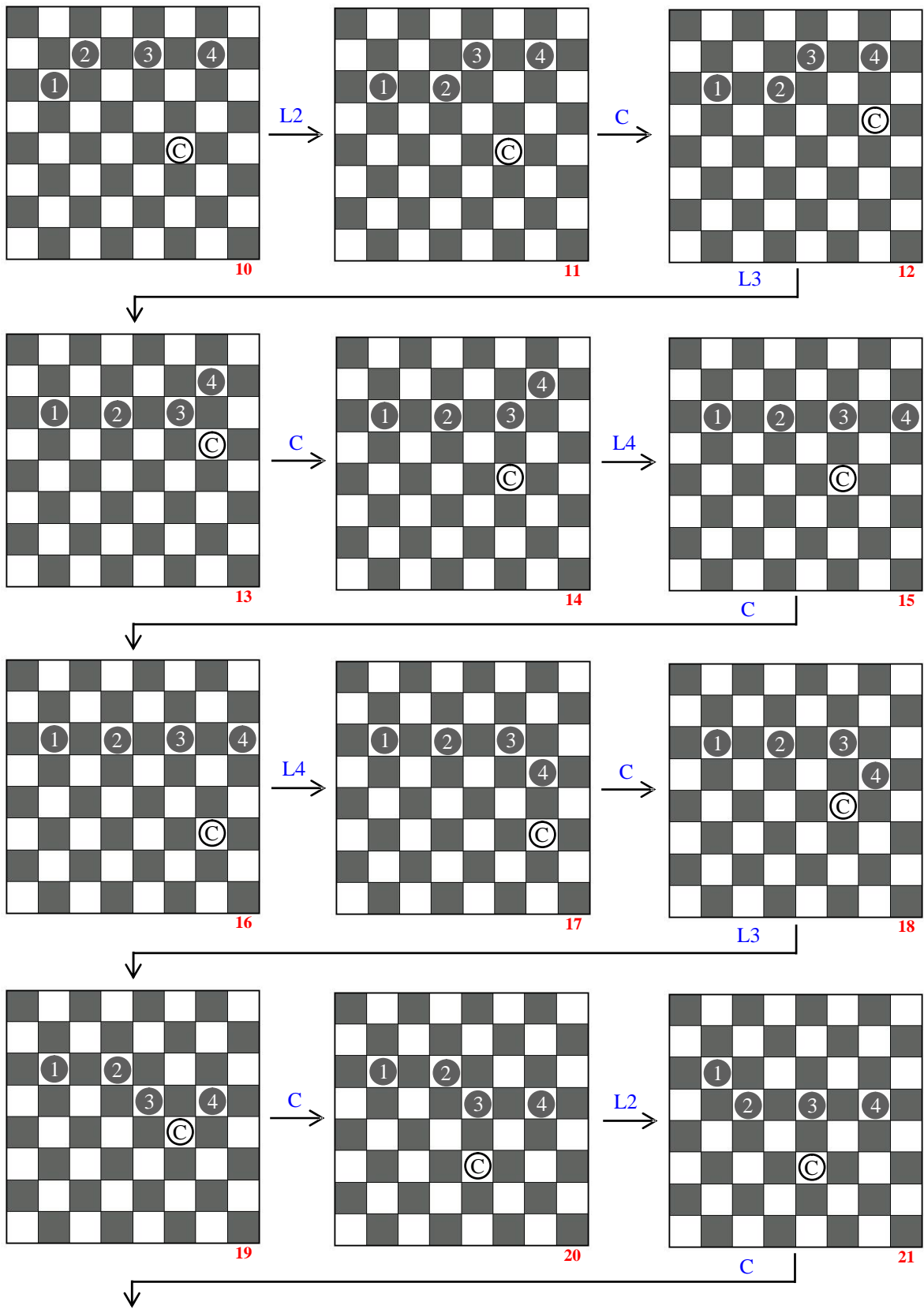


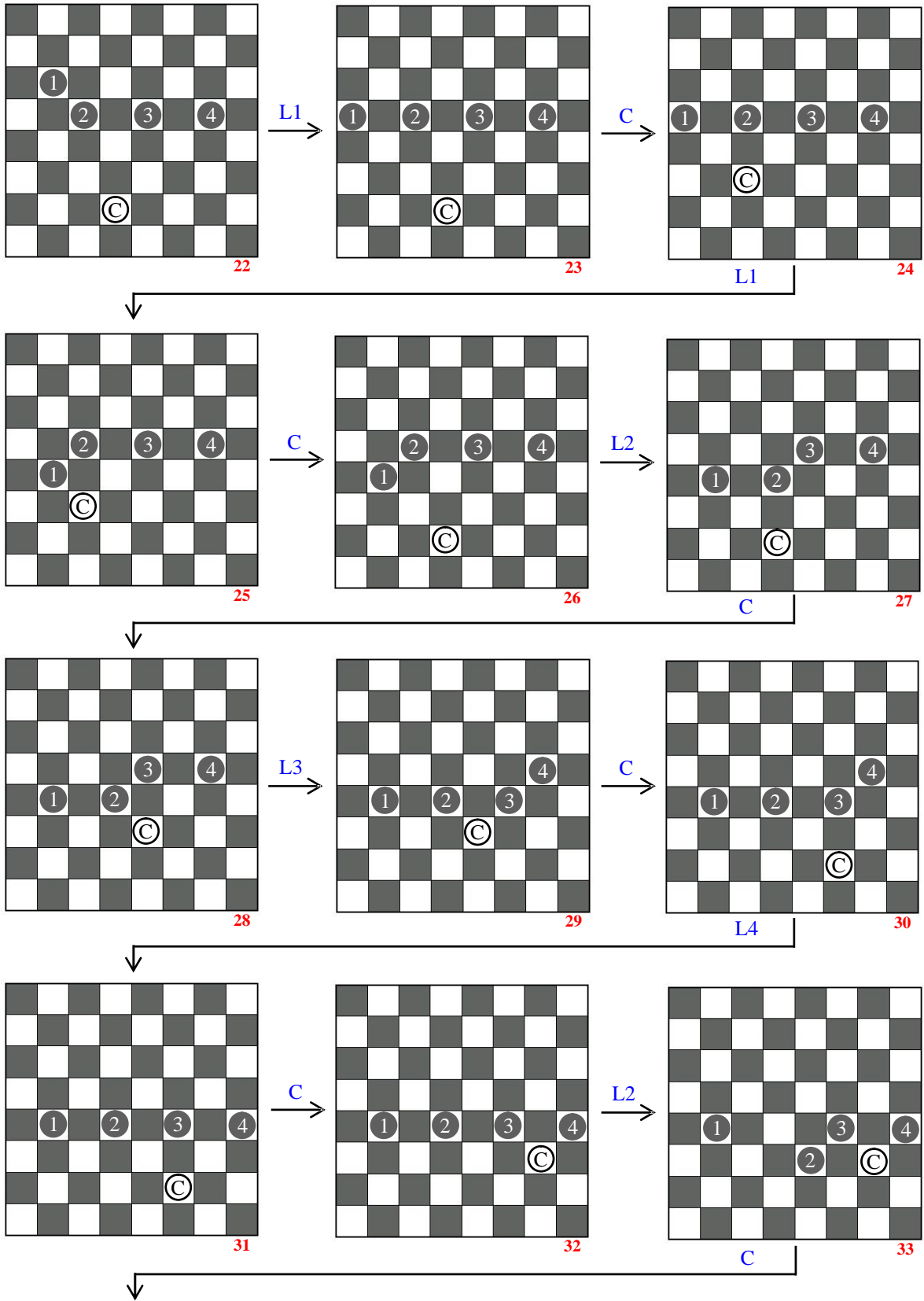
A partir desta situação (30) o cordeiro já é o vencedor!!!

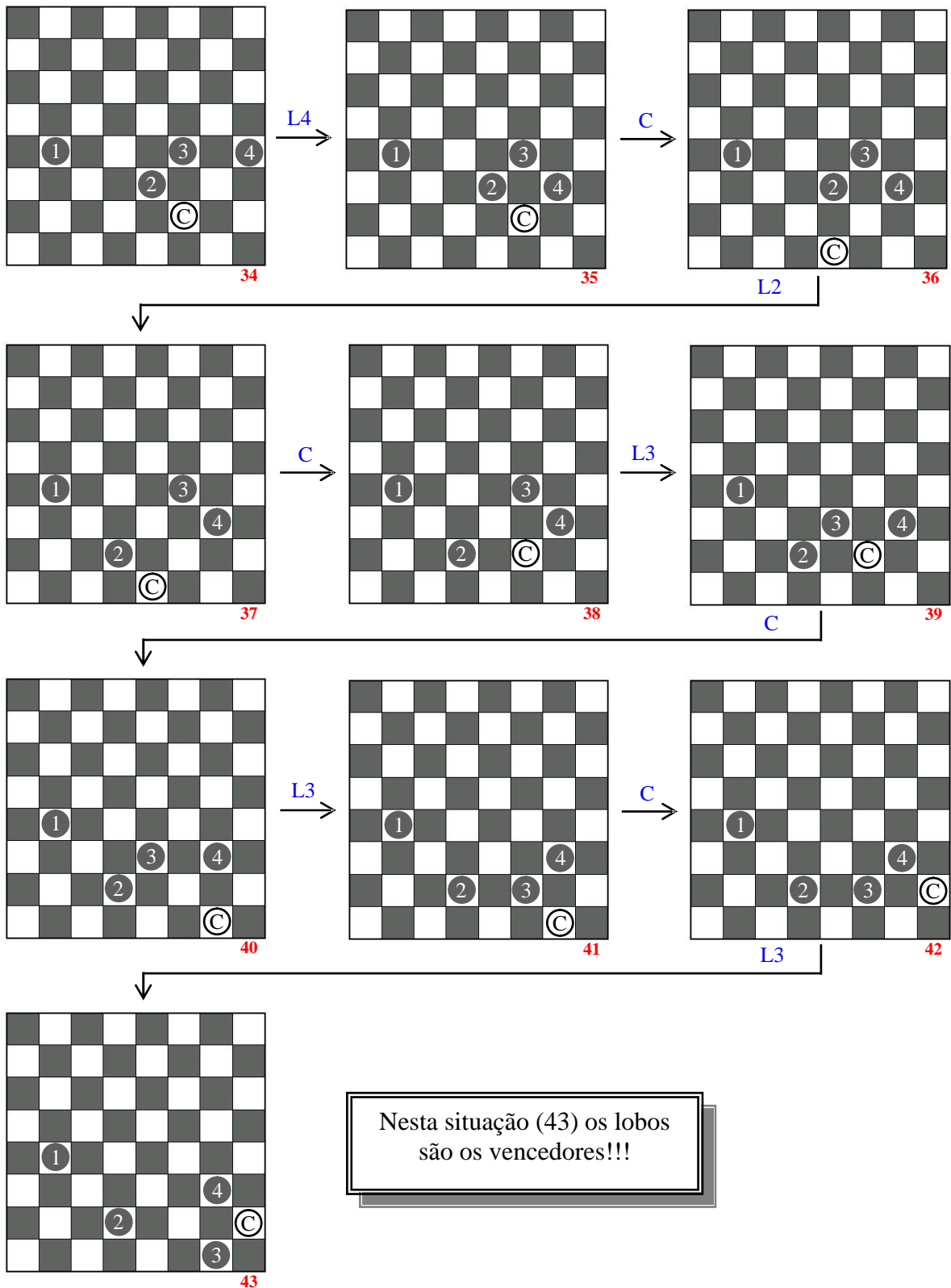
Assim, o jogo A tem a vitória do cordeiro em 30 passos alternados.

Jogo B – os lobos são os vencedores.









Assim, o jogo *B* tem a vitória dos lobos em 43 passos alternados.

4.4 As Estratégias do Jogo

Para que se possa identificar concretamente as posições dos jogadores em relação ao tabuleiro de jogo, faz-se uma representação das posições deste tabuleiro através de números atribuídos para as casas, conforme é sugerido na figura 4.4 a seguir. Isso auxiliará na definição das jogadas e na determinação das estratégias de cada um dos jogadores e facilitará a maneira de referenciar que jogador está em certa posição.

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| | 32 | | 31 | | 30 | | 29 |
| 28 | | 27 | | 26 | | 25 | |
| | 24 | | 23 | | 22 | | 21 |
| 20 | | 19 | | 18 | | 17 | |
| | 16 | | 15 | | 14 | | 13 |
| 12 | | 11 | | 10 | | 9 | |
| | 8 | | 7 | | 6 | | 5 |
| 4 | | 3 | | 2 | | 1 | |

Figura 4.4: Representação das posições dos jogadores no tabuleiro.

Pelo fato de serem utilizadas somente as casas brancas do tabuleiro para o jogo lobos e cordeiro, foram identificadas 32 posições possíveis de serem ocupadas pelos jogadores – tanto para o cordeiro quanto para os lobos – ao longo do jogo. Relembrando que se está tratando da direção de movimento do ponto de vista do cordeiro, ou seja, da parede de saída para a parede de chegada (do número 1 para o número 32).

Assim sendo, pode-se traçar um paralelo entre as situações de um jogador em relação aos demais jogadores, ou melhor, o que um jogador específico pode fazer considerando a situação dos demais jogadores envolvidos e também a posição em que este jogador se encontra. Para definir estas possibilidades teve-se base nas duas demonstrações de jogo realizadas previamente na secção 4.3 – onde em uma (jogo *A*), após 30 passos, o cordeiro vence e em outra (jogo *B*), após 43 passos, os lobos vencem. Tais possibilidades são apresentadas através da tabela 4.1 e da tabela 4.2.

Os símbolos ✓ e ✗ representam, em ambas tabela 4.1 e tabela 4.2 respectivamente, que o jogador pode ou não pode realizar um movimento na direção correspondente; as abreviações *EF*, *DF*, *EA*, *DA*, *D* e *E* indicam respectivamente, as direções de movimento esquerda frente, direita frente, esquerda atrás, direita atrás, direita e esquerda.

Tabela 4.1: Possibilidades de jogadas para os jogadores, no jogo A.

| | Cordeiro | | | | Lobo 1 | Lobo 2 | Lobo 3 | Lobo 4 | | | | |
|---------|----------|----|----|----|--------|--------|--------|--------|---|---|---|---|
| | DIREÇÕES | | | | | | | | | | | |
| JOGADAS | EF | DF | EA | DA | E | D | E | D | E | D | E | D |
| 0 | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| 1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| 2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 5 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 7 | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 8 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| 9 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 10 | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| 11 | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| 12 | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| 13 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 14 | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 15 | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 16 | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 17 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 18 | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ |
| 19 | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ |
| 20 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| 21 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 22 | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| 23 | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 24 | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 25 | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 26 | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 27 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 28 | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 29 | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 30 | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |

Tabela 4.2: Possibilidades de jogadas para os jogadores, no jogo *B*.

| JOGADAS | Cordeiro | | | | Lobo 1 | | Lobo 2 | | Lobo 3 | | Lobo 4 | |
|---------|----------|----|----|----|--------|---|--------|---|--------|---|--------|---|
| | DIREÇÕES | | | | | | | | | | | |
| | EF | DF | EA | DA | E | D | E | D | E | D | E | D |
| 0 | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| 1 | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 5 | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 7 | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 8 | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 9 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 10 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 11 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| 12 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| 13 | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| 14 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| 15 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| 16 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| 17 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| 18 | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| 19 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ |
| 20 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| 21 | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 22 | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 23 | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 24 | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 25 | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 26 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 27 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| 28 | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| 29 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ |
| 30 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| 31 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| 32 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| 33 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| 34 | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ |
| 35 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| 36 | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| 37 | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ |
| 38 | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| 39 | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| 40 | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| 41 | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| 42 | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ |
| 43 | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ |

Após ter-se chegado à tabela 4.1 e à tabela 4.2, pode-se definir algumas estratégias para as jogadas possíveis do jogo *A* e do jogo *B*, representados através de tais tabelas. Assim sendo, sabe-se que o objetivo do jogo para o cordeiro, é alcançar a parede de chegada, evitando ficar preso – seja por lobos ou por paredes – e já para os lobos o objetivo do jogo é prender o cordeiro, impedindo que ele alcance a parede de chegada.

Seguindo a idéia proposta como meta do jogo para cada um dos jogadores, é preciso que ambos jogadores tenham estratégias que possam ser seguidas por eles, até que se tenha um vencedor.

No sentido geral, as estratégias adotadas por cada um dos jogadores enquadram-se em um esquema que pode ser visto na tabela 4.3.

Tabela 4.3: Estratégias adotadas por cada um dos jogadores.

| | |
|-----------------|--|
| Cordeiro | <i>ca)</i> para que o cordeiro atinja seu objetivo de alcançar a parede de chegada sem ficar preso, é preciso que ao longo do jogo ele efetue jogadas que a princípio, o conduzam para frente e o mantenham o mais longe possível dos lobos; |
| | <i>cb)</i> pelo fato de ser necessário que o cordeiro sempre realize uma jogada quando for sua vez de fazê-la, é desejável que ele movimente-se, preferencialmente, para frente e que ele fique razoavelmente, distante dos lobos; |
| | <i>cc)</i> é indicado que o cordeiro realize jogadas que dispersem os lobos pelo tabuleiro de maneira que ao encontrar a barreira dos lobos, o cordeiro consiga ultrapassá-la, através da abertura de uma “brecha” em tal barreira (fazendo com que um(s) dos lobos saia(m) da unificação horizontal, vertical, diagonal ou circular, na qual eles se encontram), possibilitando que ele movimente-se para frente; |
| Lobos | <i>la)</i> para que os lobos atinjam seus objetivos de prender o cordeiro de alguma forma é preciso que ao longo do jogo eles efetuem jogadas que a princípio, os mantenham o mais próximo possíveis do cordeiro e também, o mais próximo possíveis entre si – formando uma espécie de “barreira” horizontal, vertical, diagonal ou circular (contendo o cordeiro) para evitar que o cordeiro consiga passar; |
| | <i>lb)</i> pelo fato de ser necessário que um dos lobos sempre realizem uma jogada quando for sua vez de fazê-la, é desejável que eles movimentem-se preferencialmente, em direção ao cordeiro e que eles fiquem razoavelmente, unidos horizontalmente, verticalmente, diagonalmente ou circularmente; |
| | <i>lc)</i> é indicado que os lobos realizem jogadas que preservem a formação de uma barreira ao longo do tabuleiro, impedindo que o cordeiro possa ultrapassá-la e fazendo com que ele movimente-se cada vez mais para trás; |

O quadro a seguir está esquematizado na forma de estratégias defensivas e ofensivas para os lobos e o cordeiro. Elas são empregadas dependendo da situação e dos jogadores em questão.

Estratégias **defensivas** e **ofensivas** dos jogadores (lobos e cordeiro)

Lobos – uma estratégia é *defensiva* para os lobos quando eles jogam no sentido de formar uma barreira horizontal, vertical, diagonal ou circular contra o cordeiro, impedindo que ele a ultrapasse. Uma estratégia é *ofensiva* para os lobos quando eles jogam no sentido de conduzir o cordeiro para trás, impedindo que ele alcance a parede de chegada (e tentando prendê-lo).

Cordeiro – uma estratégia é *defensiva* para o cordeiro quando ele joga no sentido de avançar ao longo do tabuleiro, em direção à parede de chegada (para frente) e/ou quando ele joga no sentido de retroceder ao longo do tabuleiro, em direção à parede de saída (para trás). Uma estratégia é *ofensiva* para o cordeiro quando ele joga no sentido de abrir uma “brecha” na barreira que os lobos formam, para poder ultrapassá-los.

Antagonicamente, observando a tabela 4.3, nota-se os objetivos e conseqüentemente, o intuito das jogadas entre os jogadores; isto define as estratégias que são utilizadas tanto pelo cordeiro, quanto por cada um dos lobos. As estratégias desempenham um papel fundamental para a finalização do jogo, definindo qual jogador será o vencedor. Por isso, os jogadores precisam estipular suas estratégias individuais para que consigam obter sua vitória; obviamente um deles terá estratégias dominantes e o outro portanto, terá estratégias dominadas. Como as estratégias dominantes vencem as estratégias dominadas, o resultado da aplicação destas estratégias é o jogador vencedor.

Deste modo, no jogo *A* – onde o cordeiro é o vencedor – tem-se a aplicação das estratégias dominantes do cordeiro sobre os lobos, ficando conseqüentemente, as estratégias dominadas para os lobos. Já para o caso do jogo *B* – onde os lobos são os vencedores – tem-se a aplicação das estratégias dominantes dos lobos sobre o cordeiro, ficando conseqüentemente, as estratégias dominadas para o cordeiro.

Devido ao fato de até este ponto da escrita do trabalho ter-se definido, dentre outros aspectos, o jogo diferencial lobos e cordeiro, bem como suas regras e suas estratégias, pode-se então partir para a especificação formal de tal jogo uma vez que, se possui subsídios suficientes para efetuar uma especificação significativa. A especificação formal do jogo lobos e cordeiro é realizada na seqüência da secção 4.5.

4.5 A Especificação Formal do Jogo

Para especificar formalmente o jogo lobos e cordeiro foi usada uma rede estruturada em seis camadas distintas, porém interligadas entre si (as quais foram posteriormente denominadas também, de sub-redes). Assim, obteve-se: uma camada de rede do cordeiro, uma camada de rede do controle, uma camada de rede do lobo 1, uma camada de rede do lobo 2, uma camada de rede do lobo 3 e uma camada de rede do lobo 4; futura e respectivamente, sub-rede *C*, sub-rede *Controles*, sub-rede *L1*, sub-rede *L2*, sub-rede *L3* e sub-rede *L4*. Lembra-se que foi definida uma camada de rede para cada

um dos lobos, pois eles movimentam-se de forma independente, isto é, quando o jogador que representa os lobos tem a vez para jogar, somente um dos lobos pode se movimentar naquela vez correspondente.

Foi necessário este desmembramento em seis camadas, pois caso contrário a rede tornar-se-ia muito complexa, dificultando a sua análise e o seu entendimento.

Esta especificação proposta para o jogo em questão pode ser acompanhada detalhadamente, a partir de agora.

O jogo diferencial lobos e cordeiro foi especificado formalmente através de redes de Petri condição/evento (do inglês *Condition/Event Petri Nets*), conforme definido anteriormente na secção 3.2, pois os lugares (condições) da rede podem somente acumular um único *token* a cada instante do jogo.

Lembra-se que a definição de rede de Petri condição/evento foi adaptada para o jogo diferencial lobos e cordeiro e assim, chegou-se ao seguinte esquema para o jogo proposto:

$RPJ = (C, E, pre, pos, M_0)$, onde
 C é o conjunto de condições da rede,
 E é o conjunto de eventos da rede,
 $pre: C \rightarrow E$ e $pos: E \rightarrow C$ são funções representando respectivamente, a pré e a pós-condições a serem satisfeitas para a ocorrência dos eventos da rede, e
 $M_0 \subseteq C$ é a marcação inicial da rede.

A definição da rede de Petri condição/evento definida por [REI 85] – apresentada anteriormente na secção 3.2, sofreu uma adaptação para o jogo diferencial lobos e cordeiro, facilitando o entendimento do comportamento da rede em relação ao jogo proposto.

Observa-se que conjunto M_0 foi acrescentado à definição da rede de Petri para identificar o estado inicial do jogo. Da mesma forma os *tokens* da rede são identificados por *token* jogador (ou um dos lobos, ou o cordeiro), *token* vez e *token* livre. Ao passo que a descrição for sendo realizada, cada um destes *tokens* é devidamente apresentado.

Assim, com base na definição de rede de Petri condição/evento, descreve-se nas próximas subsecções a especificação formal do jogo diferencial lobos e cordeiro, ou seja a rede RPJ (abreviação para Rede de Petri do Jogo) propriamente dita.

4.5.1 Especificação Inicial

O conjunto de condições da rede RPJ é formado por: $C = \{c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28, c29, c30, c32, c33, c34\}$. Esta definição de condições, de $c1$ até $c32$ – vistos na figura 4.5 e igualmente, na figura 4.6, segue a representação das posições encontradas no tabuleiro do jogo proposto. Já as condições $c33$ e $c34$ – vistos na figura 4.7 – são as condições que controlam a vez de jogar para cada jogador (ou lobos ou cordeiro). Com isto, respeita-se o princípio da alternância entre jogadores.

O conjunto de eventos da rede RPJ é formado por: $E = \{e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15, e16, e17, e18, e19, e20, e21, e22, e23, e24, e25, e26,$

$e_{27}, e_{28}, e_{29}, e_{30}, e_{31}, e_{32}, e_{33}, e_{34}, e_{35}, e_{36}, e_{37}, e_{38}, e_{39}, e_{40}, e_{41}, e_{42}, e_{43}, e_{44}, e_{45}, e_{46}, e_{47}, e_{48}, e_{49}, e_{50}, e_{51}, e_{52}, e_{53}, e_{54}, e_{55}, e_{56}, e_{57}, e_{58}, e_{59}, e_{60}, e_{61}, e_{62}, e_{63}, e_{64}, e_{65}, e_{66}, e_{67}, e_{68}, e_{69}, e_{70}, e_{71}, e_{72}, e_{73}, e_{74}, e_{75}, e_{76}, e_{77}, e_{78}, e_{79}, e_{80}, e_{81}, e_{82}, e_{83}, e_{84}, e_{85}, e_{88}, e_{89}, e_{92}, e_{93}, e_{96}, e_{97}, e_2', e_3', e_6', e_7', e_{10}', e_{11}', e_{14}', e_{16}', e_{17}', e_{20}', e_{21}', e_{24}', e_{25}', e_{28}', e_{30}', e_{31}', e_{34}', e_{35}', e_{38}', e_{39}', e_{42}', e_{44}', e_{45}', e_{48}', e_{49}', e_{52}', e_{53}', e_{56}', e_{58}', e_{59}', e_{62}', e_{63}', e_{66}', e_{67}', e_{70}', e_{72}', e_{73}', e_{76}', e_{77}', e_{80}', e_{81}', e_{84}', e_{86}', e_{87}', e_{90}', e_{91}', e_{94}', e_{95}', e_{98}'$. Os eventos e_1 até $e_{85}, e_{88}, e_{89}, e_{92}, e_{93}, e_{96}$ e e_{97} – encontradas na figura 4.5, representam os eventos que o jogador cordeiro pode realizar ao longo do tabuleiro do jogo, lembrando que o cordeiro pode movimentar-se tanto para frente quanto para trás. Os eventos $e_2', e_3', e_6', e_7', e_{10}', e_{11}', e_{14}', e_{16}', e_{17}', e_{20}', e_{21}', e_{24}', e_{25}', e_{28}', e_{30}', e_{31}', e_{34}', e_{35}', e_{38}', e_{39}', e_{42}', e_{44}', e_{45}', e_{48}', e_{49}', e_{52}', e_{53}', e_{56}', e_{58}', e_{59}', e_{62}', e_{63}', e_{66}', e_{67}', e_{70}', e_{72}', e_{73}', e_{76}', e_{77}', e_{80}', e_{81}', e_{84}', e_{86}', e_{87}', e_{90}', e_{91}', e_{94}', e_{95}'$ e e_{98}' – encontradas na figura 4.6, representam os eventos que o jogador lobo pode realizar ao longo do tabuleiro do jogo, lembrando que os lobos podem movimentar-se somente para frente.

As funções de pré-condição (também conhecidas como origem), $pre:C \rightarrow E$, e de pós-condição (também conhecidas como destino), $pos:E \rightarrow C$, a serem satisfeitas para a ocorrência dos eventos da rede RPJ são representadas respectivamente, por:

$$\begin{aligned}
pre(e_1) &= \{c_1\}, & pos(e_1) &= \{c_5\}, \\
pre(e_2) &= \{c_5\}, & pos(e_2) &= \{c_1\}, \\
pre(e_3) &= \{c_6\}, & pos(e_3) &= \{c_1\}, \\
pre(e_4) &= \{c_1\}, & pos(e_4) &= \{c_6\}, \\
pre(e_5) &= \{c_2\}, & pos(e_5) &= \{c_6\}, \\
pre(e_6) &= \{c_6\}, & pos(e_6) &= \{c_2\}, \\
pre(e_7) &= \{c_7\}, & pos(e_7) &= \{c_2\}, \\
pre(e_8) &= \{c_2\}, & pos(e_8) &= \{c_7\}, \\
pre(e_9) &= \{c_3\}, & pos(e_9) &= \{c_7\}, \\
pre(e_{10}) &= \{c_7\}, & pos(e_{10}) &= \{c_3\}, \\
pre(e_{11}) &= \{c_8\}, & pos(e_{11}) &= \{c_3\}, \\
pre(e_{12}) &= \{c_3\}, & pos(e_{12}) &= \{c_8\}, \\
pre(e_{13}) &= \{c_4\}, & pos(e_{13}) &= \{c_8\}, \\
pre(e_{14}) &= \{c_8\}, & pos(e_{14}) &= \{c_4\}, \\
pre(e_{15}) &= \{c_5\}, & pos(e_{15}) &= \{c_9\}, \\
pre(e_{16}) &= \{c_9\}, & pos(e_{16}) &= \{c_5\}, \\
pre(e_{17}) &= \{c_9\}, & pos(e_{17}) &= \{c_6\}, \\
pre(e_{18}) &= \{c_6\}, & pos(e_{18}) &= \{c_9\}, \\
pre(e_{19}) &= \{c_6\}, & pos(e_{19}) &= \{c_{10}\}, \\
pre(e_{20}) &= \{c_{10}\}, & pos(e_{20}) &= \{c_6\}, \\
pre(e_{21}) &= \{c_{10}\}, & pos(e_{21}) &= \{c_7\}, \\
pre(e_{22}) &= \{c_7\}, & pos(e_{22}) &= \{c_{10}\}, \\
pre(e_{23}) &= \{c_7\}, & pos(e_{23}) &= \{c_{11}\}, \\
pre(e_{24}) &= \{c_{11}\}, & pos(e_{24}) &= \{c_7\}, \\
pre(e_{25}) &= \{c_{11}\}, & pos(e_{25}) &= \{c_8\}, \\
pre(e_{26}) &= \{c_8\}, & pos(e_{26}) &= \{c_{11}\}, \\
pre(e_{27}) &= \{c_8\}, & pos(e_{27}) &= \{c_{12}\}, \\
pre(e_{28}) &= \{c_{12}\}, & pos(e_{28}) &= \{c_8\}, \\
pre(e_{29}) &= \{c_9\}, & pos(e_{29}) &= \{c_{13}\},
\end{aligned}$$

$pre(e30) = \{c13\}, pos(e30) = \{c9\},$
 $pre(e31) = \{c14\}, pos(e31) = \{c9\},$
 $pre(e32) = \{c9\}, pos(e32) = \{c14\},$
 $pre(e33) = \{c10\}, pos(e33) = \{c14\},$
 $pre(e34) = \{c14\}, pos(e34) = \{c10\},$
 $pre(e35) = \{c15\}, pos(e35) = \{c10\},$
 $pre(e36) = \{c10\}, pos(e36) = \{c15\},$
 $pre(e37) = \{c11\}, pos(e37) = \{c15\},$
 $pre(e38) = \{c15\}, pos(e38) = \{c11\},$
 $pre(e39) = \{c16\}, pos(e39) = \{c11\},$
 $pre(e40) = \{c11\}, pos(e40) = \{c16\},$
 $pre(e41) = \{c12\}, pos(e41) = \{c16\},$
 $pre(e42) = \{c16\}, pos(e42) = \{c12\},$
 $pre(e43) = \{c13\}, pos(e43) = \{c17\},$
 $pre(e44) = \{c17\}, pos(e44) = \{c13\},$
 $pre(e45) = \{c17\}, pos(e45) = \{c14\},$
 $pre(e46) = \{c14\}, pos(e46) = \{c17\},$
 $pre(e47) = \{c14\}, pos(e47) = \{c18\},$
 $pre(e48) = \{c18\}, pos(e48) = \{c14\},$
 $pre(e49) = \{c18\}, pos(e49) = \{c15\},$
 $pre(e50) = \{c15\}, pos(e50) = \{c18\},$
 $pre(e51) = \{c15\}, pos(e51) = \{c19\},$
 $pre(e52) = \{c19\}, pos(e52) = \{c15\},$
 $pre(e53) = \{c19\}, pos(e53) = \{c16\},$
 $pre(e54) = \{c16\}, pos(e54) = \{c19\},$
 $pre(e55) = \{c16\}, pos(e55) = \{c20\},$
 $pre(e56) = \{c20\}, pos(e56) = \{c16\},$
 $pre(e57) = \{c17\}, pos(e57) = \{c21\},$
 $pre(e58) = \{c21\}, pos(e58) = \{c17\},$
 $pre(e59) = \{c22\}, pos(e59) = \{c17\},$
 $pre(e60) = \{c17\}, pos(e60) = \{c22\},$
 $pre(e61) = \{c18\}, pos(e61) = \{c22\},$
 $pre(e62) = \{c22\}, pos(e62) = \{c18\},$
 $pre(e63) = \{c23\}, pos(e63) = \{c18\},$
 $pre(e64) = \{c18\}, pos(e64) = \{c23\},$
 $pre(e65) = \{c19\}, pos(e65) = \{c23\},$
 $pre(e66) = \{c23\}, pos(e66) = \{c19\},$
 $pre(e67) = \{c24\}, pos(e67) = \{c19\},$
 $pre(e68) = \{c19\}, pos(e68) = \{c24\},$
 $pre(e69) = \{c20\}, pos(e69) = \{c24\},$
 $pre(e70) = \{c24\}, pos(e70) = \{c20\},$
 $pre(e71) = \{c21\}, pos(e71) = \{c25\},$
 $pre(e72) = \{c25\}, pos(e72) = \{c21\},$
 $pre(e73) = \{c25\}, pos(e73) = \{c22\},$
 $pre(e74) = \{c22\}, pos(e74) = \{c25\},$
 $pre(e75) = \{c22\}, pos(e75) = \{c26\},$
 $pre(e76) = \{c26\}, pos(e76) = \{c22\},$
 $pre(e77) = \{c26\}, pos(e77) = \{c23\},$
 $pre(e78) = \{c23\}, pos(e78) = \{c26\},$
 $pre(e79) = \{c23\}, pos(e79) = \{c27\},$

$pre(e80) = \{c27\}, pos(e80) = \{c23\},$
 $pre(e81) = \{c27\}, pos(e81) = \{c24\},$
 $pre(e82) = \{c24\}, pos(e82) = \{c27\},$
 $pre(e83) = \{c24\}, pos(e83) = \{c28\},$
 $pre(e84) = \{c28\}, pos(e84) = \{c24\},$
 $pre(e85) = \{c25\}, pos(e85) = \{c29\},$
 $pre(e88) = \{c25\}, pos(e88) = \{c30\},$
 $pre(e89) = \{c26\}, pos(e89) = \{c30\},$
 $pre(e92) = \{c26\}, pos(e92) = \{c31\},$
 $pre(e93) = \{c27\}, pos(e93) = \{c31\},$
 $pre(e96) = \{c27\}, pos(e96) = \{c32\},$
 $pre(e97) = \{c28\}, pos(e97) = \{c32\},$
 $pre(e2') = \{c5\}, pos(e2') = \{c1\},$
 $pre(e3') = \{c6\}, pos(e3') = \{c1\},$
 $pre(e6') = \{c6\}, pos(e6') = \{c2\},$
 $pre(e7') = \{c7\}, pos(e7') = \{c2\},$
 $pre(e10') = \{c7\}, pos(e10') = \{c3\},$
 $pre(e11') = \{c8\}, pos(e11') = \{c3\},$
 $pre(e14') = \{c8\}, pos(e14') = \{c4\},$
 $pre(e16') = \{c9\}, pos(e16') = \{c5\},$
 $pre(e17') = \{c9\}, pos(e17') = \{c6\},$
 $pre(e20') = \{c10\}, pos(e20') = \{c6\},$
 $pre(e21') = \{c10\}, pos(e21') = \{c7\},$
 $pre(e24') = \{c11\}, pos(e24') = \{c7\},$
 $pre(e25') = \{c11\}, pos(e25') = \{c8\},$
 $pre(e28') = \{c12\}, pos(e28') = \{c8\},$
 $pre(e30') = \{c13\}, pos(e30') = \{c9\},$
 $pre(e31') = \{c14\}, pos(e31') = \{c9\},$
 $pre(e34') = \{c14\}, pos(e34') = \{c10\},$
 $pre(e35') = \{c15\}, pos(e35') = \{c10\},$
 $pre(e38') = \{c15\}, pos(e38') = \{c11\},$
 $pre(e39') = \{c16\}, pos(e39') = \{c11\},$
 $pre(e42') = \{c16\}, pos(e42') = \{c12\},$
 $pre(e44') = \{c17\}, pos(e44') = \{c13\},$
 $pre(e45') = \{c17\}, pos(e45') = \{c14\},$
 $pre(e48') = \{c18\}, pos(e48') = \{c14\},$
 $pre(e49') = \{c18\}, pos(e49') = \{c15\},$
 $pre(e52') = \{c19\}, pos(e52') = \{c15\},$
 $pre(e53') = \{c19\}, pos(e53') = \{c16\},$
 $pre(e56') = \{c20\}, pos(e56') = \{c16\},$
 $pre(e58') = \{c21\}, pos(e58') = \{c17\},$
 $pre(e59') = \{c22\}, pos(e59') = \{c17\},$
 $pre(e62') = \{c22\}, pos(e62') = \{c18\},$
 $pre(e63') = \{c23\}, pos(e63') = \{c18\},$
 $pre(e66') = \{c23\}, pos(e66') = \{c19\},$
 $pre(e67') = \{c24\}, pos(e67') = \{c19\},$
 $pre(e70') = \{c24\}, pos(e70') = \{c20\},$
 $pre(e72') = \{c25\}, pos(e72') = \{c21\},$
 $pre(e73') = \{c25\}, pos(e73') = \{c22\},$
 $pre(e76') = \{c26\}, pos(e76') = \{c22\},$

$$\begin{aligned}
pre(e77') &= \{c26\}, & pos(e77') &= \{c23\}, \\
pre(e80') &= \{c27\}, & pos(e80') &= \{c23\}, \\
pre(e81') &= \{c27\}, & pos(e81') &= \{c24\}, \\
pre(e84') &= \{c28\}, & pos(e84') &= \{c24\}, \\
pre(e86') &= \{c29\}, & pos(e86') &= \{c25\}, \\
pre(e87') &= \{c30\}, & pos(e87') &= \{c25\}, \\
pre(e90') &= \{c30\}, & pos(e90') &= \{c26\}, \\
pre(e91') &= \{c31\}, & pos(e91') &= \{c26\}, \\
pre(e94') &= \{c31\}, & pos(e94') &= \{c27\}, \\
pre(e95') &= \{c32\}, & pos(e95') &= \{c27\}, \\
pre(e98') &= \{c32\}, & pos(e98') &= \{c28\}.
\end{aligned}$$

Estas funções são responsáveis por satisfazerem as condições para a ocorrência dos eventos que irão movimentar os *tokens* pela rede de Petri, levando estes *tokens* de uma condição para outra, conforme o evento determinar. Nota-se que a convenção utilizada por [REI 85] – “ e ” e “ e' ” – foi adaptada respectivamente, para “*pre*” e “*pos*”.

Finalizando a definição da rede de Petri *RPJ*, define-se a sua marcação inicial: $M_0 = \{c2, c29, c30, c31, c32\}$, por exemplo. Este conjunto M_0 indica as condições ocupadas inicialmente na rede, pelos *tokens* que representam cada um dos jogadores; observa-se a figura 4.5 e a figura 4.6. Sabe-se que há quatro possibilidades de marcação inicial, pois o cordeiro pode iniciar o jogo em quaisquer das quatro posições da parede de saída, ou seja, em quaisquer das quatro condições na rede indicadas por $c1, c2, c3$ e $c4$.

Esta marcação inicial $M_0 = \{c2, c29, c30, c31, c32\}$ diz respeito à rede do cordeiro e à rede dos lobos. Falta no entanto, levar em consideração a rede da figura 4.7 que representa a vez de jogar e as condições livres. Assim, tendo-se a junção das três redes, o conjunto M_0 é formado então, por: $M_0 = \{c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28, c29, c30, c32, c33\}$.

Nesta rede precisa haver um *token* que represente o cordeiro, ilustrado na figura 4.5 pelo ponto que ocupa a condição $c2$. Igualmente, nesta rede precisa haver os *tokens* que representem cada um dos lobos (1, 2, 3 e 4), ilustrados respectivamente, na figura 4.6 pelos pontos que ocupam as condições $c32, c31, c30$ e $c29$.

Para que se possa controlar a alternância entre os jogadores é necessário haver um *token* nesta rede, que represente o jogador que está com a vez para jogar – ou lobos, ou cordeiro – apresentado na figura 4.7 pelo ponto que ocupa a condição $c33$. Finalmente, para controlar as condições ocupadas na rede (as posições ocupadas por cada um dos jogadores no tabuleiro do jogo) é necessário haver os *tokens* nesta rede, que representem as condições livres na rede (as posições que cada um dos jogadores pode assumir no tabuleiro do jogo) – apresentados também, na figura 4.7 pelos pontos que ocupam as condições $c1$ e $c3$ até $c28$. O número de *tokens* necessário para controlar as condições livres na rede, pode ser generalizado através do número de posições do tabuleiro do jogo descontado do número de jogadores, ou seja, 32 posições menos 5 jogadores, o que resultam 27 posições livres; assim, 27 *tokens*.

As figuras seguintes, figura 4.5, figura 4.6 e figura 4.7, apresentam as condições, os eventos e os *tokens* da definição para a rede de Petri *RPJ*.

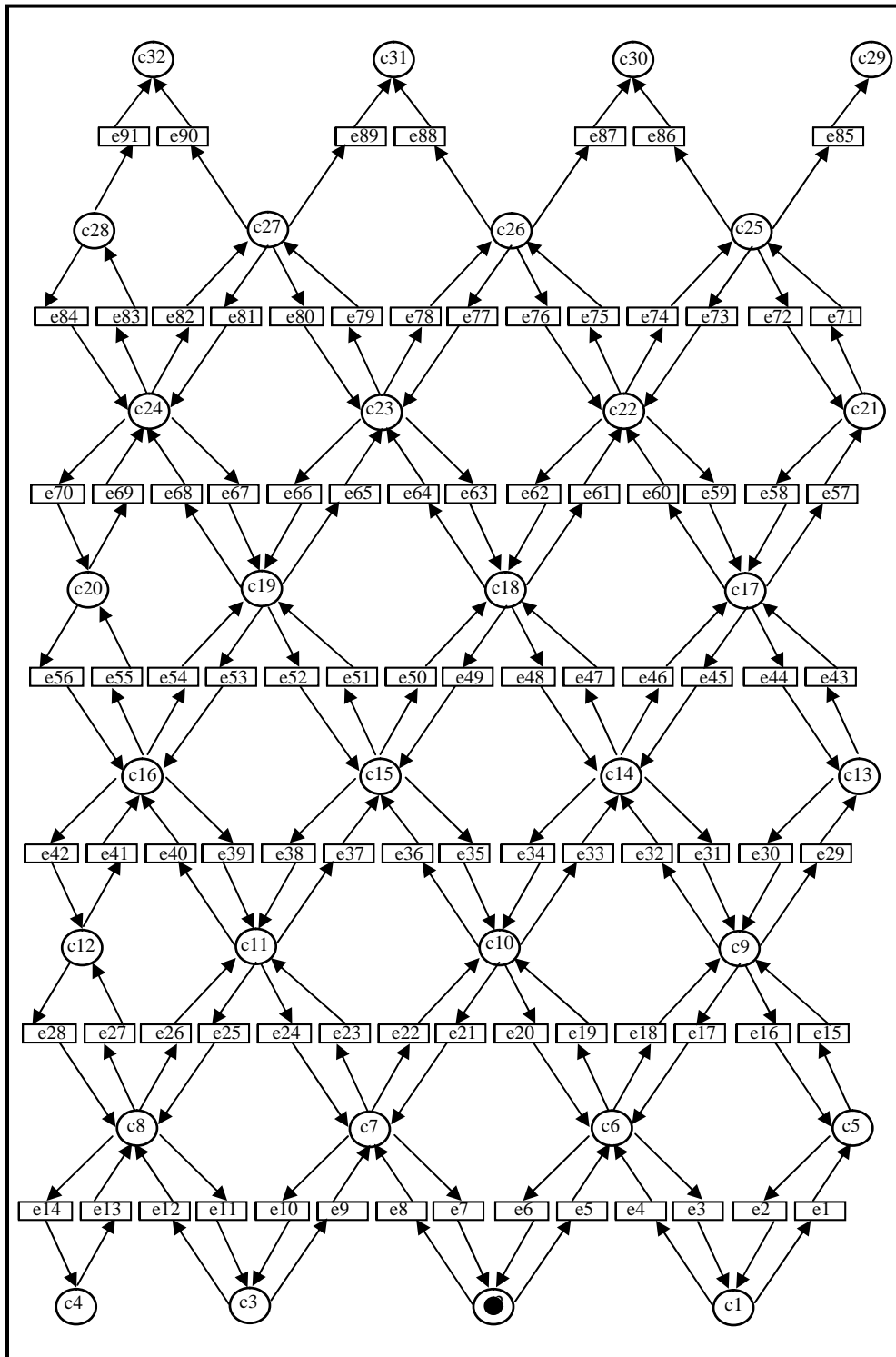


Figura 4.5: Rede de Petri representando as condições, os eventos e o *token* do cordeiro.

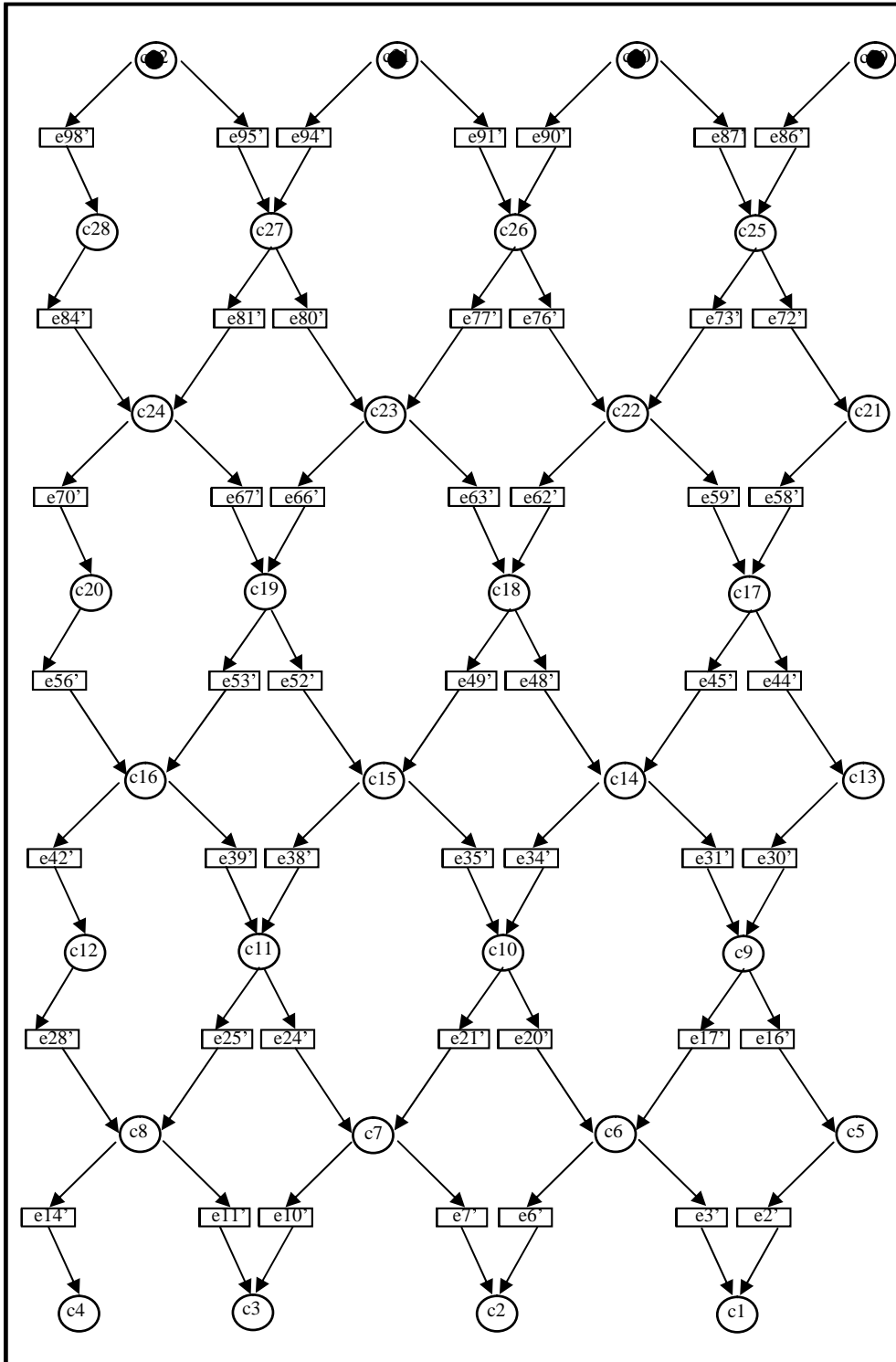


Figura 4.6: Rede de Petri representando as condições, os eventos e os *tokens* dos lobos.

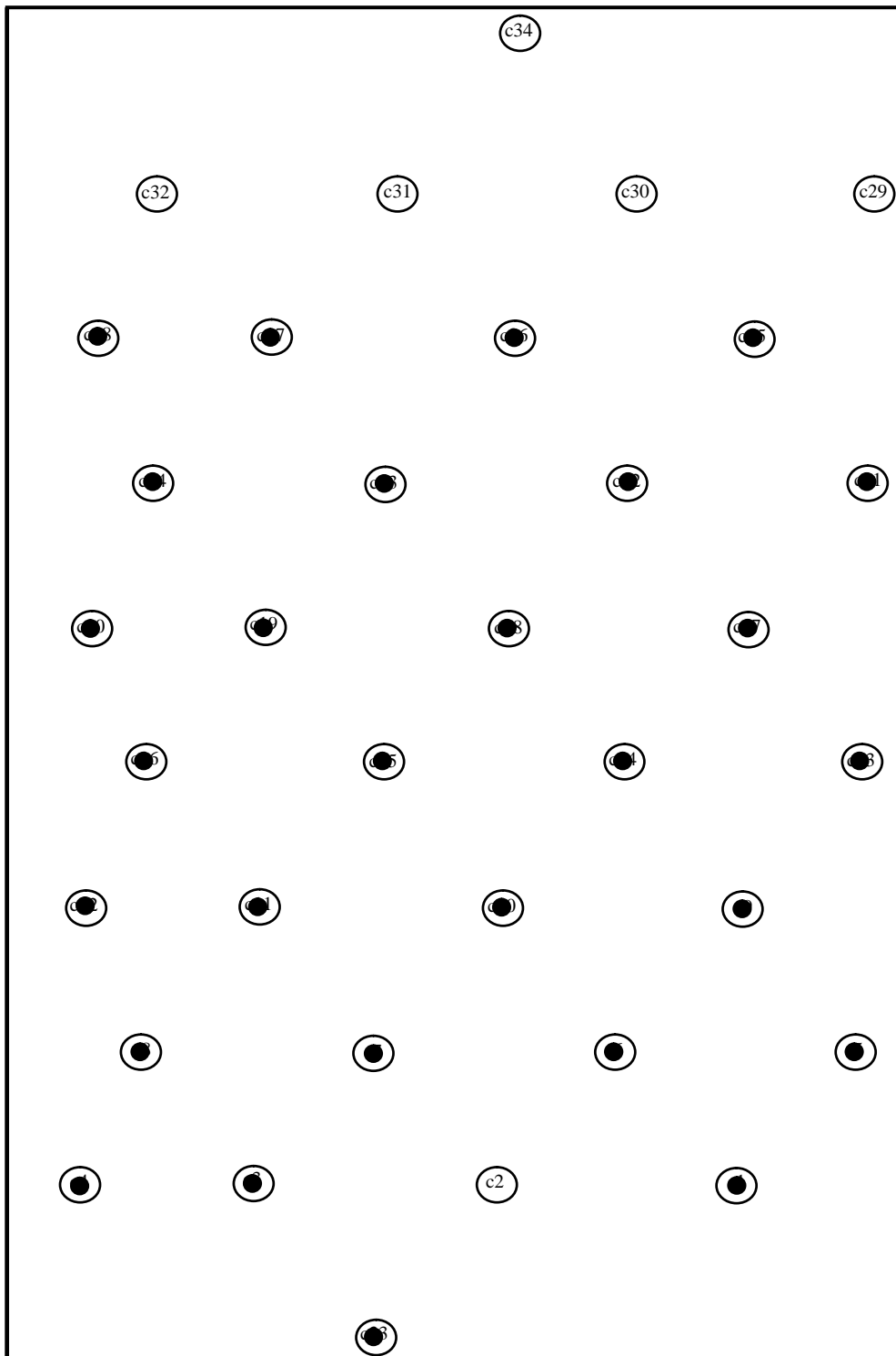


Figura 4.7: Rede de Petri representando as condições e os *tokens* da vez para jogar e das condições livres.

4.5.2 Características da Especificação

Uma vez que se pôde especificar formalmente o jogo lobos e cordeiro, através da rede de Petri condição/evento *RPJ*, consegue-se então, modelar características deste jogo nesta rede.

As figuras apresentadas a partir de agora serão partes específicas extraídas das figuras completas (figura 4.5, figura 4.6 e figura 4.7), referenciadas anteriormente, a fim de facilitar a visualização e a compreensão da rede *RPJ*. Por isso, as condições da rede serão representadas por *cnc* e *cmc*, para condições do cordeiro e por *cnl* e *cml*, para condições dos lobos, sendo *cnc*, *cnl*, *cmc* e *cml* condições quaisquer da rede *RPJ*. As condições *jl* e *jc* serão o controle respectivamente, dos lobos e do cordeiro, da vez de jogar para cada um deles. A letra *C* simboliza a rede do cordeiro (denominada sub-rede *C*) e a letra *L* simboliza a rede dos lobos (denominada sub-rede *L*). Da mesma forma, os eventos *ex*, *ey* e *ez* são arbitrários, onde *ex* e *ey* são dois eventos para o cordeiro e *ez* é um evento para os lobos.

Os círculos que indicam as condições da rede também serão diferenciados no estilo de suas linhas. Os círculos que simbolizam as condições do cordeiro (sub-rede *C*) têm uma linha simples, os círculos que simbolizam as condições dos lobos (sub-rede *L*) têm uma linha dupla e os círculos que simbolizam as condições de controle da vez de jogar dos lobos e do cordeiro têm uma linha tracejada. Igualmente, os arcos (as setas) de cada uma das sub-redes seguem o mesmo estilo de linha de suas respectivas condições.

Inicialmente, é preciso saber como a rede *RPJ* se comporta em relação à alternância entre jogadores, ou seja, ao decorrer do jogo os jogadores alternam sua vez de jogar, mais especificamente, ora movimenta-se o cordeiro, ora movimenta-se um dos lobos.

Desta maneira, levando em consideração a especificação mostrada na figura 4.5, na figura 4.6 e na figura 4.7, sabe-se que é necessário haver um controle na rede que possibilite esta alternância. Modelando-se esta característica segundo a definição da rede *RPJ*, obtém-se a figura 4.8.

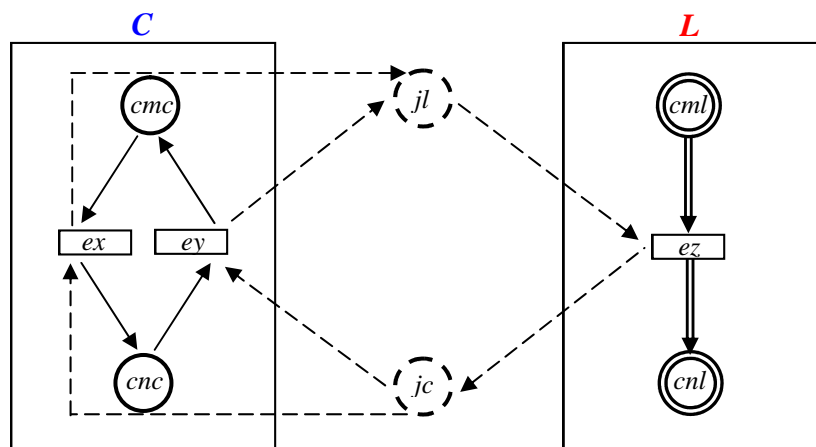


Figura 4.8: Rede *RPJ* para a característica de alternância entre jogadores.

Para facilitar a identificação dos *tokens* ao longo das descrições seguintes, utiliza-se a denominação *token* cordeiro (caracterizando o jogador que representa o cordeiro),

token lobo (caracterizando o jogador que representa os lobos), *token* vez (caracterizando a vez para jogar) e *token* livre (caracterizando as condições livres).

A figura 4.8 mostra como a rede *RPJ* está modelada para controlar a alternância entre os jogadores. O funcionamento desta rede é o seguinte: por exemplo, na vez em que um dos lobos precisa jogar (movimentar-se), a condição *jl* necessita estar com o *token* vez que é passado a um dos lobos para ele poder jogar; a condição *cml* também precisa ter um *token* lobo (figura 4.9a). Quando este lobo executar seu movimento pelo evento *ez* – o *token* lobo sendo passado à condição *cnl* – o *token* vez é repassado à condição *jc* para indicar que o cordeiro tem a vez para jogar pelos eventos *ex* ou *ey* (figura 4.9b). E assim sucessivamente, a passagem do *token* vez entre *jl* e *jc* vai se alternando até o fim do jogo.

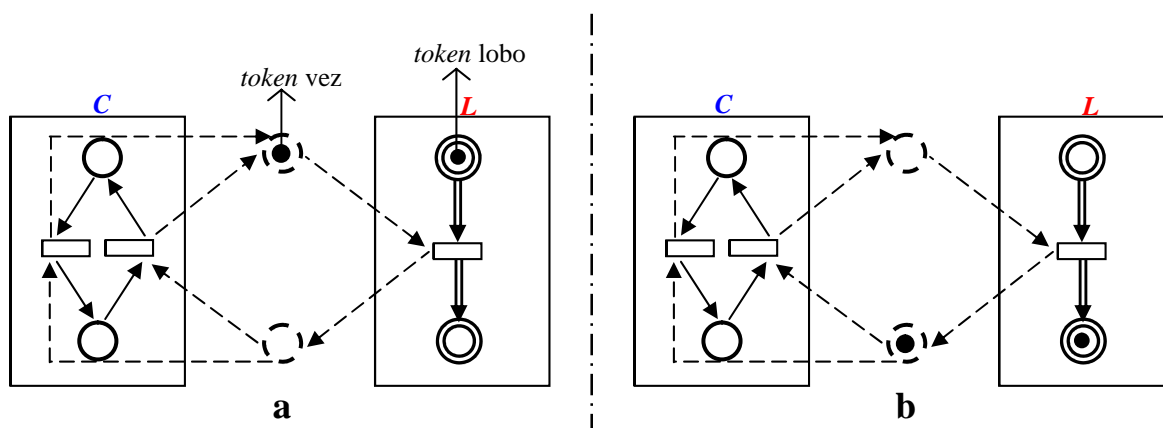


Figura 4.9: Exemplo para a característica de alternância entre jogadores.

Contudo, a rede *RPJ* da figura 4.8 precisa ser ampliada, pois nela não estão previstas as suas condições livres; precisa haver um controle de condições livres da rede para se saber quais as condições que podem ser ocupadas pelos *tokens*, representando cada um dos jogadores (*token* cordeiro e *token* lobo). Isto, porque pode haver somente um único jogador em cada condição da rede, a cada jogada.

Ampliando-se então a rede da figura 4.8, para incorporar o controle de condições livres da rede, chega-se à ilustração da figura 4.10.

Relembra-se que a sub-rede *C* está baseada na rede de Petri inicialmente projetada para o cordeiro (figura 4.5), a sub-rede *L* está baseada na rede de Petri inicialmente projetada para os lobos (figura 4.6) e a sub-rede *Controles* – agora incorporada – está baseada na rede de Petri inicialmente projetada para a vez de jogar e as condições livres (figura 4.7).

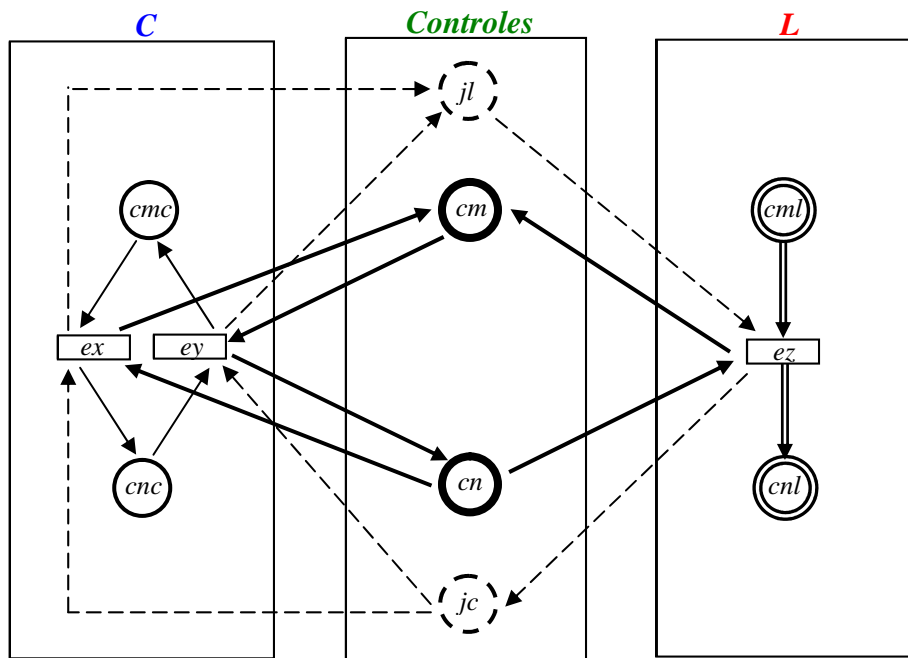


Figura 4.10: Ampliação da rede *RPJ* para o controle de condições livres.

Como se pode ver na rede *RPJ* da figura 4.10, foi necessário expandir esta rede para acrescentar uma sub-rede denominada *Controles*, que nada mais é do que a rede responsável pelo controle de alternância, já vista na figura 4.8, e a partir de agora, também responsável pelo controle de condições livres (sub-rede *Controles*). Conseqüentemente, esta rede *Controles* comunica-se com as redes *C* e *L*. A junção destas sub-redes *Controles*, *C* e *L* forma a rede *RPJ*.

Na sub-rede *Controles* há as condições *cn* e *cm* que indicam duas condições quaisquer na rede, assim como as condições *cnc*, *cnl*, *cmc* e *cml*. Seguindo a caracterização das condições e dos arcos da rede, os círculos que simbolizam as condições e as setas que simbolizam os arcos da sub-rede *Controles* possuem uma linha mais grossa (*bold*). As condições *jl* e *jc* foram anteriormente definidas e continuam com o mesmo conceito.

O funcionamento da rede *RPJ* da figura 4.10, aplicando o exemplo anterior da figura 4.9, é o seguinte: na vez em que um dos lobos precisa jogar (movimentar-se), a condição *jl* necessita estar com o *token* vez que é passado a um dos lobos para ele poder jogar e a condição *cn* necessita estar com o *token* livre para indicar que o lobo pode assumir esta condição; a condição *cml* também precisa ter um *token* lobo (figura 4.11a).

Quando este lobo executar seu movimento pelo evento *ez* – o *token* lobo sendo passado à condição *cnl* – o *token* vez é repassado à condição *jc* para indicar que o cordeiro tem a vez para jogar pelos eventos *ex* ou *ey* e o *token* livre é repassado à condição *cm* para indicar que esta condição, agora está livre e por conseqüência, a condição *cn*, agora está ocupada (figura 4.11b). E assim sucessivamente, a passagem do *token* vez entre *jl* e *jc* vai se alternando até o fim do jogo e sempre que uma condição na rede estiver livre (não ocupada por um dos jogadores), haverá um *token* livre nesta condição que indique isto. Este funcionamento fica atrelado à sub-rede *Controles*, que se comunica com as sub-redes *C* e *L*.

No total, o número de *tokens* livres da sub-rede *Controles* é de 27 – o que é definido pelo número total de condições da rede (32), descontados do número total de jogadores (5). Já o número total de *tokens* da rede *RPJ* é de 33, pois além de 27 *tokens* livre da sub-rede *Controles*, ainda há quatro (4) *tokens* lobo da sub-rede *L* e um (1) *token* cordeiro da sub-rede *C*.

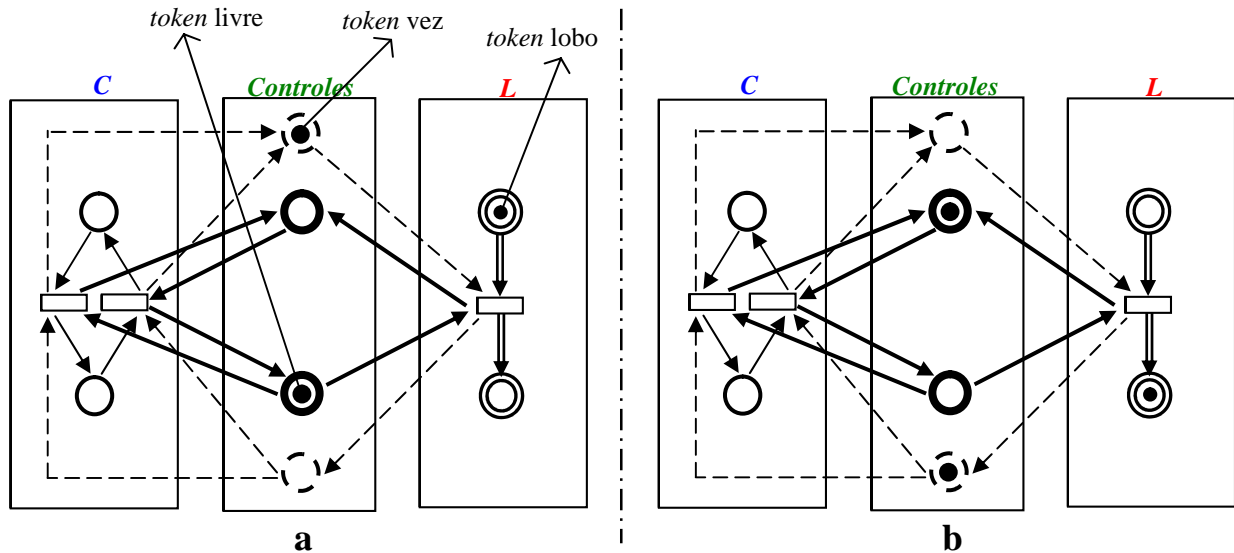


Figura 4.11: Exemplo para o controle de condições livres.

A condição j_l da sub-rede *Controles* tem por objetivo manter, se for o caso, o *token* vez que será passado a um dos lobos quando for a sua vez de jogar. No entanto, como o número de lobos é de quatro e é permitido que somente um deles jogue quando for sua respectiva vez de jogar, precisa-se entender como a condição j_l age para passar o *token* vez a um destes lobos. Para melhor visualizar esta situação do detalhamento da condição j_l , é apresentada a figura 4.12.

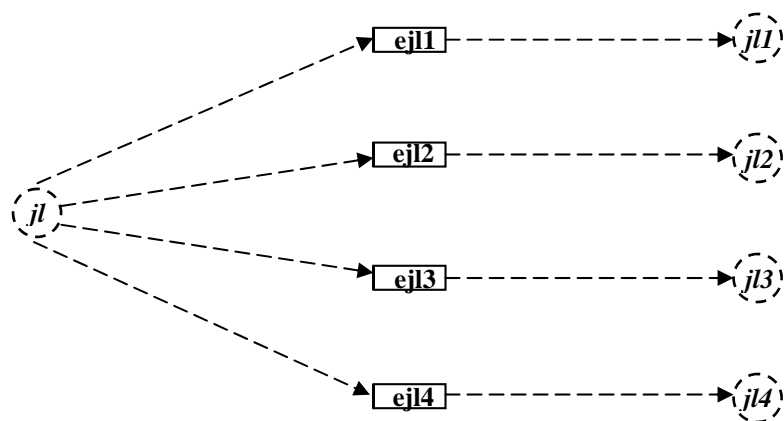


Figura 4.12: Distribuição do *token* vez para um dos lobos.

Explicando-se a figura 4.12, nota-se que no momento em que jl (pode-se chamar jl de “joga lobo”) recebe o *token* vez – figura 4.13a, o evento que primeiramente ocorrer, atribui este *token* a sua respectiva condição destino; por exemplo, se $ejl1$ ocorrer primeiro, o *token* vez é atribuído à $j11$ (pode-se chamar $j11$ de “joga lobo 1”), se $ejl2$ ocorrer primeiro, o *token* vez é atribuído à $j12$ (pode-se chamar $j12$ de “joga lobo 2”), se $ejl3$ ocorrer primeiro, o *token* vez é atribuído à $j13$ (pode-se chamar $j13$ de “joga lobo 3”) e se $ejl4$ ocorrer primeiro, o *token* vez é atribuído à $j14$ (pode-se chamar $j14$ de “joga lobo 4”).

Estas ações caracterizam o não-determinismo característico das redes de Petri. Com isso, a condição que receber o *token* vez aciona o seu lobo correspondente para jogar ($lj1$ aciona o lobo 1 para jogar – figura 4.13b, $lj2$ aciona o lobo 2 para jogar – figura 4.13c, $lj3$ aciona o lobo 3 para jogar – figura 4.13d e $lj4$ aciona o lobo 4 para jogar – figura 4.13e). É o que se pode dizer, em termos do jogo, de “quem pegar primeiro, joga”.

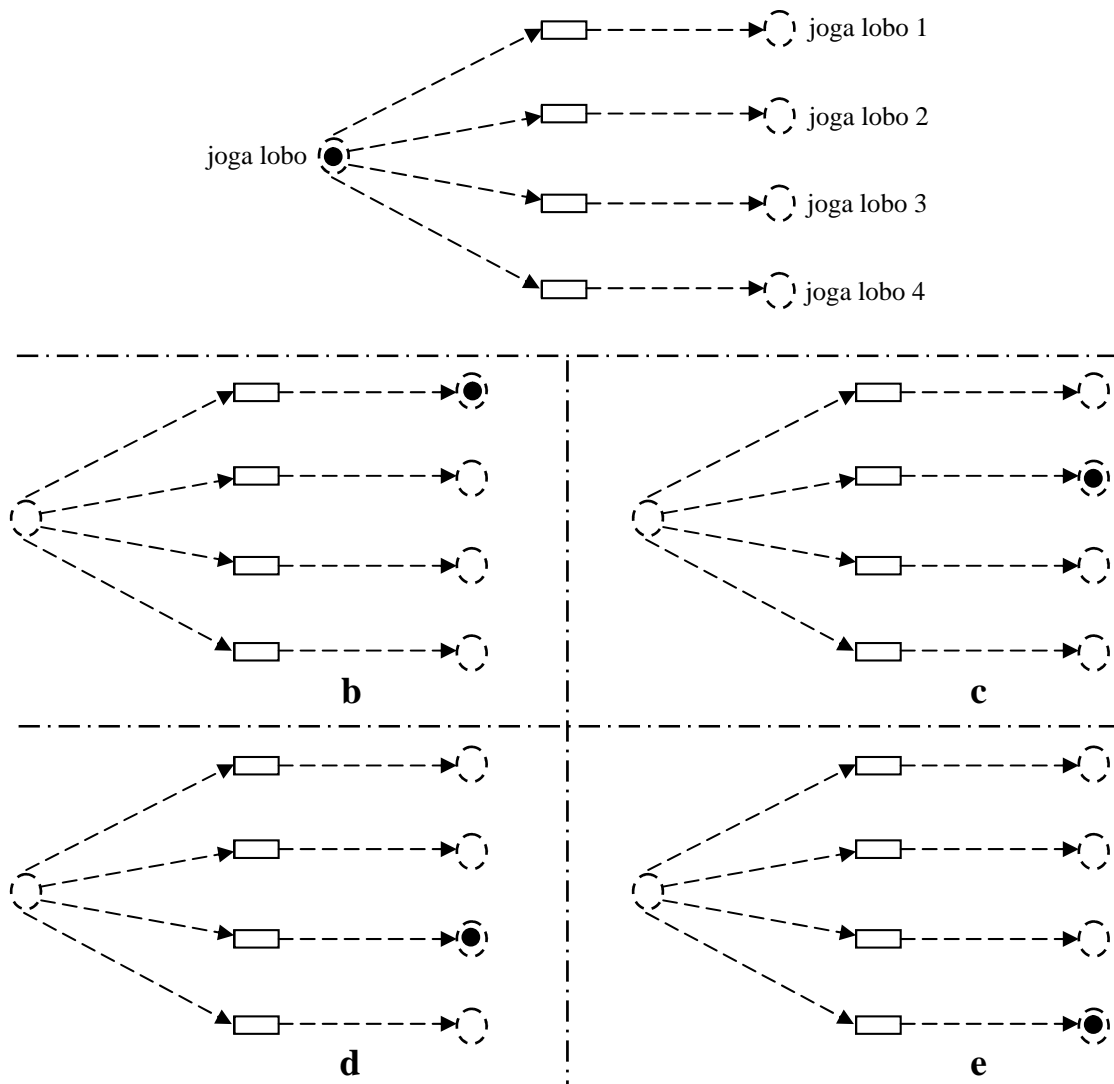


Figura 4.13: *Token* vez atribuído para um dos lobos.

Com esta definição de jl a rede RPJ sofre alterações, ficando jl ampliado conforme ilustra a figura 4.14.

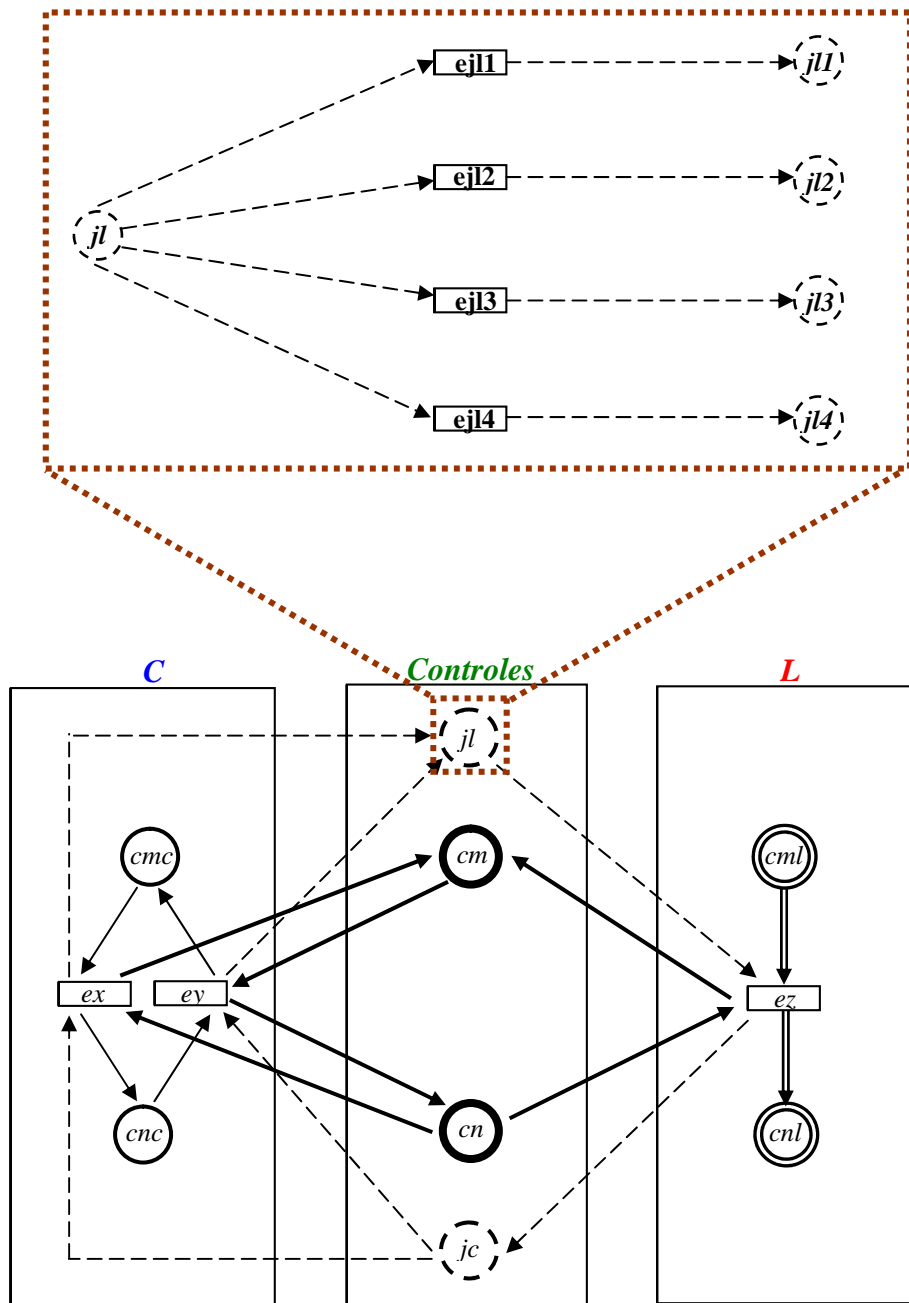


Figura 4.14: A ampliação da condição jl na rede RPJ .

A aplicação de uma estratégia para escolher o lobo que deve usar a vez de jogar para se movimentar, requer uma alteração na definição desta condição jl que repassa o *token* vez a um dos lobos $jl1$, $jl2$, $jl3$ ou $jl4$ não-deterministicamente. Isto pois, para incorporar uma estratégia é preciso especificar cada uma das configurações possíveis da rede RPJ – dependendo da situação em que estiver cada um dos jogadores, através de situações pré-definidas da rede que passem o *token* vez ao lobo mais indicado para jogar. Esta

possibilidade caracteriza um determinismo da rede e é descrita posteriormente na subsecção 4.5.4.

4.5.3 Especificação Ampliada

A partir do momento em que se definiu a rede *RPJ* não-determinística, pode-se então atualizá-la, partindo da definição anteriormente estabelecida para os conjuntos de condições e de eventos, e as funções de pré e pós-condições. A marcação inicial e os *tokens* permanecem inalterados, ficando com as mesmas definições anteriores da secção 4.5.1.

O conjunto de condições da rede *RPJ* é formado agora, por: $C = \{c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28, c29, c30, c32, c33, c34, c35, c36, c37, c38\}$. Onde as condições *c35*, *c36*, *c37* e *c38* foram acrescentadas para incorporar a ampliação da condição *jl*. Segundo a figura 4.13, *c35* é a condição que repassa o *token* vez ao lobo 1, *c36* é a condição que repassa o *token* vez ao lobo 2, *c37* é a condição que repassa o *token* vez ao lobo 3 e *c38* é a condição que repassa o *token* vez ao lobo 4.

O conjunto de eventos da rede *RPJ* é formado agora, por: $E = \{e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12, e13, e14, e15, e16, e17, e18, e19, e20, e21, e22, e23, e24, e25, e26, e27, e28, e29, e30, e31, e32, e33, e34, e35, e36, e37, e38, e39, e40, e41, e42, e43, e44, e45, e46, e47, e48, e49, e50, e51, e52, e53, e54, e55, e56, e57, e58, e59, e60, e61, e62, e63, e64, e65, e66, e67, e68, e69, e70, e71, e72, e73, e74, e75, e76, e77, e78, e79, e80, e81, e82, e83, e84, e85, e88, e89, e92, e93, e96, e97, e2', e3', e6', e7', e10', e11', e14', e16', e17', e20', e21', e24', e25', e28', e30', e31', e34', e35', e38', e39', e42', e44', e45', e48', e49', e52', e53', e56', e58', e59', e62', e63', e66', e67', e70', e72', e73', e76', e77', e80', e81', e84', e86', e87', e90', e91', e94', e95', e98', e99', e100', e101', e102'\}$. Onde os eventos *e99'*, *e100'*, *e101'* e *e102'* foram acrescentados para incorporar a ampliação da condição *jl*. Segundo a figura 4.13, *e99'* é o evento que aciona *jl1* (a condição *c35*), *e100'* é o evento que aciona *jl2* (a condição *c36*), *e101'* é o evento que aciona *jl3* (a condição *c37*) e *e102'* é o evento que aciona *jl4* (a condição *c38*).

As funções de pré-condição e de pós-condição agora, além daquelas definidas inicialmente, também são:

$$\begin{aligned} pre(e99') &= \{c34\}, & pos(e99') &= \{c35\}, \\ pre(e100') &= \{c34\}, & pos(e100') &= \{c36\}, \\ pre(e101') &= \{c34\}, & pos(e101') &= \{c37\}, \\ pre(e102') &= \{c34\}, & pos(e102') &= \{c38\}. \end{aligned}$$

Para relembrar, a rede *RPJ* pode ser vista como uma rede estruturada em seis camadas e cada uma das camadas diz respeito a uma rede correspondente: uma camada de rede do cordeiro, quatro camadas de redes dos lobos (uma camada de rede para cada um dos lobos 1, 2, 3 e 4, pois eles movimentam-se independentemente) e uma camada de rede do controle; estando todas estas camadas de redes interligadas pelos eventos da rede *RPJ*. Isto, pois há a característica da rede de Petri condição/evento, onde uma condição pode ter somente um único *token* a cada instante; em termos do jogo, há a

garantia de que dois jogadores não ocupam o mesmo lugar no tabuleiro, ao mesmo tempo. A figura 4.15 a seguir, mostra toda a rede *RPJ*.

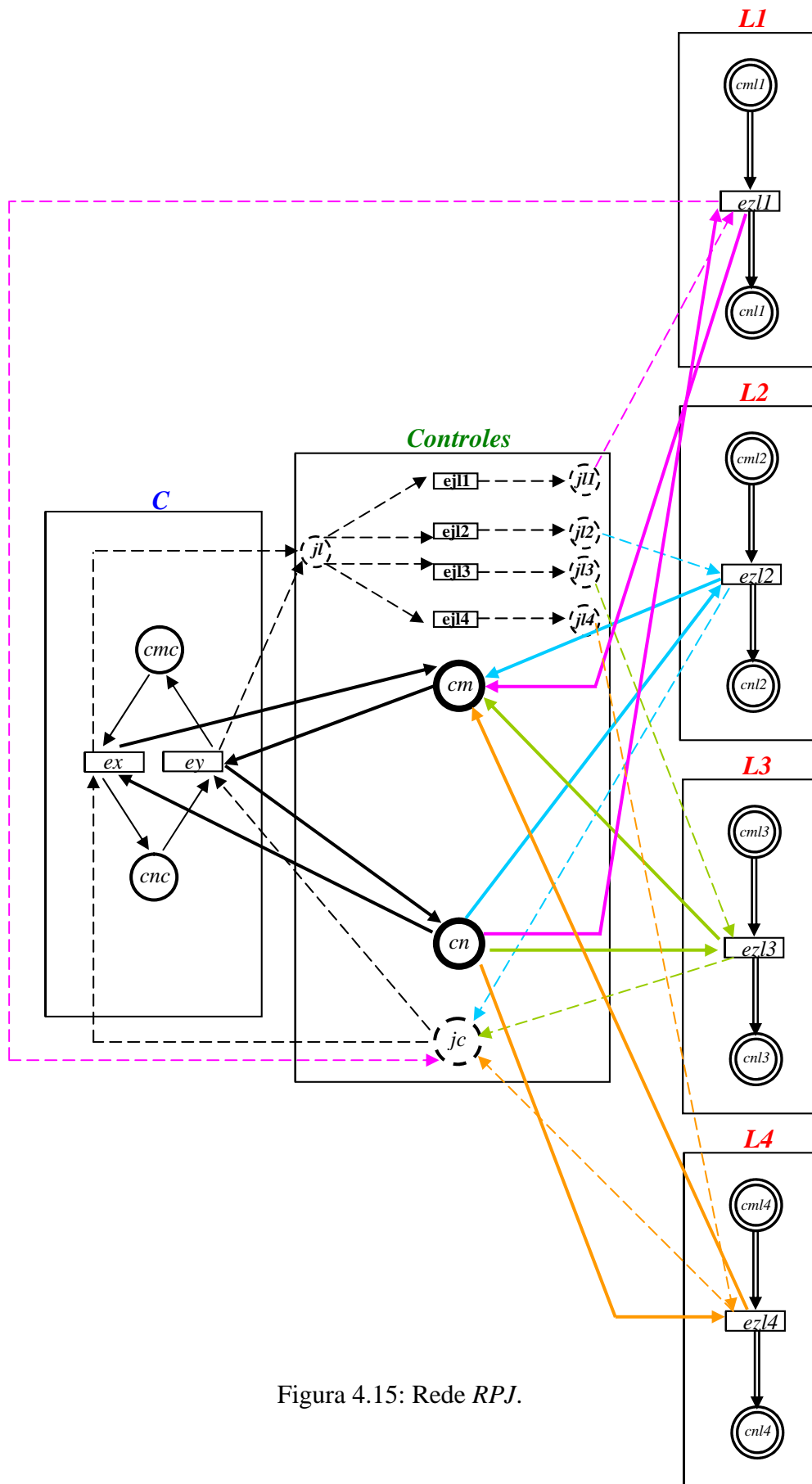


Figura 4.15: Rede RPJ.

Neste ponto da definição esclarece-se que, se a condição *jl* tem o *token* vez e o repassa a um dos lobos e assim, o lobo que receber este *token* vez estiver habilitado a realizar um movimento, ou seja, a condição para a qual ele deseja deslocar-se estiver livre (*token* livre), a rede funciona normalmente segundo a sua definição inicial. Porém, se a condição para a qual o lobo em questão deseja movimentar-se estiver ocupada, então este lobo precisa repassar o *token* vez ao próximo lobo correspondente (lobo 1 para lobo 2, lobo 2 para lobo 3, lobo 3 para lobo 4 e lobo 4 para lobo 1) sucessivamente, até que um dos lobos tenha o *token* vez e esteja possibilitado de realizar algum movimento.

Da mesma forma, quando a condição *jc* tem o *token* vez e o cordeiro está habilitado a realizar um movimento (a condição para a qual ele precisa deslocar-se está livre), a rede também funciona normalmente segundo a sua definição inicial. Um problema que novamente existe é se a condição para a qual o cordeiro precisa movimentar-se está ocupada, então é necessária a ocorrência de outro evento que corresponda a uma condição livre na rede.

Na situação em que os lobos têm a vez para jogar e assim, apenas um deles pode efetuar esta ação, estando os *tokens* dos jogadores em suas respectivas condições, a ação que será realizada pela condição que passa o *token* a um dos lobos para que este possa jogar, é uma atitude não-determinística e pode resultar no “travamento” da rede. Isto, se o lobo que for escolhido para jogar, não puder se movimentar para a sua próxima posição livre. Portanto, pela especificação construída é possível identificar tal situação e defini-la para que este travamento não ocorra.

Ao acontecer a situação de nenhum *token* lobo, ao ter o *token* vez, poder alcançar uma próxima condição da rede, significa que a rede terminou sua execução; indicativo de que os lobos perderam o jogo e que o cordeiro venceu o jogo. Igualmente, na situação em que o *token* cordeiro, ao ter o *token* vez, não puder alcançar nenhuma condição da rede (excetuando-se a situação em que ele estiver em alguma das posições que representam a parede de chegada), significa que a rede também terminou sua execução; indicativo de que o cordeiro perdeu o jogo e que os lobos venceram o jogo.

Em termos da rede *RPJ*, as duas situações descritas no parágrafo anterior (vitória do cordeiro e vitória dos lobos) equivalem às configurações que seguem:

⇒ **Vitória do cordeiro** – há quatro condições na rede *RPJ*, onde cada uma dessas condições está sendo ocupada por um lobo e a partir destas condições ocupadas, o(s) seu(s) evento(s) correspondente(s) não pode(m) ocorrer – para levar o *token* jogador para alguma condição seguinte ou simplesmente, não há evento(s) a ocorrer naquela(s) determinada(s) condição(ões). A figura 4.16 mostra um exemplo de situação que representa a vitória do cordeiro. Neste exemplo, os lobos estão ocupando as condições *c1*, *c2*, *c5* e *c6* (destacadas pela elipse na camada de rede dos lobos, figura 4.16a) e o cordeiro está ocupando a condição *c25* (destacada pela elipse na camada de rede do cordeiro, figura 4.16b).

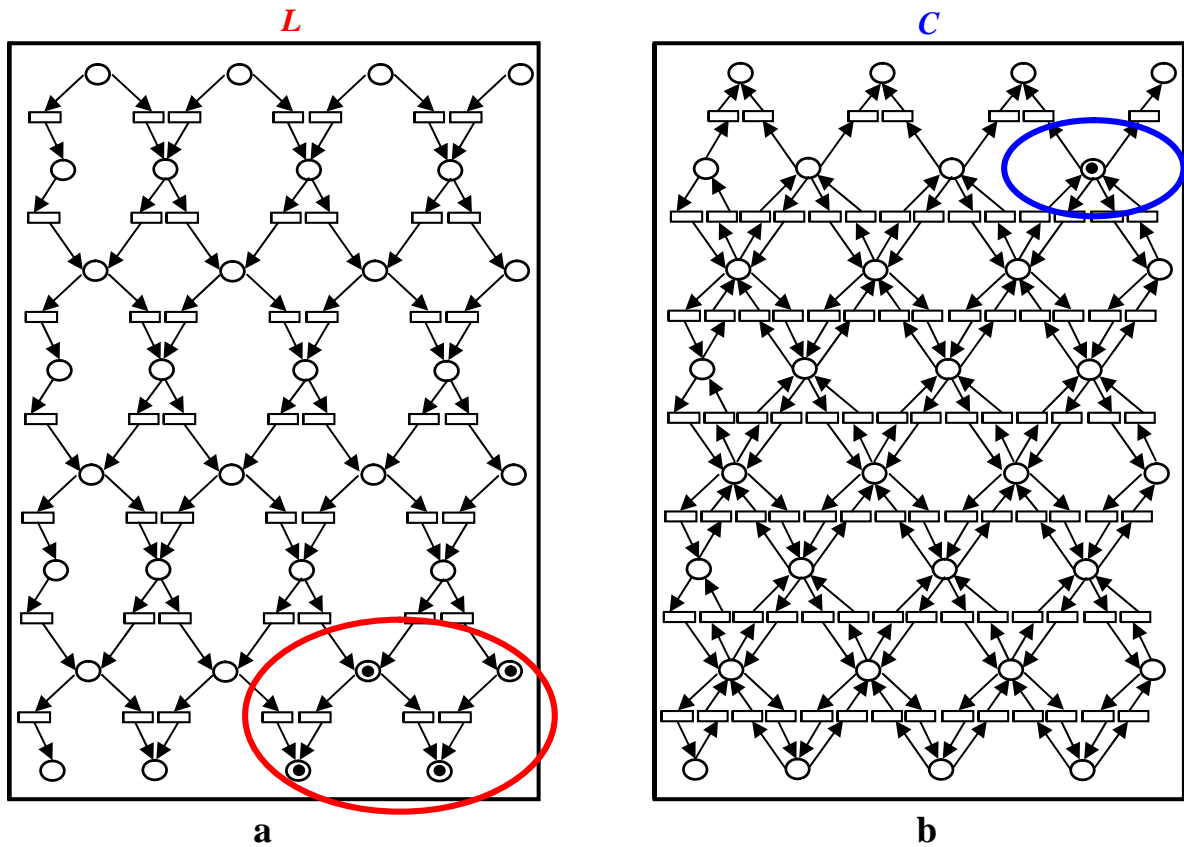


Figura 4.16: Exemplo de vitória do cordeiro.

⇒ Vitória dos lobos – há uma condição na rede *RPJ* que está sendo ocupada pelo cordeiro e a partir desta condição ocupada (exceto se ele está em alguma das condições que representam a parede de chegada – *c29*, *c30*, *c31* ou *c32*), o(s) seu(s) evento(s) correspondente(s) não pode(m) ocorrer – para levar o *token* jogador para alguma condição seguinte ou simplesmente, não há evento(s) a ocorrer naquela(s) determinada(s) condição(ões). A figura 4.17 mostra um exemplo de situação que representa a vitória dos lobos. Neste exemplo, os lobos estão ocupando as condições *c13*, *c14*, *c21* e *c22* (destacadas pela elipse na camada de rede dos lobos, figura 4.17a) e o cordeiro está ocupando a condição *c17* (destacada pela na camada de rede do cordeiro, figura 4.17b).

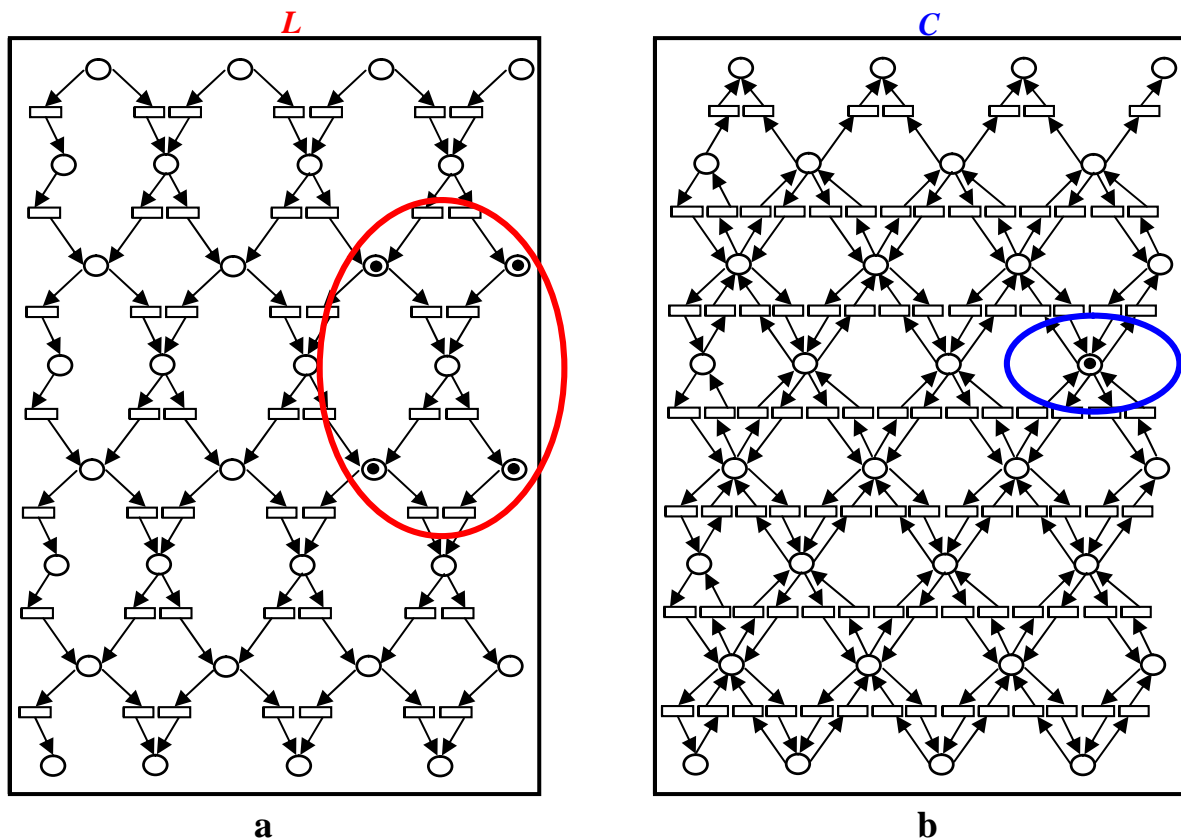


Figura 4.17: Exemplo de vitória dos lobos.

A característica do não-determinismo aplica-se a todo o funcionamento da rede *RPJ*. Portanto, até este ponto da especificação formal do jogo lobos e cordeiro está se tratando apenas das possibilidades de jogo sem o uso de quaisquer estratégias. Ou seja, quando a rede *RPJ* é inicializada, sua execução segue passos não-determinísticos até que ela chegue a um estado final, sem se preocupar em seguir qualquer esquema estratégico definido para o jogo.

A rede *RPJ* não-determinística é uma rede sem estratégias, enquanto que a rede *RPJ* determinística é uma rede com estratégias.

A rede sem estratégias é definida segundo a figura 4.15, ou seja, a execução da rede não segue a aplicação de estratégias e portanto, os *tokens* dos jogadores se movimentam pelas condições, não-deterministicamente, assim que seus eventos correspondentes puderem ocorrer.

A rede com estratégias tem a mesma definição da rede mostrada na figura 4.15, porém a sua execução segue uma verificação de estado da rede (que pode ser vista na subsecção 4.5.4), correspondendo à aplicação de estratégias, e os *tokens* dos jogadores se movimentam então para as condições definidas ao término desta verificação, através da ocorrência de eventos específicos relacionados a estas condições.

Assim, a rede sem estratégias pode ser vista como uma rede básica, enquanto que a rede com estratégias é a ampliação desta rede básica, onde se acrescenta a aplicação de estratégias e ela pode ser vista como uma rede estendida.

4.5.4 Especificação para Estratégias do Jogo

Para incorporar as estratégias do jogo a esta rede *RPJ*, precisa-se prever e definir todas as possibilidades de configuração desta rede, em relação às estratégias que o jogo tem; é definir todas as possibilidades de jogo em função de configurações possíveis da rede *RPJ*. Ou melhor, as condições *jl* e *jc* teriam que ser ampliadas para toda e qualquer possibilidade de jogo envolvendo respectivamente, as estratégias dos lobos e do cordeiro. Esta não é uma tarefa simples, pois o número de possibilidades de jogo é exponencial!

Pelo fato da exponencialidade do jogo, nesta subsecção serão apresentadas somente duas situações que ilustram o uso de estratégias na rede *RPJ*. Há mais algumas situações envolvendo estratégias, que são mostradas na secção 5.2; as demais situações podem ser representadas futuramente.

Para exemplificar o uso de uma estratégia para os lobos, tem-se a situação de jogo da figura 4.18 e a partir dela será definida a rede *RPJ* com uma estratégia para tal situação. Nesta situação os lobos (a condição *jl*) estão com a posse do *token* vez e seguindo a estratégia de formar uma barreira que impeça o cordeiro de passar, o lobo 4 (a condição *jl4*) movimenta-se para a sua próxima posição livre.

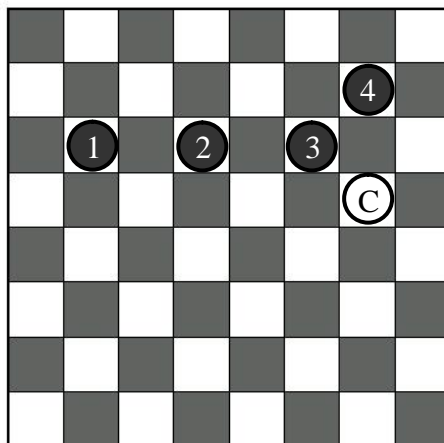


Figura 4.18: Situação de jogo para uma estratégia dos lobos.

A figura 4.19 mostra uma parte da rede *RPJ* definida para a estratégia sugerida para o jogo da figura 4.18. Observa-se nesta rede da figura 4.19, que *c24l* representa a condição ocupada pelo lobo 1, *c23l* representa a condição ocupada pelo lobo 2, *c22l* representa a condição ocupada pelo lobo 3 e *c25l* representa a condição ocupada pelo lobo 4. Já *c17c* representa a condição ocupada pelo cordeiro e *c21* representa uma condição livre na rede.

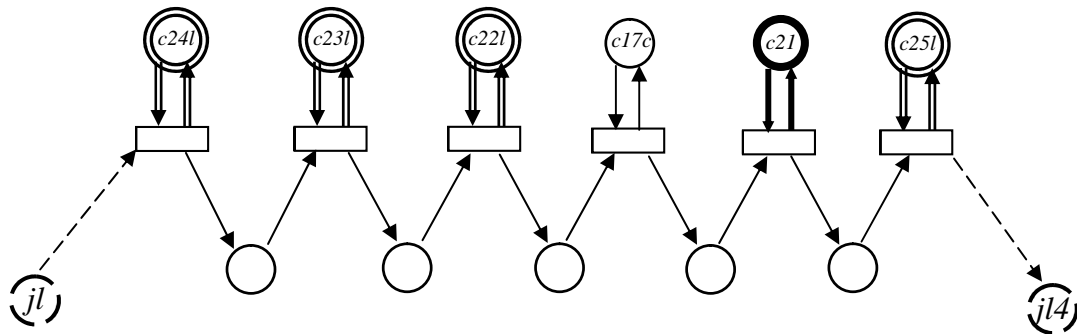


Figura 4.19: Parte da rede *RPJ* para uma estratégia de jogo dos lobos.

Pela característica da estratégia dos lobos formarem uma barreira que impeça o cordeiro de passar, pode-se concluir que se os lobos estão em uma situação semelhante àquela apresentada na figura 4.18, porém em outras posições ao longo do jogo, então eles podem adotar a mesma estratégia definida na figura 4.19, para tal situação. Exemplificando uma situação semelhante tem-se: a condição *c20l* representa a condição ocupada pelo lobo 1, a condição *c19l* representa a condição ocupada pelo lobo 2, a condição *c18l* representa a condição ocupada pelo lobo 3 e a condição *c21l* representa a condição ocupada pelo lobo 4. Já a condição *c14c* representa a condição ocupada pelo cordeiro e a condição *c17* representa uma condição livre na rede. Também nesta situação, o lobo 4 movimentar-se para a próxima posição livre.

Para exemplificar o uso de uma estratégia para o cordeiro, tem-se a situação de jogo da figura 4.20 e a partir dela será definida a rede *RPJ* com uma estratégia para tal situação. Nesta situação o lobo 4 passa o *token* vez para o cordeiro e seguindo a estratégia de passar os lobos, o cordeiro (a condição *jc*) movimentar-se para a sua próxima posição livre que possibilite esta passagem.

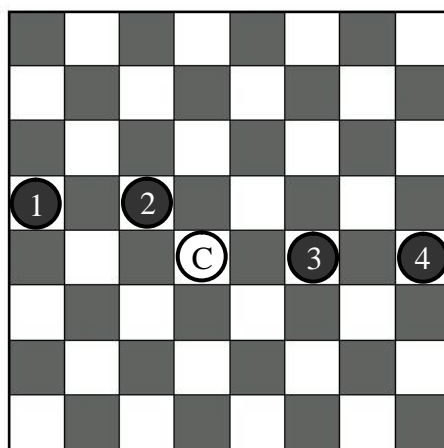


Figura 4.20: Situação de jogo para uma estratégia do cordeiro.

A figura 4.21 mostra uma parte da rede *RPJ* definida para a estratégia sugerida para o jogo da figura 4.20. Observa-se nesta rede da figura 4.21, que *c20l* representa a condição ocupada pelo lobo 1, *c19l* representa a condição ocupada pelo lobo 2, *c14l*

representa a condição ocupada pelo lobo 3 e $c13l$ representa a condição ocupada pelo lobo 4. Já $c15c$ representa a condição ocupada pelo cordeiro e $c18$ representa uma condição livre na rede.

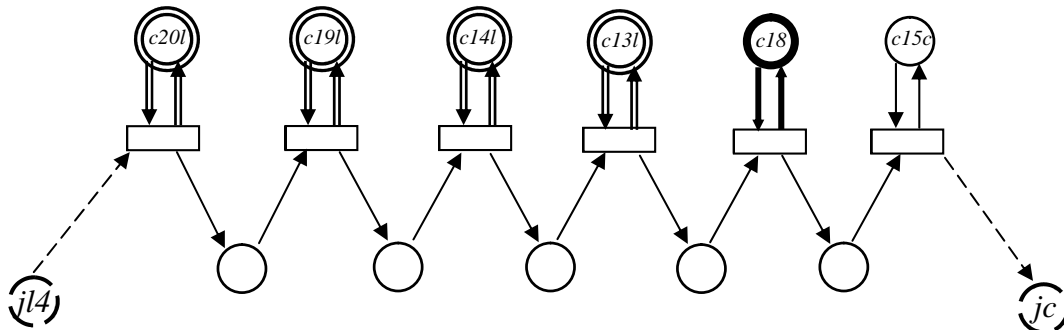


Figura 4.21: Parte da rede *RPN* para uma estratégia de jogo do cordeiro.

Pela característica da estratégia do cordeiro passar os lobos, pode-se concluir que se o cordeiro está em uma situação semelhante àquela apresentada na figura 4.20, porém em outra posição ao longo do jogo, então ele pode adotar a mesma estratégia definida na figura 4.21, para tal situação. Exemplificando uma situação semelhante tem-se: a condição $c12l$ representa a condição ocupada pelo lobo 1, a condição $c11l$ representa a condição ocupada pelo lobo 2, a condição $c14l$ representa a condição ocupada pelo lobo 3 e a condição $c13l$ representa a condição ocupada pelo lobo 4. Já a condição $c10c$ representa a condição ocupada pelo cordeiro e a condição $c15$ representa uma condição livre na rede. Também nesta situação, o cordeiro movimenta-se para a próxima posição livre que possibilite esta passagem.

As duas estratégias especificadas através da figura 4.19 e da figura 4.21 são estratégias que envolvem uma única jogada segundo um dos jogadores, que visam a uma situação imediata, não considerando possíveis jogadas realizadas pelo adversário. Essas estratégias de possíveis jogadas realizadas pelos jogadores (pode-se sugerir a denominação por estratégias de “maior visão”) são mais complicadas de se implementar, devido às 8 jogadas possíveis que os lobos podem realizar e às 4 jogadas possíveis que o cordeiro pode realizar.

Com a especificação formal do jogo diferencial lobos e cordeiro foi possível definir as características do jogo e além disso, foi possível extrair também, algumas propriedades do jogo. Estas propriedades identificadas para o jogo lobos e cordeiro dizem respeito à alcançabilidade, ao *liveness* e ao *deadlock* da rede. Por isso, a secção 4.6 mostra estas situações no jogo lobos e cordeiro.

4.6 Situações de Jogo

A especificação formal acrescenta uma visão formal de situações de jogo, que facilita seu estudo. Às vezes, são situações não desejadas que precisam ser evitadas para a real simulação do jogo. Outras vezes, são situações reais que precisam somente ser identificadas preferencialmente com antecipação.

Outro fato interessante é que essas situações são características usualmente definidas nos formalismos, podendo ser utilizados estudos já realizados. No caso do jogo lobos e cordeiro, algumas situações de jogo podem ser identificadas na especificação e ali analisadas mais eficazmente, tais como alcançabilidade, *liveness* e *deadlock*.

Assim sendo, apresenta-se cada uma destas situações de *liveness*, de alcançabilidade e de *deadlock* no jogo diferencial lobos e cordeiro, nas redes de Petri e em alguns jogos.

4.6.1 *Liveness*

Esta situação é uma pré-condição para se alcançar a vitória, mas em jogos o mais importante realmente é a alcançabilidade da situação de final de jogo. Os estudos sobre jogos em geral tratam somente do segundo caso, ou seja, a alcançabilidade.

No jogo lobos e cordeiro, o algoritmo para definir se a rede alcança uma situação de final de jogo necessita considerar todas as possibilidades de jogadas que os jogadores têm, chegando-se à vitória do cordeiro ou à vitória dos lobos. A vitória e a derrota (do adversário) são as situações de jogo decididas pelos jogadores, entretanto outras situações também se apresentam, como o “estar vivo no jogo”, identificada na rede como *liveness*.

Dizer se a rede é *live* (ou seja, a propriedade *liveness*) no jogo significa que a partida, a partir de uma situação atual, pode chegar a uma situação seguinte, ou seja, se a rede está “viva” para continuar executando.

Desta maneira, segundo a especificação proposta para o jogo lobos e cordeiro na secção 4.5, sabe-se que uma situação de rede *live* corresponde aos *tokens* dos jogadores (*token* lobos ou *token* cordeiro), estando em uma determinada condição, poderem ocupar uma próxima condição através da ocorrência de algum evento relacionado à condição em questão. Ou seja, a rede pode chegar a um próximo estado.

O algoritmo para definir se a rede é *live* no jogo lobos e cordeiro necessita considerar então, se algum dos jogadores (que esteja com a vez para jogar) tem a possibilidade de se movimentar para uma próxima posição no tabuleiro do jogo. Enquanto isto puder ser definido, significa que a rede é *live* e pode chegar a uma situação seguinte a partir de uma situação atual.

Em redes de Petri, o problema de *liveness* é decidir se esta rede é *live*, ou seja, se a rede está “viva”.

[JLL 77] apresenta duas versões para o problema *liveness*. O primeiro problema é determinar quando uma rede de Petri *free-choice* é não *live*, onde se verificou que o mesmo é NP. Já o segundo problema de determinar quando uma rede de Petri de *free-choice* é *live*, é NP-Completo. Para provar que o segundo problema é NP-Completo o autor reduz o problema *Satisfiability*, com uma conjectura no formato de fórmulas e cálculos proposicionais, aplicados ao problema *liveness* de uma rede de Petri *free-choice*.

[ESP 94] afirma que a complexidade computacional do problema *liveness* está aberta, porém existem soluções parciais para diferentes subclasses de redes: o problema *liveness* é NP-Completo para redes de Petri *1-safe*; Co-NP-Completo para redes de Petri *free-choice*; polinomial para redes de Petri limitada *free-choice* e para redes de Petri *conflict-free*.

[CHE 95] mostra que o problema *liveness* é decidível para Redes de Petri arbitrárias, PSPACE-Completo para redes de Petri *1-safe*, tempo linear para redes de Petri acíclicas, tempo constante para redes de Petri *1-safe* acíclicas, polinomial para redes de Petri *conflict-free*, polinomial para redes de Petri *1-safe conflict-free*, NP-Completo para redes de Petri *free-choice* e polinomial para redes de Petri *1-safe free-choice*.

[ESP 98b] afirma que algumas questões sobre redes de Petri *live 1-safe free-choice* são resolvíveis em tempo polinomial.

4.6.2 Alcançabilidade

Nos jogos em geral, a alcançabilidade é a situação de final de jogo, indicando que um dos jogadores é o vencedor.

Assim, a alcançabilidade para o jogo lobos e cordeiro é definida quando se pode determinar se o jogo alcança uma situação de final de jogo, ou seja, se há a vitória do cordeiro ou se há a vitória dos lobos. Desta maneira, segundo a especificação proposta para o jogo na secção 4.5, sabe-se que uma situação de vitória para o cordeiro é correspondente ao *token* cordeiro estar ocupando uma das condições *c29*, *c30*, *c31* ou *c32*, que representam a parede de chegada (a qual é o objetivo do cordeiro).

Igualmente, conforme a especificação proposta para o jogo, sabe-se que uma situação de vitória para os lobos é correspondente ao *token* cordeiro estar ocupando qualquer condição na rede e, a partir desta condição, não ser possível do cordeiro se movimentar (isto significa que os lobos prenderam o cordeiro). Como exemplo, pode-se ver a situação da figura 4.17, que é baseada na configuração da rede *RPJ* para a vitória dos lobos.

Em redes de Petri, o problema da alcançabilidade é o mais estudado e ele é fundamental para o estudo de propriedades dinâmicas de qualquer sistema. O problema da análise desta propriedade consiste em determinar se, dada uma rede de Petri $N=(R; M_0)$ e uma marcação M' , M' pode ser acessível a partir de M_0 .

[JLL 77] apresenta como resultado da análise de complexidade, que o problema da alcançabilidade para Redes de Petri *conflict-free* é NP-Difícil. A prova deste resultado foi construída através da redução ao problema *Satisfiability*. Neste artigo também é tratado o problema da alcançabilidade de Redes de Petri *k-limitadas*, o qual possui como resultado da análise de complexidade, NP-Completo.

Em [ESP 94] encontram-se vários resultados da análise de complexidade para o problema da alcançabilidade, são eles: o problema da alcançabilidade para redes de Petri com no mínimo dois arcos inibidores é indecidível, e com somente um arco inibidor é decidível. Além destes problemas, [ESP 94] apresenta resultados de complexidade do problema da alcançabilidade com diversas subclasses de redes. O problema da alcançabilidade é EXPSPACE-Completo para redes de Petri simétricas, PSPACE-completo para redes de Petri *1-safe*, PSPACE-Completo para redes de Petri *single-path*, NP-Completo para redes de Petri sem ciclos, NP-Completo para redes de Petri *sinkless*, NP-Completo para redes de Petri *conflict-free*, polinomial para redes de Petri *conflict-free* limitada e polinomial para redes de Petri *live*, limitada, cíclica e *free-choice*.

[CHE 95] afirma que o problema da alcançabilidade é decidível para redes de Petri arbitrárias, PSPACE-Completo para redes de Petri *1-safe*, NP-Completo para redes de

Petri acíclicas, NP-Completo para redes de Petri 1-*safe* acíclicas, NP-Completo para redes de Petri *conflict-free*, polinomial para redes de Petri 1-*safe conflict-free*, decidível para redes de Petri de *free-choice*, e PSPACE-Completo para redes de Petri 1-*safe free-choice*.

Em [ESP 98a] foi analisada a complexidade do problema da alcançabilidade para redes de Petri *live* e *safe free-choice*, chegando-se à conclusão de que o problema é NP-Completo. Foi citado outro problema quanto a sua complexidade: o problema da alcançabilidade para redes de Petri *live*, limitadas e cíclicas *free-choice* é de complexidade polinomial.

Segundo [ESP 98b], o problema da alcançabilidade para redes de Petri: limitada é NP-Completo, 1-*safe* é NP-Difícil, 1-*safe live free-choice* é NP-Completo.

Para uma rede de Petri condição/evento, considerada neste trabalho, os resultados podem ser diferentes. No caso particular do jogo lobos e cordeiro ser considerado como um jogo finito e pequeno, esses problemas perdem importância. No estudo de estratégias com k jogadas de avanço, viu-se que a complexidade é exponencial em k . Este resultado é confirmado no estudo de [REI 93] sobre jogos diferenciais.

Pelo algoritmo de estratégia definido posteriormente na secção 5.3 para o jogo lobos e cordeiro, sabe-se que as possibilidades a serem analisadas em cada jogada por um jogador são exponenciais.

Os resultados encontrados para jogos dizem respeito aos jogos combinatoriais, que são a classe dos jogos que engloba os jogos diferenciais e referem-se à alcançabilidade de uma situação de final de jogo. Os jogos combinatoriais são aqueles que envolvem dois jogadores com informação perfeita (sem informação “escondida” no jogo), onde estes jogadores se movem alternadamente, tendo como resultado final ou vencer ou perder ou, ainda, empatar. Outra característica dos jogos combinatoriais é o princípio estratégico.

Os jogos combinatoriais desta forma, são a classe de jogos que englobam os jogos diferenciais, uma vez que os jogos diferenciais também apresentam estas características encontradas nos jogos combinatoriais.

Os autores pesquisados não abordam os jogos diferenciais de forma geral, mas sim jogos especiais que têm aspectos semelhantes aos jogos diferenciais, no que diz respeito à maneira de jogar, ao número de jogadores, às estratégias, etc.

[BOD 89] apresenta o problema dos jogos de “colorir”. Um exemplo é o jogo onde os jogadores precisam alternadamente, colorir o menor vértice numerado descolorido, de um grafo $G = (\{1, 2, \dots, n\}, E)$, com uma cor do conjunto C , tal que dois vértices adjacentes nunca sejam coloridos com a mesma cor.

Em uma variante deste problema de [BOD 89], o primeiro jogador que estiver impossibilitado de mover-se, perde o jogo. Em outra variante, o jogador A ganha o jogo, se e somente se, o jogo termina com todos os vértices coloridos. Para ambas variantes, o problema de determinar se há uma estratégia vencedora (também é uma situação de *liveness*) para o jogador A é PSPACE-Completo, para qualquer C , onde $|C| > 3$.

Este jogo encontrado em [BOD 89] é similar ao jogo lobos e cordeiro no que diz respeito à alternância entre os jogadores e a impossibilidade de movimentação de um jogador significar a sua perda do jogo.

[REI 93] apresenta um jogo de busca poliédrico (onde os jogadores e os obstáculos são representados por poliedros – cubos, paralelepípedos, etc), tendo dois jogadores (um buscador e um evasor) que se movimentam, contornando obstáculos, em um espaço tri-dimensional, com um limite na velocidade de cada jogador, mas sem limite para a sua aceleração.

Neste jogo, [REI 93] mostra resultados através de um algoritmo aproximativo que considera uma margem de segurança definida por ε , o qual sempre encontrará uma estratégia vencedora, se existir uma estratégia ε -segura, ou seja, se é mantida uma distância ε em relação aos obstáculos do jogo. No entanto, se existir uma estratégia, mas ela não for uma estratégia ε -segura, então o algoritmo pode ou não encontrar uma estratégia vencedora.

Os resultados que são apresentados por [REI 93], mostram uma solução para o problema de alcançar a vitória de espaço polinomial e de tempo exponencial no pior caso, para este jogo. No caso do jogo lobos e cordeiro o resultado é um algoritmo exponencial que pode ser visto na secção 5.3.

Este jogo apresentado por [REI 93] tem as mesmas características do jogo lobos e cordeiro, no entanto o jogo lobos e cordeiro acontece em um espaço plano (unidimensional) sem definições de velocidade e aceleração dos jogadores. As características de velocidade e aceleração poderiam ser acrescentadas ao jogo lobos e cordeiro, sendo preciso definir que os jogadores realizassem, por exemplo, mais movimentos a cada vez que jogassem.

[STO 94] mostra o seguinte problema: dados um ponto de origem, um conjunto de obstáculos em um espaço bi-dimensional e um ponto de destino, é possível mover-se um determinado objeto, ao longo desse espaço, sem colidir-se este objeto em qualquer obstáculo, considerando o caminho mais curto?

Assim, [STO 94] definiu um algoritmo para poucos obstáculos. O algoritmo mostra-se eficiente para um tempo $O(kn)$, onde k representa o número de obstáculos, e para um espaço $O(n)$.

O jogo mostrado por [STO 94] que também é similar ao jogo apresentado por [REI 93], porém o jogo de [STO 94] acontece em um espaço bi-dimensional, pode ser comparado ao jogo lobos e cordeiro, pois uma das questões do jogo lobos e cordeiro é saber qual o número mínimo de passos (corresponde ao caminho mais curto) que um jogador precisa realizar para alcançar seu objetivo no jogo. Este número mínimo e também, máximo de passos para cada jogador foi definido na secção 5.1 e na secção 5.2.

Nestes três exemplos de jogos de [REI 93], de [STO 94] e do jogo diferencial lobos e cordeiro, tem-se que o problema de determinar a alcançabilidade de uma situação de vitória, a partir de um ponto inicial, possui complexidade de tempo polinomial.

4.6.3 Deadlock

Outra situação de jogo – não desejável – causa frustração em uma partida e é a situação de impossibilidade de jogadas, com a partida em estado não-final. Isto, em uma rede de Petri é caracterizado, como *deadlock*. No jogo lobos e cordeiro há uma situação dessas que é tratada e evitada.

Dizer que a rede está em *deadlock* no jogo lobos e cordeiro significa que os *tokens* dos jogadores não podem atingir uma condição qualquer, pois o evento correspondente não tem a possibilidade de ocorrência (indicando que a condição já está ocupada) ou então não há evento para ocorrer (indicando que o jogador está em uma das paredes definidas para o jogo, excetuando-se a situação em que o *token* cordeiro ocupa uma das condições *c29*, *c30*, *c31* ou *c32*).

Um exemplo de *deadlock* é a situação que se tinha na rede *RPJ* antes da ampliação da condição *jl*. Esta ampliação da condição *jl* pode ser vista na figura 4.14 e posteriormente, pode ser vista na rede *RPJ* da figura 4.15.

Desta maneira, segundo a especificação proposta para o jogo na secção 4.5, sabe-se que uma situação de *deadlock* faz com que a rede pare de executar mesmo não tendo fornecido ainda um resultado final (o que é uma ação não esperada em um sistema). Como exemplo disto, apresentou-se a situação em que um dos lobos possuía a vez para jogar, mas não tinha a possibilidade de se movimentar para uma próxima posição porque ela estava ocupada por um outro jogador (ou um dos lobos ou o cordeiro).

Pela identificação desta situação na rede, pôde-se defini-la de modo que no momento em que esta situação acontecesse, o lobo que estivesse com a vez para jogar deveria repassá-la para um outro lobo e assim sucessivamente, até que um dos lobos pudesse se movimentar e estivesse com a vez para jogar. Assim, para esta situação pode-se dizer que a rede *RPJ* é *deadlock-free*, pois ela está livre da ocorrência de *deadlock*. No entanto, quando todos os lobos, já tendo repassado o *token* vez entre si, não têm a possibilidade de realizar algum movimento, significa que há uma situação de final de jogo onde há a vitória do cordeiro (neste caso, não se caracteriza a situação de *deadlock*).

Já a situação em que o cordeiro possui a vez para jogar, mas não tem a possibilidade de se movimentar para qualquer próxima posição porque ela está ocupada por um dos lobos, não caracteriza um *deadlock* e sim, uma situação de final de jogo onde há a vitória dos lobos (ou seja, os lobos prenderam o cordeiro).

Assim, o algoritmo para definir se a rede é *deadlock-free* necessita considerar se algum dos lobos, estando com a vez para jogar, tem a possibilidade de se movimentar para uma próxima posição livre no tabuleiro do jogo.

Em redes de Petri, uma marcação de uma rede é *deadlock* se esta não habilita uma transição (ou no caso da rede de Petri condição/evento, se não há ocorrência de um evento). O problema *deadlock* em uma rede de Petri é o problema de decidir se quaisquer das marcações alcançáveis são *deadlock*.

Em [ESP 94] são apresentados resultados de complexidade para redes de Petri *deadlock-free*, ou seja, toda marcação alcançável habilita alguma transição. Assim, uma rede de Petri *deadlock-free* pode ser reduzida em tempo polinomial ao problema da alcançabilidade. O problema *deadlock-free* é PSPACE-Completo para redes de Petri 1-*safe single-path*, NP-Completo para redes de Petri 1-*safe free-choice* e polinomial para redes de Petri *conflict-free*.

[CHE 95] afirma que o problema *deadlock* é decidível para redes de Petri arbitrárias, PSPACE-Completo para redes de Petri 1-*safe*, tempo linear para redes de Petri acíclicas, tempo constante para redes de Petri 1-*safe* acíclicas, polinomial para redes de Petri *conflict-free*, polinomial para redes de Petri 1-*safe conflict-free*, NP-Completo para redes de Petri *free-choice* e NP-Completo para redes de Petri 1-*safe free-choice*.

[DEM 2001] faz um levantamento de alguns jogos combinatoriais, dividindo-os em jogos com *loop* e jogos sem *loop*. Os jogos com *loop* são aqueles onde é possível retornar a posições anteriormente assumidas, como acontece por exemplo, em um jogo de xadrez. Os jogos sem *loop* por sua vez, são aqueles em que os jogadores têm os mesmos movimentos disponíveis, sem repeti-los.

Assim, [DEM 2001] conclui que muitos jogos sem *loop* são PSPACE-Completo, o que resulta de um tempo exponencial e um espaço polinomial. Por outro lado, os jogos com *loop* são identificados como EXPTIME-Completo (que é a subclasse dos problemas mais difíceis entre os problemas que têm complexidade EXPTIME), sendo este resultado é obtido em um tempo exponencial no pior caso.

Esta denominação de jogo com *loop* e jogo sem *loop* apresentada por [DEM 2001], pode ser comparada ao jogo lobos e cordeiro da seguinte maneira: pelo fato do cordeiro poder movimentar-se tanto para frente quanto para trás é possível que ele venha a assumir posições que já tenha assumido anteriormente, significando um jogo com *loop*. No caso dos lobos, pelo fato de poderem movimentar-se somente para frente não é possível que eles venham a reassumir posições anteriores, significando um jogo sem *loop*.

A situação de *loop* pode ser alterada, ou seja, é possível executar um *loop* e depois deixar este *loop*. Assim, a volta de um jogador a mesma posição pode até ser estratégico e útil. Apesar do fato de jogos com esta característica precisarem de um cuidado especial, somente o *loop* de estado – isto é, não seria a volta de um jogador a mesma posição, mas sim seria a volta do jogo ao mesmo estado – é uma situação não desejável e se caracteriza como um *deadlock*.

Uma vez que o jogo diferencial lobos e cordeiro foi especificado formalmente e algumas situações puderam ser identificadas, chega-se a conclusões que são interessantes de serem apresentadas. Por isso, na secção 4.7 a seguir, mostra-se tais conclusões a fim de reforçar o conhecimento adquirido.

4.7 Conclusões

Conforme apresentado neste capítulo, inicialmente foram definidos o espaço do jogo e os jogadores do jogo diferencial lobos e cordeiro, partindo-se logo após para as regras do jogo. Na seqüência, foram mostrados dois exemplos de partidas para o jogo proposto. Depois disso, houve a definição das estratégias para este jogo. Finalmente, apresentou-se a especificação formal do jogo lobos e cordeiro.

Com isso, na secção 4.1 pôde-se definir o jogo lobos e cordeiro propriamente dito, que foi o ponto de partida para que se pudesse estabelecer o procedimento de estudo dos jogos diferenciais. Para definir o jogo foi necessário identificar os elementos do jogo como, os jogadores, o espaço do jogo, as características de cada jogador, etc. Este jogo foi escolhido pelo fato de ser simples e assim, poder analisar melhor as características de um jogo.

A secção 4.2 apresentou as regras para simulação do jogo proposto, as quais possibilitaram a realização do jogo pelos jogadores segundo estas regras estabelecidas. Tais regras são definidas de forma a ter-se um jogador vencedor no final do jogo.

Na secção 4.3 foram ilustrados dois exemplos de partidas – mostrando as situações que ocorriam em cada um delas – para que se pudesse usar a definição e as regras

instituídas para o jogo lobos e cordeiro. A evolução destas situações de jogo aconteceu até que se tivesse um jogador vencedor (no jogo *A* o vencedor foi o cordeiro e no jogo *B* o vencedor foram os lobos) e então se teve uma finalização do jogo.

A secção 4.4 abordou as estratégias que cada jogador poderia seguir durante o jogo e assim, observou-se que os jogadores possuem estratégias ofensivas e defensivas, as quais foram identificadas segundo os objetivos de cada um dos jogadores.

Na secção 4.5 definiu-se a especificação formal para o jogo diferencial lobos e cordeiro. Esta especificação permitiu modelar através de redes de Petri condição/evento, as características do jogo, bem como o seu comportamento. Desta especificação para o jogo surgiu a rede *RPJ*, com a qual pôde-se observar propriedades do jogo como, alcançabilidade, *liveness* e *deadlock*.

A especificação proposta para o jogo lobos e cordeiro tem por finalidade identificar as situações que acontecem durante o jogo, mas que não são diretamente percebidas quando se está jogando, e prever como o jogo deve proceder quando tais situações ocorrem.

Futuramente, a especificação formal proposta auxiliará no projeto de algoritmos que serão desenvolvidos para o jogo lobos e cordeiro e também, pode-se utilizar esta especificação para modelar outros jogos mais complexos, já que o jogo lobos e cordeiro possui características destes jogos mais complexos.

A secção 4.6 apresentou algumas situações do jogo diferencial lobos e cordeiro, baseadas na especificação formal proposta para o jogo e que caracterizaram a alcançabilidade, o *liveness* e o *deadlock*. Igualmente, foram mostrados alguns resultados encontrados na literatura para as redes de Petri e para os jogos, no que diz respeito a estas propriedades de alcançabilidade, de *liveness* e de *deadlock*, fazendo-se a comparação das características destes jogos com o jogo diferencial lobos e cordeiro.

Com base nas características de outros jogos, apresenta-se na subsecção 4.7.1 a seguir o que tais características representariam para o jogo diferencial lobos e cordeiro.

4.7.1 Características de Jogos em Geral

Há características de jogos que são bastante comuns. Algumas estão presentes no jogo lobos e cordeiro, outras são facilmente simuladas. Esse fato torna o estudo de um jogo simples como o jogo lobos e cordeiro de grande utilidade.

Duas características de jogos em geral podem ser facilmente simuladas para o jogo lobos e cordeiro, tais como:

- Velocidades diferentes para os jogadores (onde seria possível, por exemplo, que os lobos realizassem duas jogadas consecutivas para cada uma jogada realizada pelo cordeiro).
- Obstáculos existentes no espaço do jogo é um caso já tratado, uma vez que, por exemplo, um lobo é um obstáculo para o cordeiro; este obstáculo é móvel e precisa ser contornado.

Estas características seriam implementadas da seguinte forma na rede de Petri proposta para o jogo lobos e cordeiro:

- Velocidades diferentes para os jogadores → ao invés do jogador passar o *token* vez ao próximo jogador (o jogador adversário), logo após ter realizado um movimento, ele permaneceria com esse *token* vez e realizaria, por exemplo, dois movimentos consecutivos. Ou seja, o evento que passaria o *token* vez para o outro jogador, ao contrário disto, passaria o *token* vez novamente para o mesmo jogador, dando a idéia de maior velocidade. Exemplificando esta situação, com base na rede *RPJ* da figura 4.15 tem-se a figura 4.22.

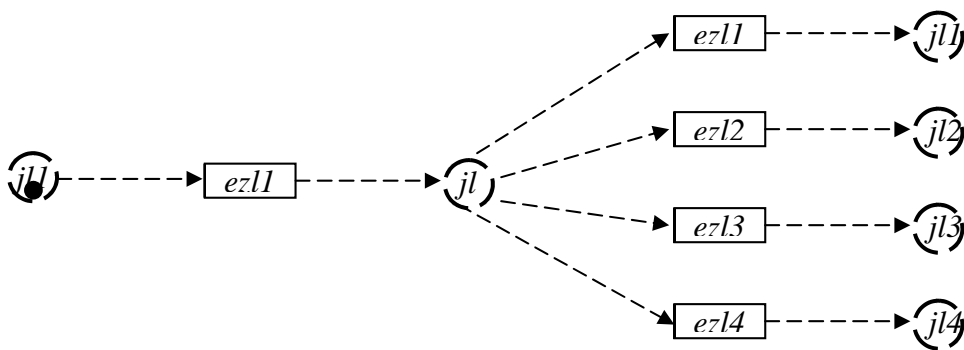


Figura 4.22: Rede *RPJ* para velocidades diferentes para os jogadores.

- Obstáculos existentes no espaço do jogo → o jogador se movimenta para ocupar condições que estejam livres na rede e caso as condições estejam ocupadas, o jogador tem que se movimentar para uma condição próxima àquela que está ocupada, dando a idéia de contornar o obstáculo que há pela frente. Exemplificando esta situação, tem-se a figura 4.23, onde a condição *c18c* está ocupada (figura 4.23a), então é realizado o movimento para a condição *c19c* (figura 4.23b). Lembre-se que *c15c*, *c18c* e *c19c* referem-se à rede do cordeiro e *c18* e *c19* referem-se à rede de controle da vez de jogar e das condições livres.

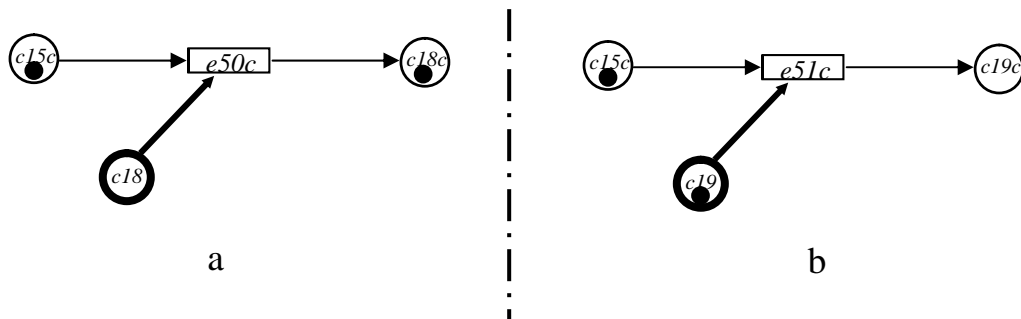


Figura 4.23: Rede *RPJ* para obstáculos existentes ao longo do espaço do jogo.

5 PROBLEMAS RESOLVIDOS DO JOGO DIFERENCIAL “LOBOS E CORDEIRO”

Ao definir-se um jogo, sabe-se que há questões relevantes sobre o mesmo jogo (por exemplo, as questões dos jogos diferenciais encontradas na secção 2.2) e conseqüentemente, estas questões precisam ser respondidas; isto, para que se possa realizar uma análise mais completa das situações de jogo. Como exemplo destas questões pode-se citar aquela que diz respeito à vitória de um dos jogadores ou até mesmo, de ambos jogadores envolvidos no jogo.

A partir de agora, apresentam-se os problemas do jogo diferencial lobos e cordeiro, os quais precisam ser resolvidos.

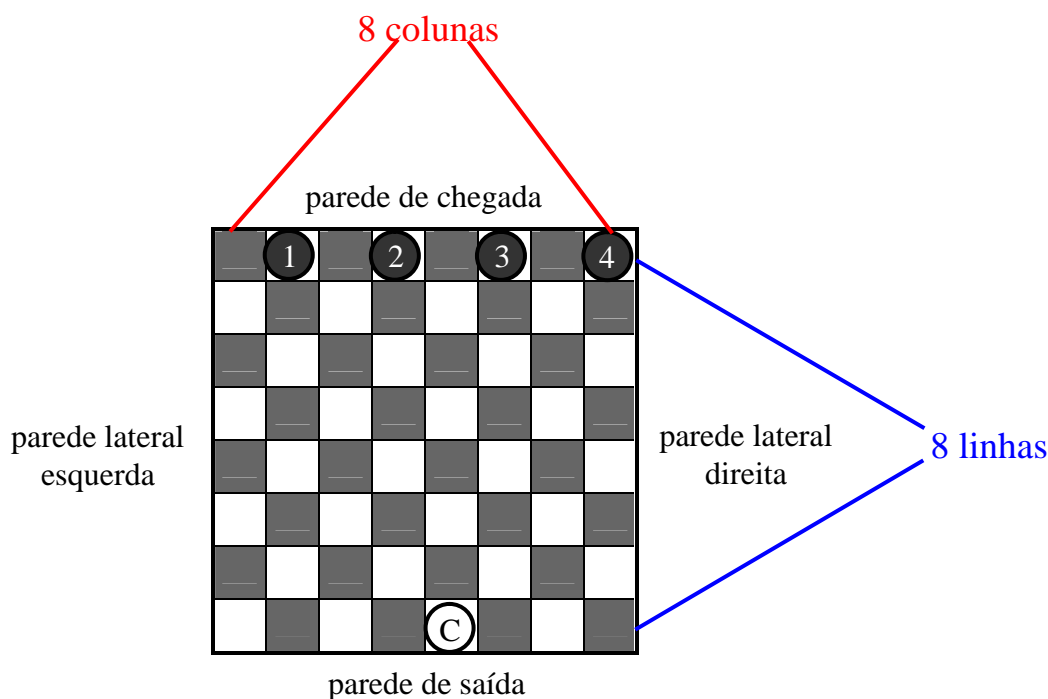


Figura 5.1: Tabuleiro do jogo “lobos e cordeiro”, de dimensão 8x8.

Algumas variações nas regras do jogo podem ser estudadas e as alterações no sistema indicadas, já que uma das utilidades da especificação formal é a capacidade de simular tais situações com poucas e bem localizadas (determinadas) alterações.

Inicialmente, no que diz respeito ao jogo lobos e cordeiro, sabe-se que para atender as suas definições é necessário que o tamanho do tabuleiro seja finito, ou seja, suas dimensões precisam ser pré-estabelecidas. Tal conclusão foi encontrada pois:

- os lobos somente “prendem” o cordeiro com o auxílio das paredes laterais esquerda e direita e/ou da parede de saída (vistas na figura 5.1), uma vez que eles não podem se movimentar para trás. Assim sendo, sem a existência destas paredes a ação de “prender” seria mais difícil de ser possível;

- o cordeiro poderia movimentar-se infinitamente para trás, à medida que os lobos se movimentassem em sua direção, se não houvesse a parede de saída (mostrada na figura 5.1). Desta forma, os lobos não conseguiriam prender o cordeiro, fazendo, por exemplo, com que ambos – lobos e cordeiro – movimentassem-se infinitamente para trás, ou seja, na direção em que o cordeiro se deslocaria. Esta situação caracterizaria um jogo sem término;

- o cordeiro não alcançaria a parede de chegada (mostrada na figura 5.1), enquanto se movimentasse, se não houvesse a parede de chegada. O cordeiro poderia “passar” pelos lobos, mas não conseguiria atingir a parede de chegada, porque ela simplesmente não existiria!

O que poderia estar previsto são diferentes versões do jogo lobos e cordeiro. Ou seja, poder-se-ia ter tamanhos de tabuleiros diferentes daquele definido originalmente para o jogo, tendo variações no seu número de linhas e no seu número de colunas. Da mesma forma, existiria a possibilidade de aumentar-se ou diminuir-se, conforme se desejasse, o número de jogadores – tanto lobos, quanto cordeiro.

Seria possível por exemplo, ter tabuleiros com dimensões 10x10 (10 colunas e 10 linhas), 15x9 (15 colunas e 9 linhas), ou quaisquer outras, dependendo da situação que se desejasse jogar.

É interessante no entanto, manter-se uma proporção entre o número de colunas do tabuleiro e o número de lobos do jogo, uma vez que os lobos somente conseguem formar uma “barreira” (segundo definição de uma de suas estratégias na seção 4.4), se esta proporção for obedecida.

Poder-se-ia também, na medida em que se aumentassem as dimensões do tabuleiro, aumentar, além do número de lobos, o número de cordeiros, dependendo da situação que se desejasse jogar.

Em se tratando de problemas encontrados no jogo lobos e cordeiro, pode-se dizer que estes problemas são divididos em duas categorias: os problemas simples e os problemas não simples. Na seqüência, descreve-se cada uma das categorias de problemas citadas.

5.1 Problemas Simples (não dependentes de estratégias)

A seguir são descritas situações de jogo levando-se em consideração o número de passos, mínimo e máximo, de cada jogador e o objetivo de cada um deles em relação a este número de passos. Estas situações são aplicadas em um tabuleiro de dimensão 8x8,

aos problemas simples – aqueles que não dependem das estratégias de jogo de cada jogador.

A figura 5.2 mostra a situação inicial do jogo que é a base utilizada para prever o número de passos mínimos e máximos para o cordeiro e os lobos vencerem, tanto para os problemas simples quanto para os problemas não simples. Observa-se que esta figura 5.2 foi previamente explicada na secção 4.1.

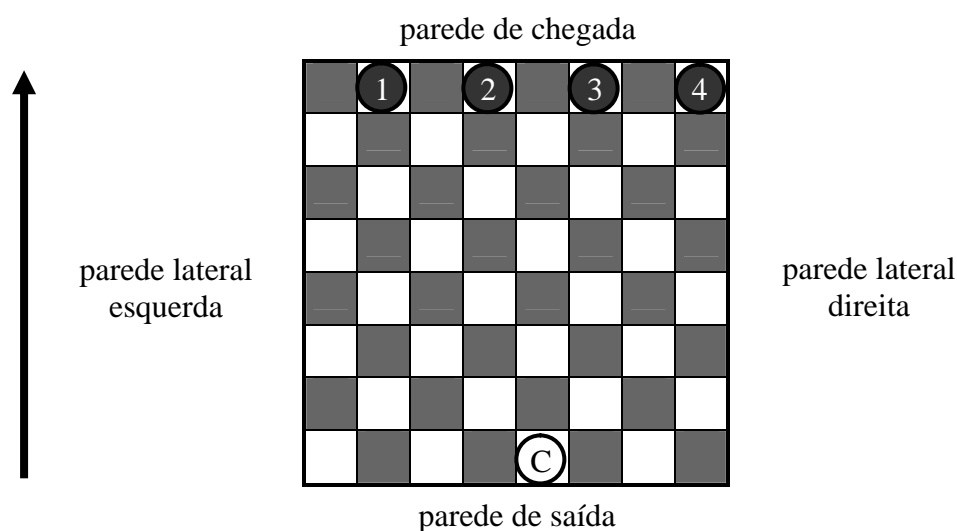


Figura 5.2: Situação inicial do jogo.

As seguintes figuras que ilustram as diferentes situações de jogo, apresentam ao longo do tabuleiro a letra C que representa os movimentos do cordeiro e os números 1, 2, 3 e 4 que representam respectivamente, os movimentos do lobo 1, do lobo 2, do lobo 3 e do lobo 4.

A) Número de passos mínimos para o cordeiro vencer o jogo:

Alcançar a parede de chegada – Com os lobos movimentando-se em direção a uma das paredes laterais, esquerda ou direita - sem a intenção de formar uma barreira que impeça o cordeiro de passar - (poderia dizer-se “burramente”), o cordeiro alcança, pelo lado da parede lateral oposta àquela que os lobos se movimentaram, a parede de chegada em 7 jogadas, movimentando-se somente para frente e não encontrando lobo algum como obstáculo, ou seja, em sua frente. Esta situação de jogo é ilustrada na figura 5.3.

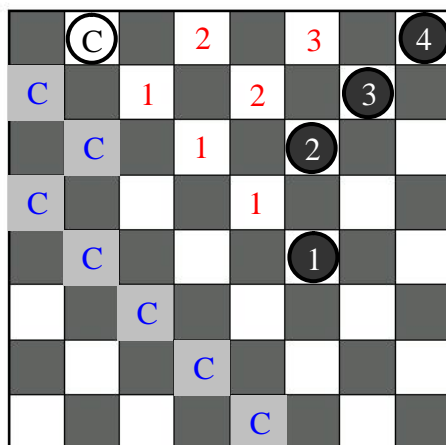


Figura 5.3: Final de jogo: número de passos mínimos para o cordeiro vencer o jogo.

Poder-se-ia dizer então, que o número mínimo de passos para o cordeiro vencer o jogo é atribuído ao número mínimo de posições do tabuleiro, que o cordeiro precisa percorrer até alcançar a parede de chegada; isto é, o número de linhas que o cordeiro vem a assumir descontado da linha que o cordeiro parte (8 linhas – 1 linha = 7 linhas), generalizando tem-se $li-1$, onde li é o número de linhas.

B) Número de passos mínimos para os lobos vencerem o jogo:

Prender o cordeiro – Com o cordeiro movimentando-se para frente e para trás - sem a intenção de passar os lobos - (poderia dizer-se burramente), os lobos prendem o cordeiro na parede lateral esquerda em 6 jogadas. Esta situação de jogo é ilustrada na figura 5.4.

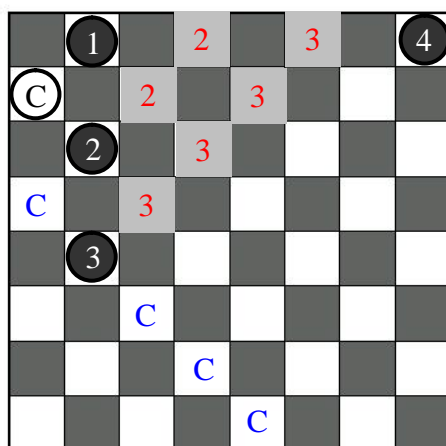


Figura 5.4: Final de jogo: número de passos mínimos para os lobos vencerem o jogo.

Poder-se-ia dizer então, que o número mínimo de passos para os lobos vencerem o jogo é atribuído ao número mínimo de posições do tabuleiro, as quais os lobos precisam percorrer até prender o cordeiro; isto é, o número de linhas que o cordeiro vem a assumir descontado das linhas que o cordeiro e os lobos partem (8 linhas – 2 linhas = 6 linhas), generalizando tem-se $li-2$, onde li é o número de linhas.

C) Número de passos máximos para o cordeiro vencer o jogo:

Alcançar a parede de chegada – Com os lobos movimentando-se ao longo do tabuleiro - sem a intenção de formar uma barreira que impeça o cordeiro de passar - (poderia dizer-se burramente), o cordeiro alcança a parede de chegada em 28 jogadas, movimentando-se para frente e para trás. Esta situação de jogo é ilustrada na figura 5.5.

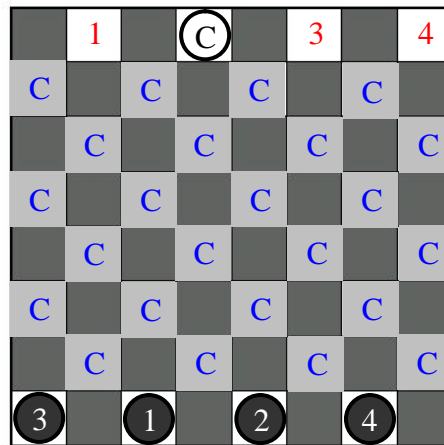


Figura 5.5: Final de jogo: número de passos máximos para o cordeiro vencer o jogo.

Poder-se-ia dizer então, que o número máximo de passos para o cordeiro vencer o jogo é atribuído ao número máximo de posições do tabuleiro, as quais o cordeiro precisa percorrer até alcançar a parede de chegada; isto é, o número total de posições do tabuleiro descontado do número de posições ocupadas pelos lobos (32 posições – 4 posições = 28 posições), generalizando tem-se $n-4$, onde n é o tamanho do tabuleiro, ou seja, o seu número de posições.

D) Número de passos máximos para os lobos vencerem o jogo:

Prender o cordeiro – Com o cordeiro movimentando-se para frente e para trás - sem a intenção de passar os lobos - (poderia dizer-se burramente), os lobos prendem o cordeiro na parede de saída em 26 jogadas. Esta situação de jogo é ilustrada na figura 5.6.

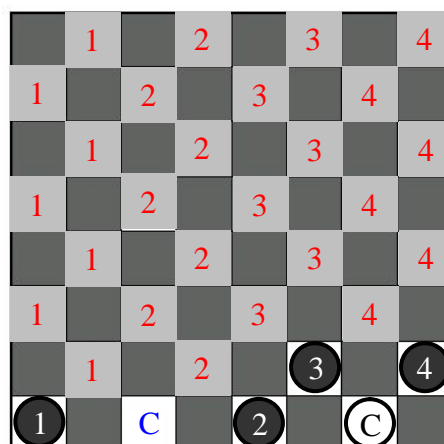


Figura 5.6: Final de jogo: número de passos máximos para os lobos vencerem o jogo.

Poder-se-ia dizer então, que o número máximo de passos para os lobos vencerem o jogo é atribuído ao número máximo de posições do tabuleiro, as quais os lobos precisam percorrer até prenderem o cordeiro no ponto mais distante deste tabuleiro (a parede de saída). Isto é, o número total de posições do tabuleiro descontado do número de posições ocupadas pelos lobos, descontado do número de posições ocupadas pelos lobos que ficaram na penúltima linha do tabuleiro ($(32 \text{ posições} - 4 \text{ posições}) - 2 \text{ posições} = 26 \text{ posições}$), generalizando tem-se $(n-4)-2$, onde n é o tamanho do tabuleiro, ou seja, o seu número de posições.

Estas situações de jogo apresentadas na figura 5.3, na figura 5.4, na figura 5.5 e na figura 5.6 foram analisadas para o número mínimo e máximo de passos necessários para o cordeiro e os lobos vencerem o jogo, sem o uso da especificação formal proposta para o jogo. Por isso, o próximo passo é transpor estas situações de jogo para as definições da rede *RPJ* (segundo as definições da secção 4.5).

A especificação formal do jogo diferencial lobos e cordeiro é importante, porque possibilita identificar as situações que acontecem ao longo do jogo que executa na rede *RPJ* e assim, pode-se definir condições e eventos, por exemplo, para a rede não entrar em *deadlock*, ou também, para a rede permanecer “viva” (*live*) para chegar a uma próxima situação do jogo.

A figura 5.7 ilustra a situação da figura 5.3 aplicada à rede *RPJ*. Nesta situação, os lobos estão ocupando as condições *c29*, *c25*, *c22* e *c14* (representadas na camada de rede dos lobos, figura 5.7a) e o cordeiro está ocupando a condição *c32* (representada na camada de rede do cordeiro, figura 5.7b).

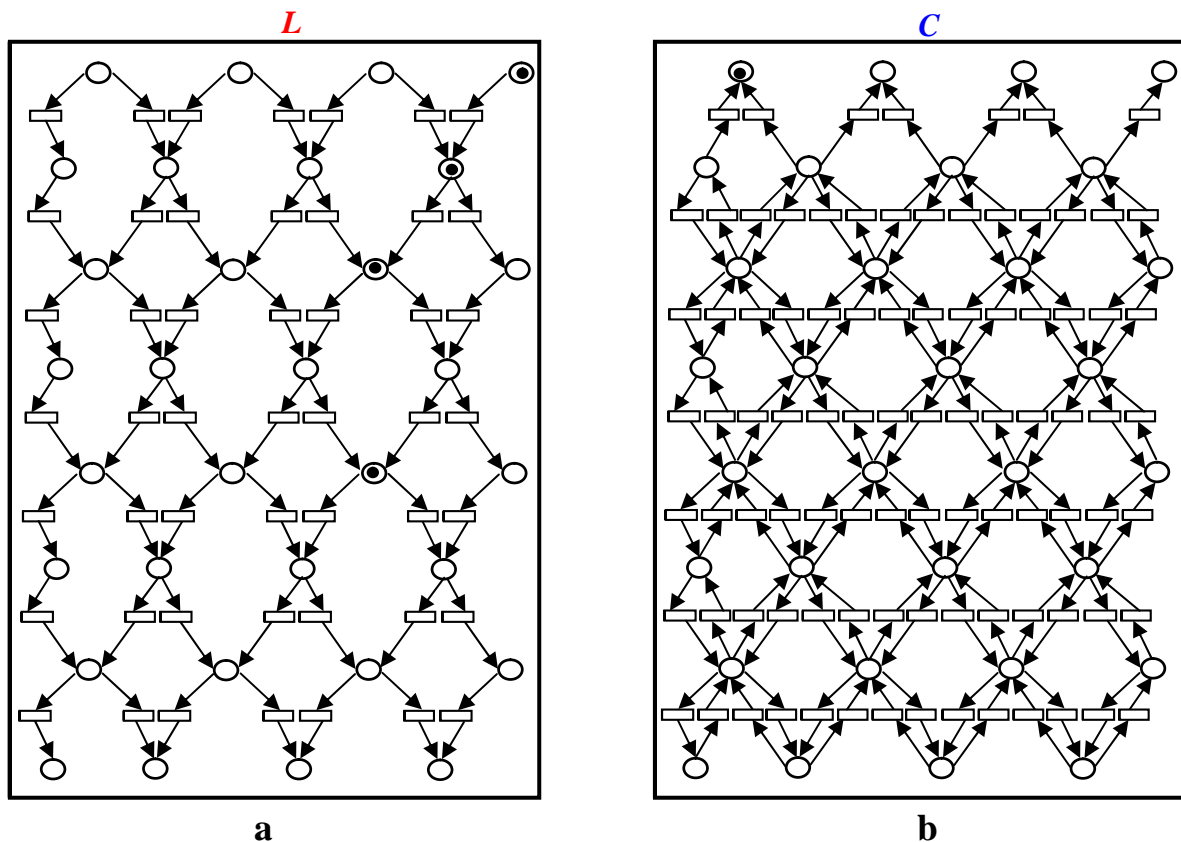


Figura 5.7: Situação da figura 5.3 aplicada à rede *RPJ*.

A figura 5.8 ilustra a situação da figura 5.4 aplicada à rede *RPJ*. Nesta situação, os lobos estão ocupando as condições *c32*, *c29*, *c24* e *c16* (representadas na camada de rede dos lobos, figura 5.8a) e o cordeiro está ocupando a condição *c28* (representada na camada de rede do cordeiro, figura 5.8b).

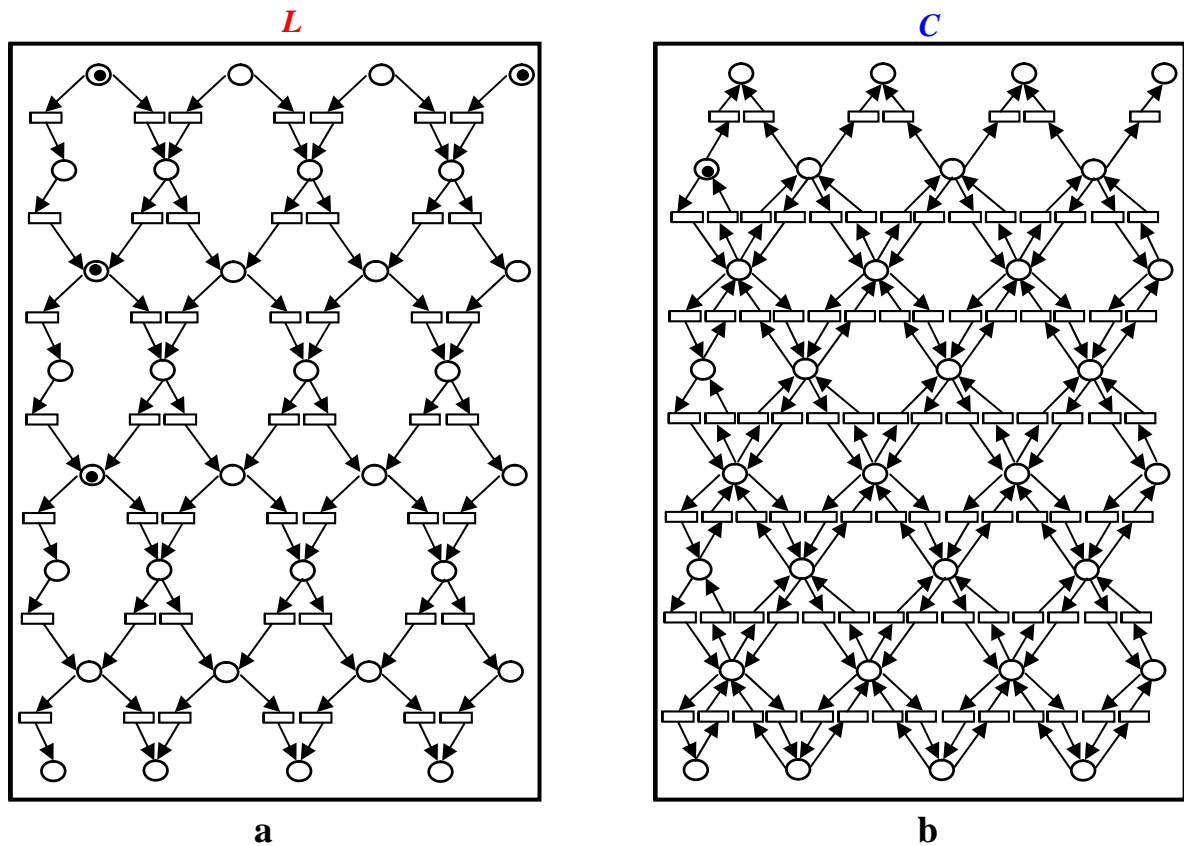


Figura 5.8: Situação da figura 5.4 aplicada à rede *RPJ*.

A figura 5.9 ilustra a situação da figura 5.5 aplicada à rede *RPJ*. Nesta situação, os lobos estão ocupando as condições *c4*, *c3*, *c2* e *c1* (representadas na camada de rede dos lobos, figura 5.9a) e o cordeiro está ocupando a condição *c31* (representada na camada de rede do cordeiro, figura 5.9b).

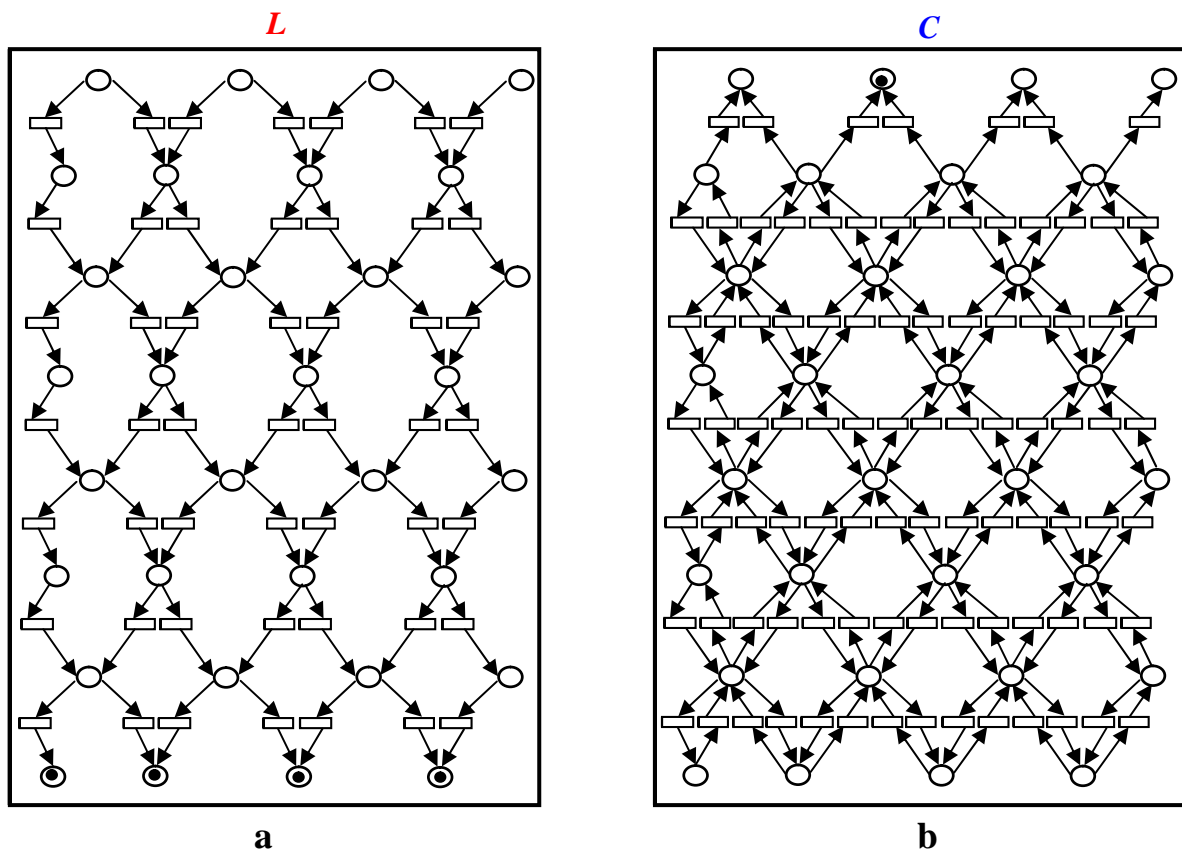


Figura 5.9: Situação da figura 5.5 aplicada à rede *RPJ*.

A figura 5.10 ilustra a situação da figura 5.6 aplicada à rede *RPJ*. Nesta situação, os lobos estão ocupando as condições *c6*, *c5*, *c4* e *c2* (representadas na camada de rede dos lobos, figura 5.10a) e o cordeiro está ocupando a condição *p1* (representada na camada de rede do cordeiro, figura 5.10b).

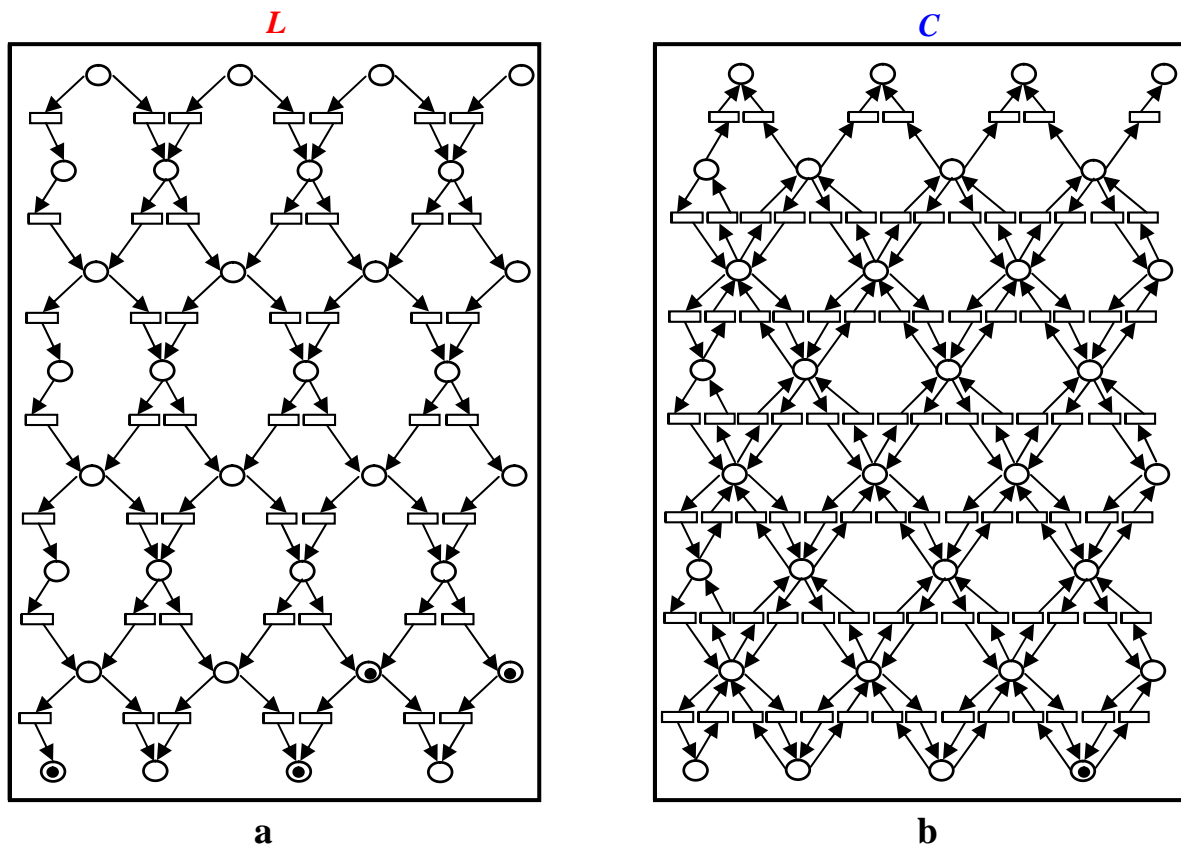


Figura 5.10: Situação da figura 5.6 aplicada à rede *RPJ*.

5.2 Problemas Não Simples (dependentes de estratégias)

A seguir são descritas situações de jogo levando-se em consideração o número de passos, mínimo e máximo, de cada jogador e o objetivo de cada um deles em relação a este número de passos. Estas situações são aplicadas em um tabuleiro de dimensão 8x8, aos problemas não simples – aqueles que dependem das estratégias de jogo de cada jogador.

A) Número de passos mínimos para o cordeiro vencer o jogo:

Alcançar a parede de chegada – Com os lobos movimentando-se com a intenção de formar uma barreira, o cordeiro passa pelos lobos e alcança a parede de chegada em 9 jogadas. Esta situação de jogo é ilustrada na figura 5.11.

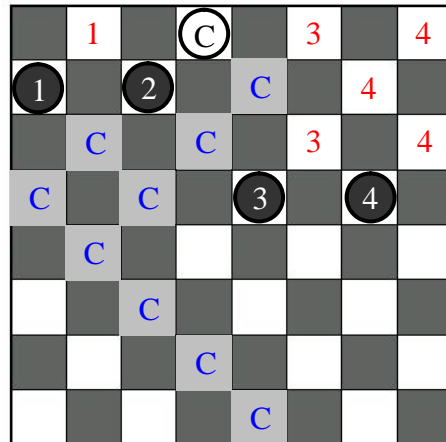


Figura 5.11: Final de jogo: o cordeiro alcança a parede de chegada, após 9 jogadas.

O cordeiro consegue passar pelos lobos, fazendo com que eles abram uma “brecha” na barreira por eles formada.

Passar os lobos – Com os lobos movimentando-se com a intenção de formar uma barreira, o cordeiro passa pelos lobos em 8 jogadas. Nesta situação o cordeiro ainda pode alcançar a parede de chegada, pois além dele, os lobos também podem realizar jogadas. Esta situação de jogo é ilustrada na figura 5.12.

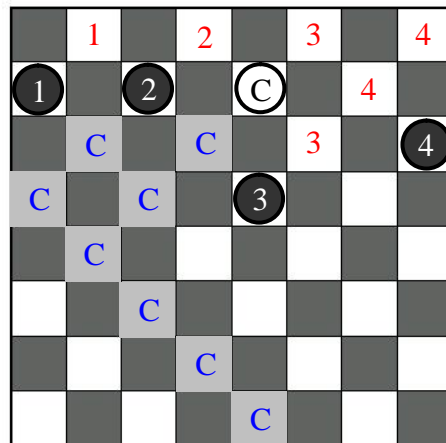


Figura 5.12: O cordeiro passa os lobos.

Para chegar-se à situação da figura 5.12, adotou-se a estratégia representada por parte da rede *RPJ* da figura 5.13.

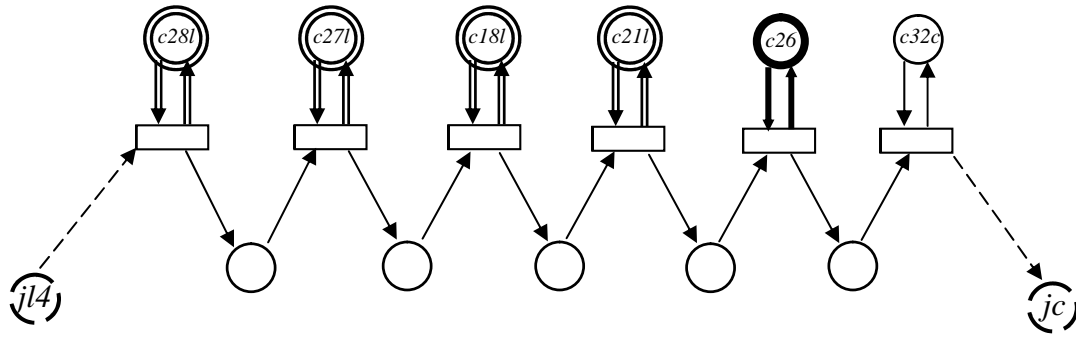


Figura 5.13: Parte da rede *RPJ* para chegar à situação de jogo da figura 5.12.

Para relembrar, sabe-se que para toda e qualquer estratégia que se queira aplicar no jogo é necessário, no contexto da rede *RPJ*, definir todas as situações que envolvam tais estratégias. Ou seja, é preciso prever as condições e os eventos que serão utilizados para que a rede possa seguir sua execução, segundo uma estratégia estabelecida.

Assim, a especificação formal do jogo é importante também quando se modelam as estratégias, pois através das condições e dos eventos da rede *RPJ*, pode-se identificar se a rede chega a determinada configuração e a partir desta configuração, aplicar uma determinada estratégia do jogo para alcançar assim, uma situação de vitória para um dos jogadores.

Como já se observou, as estratégias podem ser implementadas na rede *RPJ*, pois uma vez que se tem uma determinada situação nesta rede, define-se então as condições e os eventos que irão atuar naquele determinado momento para que a rede assuma uma outra situação, por intermédio da aplicação de uma estratégia.

B) Número de passos mínimos para os lobos vencerem o jogo:

Prender o cordeiro – Com o cordeiro movimentando-se para frente e para trás - com a intenção de passar os lobos - os lobos prendem o cordeiro na parede lateral esquerda em 6 jogadas. Esta situação de jogo é ilustrada na figura 5.14.

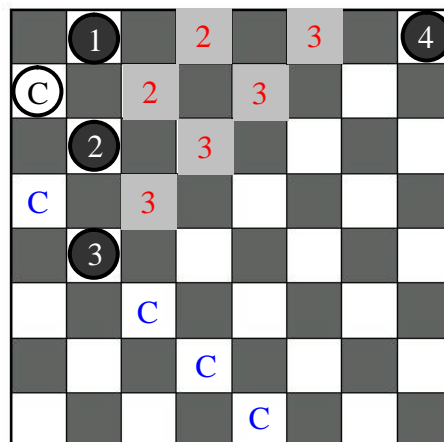


Figura 5.14: Final de jogo: os lobos prendem o cordeiro, após 6 jogadas.

Os lobos conseguem formar uma barreira, movimentando-se horizontalmente (ou verticalmente, ou diagonalmente, dependendo do caso) na medida em que se deslocam pelo tabuleiro.

Formar uma barreira – Com os lobos movimentando-se para formar uma barreira são necessárias 4 jogadas. Esta situação de jogo é ilustrada na figura 5.15.

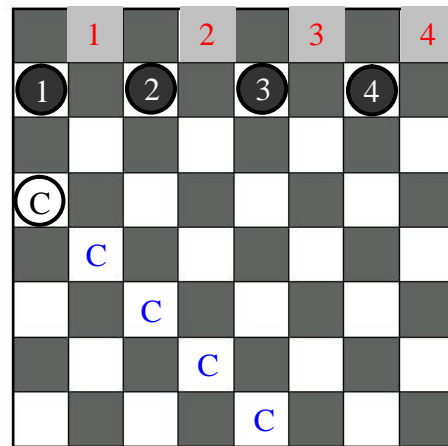


Figura 5.15: Os lobos formam uma barreira, após 4 jogadas.

Para chegar-se à situação da figura 5.15, adotou-se a estratégia representada por parte da rede *RPJ* da figura 5.16.

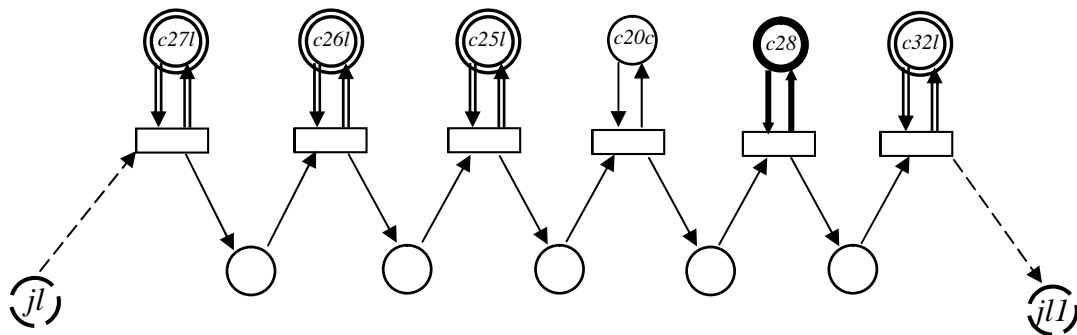


Figura 5.16: Parte da rede *RPJ* para chegar à situação de jogo da figura 5.15.

C) Número de passos máximos para o cordeiro vencer o jogo:

Alcançar a parede de chegada – Com os lobos movimentando-se com a intenção de formar uma barreira, o cordeiro passa pelos lobos e alcança a parede de chegada em 28 jogadas. Esta situação de jogo é ilustrada na figura 5.17.

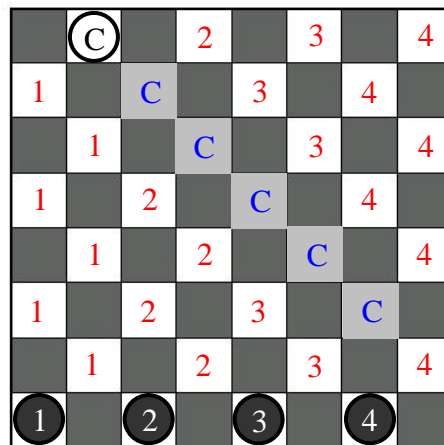


Figura 5.17: Final de jogo: o cordeiro alcança a parede de chegada, após 28 jogadas.

Passar os lobos – Com os lobos movimentando-se com a intenção de formar uma barreira, o cordeiro passa pelos lobos em 26 jogadas. Nesta situação o cordeiro não pode alcançar a parede de chegada – mesmo ainda tendo posições a percorrer, pois os lobos não conseguem realizar mais do que duas jogadas, pois estão no seu limite de jogo. Esta situação de jogo é ilustrada na figura 5.18.

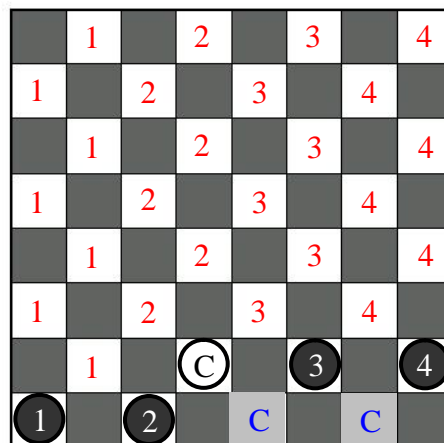


Figura 5.18: O cordeiro passa os lobos, mas não pode alcançar a parede de chegada.

Para chegar-se à situação da figura 5.18, adotou-se a estratégia representada por parte da rede *RPJ* da figura 5.19.

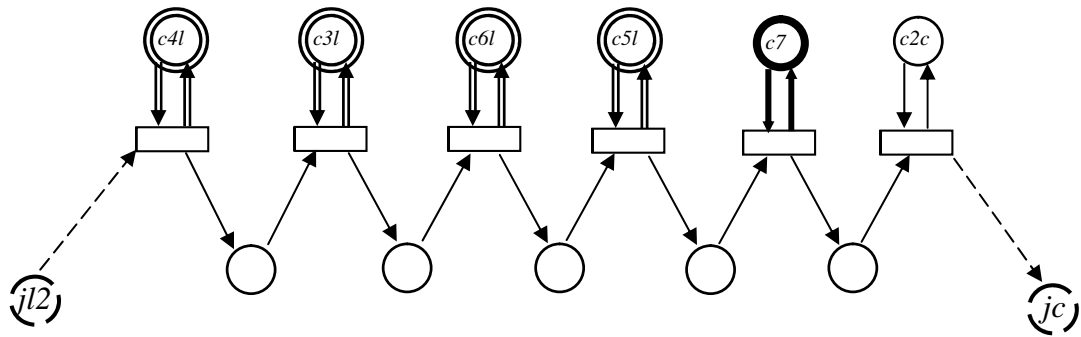


Figura 5.19: Parte da rede *RPJ* para chegar à situação de jogo da figura 5.18.

D) Número de passos máximos para os lobos vencerem o jogo:

Prender o cordeiro – Com o cordeiro movimentando-se para frente e para trás - com a intenção de passar os lobos - os lobos prendem o cordeiro na parede de saída em 26 jogadas. Esta situação de jogo é ilustrada na figura 5.20.

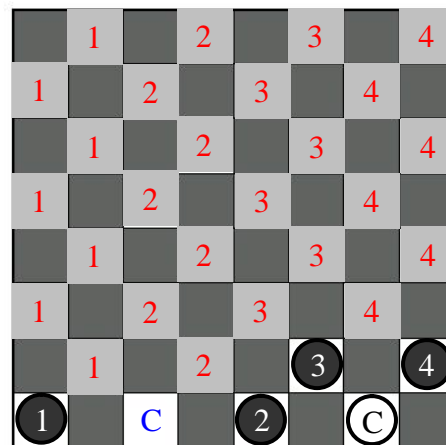


Figura 5.20: Final de jogo: os lobos prendem o cordeiro, após 26 jogadas.

Formar uma barreira – Com os lobos movimentando-se para formar uma barreira são necessárias 24 jogadas. Nesta situação também, o cordeiro já está preso. Esta situação de jogo é ilustrada na figura 5.21.

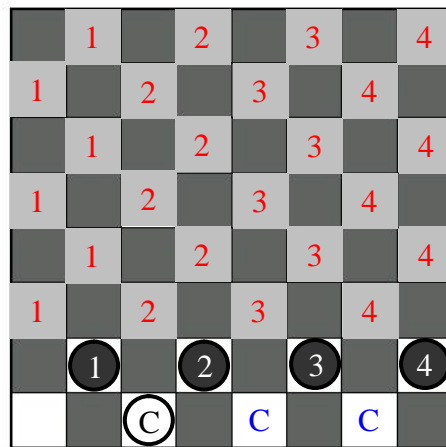


Figura 5.21: Os lobos formam uma barreira, após 24 jogadas.

Para chegar-se à situação da figura 5.21, adotou-se a estratégia representada por parte da rede *RPJ* da figura 5.22.

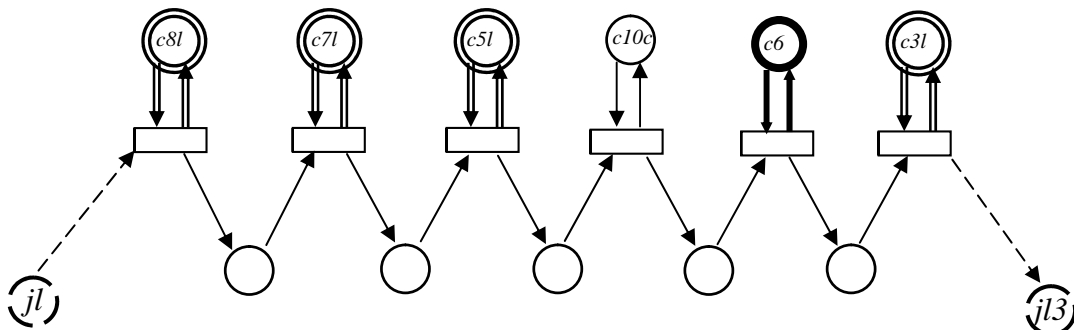


Figura 5.22: Parte da rede *RPJ* para chegar à situação de jogo da figura 5.21.

Neste ponto do trabalho, já que se está tratando de estratégias do jogo, pode-se notar o uso das camadas em que o jogo lobos e cordeiro foi dividido na sua especificação formal de redes de Petri.

As diferentes camadas da rede de Petri facilitariam, por exemplo, a especificação de um jogo mais complexo, porque poderia ter-se diversas visões do jogo desejado e assim, analisar-se separadamente cada uma destas visões para extrair o comportamento que a rede de Petri devesse possuir, conforme o caso desejado. Além disso, para o caso de um jogo mais complexo, conseguiria definir-se as suas estratégias de modo a entender o que elas realmente refletiriam na execução da rede.

Assim, os algoritmos de estratégias são baseados nas estratégias de cada jogador, considerando também as possíveis jogadas realizadas por cada um deles. A próxima secção 5.3 tem por finalidade mostrar a idéia que fundamenta estes algoritmos de estratégias.

5.3 Algoritmos de Estratégias

Uma boa estratégia de jogo considera várias jogadas de avanço, ao tomar uma decisão para escolher o próximo movimento a ser realizado no jogo.

No contexto do jogo diferencial lobos e cordeiro, adota-se a denominação dos algoritmos de estratégias, porque se tem por base as possíveis jogadas realizadas por cada um dos jogadores.

Iniciando-se então, pelos lobos constata-se o seguinte: cada um dos quatro lobos tem, em uma jogada, duas possibilidades de movimentação para ocupar uma determinada posição, o que corresponde a duas posições possíveis de serem ocupadas por um lobo a cada jogada. Assim, o resultado das possibilidades de movimentação para os lobos em cada jogada é obtido pelo número de lobos – quatro – multiplicado pelas possibilidades de movimentação de cada um dos lobos – duas. Ou seja, *4 lobos x 2 possibilidades de movimentação de cada lobo = 8 possibilidades de movimentação em cada jogada*.

Na sequência, partindo-se para o cordeiro e constata-se o seguinte: o cordeiro tem, em uma jogada, quatro possibilidades de movimentação para ocupar uma determinada posição, o que corresponde a quatro posições possíveis de serem ocupadas pelo cordeiro a cada jogada. Assim, o resultado das possibilidades de movimentação para o cordeiro em cada jogada é obtido simplesmente pelas possibilidades de movimentação do cordeiro. Ou seja, *4 possibilidades de movimentação em cada jogada*.

Tendo-se então este número de possibilidades de movimentação dos jogadores em cada jogada, ao estar-se aplicando estratégias, é preciso identificar as possibilidades de movimentação em cada jogada considerando-se todos os jogadores (ou seja, prever as possíveis jogadas futuras que os jogadores podem realizar).

Assim, como os lobos têm 8 possibilidades de jogadas e o cordeiro tem 4 possibilidades de jogadas, o jogador que aplicar alguma estratégia necessita analisar todas as possibilidades de movimentação que podem acontecer em uma jogada para todos os jogadores. Chega-se então, no jogo lobos e cordeiro, às 32 possibilidades de movimentação que precisam ser analisadas em cada jogada (*8 possibilidades de movimentação em cada jogada para os lobos x 4 possibilidades de movimentação em cada jogada para o cordeiro = 32 possibilidades de movimentação analisadas em cada jogada*).

Além disso, é preciso também, prever o número máximo de jogadas de avanço que os lobos podem realizar ao longo do jogo. Identificou-se então no jogo lobos e cordeiro, que o número máximo de jogadas de avanço está diretamente relacionado ao número de linhas que os lobos podem percorrer ao longo do jogo, ou seja, sete linhas que correspondem às 7 jogadas de avanço ao longo do jogo (onde o número máximo de jogadas de avanço é representado por k).

Igualmente como aconteceu para os lobos, é preciso prever o número máximo de jogadas de avanço que o cordeiro pode realizar ao longo do jogo. Identificou-se então, que o número máximo de jogadas de avanço também está diretamente relacionado ao número de linhas que o cordeiro pode percorrer ao longo do jogo, ou seja, sete linhas que correspondem às 7 jogadas de avanço ao longo do jogo.

Com isso, se desejasse-se prever as possíveis jogadas futuras de um jogador qualquer do jogo lobos e cordeiro, levando em consideração, por exemplo, duas jogadas

de avanço, teria que se analisar 32^2 possibilidades de movimentação, ou seja, $8 \times 4 \times 8 \times 4 = 32 \times 32 = 1.024$ possibilidades de movimentação.

Neste caso, para considerar sete jogadas de avanço (que é o número máximo de jogadas de avanço que pode ser previsto, já que o jogo é finito) teria que se analisar 32^7 possibilidades de movimentação, ou seja, $8 \times 4 \times 8 \times 4 \times 8 \times 4 \times 8 \times 4 \times 8 \times 4 \times 8 \times 4 = 32 \times 32 \times 32 \times 32 \times 32 \times 32 \times 32 = 34.359.737.368$ possibilidades de movimentação.

Pode-se então generalizar estas possibilidades de movimentação nos seguintes termos: *possibilidades de movimentação do jogador A, elevadas a k* *x* *possibilidades de movimentação do jogador B, elevadas a k*. Assim, a solução do jogo depende de um algoritmo trivial exponencial, que é um algoritmo ineficiente, não razoável.

Entretanto, a complexidade real pode ser consideravelmente menor. Assim, um estudo de amortização ou de complexidade média é conveniente, pois há a limitação das possibilidades de movimentos segundo uma estratégia de jogo; assim são consideradas somente as posições para as quais é estratégico o jogador se movimentar, sendo as demais posições desconsideradas. Por exemplo, como o cordeiro tem o objetivo de alcançar a parede de chegada, segundo a sua estratégia, então ele não vai se movimentar para trás, quando ele tem a possibilidade de se movimentar para frente.

Um estudo estatístico também pode diminuir as possibilidades de movimentos (seja segundo uma estratégia de jogo ou seja pelo andamento do jogo); por isso, uma simulação do jogo é útil para definir-se as estratégias do jogo. Com resultados estatísticos ter-se-ia condições de desenvolver um algoritmo aproximativo.

Em termos da especificação formal proposta para o jogo diferencial lobos e cordeiro, as estratégias que prevêm possíveis jogadas que podem ser realizadas pelos jogadores, são definidas com base nas condições ocupadas pelos *tokens* dos jogadores e assim, estas condições que estão ocupadas são então desconsideradas como jogadas possíveis, pois os seus eventos correspondentes às pré-condições não irão ocorrer devido ao fato desta condição já possuir um *token*.

A especificação formal proposta auxilia na execução do jogo, porque se pode prever situações quaisquer para aplicar estratégias do jogo, as situações de fim de jogo podem ser alcançadas e as situações de *deadlock* podem ser evitadas (propriedades apresentadas na secção 4.6).

Desta maneira, uma vez que os problemas resolvidos para o jogo lobos e cordeiro foram mostrados e a idéia dos algoritmos de estratégias foi apresentada, tem-se algumas conclusões que são interessantes de serem descritas. Por isso, a secção 5.4 apresenta tais conclusões a fim de reforçar o conhecimento adquirido.

5.4 Conclusões

Neste capítulo 5 foram apresentados os problemas resolvidos para o jogo diferencial lobos e cordeiro, tendo por base o número mínimo e máximo de jogadas para cada jogador – lobos e cordeiro. Chegou-se portanto, aos problemas simples (vistos na secção 5.1), ou aqueles que não dependem de estratégia e aos problemas não simples, ou aqueles que dependem de estratégia (vistos na secção 5.2). Também foram apresentadas as idéias dos algoritmos de estratégias, os quais tiveram como base as estratégias utilizadas pelos jogadores durante o jogo lobos e cordeiro.

Na secção 5.3 mostrou-se a ideia dos algoritmos de estratégias para o jogo lobos e cordeiro. Estes algoritmos precisam considerar todas as possibilidades de jogadas dos jogadores ao longo do jogo e em termos genéricos representaria: *possibilidades de movimentação do jogador A, elevadas a k x possibilidades de movimentação do jogador B, elevadas a k* .

Além disso, esclarece-se que o jogo diferencial lobos e cordeiro foi formalmente especificado usando redes de Petri, porque elas possibilitam a verificação de propriedades que são importantes em um sistema qualquer – neste caso o jogo proposto. Pôde-se então traduzir os problemas apresentados neste capítulo 5 para redes de Petri e analisar determinadas situações, constatando algumas propriedades como alcançabilidade, *liveness* e *deadlock* (conforme mostrado na secção 4.6).

Portanto, as questões inicialmente levantadas em jogos diferenciais (secção 2.2) puderam igualmente ser respondidas, pois as estratégias puderam ser definidas e especificadas, tanto para o cordeiro quanto para os lobos e desta forma, situações de vitória para ambos jogadores puderam ser identificadas.

Na seqüência da subsecção 5.4.1 apresentam-se as questões respondidas para o jogo diferencial lobos e cordeiro. Lembra-se que estas questões foram inicialmente levantadas para os jogos diferenciais na secção 2.2.

5.4.1 Questões Respondidas

Para as questões levantadas nos jogos diferenciais (questões definidas anteriormente na secção 2.2) são apresentados os resultados para estas questões no jogo diferencial lobos e cordeiro. Estas questões dizem respeito às estratégias aplicadas pelos jogadores, à vitória dos jogadores, ao menor ou maior período de tempo que um jogo acontece e a como identificar as estratégias do jogo.

Estes resultados foram obtidos com base: nas partidas que foram mostradas na secção 4.3, onde nestas simulações de partidas as estratégias do jogo foram identificadas para cada um dos jogadores; na especificação formal do jogo que foi feita na secção 4.5, onde foram definidas as situações de vitória para cada um dos jogadores; e nas definições da rede de Petri para que se chegasse às situações ilustradas pelas figuras (da figura 5.3 até a figura 5.22) da secção 5.1 e da secção 5.2, onde o número de passos mínimos e máximos foi definido para cada um dos jogadores alcançar uma determinada situação no jogo (seja de vitória, seja para aplicar uma estratégia).

Assim sendo, os resultados são:

- Para as estratégias aplicadas pelos jogadores → cada um dos jogadores tem estratégias específicas que são aplicadas ao longo do jogo; deste modo uma estratégia para os lobos é formar uma “barreira” horizontal, vertical, diagonal ou circular, para impedir que o cordeiro passe pelos lobos e alcance a parede de chegada. Uma estratégia para o cordeiro é abrir uma “brecha” na barreira dos lobos, para o cordeiro poder passar pelos lobos, podendo alcançar a parede de chegada. Isto foi definido na secção 4.4. Essas estratégias foram implementadas, podendo ser aplicadas em outros jogos.

- Para a vitória dos jogadores → cada um dos jogadores tem uma situação de vitória que é alcançada no final do jogo por um deles (ou cordeiro ou lobos). Os lobos vencem o jogo quando eles prendem o cordeiro e o cordeiro vence o jogo

quando ele alcança a parede de chegada ou quando ele passa pelos lobos, deixando-os “para trás”. Estas situações de vitória para os jogadores foram apresentadas na subsecção 4.5.2 e foram baseadas na especificação formal proposta para o jogo lobos e cordeiro. Estas se constituem no problema da alcançabilidade.

- Para o menor ou maior período de tempo → cada um dos jogadores pode realizar suas jogadas em um maior ou um menor período de tempo no jogo. Este tempo é definido em números de passos que são executados para se chegar, por exemplo, a uma situação de vitória. Através da simulação do jogo foi possível definir estes passos e em seguida foi possível identificar as situações resultantes da simulação na rede de Petri proposta para o jogo. Isto foi definido na secção 5.1 e na secção 5.2.

- Para identificar as estratégias do jogo → no jogo lobos e cordeiro cada um dos jogadores aplica estratégias próprias para alcançar seus objetivos. Estas estratégias foram inicialmente identificadas através de simulações do jogo apresentadas na secção 4.3 e posteriormente, estas estratégias foram definidas na rede *RPJ*, na secção 4.5.4.

É importante salientar que estas questões em jogos mais complexos têm tratamento análogo, portanto o estudo feito neste trabalho é também útil para análises futuras de jogos mais complexos.

6 CONCLUSÃO

Os conceitos dos jogos diferenciais são muito utilizados em geral, nos campos da economia, da pesquisa operacional, das estratégias militares (principalmente, no que diz respeito aos mísseis), da política, dentre outros encontrados na literatura. Especificamente, na ciência da computação, são empregados nas áreas da inteligência artificial, da computação distribuída, da segurança e privacidade, entre outras.

O jogo lobos e cordeiro é um exemplo simples de jogo diferencial que envolve quatro lobos e um cordeiro, que se movimentam alternadamente, em um tabuleiro de 64 casas (8 linhas e 8 colunas; onde destas 64 casas são utilizadas somente 32 casas, que representam as 32 posições que os jogadores podem assumir ao longo do jogo), não havendo restrições como velocidade e aceleração.

Este jogo foi escolhido para exemplificação de formalização justamente, pela simplicidade e pela facilidade de versões alternativas com características de sequencialidade (característica dos jogos usuais), de paralelismo com velocidades diferentes (os jogadores podem jogar independentemente), de número de peças variável (mais de um cordeiro, mais ou menos lobos), de espaço ilimitado (o tabuleiro pode ser estendido) e outras mais.

Até esse momento está especificado o padrão mais simples de jogo, no entanto quaisquer das alternativas citadas no parágrafo anterior são facilmente incluídas (como se mostrou algumas características na subsecção 4.7.1), conforme foi sugerido para algumas variações no capítulo 5. Estas variações seriam definidas na rede de Petri proposta para o jogo, por exemplo, acrescentando-se ou tirando-se condições para o caso de se aumentar ou diminuir o espaço do jogo (conseqüentemente, influenciaria na definição do número de eventos para tais condições) ou então se acrescentando ou tirando-se jogadores (o que influenciaria no número de *tokens* que a rede teria).

Sabe-se a importância de poder-se representar um sistema do “mundo real” através de uma especificação formal, pois ela permite a descrição de sistemas, a verificação e a análise de propriedades, a verificação da correção de implementações, a otimização, a descrição e a verificação do processo de desenvolvimento do sistema, além de outras [DÉH 2000]. Assim sendo, dentre os modelos encontrados, as redes de Petri são de grande destaque, pelo fato de possibilitarem esta especificação formal de sistemas e com isso, identificar as propriedades de tais sistemas, podendo ser feita uma melhor análise do sistema a ser desenvolvido.

De modo geral, a especificação formal vem ao encontro dos objetivos da modelagem de sistemas, através das propriedades que podem ser extraídas da modelagem, o que possibilita a identificação de determinadas situações ao longo da execução do sistema. Com isso, sabe-se que um método de especificação formal dá uma descrição precisa de um sistema em uma notação com uma sintaxe e uma semântica bem definidas.

Uma vez escolhida a formalização para representar o jogo lobos e cordeiro através de redes de Petri, mais especificamente a do tipo condição/evento, verificou-se que este jogo pôde ser especificado utilizando tal formalização (conforme apresentado na secção 4.5). O aspecto não-determinístico da rede *RPJ* igualmente pôde ser especificado, uma vez que o jogo definido é finito.

Através da especificação formal proposta para o jogo lobos e cordeiro pôde-se constatar que as situações antes vistas somente pelos aspectos intrínsecos do jogo, puderam ser identificadas e modeladas segundo esta especificação proposta; por exemplo, pôde-se especificar na rede *RPJ*, o que acontece na situação em que os lobos têm a vez para jogar e assim, apenas um deles pode efetuar esta ação.

Tendo-se então o exemplo descrito no parágrafo anterior, chegou-se à descrição que segue: pela situação que se vê no tabuleiro do jogo, estando os jogadores colocados em posições distintas neste tabuleiro, a ação que será realizada pelo jogador que representa os lobos é movimentar aquele lobo que tenha uma posição livre a sua frente. O jogo acontece então normalmente! Já pela situação que se tem na rede de Petri definida para o jogo, a escolha do lobo que irá jogar é uma ação não-determinística que é definida pela rede ou então, se for aplicada alguma estratégia, será definida uma ação determinística que represente a estratégia desejada na rede.

Além disso, as situações que caracterizam o jogo e que sem a especificação formal são eventualmente constatadas no jogo (quer dizer que as situações podem ou não ser diretamente detectadas), são identificadas com precisão através da especificação formal proposta para o jogo lobos e cordeiro; como exemplo destas situações tem-se, alcançar uma situação de fim do jogo – seja com a vitória do cordeiro ou dos lobos – ou detectar um possível travamento do jogo (situações estas que podem ser vistas nas propriedades apresentadas na secção 4.6).

Segundo a especificação proposta para o jogo lobos e cordeiro, as estratégias puderam ser definidas primeiramente, identificando-se as estratégias do jogo a serem utilizadas pelos jogadores (mostradas na secção 4.4) e então, modelando-se tais estratégias segundo as definições da rede *RPJ* (mostradas na subsecção 4.5.4).

Deste modo, o que demandou mais esforço na especificação do jogo foram as possibilidades de jogadas de cada jogador combinadas com suas respectivas estratégias. Assim, esta questão que envolve as estratégias pode ser igualmente aprofundada como um trabalho futuro, pois é uma tarefa que requer um estudo mais detalhado das situações do jogo.

A utilização das redes de Petri permitiu identificar os elementos do jogo, as dificuldades de execução e as características de estratégias. Por isso, tem-se que:

- pelo fato deste trabalho apresentar um protótipo simples de um jogo diferencial, pode-se então antever como se especificaria um outro jogo com características mais complexas;
- há problemas que já foram resolvidos através deste estudo e que estão presentes em jogos mais complexos, como por exemplo o problema de dar a vez de jogar e da escolha não-determinística, com restrições (escolher um dos lobos para jogar e ele poder se movimentar para uma posição que esteja livre);
- pode-se fazer a simulação de estratégias do jogo, através da especificação de estratégias pré-definidas na rede;

- pela generalização do jogo diferencial proposto é possível especificar formalmente a alternância entre os jogadores para a vez de jogar;
- as características do jogo como o espaço de jogo, o número de jogadores, os movimentos permitidos, as posições assumidas, etc, foram identificadas e transformadas, segundo especificação das redes de Petri condição/evento, de forma simples e direta;
- o refinamento da especificação do jogo, através de camadas distintas possibilitou a modularização da rede de Petri e isto pode ser aproveitado em outros jogos.

Na especificação formal do jogo lobos e cordeiro foi definida uma rede básica (sem estratégias - definida na subsecção 4.5.3) e uma rede estendida (com estratégias - definida na subsecção 4.5.4), onde a rede estendida é a ampliação da rede básica, mostrada na figura 4.15.

Através da especificação formal da rede estendida, uma situação como a que um lobo específico é o mais indicado para se movimentar é identificada, e uma estratégia pré-definida pode ser implementada na rede *RPJ* para que este lobo específico possa realmente se movimentar.

Com isso, a proposta da especificação formal do jogo diferencial lobos e cordeiro através de redes de Petri vem a contribuir para melhor visualizar o ambiente do jogo, as possíveis situações do jogo, as estratégias adotadas pelos jogadores, entre outras.

Este trabalho além de especificar o jogo diferencial lobos e cordeiro, tratou outras características comuns em jogos diferenciais como velocidades diferentes e obstáculos no espaço do jogo, contribuindo então na especificação de jogos mais complexos.

Futuramente, por meio desta especificação formal há a possibilidade da implementação deste jogo, levando em consideração aspectos capturados ao longo da sua especificação e que dizem respeito principalmente, a como o jogo se comporta diante do acontecimento de determinadas situações.

Portanto, sob uma visão geral, os algoritmos que serão desenvolvidos para a implementação deste e de outros jogos, podem ser projetados com maior facilidade uma vez que as situações decorrentes em um jogo são previamente analisadas usando a especificação formal de sistemas.

Identificou-se que a duração do jogo é de $O(n)$ jogadas, onde n refere-se ao número de posições que podem ser ocupadas pelos jogadores no tabuleiro do jogo, neste caso 32 posições. Uma estratégia do jogo é um algoritmo que precisa prever também, possíveis jogadas que podem ser realizadas pelos jogadores. Este cálculo é feito sabendo-se o número de movimentações possíveis em uma jogada – que são 8 jogadas para os lobos e 4 jogadas para o cordeiro.

Deste modo, torna-se importante desenvolver um algoritmo aproximativo que não considere todas as possibilidades de movimentação, eliminando as possibilidades menos promissoras.

Ao considerar a diversificação das características do jogo lobos e cordeiro, conforme já comentado anteriormente neste capítulo, identificou-se a possibilidade de estender também esta diversificação para a definição de redes de Petri proposta neste trabalho (conforme se apresentou algumas características na subsecção 4.7.1). Isto pois, uma vez

que se constatarem as características que um jogo tem, então é possível aplicar um modelo de especificação formal – como as redes de Petri – para o sistema que se deseja implementar.

Neste trabalho, destacam-se três aspectos que seguem. O primeiro está relacionado à escolha do jogo diferencial lobos e cordeiro (definido na seção 4.1 e na seção 4.2): este jogo foi escolhido porque é um exemplo simples, que no entanto apresenta características de jogos mais complexos. O segundo diz respeito à especificação formal do jogo proposto (definida na seção 4.5): realizou a formalização deste jogo porque, segundo mostrado nas motivações deste trabalho, os jogos apresentam problemas difíceis de serem resolvidos com algoritmos de complexidade baixa (conforme visto nos resultados da seção 4.6). E o terceiro está relacionado à escolha das redes de Petri condição/evento como modelo de especificação do jogo (definidas na seção 3.2): esta formalização foi escolhida porque as redes de Petri mostraram-se apropriadas para modelar as características que o jogo apresenta, principalmente no que se refere à alternância entre os jogadores.

Finalmente, pode-se dizer com certeza, que a experiência adquirida ao longo das pesquisas e da escrita deste trabalho originou o esclarecimento de questões acerca da teoria da computação, mais especificamente no que diz respeito à complexidade de algoritmos e à especificação formal. O fato do exemplo de jogo utilizado ser simples permitiu estudar características de jogo que talvez fossem difíceis de serem visualizadas em um exemplo mais complexo de jogo.

Com isso, o desenvolvimento deste trabalho deu sua contribuição, pois os problemas usuais dos jogos diferenciais puderam ser identificados e tratados neste exemplo simples de jogo e isso é útil para o estudo de outros jogos mais complexos, já que a aprendizagem incorporada com este estudo serve de base para o melhor entendimento de um problema mais complexo a ser resolvido. Neste trabalho não foi possível estudar com mais ênfase os problemas mais complexos, mas se pôde antever como podem ser tratados.

Portanto, a contribuição deste trabalho está no sentido de ter proposto uma especificação completa para um jogo diferencial através de redes de Petri, para que se pudesse observar o comportamento deste jogo sem perder de vista outras características usuais em jogos diferenciais que foram tratadas e sugeridas implementações.

Sugere-se como um trabalho futuro a diversificação do jogo proposto inicialmente, para incorporar além de outras, as diferentes características apresentadas até este momento.

Outro trabalho que seria de grande utilidade, seria o desenvolvimento de um simulador de redes de Petri para jogos diferenciais. Estudando simuladores existentes [DAI 2005], pôde-se verificar que são sempre desenvolvidos para um certo tipo de rede e se tornam de difícil uso em circunstâncias não previstas pelo seu idealizador.

Um simulador permitiria a simulação de jogadas, o estudo estatístico da eficácia de jogadas, e baseado nestes dados, estratégias mais eficazes poderiam ser desenvolvidas.

REFERÊNCIAS

- [BAR 2001] BARROS, J. P. M. P. R. **Introdução à modelação de sistemas utilizando redes de Petri**. [S.l.]: Instituto Politécnico de Beja - Escola Superior de Tecnologia e Gestão, 2001. Disponível em: <<http://www.estig.ipbeja.pt/~jpb/textos/pc/pn.pdf>>. Acesso em: 18 fev. 2005.
- [BAZ 2001] BAZZAN, A. L. C. Coordenação de Agentes com Uso de Técnicas de Teoria dos Jogos. In: JORNADA DE ATUALIZAÇÃO EM INTELIGÊNCIA ARTIFICIAL, 1., 2001, Fortaleza. **Anais...** Fortaleza: SBC, 2001.
- [BOD 89] BODLAENDER, H. L. **On the Complexity of some Coloring Games**. [S.l.]: Department of Computer Science, Utrecht University, nov. 1989.
- [BOU 99] BOUZY, B. **Complex Games in Practice**. Paris: UFR de mathématiques et d'informatique, Université René Descartes (Paris V), 1999.
- [CHE 95] CHENG, A.; ESPARZA, J.; PALSBERG, J. Complexity results for 1-Safe Nets. **Theoretical Computer Science**, Amsterdam, v. 147, p. 117-136, 1995.
- [DAI 2005] DAIMI – Department of Computer Science, Faculty of Science, University of Aarhus, Denmark. Página Oficial da Universidade. Disponível em: <<http://www.daimi.au.dk/PetriNets/tools>>. Acesso em: 05 jan. 2005.
- [DÉH 2000] DÉHARBE, D.; MOREIRA, A. M.; RIBEIRO, L.; RODRIGUES, V. M. Introdução a Métodos Formais: Especificação, Semântica e Verificação de Sistemas Concorrentes. **Revista de Informática Teórica e Aplicada (RITA)**, Porto Alegre, v. 7, n.1, p. 5-48, set. 2000.
- [DEM 2002] DEMAINE, E. D.; HOHENBERGER, S.; LIBEN-NOWELL, D. **Tetris is Hard, Even to Approximate**. [S.l.]: Massachusetts Institute of Technology, 2002. (Technical Report MIT-LCS-TR-865).
- [DEM 2001] DEMAINE, E. D. **Playing Games with Algorithms: Algorithmic Combinatorial Game Theory**. [S.l.]: Department of Computer Science, University of Waterloo, 2001.

- [EPP 2002] EPPSTEIN, D. **Computational Complexity of Games and Puzzles**. UC Irvine: Department of Informatics and Computer Science. Disponível em: <<http://www.ics.uci.edu/~eppstein/cgt/hard.html>>. Acesso em: 01 nov. 2002.
- [ESP 94] ESPARZA, J.; NIELSEN M. Decidability Issues for Petri Nets. **J. Inform. Process**, [S.l.], v. 30, n. 3, p. 143-160, 1994.
- [ESP 98a] ESPARZA, J. Reachability in Live and Safe Free-Choice Petri Nets is NP-complete. **Theoretical Computer Science**, [S.l.], v. 198, p. 211-224, 1998.
- [ESP 98b] ESPARZA, J. **Decidability and Complexity of Petri Net Problems - An Introduction**. Berlin: Springer, 1998. p. 374-428. (Lecture Notes in Computer Science, v. 1491).
- [FRA 97] FRAENKEL, A. S. **Combinatorial Games: Selected Bibliography With a Succinct Gourmet Introduction**. Israel: Department of Applied Mathematics and Computer Science, Weizmann Institute of Science, 1997.
- [GAR 79] GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. San Francisco: W. H. Freeman, 1979.
- [HOR 78] HOROWITZ, E.; SAHNI, S. **Fundamentals of Computer Algorithms**. [S.l.]: Computer Science Press, 1978.
- [ISA 99] ISAACS, R. **Differential Games: A Mathematical Theory With Applications to Warfare and Pursuit, Control and Optimization**. [S.l.]: Dover Publications, Inc, 1999.
- [JON 77] JONES, N. D.; LANDWEBER, L. H.; LIEN, Y. E. Complexity of Some Problems in Petri Nets. **Theoretical Computer Science**, [S.l.], v. 4, p. 277-299, 1977.
- [KOL 97] KOLLER, D.; PFEFFER, A. Representations and Solutions for Game-Theoretic Problems. **Artificial Intelligence**, [S.l.], v. 94, n. 1, p. 167-215, July 1997.
- [KOR 94] KORFF, M.; RIBEIRO, L. Formal Relationship between Graph Grammars and Petri Nets. In: CUNY, L. et al. (Ed.). **Graph Grammars and Their Application to Computer Science**. [S.l.: s.n.], 1994. p. 288-303. (Lecture Notes in Computer Science, 1073).
- [LEW 2000] LEWIS, H. R.; PAPADIMITRIOU, C. H. **Elementos de Teoria da Computação**. 2. ed. Porto Alegre: Bookman, 2000.

- [LOR 2001] LORETO, A. B.; TOSCANI, L. V.; RIBEIRO, L. Análise de Gramática de Grafos Baseada em Propriedades de Redes de Petri. In: CONGRESSO DE MATEMÁTICA APLICADA E COMPUTACIONAL - CNMAC, 24., 2001, Belo Horizonte. **Resumo das comunicações**. Belo Horizonte: UNI-BH, 2001.
- [MAC 96] MACIEL, P. R. M.; LINS, R. D.; CUNHA, P. R. F. **Introdução às Redes de Petri e Aplicações**. Campinas: Instituto de Computação, UNICAMP, 1996. 197p.
- [PAP 94] PAPADIMITRIOU, C. **Computational Complexity**. [S.l.]: Addison-Wesley, 1994.
- [PET 2000] PETERSON, G.; REIF, J.; AZHAR, S. **Decision Algorithms for Multiplayer Non-Cooperative Games of Incomplete Information**. [S.l.]: Computer Science Department, Georgia Institute of Technology and Duke University, 2000.
- [REI 93] REIF, J. H.; TATE, S. R. **Continuous Alternation: The Complexity of Pursuit in Continuous Domains**. Durham: Computer Science Department of Duke University, Durham, 1993.
- [REI 85] REISIG, W. **Petri Nets – An Introduction**. [S.l.]: Springer-Verlag, 1985.
- [RIB 97] RIBEIRO, L.; KORFF, M. Métodos Formais para Especificação: Gramática de Grafos. In: ESCOLA DE MÉTODOS FORMAIS PARA QUALIDADE DE SOFTWARE, 1., 1997, Pelotas. **Anais...** Pelotas: UFPel, 1997.
- [STO 94] STORER, J. A.; REIF, J. H. Shortest Path in the Plane with Polygonal Obstacles. **Journal of the Association for Computing Machinery**, [S.l.], v. 41, n. 5, p. 982-1012, Sept. 1994.
- [SUL 2002] SULZBACH, S. I. **Um Estudo da Computabilidade e da Tratabilidade dos Problemas de Verificação em Gramáticas de Grafos**. 2002. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [SUL 2003] SULZBACH, S. I. **Análise de Complexidade em Jogos Diferenciais**. 2003. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [SUL 2004] SULZBACH, S. I.; TOSCANI, L. V.; RIBEIRO, L. Modelo de Especificação de Jogo Diferencial. In: WORKCOMP-SUL, 1., 2004, Florianópolis. **Anais...** Florianópolis: UNISUL, 2004.
- [SZU 96] SZURLEY, D.; ABUOFADEL, E. **Research in Differential Games**. 1996. Disponível em: <<http://faculty.gvsu.edu/aboufadel/web/szurley.htm>>. Acesso em: 14 jan. 2003.

- [TOS 2000] TOSCANI, L. V.; VELOSO, P. A. S. **Complexidade de Algoritmos**. Porto Alegre: Instituto de Informática da UFRGS: Sagra Luzzatto, 2000.
- [VON 99] VON STENGEL, B. Computing Equilibria for Two-Person Games. In: AUMANN, R. J.; HART, S. **Handbook of Game Theory**. Amsterdam: North-Holland, 1999. v. 3.
- [YU 2003] YU, I. C. **Decidability and Complexity of Petri Net Problems**. Mar. 2003. Disponível em: <<http://www.ifi.uio.no/dbsem/foils2003vaar/complexity.pdf>>. Acesso em: 20 jan. 2005.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.