

3

Máquinas de Estados Hierárquicas

Máquinas de Estados têm se apresentado até aqui como uma ferramenta poderosa para a especificação e controle do comportamento de agentes inteligentes em jogos eletrônicos. Este capítulo discute algumas limitações da utilização de Máquinas de Estados Finitos e *Fuzzy* em jogos, e introduz a elas o conceito de hierarquia como uma proposta para tratar dessas limitações.

3.1.

Limitações das Máquinas de Estados Finitos e *Fuzzy*

Os tipos de Máquinas de Estados discutidos até aqui mostraram-se adequados e didáticos em exemplos simples. Entretanto, basta adicionar mais alguns estados e transições para que as representações em figuras se tornem caóticas e intratáveis.

A grande limitação das Máquinas de Estados como formalismo visual para a descrição de sistemas complexos se deve à explosão do número de estados e transições conforme o sistema vai se tornando mais elaborado, como discutido em Harel (1987), Drusinsky (1989) e Rocha (1999). No caso de sua utilização em jogos eletrônicos, além da dificuldade na especificação de comportamento complexo, Máquinas de Estados Finitos e *Fuzzy* sofrem ainda com o elevado tempo de processamento (em termos de tempo real) necessário para a avaliação de um grande número de condições de transição. Máquinas de Estados *Fuzzy* em especial são muito prejudicadas por transições em grande número, uma vez que, no pior caso, todas as transições são consideradas em uma mesma iteração (como visto na **Seção 2.2.3**).

Dessa maneira, é desejável que haja mecanismos que possibilitem a especificação de Máquinas de Estados complexas de maneira mais gerenciável e didática. Além disso, uma maneira de diminuir o elevado número de transições, típico de sistemas complexos, mantendo-se o mesmo comportamento, seria altamente vantajosa.

3.2. Hierarquia: Super-Máquinas e Sub-Máquinas

Considere-se a Máquina de Estados exemplificada na **Figura 2.2** do **Capítulo 2**. Para que o comportamento do agente seja capaz de torná-lo suficientemente competitivo, faz-se necessário refinar o estado *Atacando um Inimigo*, gerando vários outros estados para tratar das diversas situações de combate, como posicionamento, ataque com golpe ou tiro, esquiva, etc.

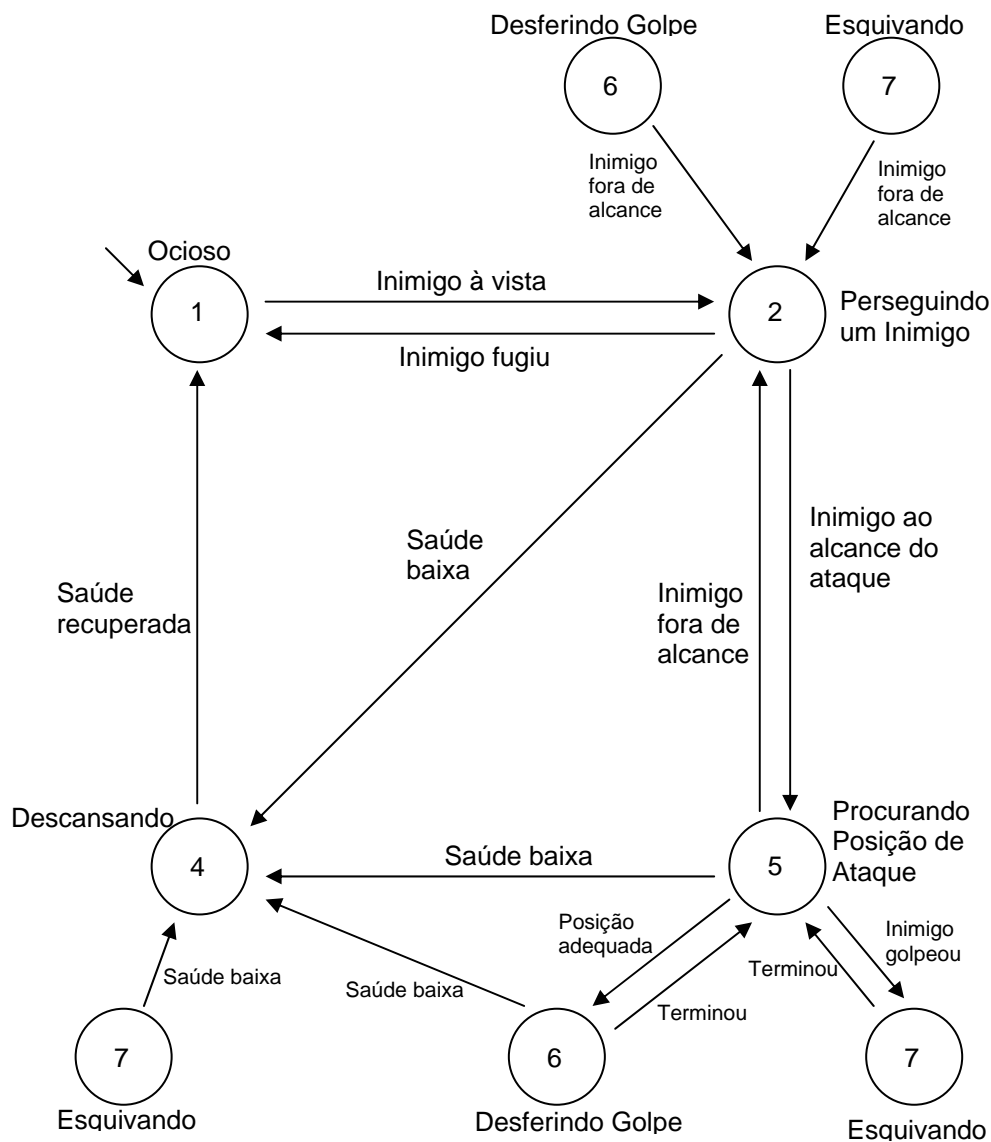


Figura 3.1 Máquina de Estados resultante da substituição do estado *Atacando um Inimigo* pelos estados *Procurando Posição de Ataque*, *Desferindo Golpe* e *Esquivando*.

A opção mais imediata seria a criação de novos estados que substituiriam o estado *Atacando um Inimigo*. Com isso, as transições entre o estado extinto e os demais estados pré-existentes teriam que ser replicadas em cada um dos novos estados. A **Figura 3.1** mostra a Máquina de Estados resultante desse refinamento.

A substituição do estado *Atacando um Inimigo* por três novos estados tornou a representação gráfica da Máquina de Estados muito mais difícil de ser construída e entendida. Note-se, em primeiro lugar, que alguns estados tiveram que ser replicados na figura, para que fosse possível explicitar todas as suas transições. Além desse inconveniente meramente representacional, outra característica que salta aos olhos é o grande aumento no número de transições. As transições que levavam do estado *Atacando um Inimigo* aos estados *Perseguindo um Inimigo* e *Descansando* na Máquina original tiveram que ser replicadas nos novos estados da nova Máquina.

Considere-se, também, que o estado *Descansando* tivesse que ser substituído por novos estados, que lidassem com os detalhes da ação de descansar, e que o mesmo acontecesse com o estado *Perseguindo um Inimigo*. Essa nova Máquina, perfeitamente plausível para um agente de um jogo de ação típico dos dias de hoje (Lopes, 2003), teria se tornado praticamente impossível de gerenciar e representar de maneira clara e organizada.

Uma opção mais adequada para a substituição do estado *Atacando um Inimigo* seria associar *uma outra Máquina de Estados* a esse estado. Essa *sub-Máquina* conteria os estados que tratam das ações minuciosas envolvidas no contexto do estado *Atacando um Inimigo*, que correspondem aos novos estados introduzidos no exemplo anterior.

Uma estrutura desse tipo, onde sub-Máquinas podem ser associadas a estados de uma Máquina de Estados, é chamada *Máquina de Estados Hierárquica*. É importante notar que a hierarquia não se limita a dois níveis; quaisquer estados da sub-Máquina poderiam ter outras sub-Máquinas associadas a eles, e a noção de hierarquia se estende trivialmente. A **Figura 3.2** mostra o refinamento do estado *Atacando um Inimigo* da **Figura 2.2** através de hierarquia.

Fundamentalmente, a hierarquia não adiciona poder de expressão ao modelo formal de Máquinas de Estados, e nem tampouco reduz o número de estados; mas pode reduzir significativamente o número de transições (Girault *et al*, 1999). Alguns autores, como Rocha (1999), estabelecem uma definição de Máquinas de

Estados Hierárquicas diversa à das Máquinas de Estados convencionais, adicionando uma nova função que trata da transição entre níveis da hierarquia⁵; Outros, como Harel (1987) e Girault *et al* (1999), preferem abster-se de uma nova definição; esta última abordagem é preferida, uma vez que qualquer Máquina de Estados Hierárquica pode ser re-escrita como uma Máquina de Estados sem a presença da hierarquia (Harel, 1987).

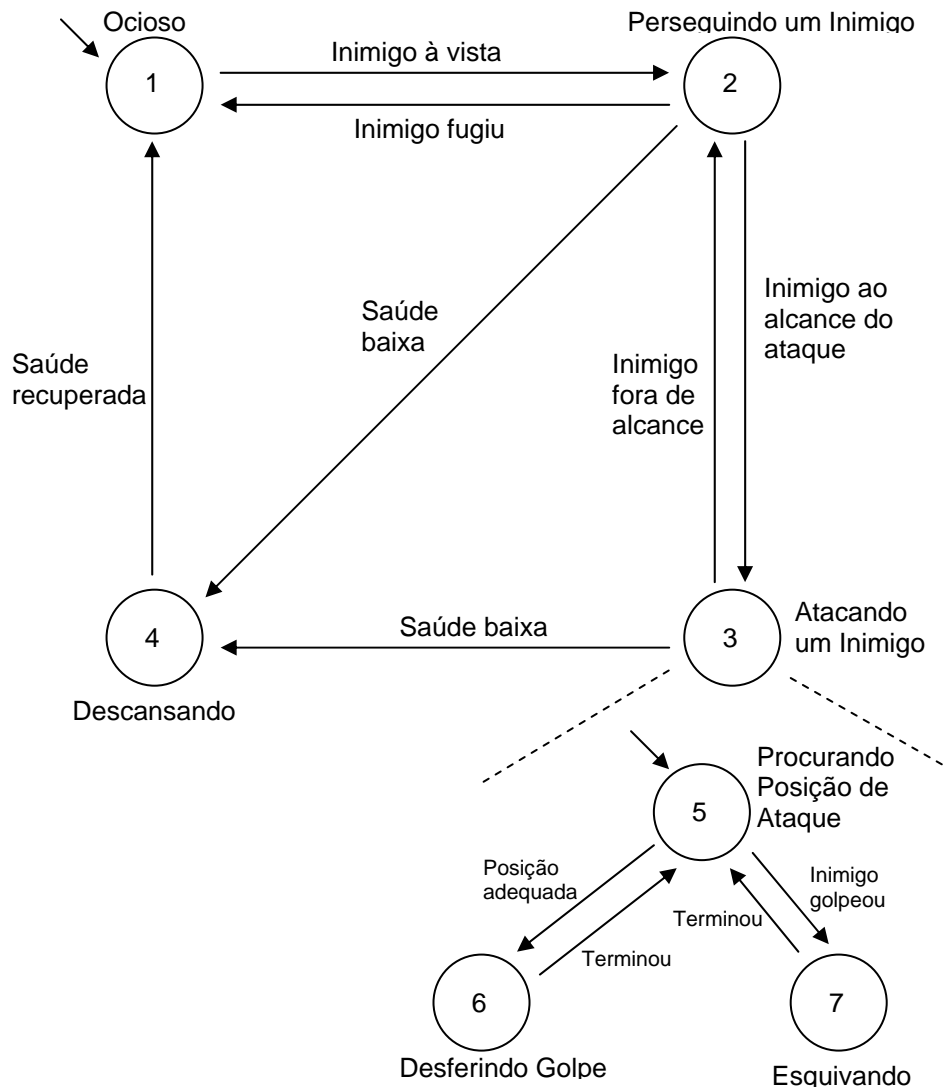


Figura 3.2 Máquina de Estados Hierárquica resultante do refinamento do estado *Atacando um Inimigo*.

⁵ A definição de uma nova função por Rocha (1999) se justifica porque as transições entre níveis apresentam importância especial no contexto do seu trabalho, que envolve o uso de Máquinas de Estados na especificação e construção de unidades de controle.

O funcionamento de uma Máquina de Estados Hierárquica, a cada iteração, ocorre da seguinte maneira: primeiramente, a Máquina superior na hierarquia é executada; se o estado corrente tiver uma sub-Máquina associada, essa sub-Máquina é executada; o processo continua descendo na hierarquia até que não haja uma sub-Máquina associada ao estado corrente da Máquina atual; as ações realizadas pelo agente serão as que correspondem a esse estado. Na próxima iteração, a execução inicia novamente na Máquina superior, diferentemente da execução das Máquinas Hierárquicas propostas por Houlette *et al* (2001) e Houlette e Fu (2003), nas quais a execução permanece na sub-Máquina por várias iterações, até que ela atinja um estado final.

A adição de hierarquia às Máquinas de Estados atende às limitações discutidas na **Seção 3.1**, possibilitando a construção de Máquinas de Estados através de uma representação em vários níveis, tornando plausível e gerenciável a definição gráfica de Máquinas de Estados complexas (Houlette e Fu, 2003). Além disso, a hierarquia fornece também uma desejável redução do número de transições quando são comparadas Máquinas de Estados Hierárquicas e Não-Hierárquicas que representam o mesmo comportamento. A reutilização e o compartilhamento de sub-Máquinas em estados diferentes da super-Máquina também são características desejáveis desse modelo.

Hierarquia não é a única maneira possível de se conceber Máquinas de Estados combinadas. Máquinas de Estados Concorrentes também são uma alternativa muito utilizada em síntese de circuitos digitais e outras áreas, e poderiam ser utilizadas na representação do comportamento de agentes inteligentes em jogos. Entretanto, as limitações desse tipo de Máquina de Estados inerentes tanto à representação visual (Harel, 1987) quanto à descrição e ao funcionamento em sistemas de jogos (Houlette e Fu, 2003) dificultam sua consideração nesse contexto, excluindo-as do objetivo e do escopo desta obra.

3.3. Máquinas de Estados *Fuzzy* Hierárquicas

Tudo o que foi discutido até agora sobre a adição de hierarquia se aplica tanto às Máquinas de Estados Finitos quanto às Máquinas de Estados *Fuzzy*. No

caso das Máquinas de Estados Finitos Hierárquicas, seu funcionamento é trivialmente derivado da descrição apresentada na **Seção 3.2**. Máquinas de Estados *Fuzzy* Hierárquicas, em contrapartida, necessitam que esse funcionamento seja estendido para lidar com o caráter *fuzzy* do estado corrente.

O funcionamento de uma Máquina de Estados *Fuzzy* Hierárquica ocorre da seguinte maneira: a cada iteração, são avaliadas as transições que partem de todos os estados que pertençam ao conjunto estado corrente, isto é, cujo grau de pertinência seja não-nulo; transições podem ser efetuadas, dependendo do valor de verdade de suas condições, substituindo-se as variáveis pelos valores de entrada; após as transições, terá ocorrido uma redistribuição dos graus de pertinência, reconfigurando o conjunto estado corrente. A partir daí, a execução desce um nível na hierarquia, considerando todas as sub-Máquinas associadas aos estados cujo grau de pertinência ao conjunto estado corrente é não-nulo; o processo é repetido para cada uma dessas sub-Máquinas, até que não haja mais sub-Máquinas associadas aos estados pertencentes ao conjunto estado corrente em nenhuma das Máquinas sendo consideradas.

O grau de pertinência ao conjunto estado corrente dos estados de uma sub-Máquina são multiplicados pelo grau de pertinência do seu estado-pai (isto é, o estado da Máquina imediatamente superior ao qual está associada a sub-Máquina). Dessa maneira, o grau de pertinência do estado-pai é distribuído pelos estados da sub-Máquina associada. Isso permitirá interpretar os valores dos graus de pertinência como probabilidades, viabilizando a estratégia de sorteio descrita na **Seção 2.2.3**. O exemplo a seguir ajudará na compreensão desse funcionamento e da interpretação dos graus de pertinência como probabilidades em Máquinas de Estados *Fuzzy* Hierárquicas.

Considere-se uma Máquina de Estados *Fuzzy* Hierárquica cujos estados da Máquina superior sejam a , b e c , e que esteja associada ao estado b uma sub-Máquina com estados d , e , f e g . A **Figura 3.3** mostra a representação gráfica desse exemplo.

Suponha-se que, ao final de uma determinada iteração, os valores dos graus de pertinência ao conjunto estado corrente dos estados da Máquina superior sejam os seguintes:

$$\mu_a = 0.3$$

$$\mu_b = 0.5$$

$$\mu_c = 0.2$$

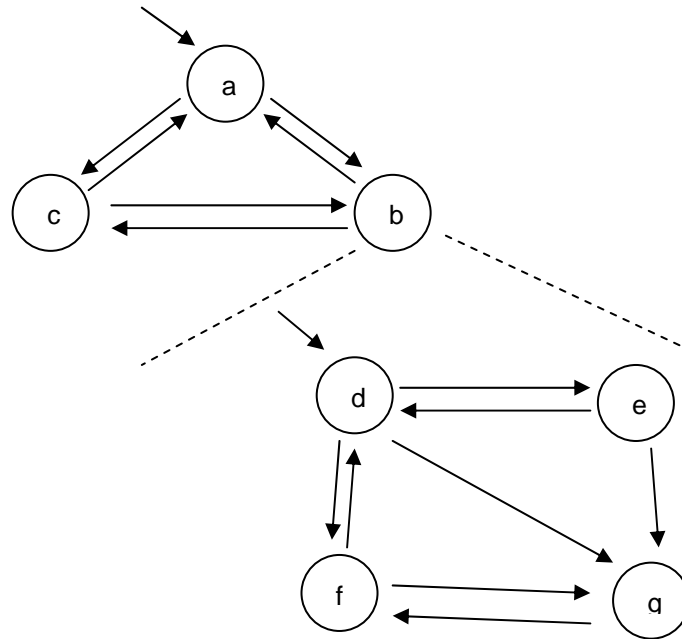


Figura 3.3 Exemplo simplificado de Máquina de Estados *Fuzzy* Hierárquica.

Como o estado *b* possui grau de pertinência não-nulo, a sub-Máquina associada a ele será executada. Suponha-se então que, ao término dessa iteração, os valores dos graus de pertinência ao conjunto estado corrente dos estados da sub-Máquina sejam os seguintes:

$$\mu_d = 0.1 \qquad \mu_e = 0 \qquad \mu_f = 0.6 \qquad \mu_g = 0.3$$

Distribuindo-se o grau de pertinência de *b* aos estados de sua sub-Máquina associada, ter-se-á que a probabilidade de cada estado no sorteio será a seguinte:

$$\begin{aligned} P_a &= \mu_a = 0.3 & P_e &= \mu_b \mu_e = 0 \\ P_c &= \mu_c = 0.2 & P_f &= \mu_b \mu_f = 0.3 \\ P_d &= \mu_b \mu_d = 0.05 & P_g &= \mu_b \mu_g = 0.15 \end{aligned}$$

$$\Sigma P_i = 1$$

As ações executadas pelo agente corresponderão ao estado sorteado.