PCS 3216 – Sistemas de Programação

Aula 08 – Técnicas complementares
para a simulação reativa de fenômenos
sequenciais, guiada por eventos

Introdução

- Nas aulas anteriores sobre simulação guiada por eventos foram apresentados diversos assuntos associados a fenômenos de diferentes naturezas, com foco principalmente nos elementos comuns a todos esses simuladores.
- Aqui pretende-se completar esse estudo dedicando esta aula aos fenômenos cujos simuladores diferem entre si por exigirem tratamentos específicos, conforme o caso.
- É interessante notar que, em muitos casos, esses fenômenos podem ser reagrupados segundo características que lhes são comuns, criando famílias de fenômenos passíveis de serem simulados segundo a técnica proposta, apesar de divergirem do padrão ideal, em um ou outro aspecto, do inicialmente proposto para a simulação reativa de fenômenos discretos sequenciais guiada por eventos temporais.

- Muitas vezes surgem questionamentos quanto ao uso da técnica estudada:
  - Dá para fazer de outra forma?
  - É preciso mesmo ser feito assim?
  - Posso continuar fazendo do meu jeito?
- Em programação nada tem solução única. Pelo contrário, qualquer programa pode ser substituído por infinitos outros, equivalentes. Cabe ao programador optar (se souber) pelo mais adequado.
- Cada técnica de programação adotada pode, naturalmente, apresentar vantagens e desvantagens em cada situação. Ter fluência em várias técnicas permite ao profissional escolher em cada caso aquela que melhor atenda às suas necessidades, para obter eficiência, economia e padronização.
- Ao programador, é naturalmente desejável ter à disposição métodos e técnicas diversas que permitam fazer sua melhor escolha, em lugar de ignorar possíveis alternativas e usar sempre soluções triviais, ou então, baseadas em tentativas e erros, ou ainda, “resolver” o problema “caçando” soluções prontas.

- Uma questão fundamental: como determinar o momento adequado para interromper o processo, ou seja, como saber o grau de abstração em que se pode dar por terminada a etapa de refinamento?
- A técnica de refinamentos sucessivos tem a flexibilidade de permitir ao seu usuário determinar o ponto certo para considerar terminado o seu trabalho. Isso permite trabalhar de forma incremental, adicionando progressivos detalhes em cada passo do refinamento das abstrações.
- Dependendo do uso a ser feito do simulador desenvolvido, pode ser conveniente manter um menor grau de detalhamento ou então adicionar detalhes à sua modelagem.
- O refinamento deve prosseguir até que todos os detalhes relevantes tenham sido incluídos, ou seja, da inclusão de detalhes em excesso pode resultar uma simulação mais lenta, ineficiente, complexa, portanto mais onerosa. Recomenda-se não ultrapassar o grau de abstração estritamente necessário para que a simulação permita observar em suficiente detalhe todos os elementos exigidos pela aplicação específica que se está desenvolvendo.

Extração de eventos

- Eventos não originados em fenômenos físicos externos costumam depender dos seguintes elementos:
  - Estado corrente do dispositivo simulado
  - Último evento recebido
  - Tabela de transições para determinar próximo estado
- O motor de eventos é alimentado pelos eventos que são extraídos a cada passo, nesta operação.
- Os eventos seguem sempre uma relação sequencial de precedência, ligada à sequência em que devem ser simulados, mesmo quando sua natureza não é temporal
- Na simulação, torna-se natural e conveniente, por essa razão, o emprego de listas ordenadas de eventos programados para serem tratados em passos futuros de simulação.
- Nesse caso, a extração do evento a ser simulado em cada passo de simulação é feita a partir dessa lista de eventos.
- Eventos com tratamento complexo (p.ex. códigos de máquina), utilizam algoritmos especiais para determinar, a cada passo de simulação, o evento seguinte a ser tratado.

Eventos não sequenciais

- Entradas paralelas
- Ocorrências simultâneas
- Processos independentes, mas comunicantes
- Critérios (geralmente arbitrários) de desempate para sequencializar eventos simultâneos
- Técnicas para a simulação do tempo
  - Discretização de instantes em intervalos uniformes
  - Discretização de instantes em intervalos variáveis
- Sequencialização de eventos futuros “simultâneos”
  - Fila de eventos futuros ordenada pelo instante previsto
  - Critérios de preferência, no caso de empate: ordem de chegada à fila, prioridade, sorteio, etc.

Simulação de paralelismo

- Pode ocorrer coincidência temporal de eventos extraídos de meios de entrada, p.ex. arquivos de dados, e também de eventos dependentes (gerados pelas rotinas de reação executadas durante a simulação), quando programados para um instante futuro de simulação, estipulado pelas rotinas de tratamento.
- Devido à característica sequencial do simulador, não pode existir paralelismo verdadeiro. Por isso, todas as ocorrências futuras são organizadas em uma fila única de eventos, ordenada segundo seu instante previsto de ocorrência.
- Sistemas reativos geram eventos dependentes, programam seu instante de ocorrência e os inserem nessa lista única de eventos.
- Assim, ocorrências programadas como paralelas na prática deixam de ser paralelas quando utilizam essa fila única, da qual apenas o primeiro elemento é extraído de cada vez, mesmo se o seguinte estiver programado para o mesmo instante, forçando assim o tratamento sequencial.
- Se o instante previsto para um evento for menor que o instante corrente de simulação, o simulador deve considerá-lo resultante de demoras na fila (= interrupções simultâneas do hardware).

Eventos independentes próprios

- Fenômeno reativos puros não agem por si mesmos, mas apenas reagem respondendo à ocorrência de eventos
- Há, entretanto, fenômenos ativos, capazes de criar espontaneamente seus próprios eventos independentes (são fontes de eventos aleatórios)
- A simulação dessa geração espontânea de eventos independentes pode ser feita com facilidade a partir do sorteio de números aleatórios, que representem o número de eventos a gerar, o instante de simulação em que tal geração deve ser feita, o tipo e as características dos eventos aleatórios a serem gerados.
- Todos esses eventos são inseridos devidamente na lista de eventos do simulador, e tratados normalmente como os demais eventos já comentados.

Eventos ligados ao tempo de simulação

- Um importante caso de geração espontânea de evento independente ocorre na simulação de ***timers***.
- O mais direto gera uma base periódica microscópica de tempo para o simulador. A cada evento gerado (= uma interrupção de relógio), a próxima interrupção é programada para um instante futuro dado pelo período do relógio, e quando esta interrupção for tratada, a seguinte é programada, etc.
- Em geral, os simuladores guiados por eventos evitam essas bases de tempo periódicas, preferindo usar intervalos variáveis entre as atualizações do seu relógio, coincidentes com os momentos em que ocorrem outros eventos.
- Nesse caso o relógio do simulador é atualizado na ocasião da extração dos eventos.
- Isso permite economizar o processamento de muitas interrupções, que em sua maioria não apresentam efeitos reativos úteis, e apenas desperdiçam esforço computacional e tornam mais lento o simulador.

Se a especificação não for por regras

- É preciso converter a especificação disponível para o formato estabelecido, para que possa ser para ela elaborado um simulador guiado por regras
- Caso o fenômeno esteja descrito em seu comportamento por meio de outro tipo de formalismo que não seja a de um conjunto de regras do tipo ***se...então***, torna-se necessário converter sua especificação em outra equivalente, que siga essa formulação.
- Isso pode mostrar-se artificioso quando a formulação já não se apresentar originalmente discretizada.
- Nesse caso, é necessário primeiramente efetuar uma discretização dessa especificação, para depois, criar para cada um dos casos (agora um conjunto finito) uma regra ***se...então*** equivalente, esta compatível com a formulação adotada.

A calculadora é muito limitada

- As limitações da calculadora não permitem que seja imitada nas aplicações de meu interesse.
- É verdade que se trata do caso mais limitado das simulações, pois é a mais simples delas, exatamente por ser integralmente aderente a todas as hipóteses adotadas para a aplicação do método: variedade
  - É um fenômeno discreto (com um número finito de situações)
  - Seus eventos são todos individualizados (evento=tipo)
  - Só usa eventos independentes verdadeiros (temporais e sequenciais)
  - É um fenômeno reativo puro (só responde a eventos)
  - É integralmente guiado por regras
- De fato, fenômenos tão bem comportados são raros, porém há artifícios que permitem contornar com certa facilidade as situações que se afastam dessa condição ideal.

O fenômeno inclui aspectos não discretos

- Neste caso, a variedade de situações pode ser infinita, e técnicas híbridas são utilizadas em seu tratamento.
- Separar então as partes discretas para tratamento convencional, e decompõem-se as partes contínuas em um pequeno número finito de subclasses finitas homogêneas.
- O tempo físico é uma variável independente contínua que também deve ser simulada de forma discreta.
- Uma técnica para simular o tempo se baseia no uso de uma lista de eventos futuros, ordenada pelo instante previsto para sua ocorrência.
- Eventos temporais, de tempo real (tempo contínuo) devem passar por uma discretização, para receberem tratamento similar ao que é usualmente aplicado às interrupções de hardware, de modo que a reação do simulador à sua ocorrência possa ser gerada o mais imediatamente possível

Reclassificação de eventos

- Em certos casos, é possível que os eventos recebidos pelo simulador cheguem classificados de forma não ideal para o tratamento que lhes é dado pelo simulador.
- Assim, pode acontecer que um grupo de eventos que são recebidos com identificações diferenciadas em seus tipos requeiram um tratamento único. Nesse caso, a menos de parâmetros diferenciados, é possível usar uma mesma rotina de tratamento para todos eles.
- Na situação oposta, pode ser que um único tipo de interrupção se refira a um conjunto de casos que necessitem receber tratamentos diferentes. Neste caso, a única rotina de tratamento precisa incorporar uma classificação adicional para a ocorrência, decompondo-a em diversos eventos, cada qual atendido por sua rotina própria de tratamento.

Eventos não temporais

- Muitas vezes, a origem dos eventos não é física (nem sinal elétrico, nem sinal detectado e/ou transmitido por alguma fonte externa).
- Eventos de origem lógica (informações extraídas de textos ou de áreas de dados, ou de repositórios de código) geralmente resultam de uma conversão prévia dessas informações para a forma de eventos tratáveis pelo simulador.
- Extratores e classificadores são usados para isso, como é o caso, por exemplo, dos analisadores léxicos, que operam no pré-processamento de arquivos de entrada escritos em alguma linguagem de programação, para que tais elementos sejam utilizados como eventos.

Extração de eventos não temporais

- Principais fontes de origem para os eventos não temporais:
  - **Memória** de um processador, de onde são extraídas **instruções de máquina** (real ou virtual) para serem *executadas pelo simulador*.
  - O pré-precessamento aqui consiste em determinar o endereço do material de interesse em cada momento, extraí-lo e entregá-lo ao *dumper*, ao *interpretador* da linguagem de máquina ou a outro programa que utilize como entrada esse código binário.
  - **Textos-fonte** de linguagens simbólicas ou de alto nível, empregando-se para isso *analisadores léxicos* como pré-processadores, que alimentam os compiladores e interpretadores da linguagem de entrada.
  - **Arquivos** contendo **códigos numéricos** de programas absolutos e relocáveis em formato carregável. Aqui o pré-processamento consiste em extrair os metadados desse material e repassá-los ao *loader*, ao *disassembler* ou outro programa de sistema que utilize tais códigos como entradas.

Extração de eventos da memória

- A extração de eventos a partir de repositórios de dados numéricos, como é o caso da memória, envolve manipular os seguintes elementos:
  - Posição do último dado recém-extraído
  - Critério de localização do dado associado ao próximo evento a extrair
  - Extração e classificação do evento associado ao dado extraído
- O programa de sistema a que se destina o evento assim extraído é em geral o *dumper*, ou então um *interpretador* de linguagem binária, tal como um simulador de *máquina virtual*.
- Nesse caso, o loop de simulação deve incluir um passo adicional, referente à extração e classificação do próximo evento a ser simulado, o qual se refere ao código numérico da próxima instrução a ser executada pelo interpretador.
- A simulação de cada evento extraído deve preparar o simulador para a extração do próximo evento a ser simulado. Na prática, isso é feito caso a caso, determinando-se, após a simulação de cada instrução, o endereço da instrução seguinte a ser simulada.

Extração de eventos de textos-fonte

- Textos-fonte suscitam a extração de eventos das mais variadas formas, de acordo com o tipo de estrutura apresentada pelo material textual neles contido.
- É comum encontrar textos fonte contendo:
  - **Textos puros**, na forma de uma sequência de *caracteres*
  - Textos representando na forma de **scripts** ou de programas escritos *linguagem de programação* simbólica ou de alto nível
  - Textos formatados como **arranjos bidimensionais** de caracteres, formando documentos textuais, tabelas (planilhas), desenhos em formato semi-gráfico
  - Textos redigidos em *linguagem natural*
- Tais textos apresentam, em comum, seus elementos básicos na forma de subsequências de caracteres com lei de formação conhecida. Essas sequências costumam ser denominadas **itens léxicos** ou lexemas, e os programas que os extraem e classificam segundo sua estrutura são os *analisadores léxicos.*
- Os itens léxicos extraídos alimentam programas responsáveis por seu processamento (compiladores, interpretadores de linguagens de alto nível, processadores de texto, editores, etc)

Extração de eventos de códigos-objeto

- Códigos-objeto podem, analogamente, ser decompostos em eventos que representem seus componentes mais elementares de acordo com o formato adotado para a sua representação.
- Esses componentes abrangem:
  - O **código numérico** propriamente dito
  - **Metadados** que o acompanham, indicando
    - **Endereço** associado (local associado ao dado, na memória)
    - **Comprimento** dos dados (número de bytes do dado)
    - **Tipo** do particular código (de instrução, de endereço, de dado)
    - **Informação de relocação** (absoluto, base de relocação)
- O extrator de eventos, neste caso, deve:
  - **Separar** cada um dos códigos juntamente com metadados a ele associados, formando um grupo a ser usado como evento
  - **Classificar** o evento conforme o tipo de dado correspondente ao grupo extraído
  - **Repassar** o evento assim extraído ao programa responsável por seu processamento (loader, relocador, linker, desmontador, etc.)

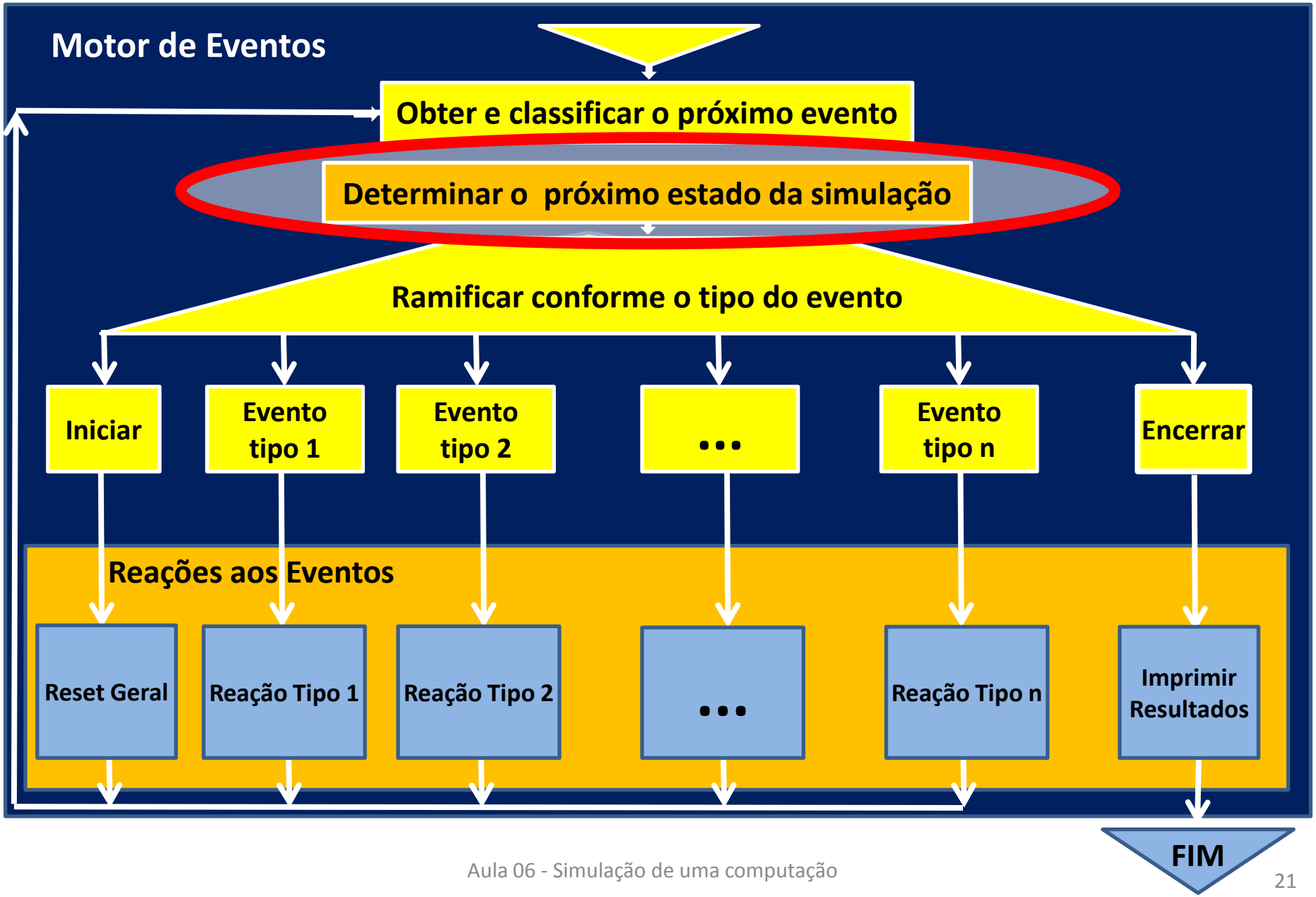
Extração a partir de outras fontes

- Muitas outras fontes de informações existem, e seus elementos constituintes podem ser extraídos de forma análoga ao que foi exposto nos casos apresentados.
- Naturalmente, em cada caso a sequência de componentes básicos varia em tamanho e forma, mas deve formar um grupo que tenha um papel específico no processamento do material em questão, constituindo um evento próprio para sua simulação.
- A classificação do evento está associada à estrutura do grupo de elementos que o constitui.
- Após a extração e classificação, o evento assim obtido deve ser repassado ao loop de simulação, integrando assim a lógica do particular simulador guiado por eventos.

Máquinas de estados no controle da situação do dispositivo

- A cada iteração do loop do simulador (passo de simulação), o simulador (ou, igualmente, o dispositivo simulado) efetua uma **transição** que o leva de sua situação (ou estado) corrente para uma próxima situação (ou estado).
- É de grande utilidade que essas mudanças de estado estejam muito bem planejadas *antes* da implementação do simulador, pois isso poupa muito trabalho em sua depuração e dá ao projetista uma confiança maior no programa que está sendo construído.
- Por essa razão, recomenda-se fortemente que seja formalizada a dinâmica das mudanças de estado do fenômeno simulado através do levantamento rigoroso e completo de todos os movimentos possíveis do seu simulador.
- Dessa etapa do projeto deve emergir uma **máquina de estados** que seja capaz de indicar, após cada iteração do simulador, o *estado* em que ele passa a se encontrar, garantindo que não seja feita nenhuma transição indevida, especialmente em situações muito similares e de difícil discernimento.
- O acionamento dessa máquina de estado pode ser feita logo após a extração do próximo evento, de maneira que a reação associada possa tirar proveito do conhecimento do estado formal em que o simulador se encontra em cada ocasião.

Complementando o diagrama da simulação



SIMULAÇÃO DISCRETA DO TEMPO

Enquanto a **lista de eventos não estiver vazia**:

- **Extrair o primeiro evento** da lista, ou seja, aquele que estiver **programado para ocorrer mais cedo**; seja ele um evento (i,t) do tipo i, programado para ocorrer no instante t de simulação.
- **Tratar o evento extraído** da mesma forma como um sistema operacional trataria uma interrupção: **executando a i-ésima rotina de tratamento extraída de um vetor de interrupções**
- Atualizar o instante corrente de simulação
- **Coletar informações** de controle da simulação;
- **Imprimir informação** de acompanhamento;

Terminada a simulação, consolidar, formatar e **apresentar as informações colhidas** ao longo do processo de simulação.

Simulação do tempo

- **AGORA** = instante de simulação corrente;
- **Lista de eventos** = lista de pares ordenados, da forma (*tipo*, *instante*), em ordem crescente de seu *instante* de ocorrência;
- O último evento dessa lista obrigatoriamente deve ser do *tipo* "**fim de simulação**".
- Para determinar o **próximo evento** a tratar, basta extrair o **primeiro elemento da lista**.
- Então atualiza o instante de simulação corrente para o **maior dentre os valores de AGORA e do *instante* programado para o evento**;

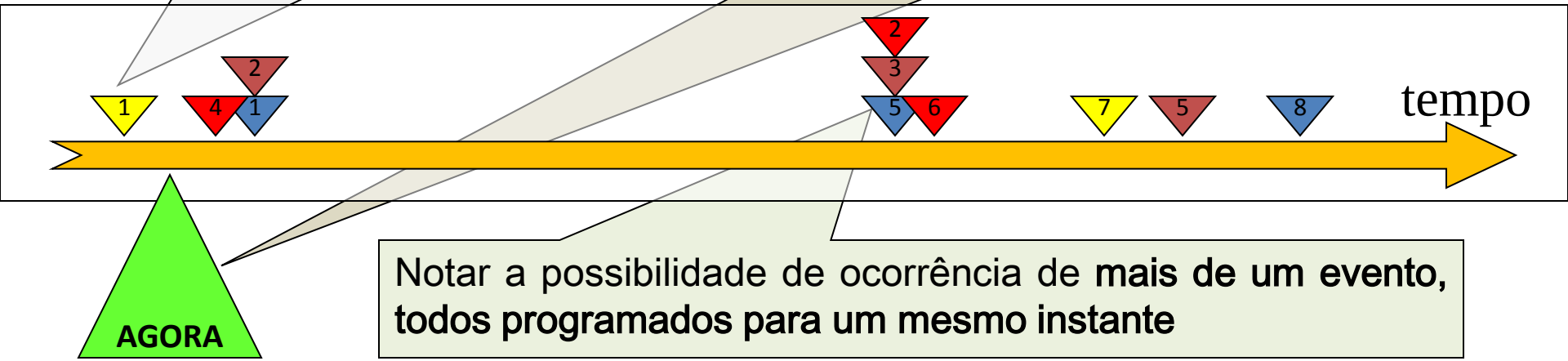
Notação gráfica para os eventos

- O próximo slide apresenta uma figura em que está representada graficamente a fila de eventos futuros, parecida com a que estamos usando.
- A barra laranja é uma escala linear de tempo, iniciada nas imediações do instante AGORA, que designa o instante corrente de simulação.
- Os triângulos representam eventos programados ainda não tratados. São posicionados, na linha de tempo, no momento previsto para sua ocorrência.
- Havendo mais de um evento programado para o mesmo instante, esses eventos são dispostos verticalmente, um sobre o outro, na mesma abcissa.
- Eventos à direita de AGORA são eventos programados para ocorrerem em instantes de simulação futuros.
- Eventos à esquerda de AGORA haviam sido programados para ocorrerem no instante passado de simulação em que o evento está colocado, mas que não foi tratado a tempo, devido a sobrecargas no processo simulado ou então devido ao tratamento de evento(s) programado(s) para o mesmo instante.
- O loop de simulação extrai e trata sempre o evento mais à esquerda e mais acima, nessa figura, sem alterar os demais.

Lista de eventos

Notar que em qualquer ocasião podem ocorrer na lista eventos, programados para **instantes anteriores, mas ainda não tratados**.

Esse símbolo indica sobre o eixo do tempo a posição do **instante corrente de simulação**.



AGORA

Notar a possibilidade de ocorrência de **mais de um evento, todos programados para um mesmo instante**

Esse símbolo indica um **evento programado** para o instante correspondente à sua posição sobre o eixo do tempo.
i – corresponde ao tipo do evento.
A cor – indica o elemento simulado ao qual o evento se refere.

Observação sobre o tempo

- **No mundo real o tempo flui continuamente**, constatando-se situações com longos períodos sem atividade;
- Em simulações discretas, como as dirigidas por eventos, **é desnecessário simular ativamente os períodos de inatividade**, podendo-se poupar processamento desprezando-se tais períodos sem efetuar ação alguma de simulação;
- Assim, o valor da variável **AGORA evoluirá de forma discreta**, saltando de um valor para outro ao longo da simulação, de acordo com os *instantes* de ocorrência dos eventos da lista.

Simulação dos eventos

- Extraído no instante de simulação AGORA o evento **(i, t)**, executa-se a rotina de tratamento **i**, que deve efetuar toda a computação responsável pelas atividades ligadas a ocorrências de eventos do tipo **i**;
- Se **t < AGORA**, significa que, devido a algum *overhead*, **não foi possível atender o evento no instante programado**, por isso seu tratamento acabou sendo postergado para o instante AGORA.
- Entre outras atualizações das estruturas de dados de simulação, **deve ser atualizado o novo valor do instante de simulação corrente AGORA**.
- Para não onerar o processo, **essa atualização não precisa ser contínua**, podendo saltar os intervalos sem atividade.

Conclusão

- Os assuntos tratados nesta apresentação complementam as informações estudadas nas aulas anteriores acerca dos motores de eventos e sua aplicação à implementação de simuladores para aplicação na elaboração de programas de sistema.
- Com essas informações, espera-se que seja mais confortável usar essa importante técnica para o desenvolvimento dos programas estudados nesta disciplina.
- Relembramos que essa técnica é muito geral, e pode ser utilizada no desenvolvimento de softwares variados, nas mais diversas aplicações, constituindo uma ferramenta de grande valor no exercício de atividades de projeto e desenvolvimento de software em geral.