




Laboratório de Engenharia de Software

Máquinas de estado

Arndt von Staa
Departamento de Informática
PUC-Rio
Setembro 2017

Especificação

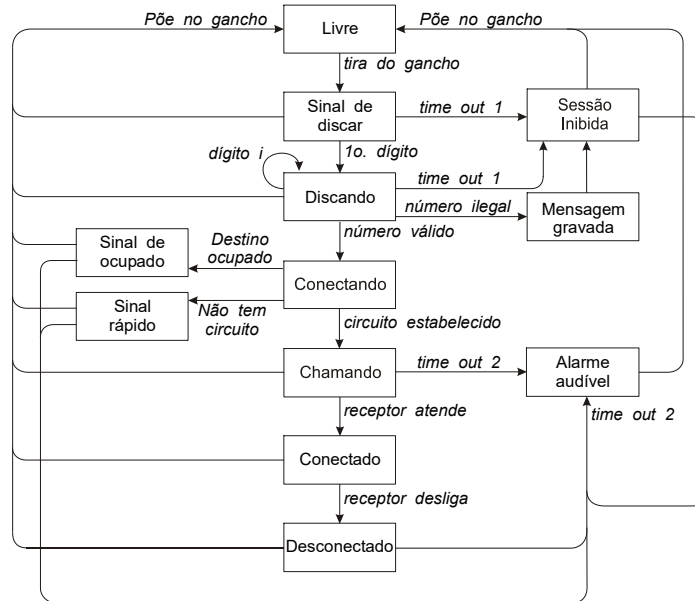


Laboratório de Engenharia de Software

- Objetivo desta aula
 - Apresentar máquinas de estado e seu uso ao gerar testes funcionais
- Justificativa
 - Muitos testes dependem de uma sequência grande e complexa de decisões.
 - Determinar que dados devem ser fornecidos, em que ordem e segundo que condições é uma tarefa complexa e propensa a enganos.
 - Gerar e fornecer todos esses dados ao programa sob teste também tende a ser uma tarefa complexa e propensa a enganos.
 - Consequentemente, deseja-se estabelecer uma forma para gerar (quase) automaticamente os dados de teste
- Texto
 - Pezzè, M.; Young, M.; Teste e Análise de Software; Porto Alegre, RS: Bookman; 2008, capítulo 14

Set 2017Arndt von Staa© LES/DI/PUC-Rio2

Máquina de estados, exemplo telefone

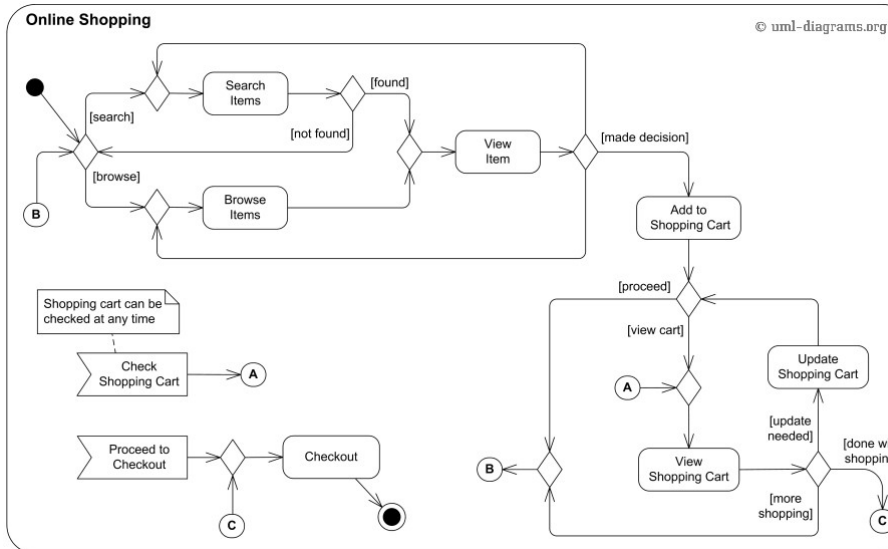


Set 2017

Arndt von Staa© LES/DI/PUC-Rio

3

Exemplo: diagrama de atividades



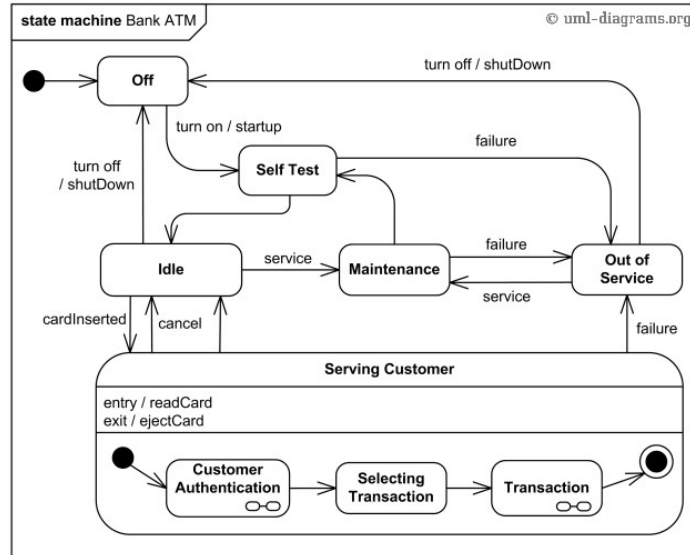
<http://www.uml-diagrams.org/online-shopping-uml-activity-diagram-example.html?context=activity-examples>

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

4

Exemplo: UML State chart



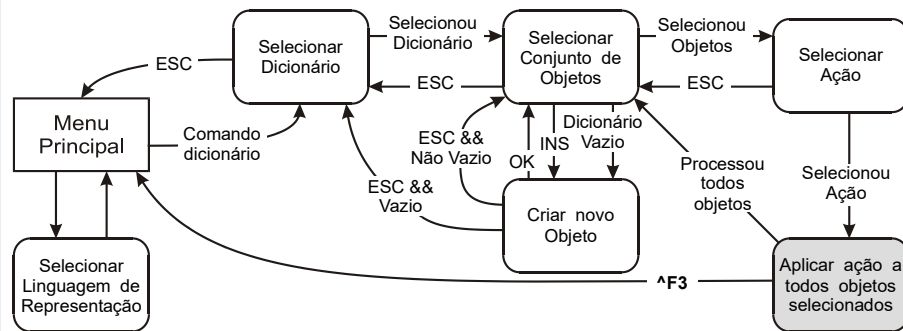
<http://www.uml-diagrams.org/bank-atm-uml-state-machine-diagram-example.html>

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

5

Exemplo: Interface humano computador



- Cada linguagem de representação possui $n > 0$ classes de objetos
- Cada classe de objetos pode aparecer em várias linguagens
- Para cada classe de objetos existe um dicionário
- A visualização de um dicionário exibe os nomes de todos os objetos da correspondente classe existentes na base de dados
- Cada linguagem de representação define um conjunto de ações que se pode realizar com os elementos de determinada classe

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

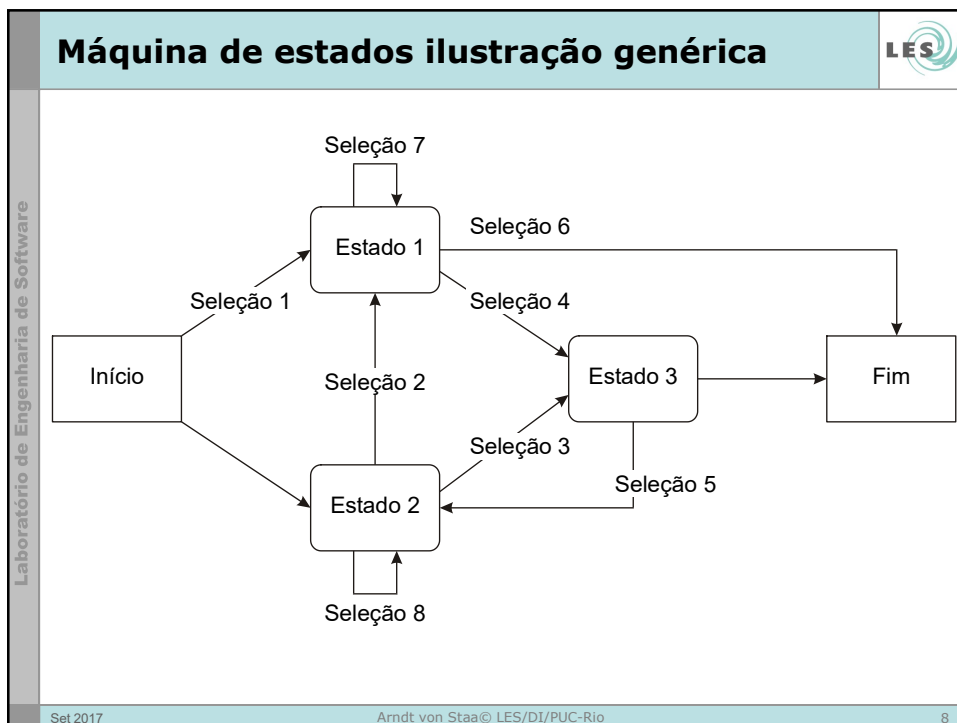
6

Casos de teste criados (incompleto)						LES
Ação	Entrada	Memória	Saída	Memória modificada	Risco de evolução	
Selecionar linguagem	Menu princ: Linguagem	Tabela linguagens	Exibe linguagem selec	Linguagem corrente	Tabela linguagens pode evoluir	
Ir dicionário	Menu princ: dicionário	Tabela dicionários	Exibe classes da linguagem			
Selecionar classe	Menu dic sel: classe	Tabela classes	Realça classe	Classe corrente	Tabela classes pode evoluir	
Ir classe	Menu dic: aciona	Objetos classe corr	Exibe parte dos objetos			
Selecionar objeto	Menu classe marca: obj	Lista objetos selec	Objeto ticado	Obj adicionado lista	O conjunto de objetos evolui durante o uso	
Ir objeto	Menu classe: aciona obj	Lista objetos selec	Menu ações	Obj acionado adicionado		
Selecionar ação	Acionar ação	Lista objetos	Janela ação	- Lista objetos - Ação a ser executada	Conjunto de ações depende da linguagem	

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

7



Set 2017

Arndt von Staa© LES/DI/PUC-Rio

8

Máquina de estados

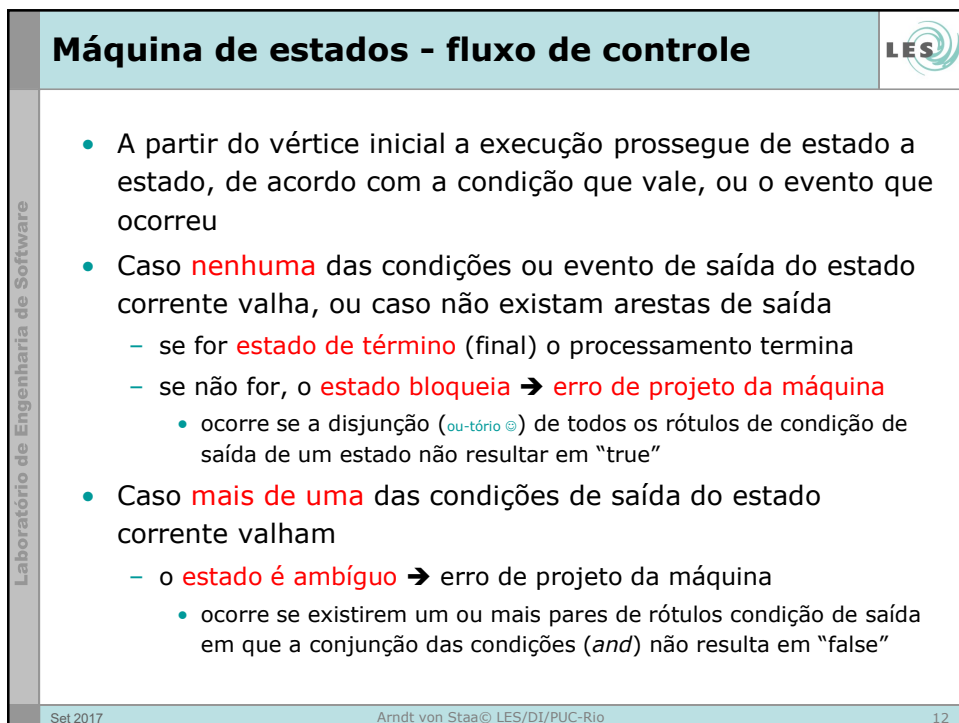
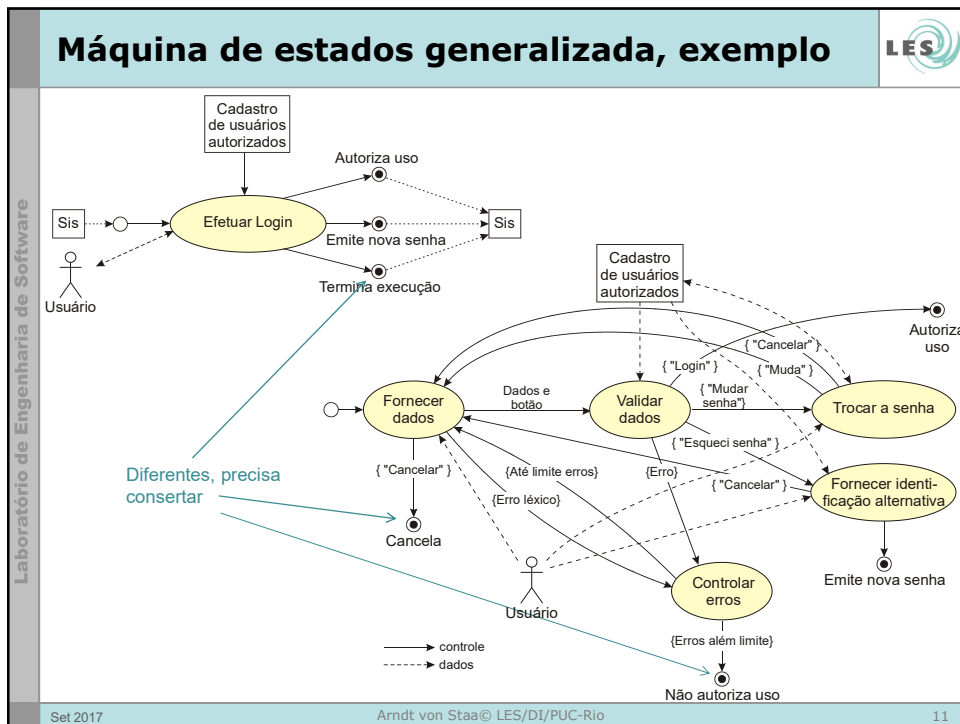


- Uma máquina de estados é um **grafo dirigido**
 - cada **vértice** é um estado
 - podem existir dois tipos de vértices especiais: **início** e **término**
 - nem sempre existem ⊗
 - cada **aresta** é uma transição
 - cada aresta possui zero ou um **rótulo condição** que designa a **condição** ou o **evento** que permite seguir por aquela aresta
 - uma aresta sem rótulo condição corresponde a um “else”
 - no máximo uma aresta de saída de um estado pode estar sem rótulo

Máquinas de estados generalizadas



- Em máquinas de estado generalizadas:
 - os estados podem conter **código executável**
 - as transições podem conter, além de condições ou eventos, **ações** a serem efetuadas caso a máquina transite por aquela aresta
 - os fragmentos de código contidos nos estados, e/ou nas arestas podem **fazer uso de memória**
 - ex. carrinho de compras sendo preenchido
 - ex. tabelas de símbolos para **filtros léxicos**
 - estados podem ser **decompostos** em máquinas de estado mais detalhadas.
 - nas máquinas resultado da decomposição
 - tudo que atinge o estado decomposto deve aparecer como origem
 - e tudo que sai do estado decomposto deve aparecer como término
 - pode-se criar uma **máquina nível zero** formada por um único estado e as origens e termos do processamento como um todo



Máquinas de estados - fluxo de eventos



- Em máquinas de eventos a progressão de estado para estado é dada por um evento
 - usualmente o **conjunto de eventos** que podem ocorrer em um determinado estado é **conhecido**
 - podem existir eventos genéricos, ex. cancela
 - a **progressão espera** até que ocorra um evento
 - caso a espera possa ser potencialmente ilimitada pode ser conveniente inserir um evento *time-out*
 - no exemplo do telefone
 - caso a máquina esteja no estado *livre*, o telefone espera indefinidamente pelo evento *tirar do gancho*
 - caso a máquina esteja no estado *digitando*, a espera é limitada por um *time-out*
 - máquinas de fluxo de eventos permitem projetar a sincronização de sistemas *multi-threading* (redes de petri, state charts)

Máquinas de estado: aspectos positivos



- Permitem
 - visualizar e verificar se as transições estão completas e corretas
 - exibir condições compostas complexas
 - gerar código diretamente a partir do diagrama
 - máquinas de estado generalizadas são "código" em nível de abstração mais alto
 - verificar as condições (assertivas) de entrada e saída
 - verificação de modelos
 - controle dinâmico da execução
- Grande parte das ações envolvendo máquinas de estado podem ser automatizadas
 - as anotações envolvendo assertivas e código podem ser criadas localmente sem necessitar conhecer o diagrama inteiro

- Set 2017

LES

Laboratório de Engenharia de Software

1. O componente limpa os campos e gera o captcha
2. O usuário digita sua identificação, senha e captcha
3. Quando o usuário selecionar a ação "Login" então
 - 3.1 O controle de acesso verifica sintaticamente os dados fornecidos
 - 3.2 O controle de acesso verifica se <usuario, senha> existe
 - 3.3 O controle de acesso retorna ao sistema **sis**, fornecendo a condição "autorizar uso" e os direitos de uso correspondentes a <usuário, senha>
- Fim quando
4. Quando o usuário selecionar a ação "Mudar senha" então
 - 4.1 O controle de acesso verifica sintaticamente os dados fornecidos
 - 4.2 O controle de acesso verifica se <usuario, senha> existe
 - 4.3 O controle de acesso **ativa** o caso de uso "Trocar a senha"
 - 4.4 Repete a partir de 1
- Fim quando
5. Quando o usuário selecionar a ação "Esqueci a senha" então
 - 5.1 O controle de acesso verifica sintaticamente os dados fornecidos
 - 5.2 O controle de acesso verifica se usuário existe
 - 5.3 O controle de acesso ativa o caso de uso "Fornecer identificação alternativa"
 - 5.4 Se retornar do caso de 5.3, repete a partir de 1
- Fim quando
6. Quando o usuário selecionar a ação "Cancelar" então
 - 6.1 O controle de acesso retorna ao sistema **sis**, fornecendo a condição "cancelar uso" e direitos de uso nulo
- Fim quando

Componente Login: especificação		LES
fluxos alternativos	<p>Evento 1/3.1, 4.1, 5.1 : O usuário digitou incorretamente a identificação ou o captcha</p> <p>E1.1 Se for a quarta ou mais vez que ocorreu um evento de erro então</p> <p>E1.1.1. O controle de acesso emite a mensagem "Acesso não autorizado"</p> <p>E1.1.2. O controle de acesso retorna ao sistema sis, fornecendo a condição "não autorizar uso" e direitos de uso nulo</p> <p>Fim se</p> <p>E1.2. O controle de acesso emite a mensagem "Dados incorretos"</p> <p>E1.3. O controle de acesso retorna ao passo 1</p> <p>Fim evento E1</p>	
	<p>Evento 2/3.2, 4.2 : O par <usuário, senha> não está definido</p> <p>E2.1 Se for a quarta ou mais vez que ocorreu um evento de erro então</p> <p>E2.1.1. O controle de acesso emite a mensagem "Acesso não autorizado"</p> <p>E2.1.2. O controle de acesso retorna ao sistema sis, fornecendo a condição "não autorizar uso" e direitos de uso nulo</p> <p>Fim se</p> <p>E2.2. O controle de acesso emite a mensagem "Usuário desconhecido"</p> <p>E2.3. O controle de acesso retorna ao passo 1</p> <p>Fim evento E2</p>	

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

17

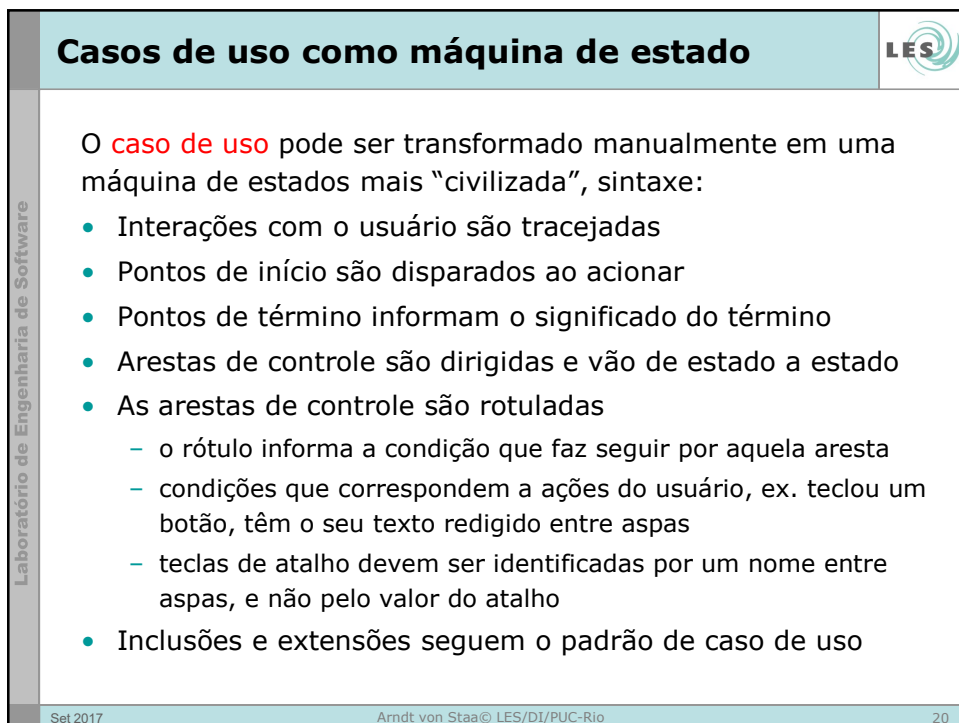
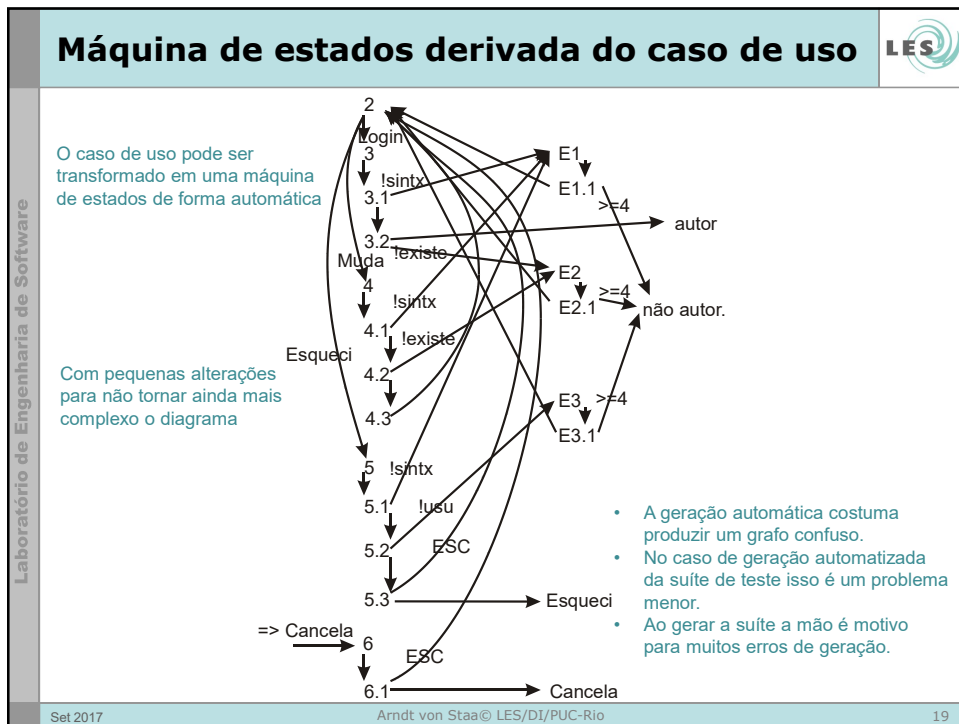
Componente Login: especificação		LES
fluxos alternativos	<p>Evento 3/5.2 : A identificação do usuário não existe no cadastro</p> <p>E3.1 Se for a quarta ou mais vez que ocorreu um evento de erro então</p> <p>E3.1.1. O controle de acesso emite a mensagem "Acesso não autorizado"</p> <p>E3.1.2. O controle de acesso retorna ao sistema sis, fornecendo a condição "não autorizar uso" e direitos de uso nulo</p> <p>Fim se</p> <p>E3.2. O controle de acesso emite a mensagem "Usuário desconhecido"</p> <p>E3.3. O controle de acesso retorna ao passo 1</p> <p>Fim evento E3</p>	
	<p>Evento E4: o usuário clica "Cancelar" em qualquer lugar</p> <p>E4.1 O sistema solicita confirmação do cancelamento</p> <p>E4.2 Se usuário confirma o cancelamento</p> <p>E4.2.1 O controle de acesso retorna fornecendo o conjunto "cancelar uso" ao sistema sis</p> <p>Fim se</p> <p>E4.3 O controle de acesso retorna ao passo 1</p> <p>Fim evento E4.</p>	

Veja Aula 04 Especificações, resumo

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

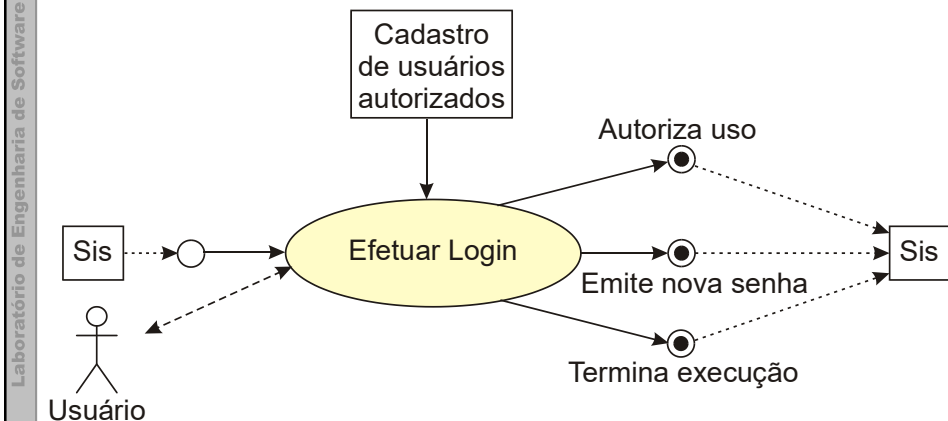
18



Máquina de estados criada diretamente



- Contexto do caso de uso → diagrama nível 0



Set 2017

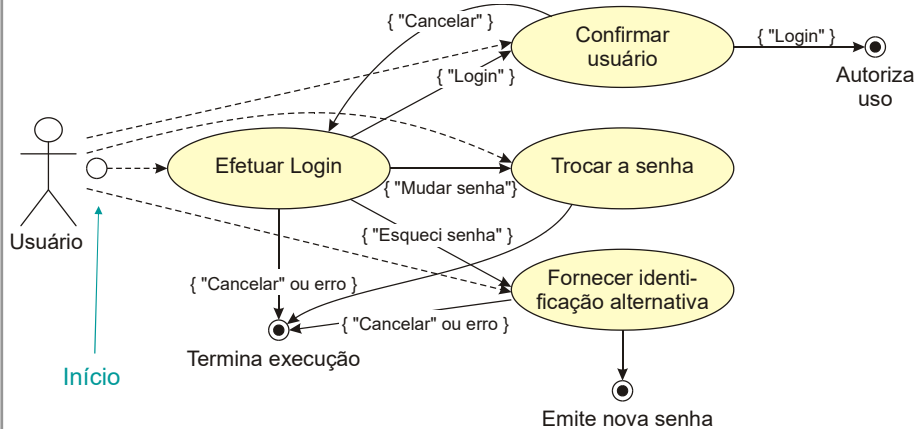
Arndt von Staa© LES/DI/PUC-Rio

21

Componente Login, máquina de estados



Versão inicial



- A máquina pode ser melhorada?
- Considere a separação da entrada de dados da verificação dos dados
 - verificação léxica pode ser realizada imediatamente
 - verificação envolvendo bases de dados, métodos etc. devem estar separados da entrada

Set 2017

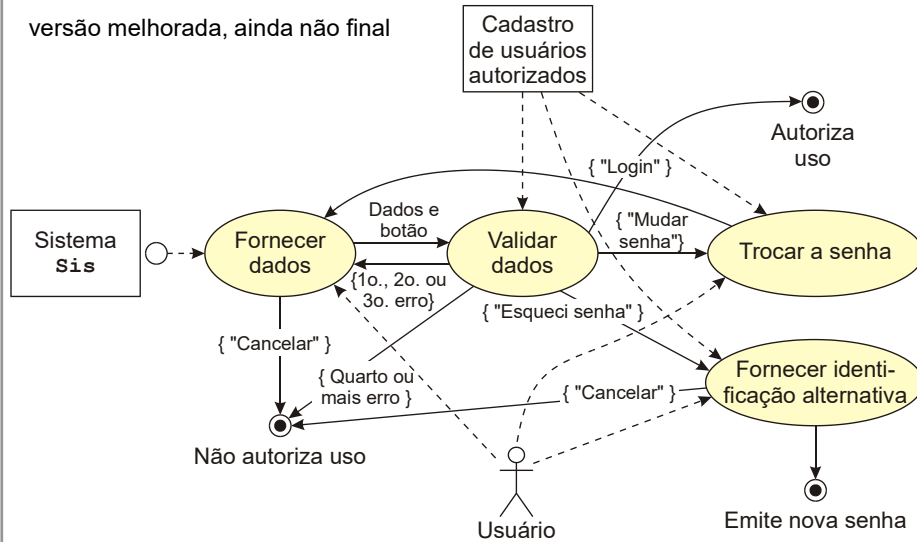
Arndt von Staa© LES/DI/PUC-Rio

22

Componente Login, máquina de estados



versão melhorada, ainda não final



- A máquina pode ser melhorada mais ainda?
- Considere fatorar o controle de número de erros.

Set 2017

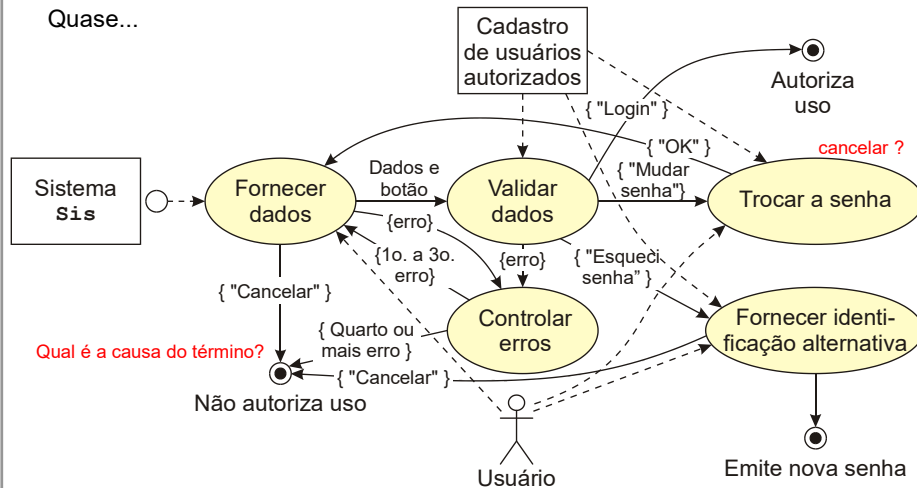
Arndt von Staa© LES/DI/PUC-Rio

23

Componente Login, máquina de estados



Quase...



- Ainda tem problemas a serem resolvidos?
- Considere a separação de "cancelar" e "não autorizar", cancelar troca de senha, ...
 - cada saída de um estado ou da máquina deve explicitar a causa da escolha

Set 2017

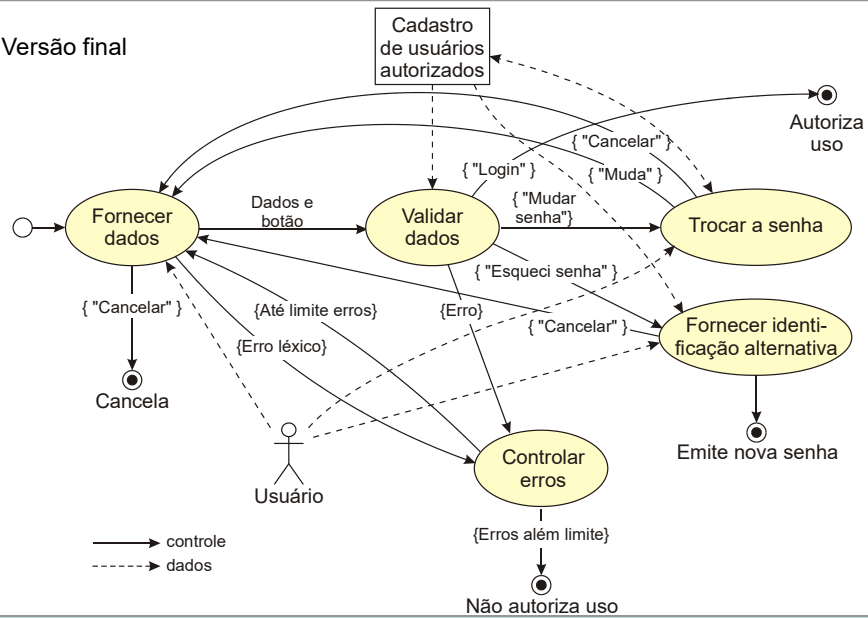
Arndt von Staa© LES/DI/PUC-Rio

24

Componente Login, máquina de estados



Versão final

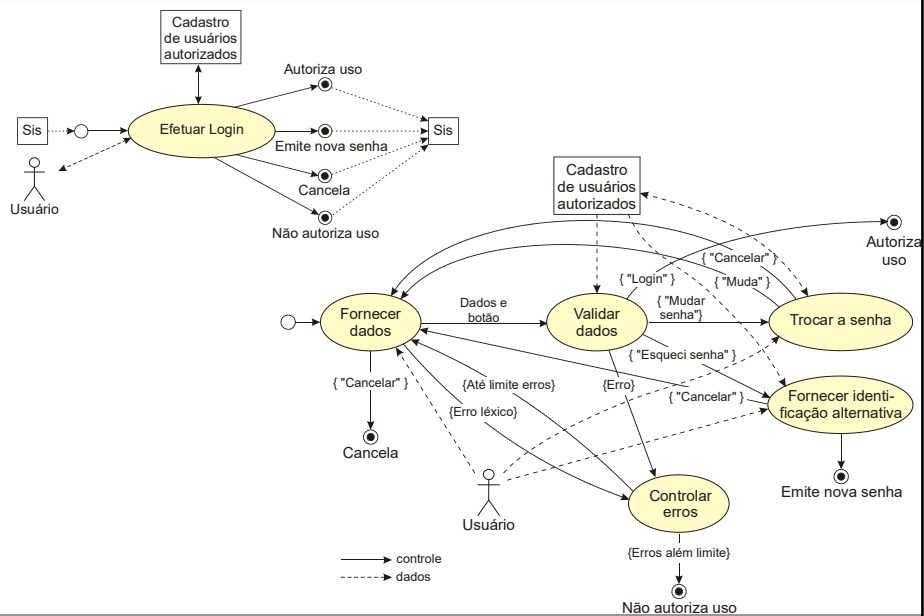


Set 2017

Arndt von Staa© LES/DI/PUC-Rio

25

Componente Login, máquina de estados



Set 2017

Arndt von Staa© LES/DI/PUC-Rio

26

Exemplo: Fornecer dados 1 / 4

Estado	Fornecer dados
Resumo	Recebe os dados do usuário e efetua a verificação independente de cadastro (<i>verificação léxica</i>)
Escopo	não se aplica a estados
Ator principal	não se aplica a estados
Interessados	não se aplica a estados
Invariante	O número de erros identificados na presente instância de uso é menor ou igual a três.
Pré condição	
Acionamento	<i>Obter dados</i> inicia quando <ul style="list-style-type: none">• ou o sistema sistema solicitar dados do usuário para autorizar o uso• ou o estado validar dados encontra dados ilegais e solicitar novos dados

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

27

Exemplo: Fornecer dados 2 / 4

The logo for LES (Laboratório de Engenharia de Software) is located in the top right corner. It consists of the letters "LES" in a bold, sans-serif font, followed by a circular graphic containing several concentric arcs.

Fluxo principal

1. O controle de acesso limpa os campos de entrada
2. O controle acesso gera o captcha exibido
3. O usuário digita sua identificação e senha, e o captcha
4. O usuário clica a ação a ser realizada
5. O controle de acesso valida a corretude léxica de idUsuario
6. O controle de acesso verifica o captcha digitado
7. O controle de acesso *ativa o caso de uso Validar Dados*


Laboratório de Engenharia de Software

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

28

Exemplo: Fornecer dados 3 / 4



Laboratório de Engenharia de Software


Fluxos alternativos	<p>Evento E1: o usuário clica "cancelar"</p> <p>E1.1 O controle de acesso fecha a janela de identificação de usuário</p> <p>E1.2 O controle de acesso fornece o controle "cancelar uso" ao sistema Sis</p> <p>Fim evento E1.</p> <p>Evento E2/6: o usuário digita caracteres de controle incorretos</p> <p>E2.1 O controle de acesso emite a mensagem "Caracteres de controle incorretos"</p> <p>E2.2 Ativa o estado "controlar erros"</p> <p>Fim evento E2.</p> <p>Evento E3/5: o usuário fornece identificação usuário lexicamente incorreta</p> <p>E3.1 O controle de acesso emite a mensagem "Usuário incorreto"</p> <p>E3.2 Ativa o estado "controlar erros"</p> <p>Fim evento E3.</p>
---------------------	--

E1. Melhor:
E1.1 O controle acesso prepara o retorno <Cancelar, direitos: vazio>
E1.2 Termina

E3 - Especificação léxica de usuário ver: regra de negócio idUsuario

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
29

Exemplo: Fornecer dados 4 / 4



Laboratório de Engenharia de Software

Pós condições	Dados e ação a executar fornecidos ao caso de uso <i>Validar dados</i>
Garantia mínima	<ul style="list-style-type: none"> N/A
Requisitos	<ul style="list-style-type: none"> -.-
Regras de negócio	<ul style="list-style-type: none"> idUsuario deve ter entre 5 e 30 caracteres idUsuario não deve conter letras diacríticas idUsuario não deve conter dígitos idUsuario não deve conter brancos idUsuario pode conter somente os caracteres especiais: '-' e '.'
Casos de uso correlatos	Validar dados Reconfirmar usuário corrente Trocar a senha Fornecer identificação alternativa

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
30

Laboratório de Engenharia de Software

Conversão "Fornecer dados" para tabela de decisão

		1	2	3	4	5	6	7	8	9	10
	Usuário correto	-	s	s	s	n	n	n	-	-	-
	Captcha correto	-	s	s	s	s	s	s	n	n	n
xorob	Tecla "Login"	n	s	n	n	s	n	n	s	n	n
xorob	"Mudar senha"	n	-	s	n	-	s	n	-	s	n
xorob	"Esqueci senha"	n	-	-	s	-	-	s	-	-	s
xorob	"Cancelar"	s	n	n	n	n	n	n	n	n	n
	Ativar "validar"		x	x	x						
	Ativar "erro"					x	x	x	x	x	x
	Erro "caracteres controle"								x	x	x
	Erro "léxico"					x	x	x			
	Termina "cancela"	x									

A tabela está correta?

xorob – exclusive or obrigatório → exatamente uma das condições deve ser *true*

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
31

Laboratório de Engenharia de Software

Conversão "Fornecer dados" para tabela de decisão

		1	2	3	4	5	6	7	S
	Usuário lex correto	-	s	s	s	n	s	s	
	Captcha correto	-	s	s	s	-	n	s	
xorob	"Login"	-	s	n	n	-	-	n	
xorob	"Mudar senha"	-	-	s	n	-	-	n	
xorob	"Esqueci senha"	-	-	-	s	-	-	n	
xorob	"Cancelar"	s	n	n	n	n	n	n	
	Ativar "validar"		x	x	x				
	Ativar "erro"					x	x		
	Erro "caracteres controle"						x		
	Erro "léxico"					x			
	Termina "cancela"	x							
	Impossível							x	
	Contagem	32	4	2	1	16	8	1	64

Não seria melhor tratar explicitamente cada botão?

A tabela está correta?

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
32

Exemplo: Validar dados 1 / 3	
Laboratório de Engenharia de Software	Estado
	Validar dados e ação
	Resumo
	Valida os dados e a ação solicitada com relação ao cadastro de usuários autorizados
	Invariante
Laboratório de Engenharia de Software	Pré condição
	Nome correto, senha qq, e ação selecionada é uma de { login, mudaSenha, novaSenha } resultado do estado fornecer dados
	Acionamento
	<i>Validar dados</i> inicia ao receber o controle de <i>Obter dados</i>
	Fluxo principal
	<ol style="list-style-type: none"> 1. O Controle de acesso busca os dados do idUsuario no cadastro 2. Se a ação solicitada for "Esqueci senha" <ul style="list-style-type: none"> Então 2.1 Ativa o estado <i>Fornecer identificação alternativa</i> FimSe 3. O Controle de acesso verifica se a senha fornecida corresponde a uma das registradas para este usuário 4. Se a ação solicitada for "Login" <ul style="list-style-type: none"> Então 4.1.1 O Controle de acesso prepara o retorno <autorizado, direitos: de < idUsuario, senha>> 4.1.2 Termina Senão 4.2 O Controle de acesso ativa o estado <i>Trocar senha</i> FimSe

Não deveria ser : se ação solicitada é "trocar senha" ? →


Exemplo: Validar dados 2 / 3	
Laboratório de Engenharia de Software	Fluxos alternativos
	E1. Evento: Identificação do usuário não existe no Cadastro E1.1. O Controle de acesso exibe a mensagem "Usuário desconhecido" E1.2. O Controle de acesso ativa o estado "Controlar erros" Fim evento
	E2. Evento: Senha não fornecida, ou lexicamente errada ou não corresponde a qualquer uma das senhas de idUsuario E2.1. O Controle de acesso exibe a mensagem "Usuário desconhecido" E2.2. O Controle de acesso ativa o estado "Controlar erros" Fim evento

Razões de segurança indicam que as duas mensagens devem ser alguma coisa similar a "Usuário desconhecido". Um possível agressor ficará na dúvida se o problema é do idUsuario ou da Senha.

Problema 1: como discernir o erro de dados observado durante os testes?

Problema 2: a condição do evento E2 contém uma expressão lógica composta. Deveria ser simples para que se possa criar tabelas de decisão com condições binárias.

Exemplo: Validar dados 3 / 3



Laboratório de Engenharia de Software

Pós condições	
Garantia mínima	
Requisitos	<ul style="list-style-type: none"> • Não deve ser possível discernir se o erro foi idUsuario incorreto ou se senha foi incorreta
Regras de negócio	
Casos de uso correlatos	Fornecer dados Reconfirmar usuário corrente Trocar a senha Fornecer identificação alternativa

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
35

Conversão “Validar dados” para tabela de decisão



Laboratório de Engenharia de Software

		1	2	3	4	5	6	7	8	9
	Usuário conhecido	s	s	s	s	s	n	n	n	-
	Senha corresponde	s	s	-	n	n	-	-	-	-
xorob	Tecla “Login”	s	n	n	s	n	s	n	n	n
xorob	“Mudar senha”	-	s	n	-	s	-	s	n	n
xorob	“Esqueci senha”	-	-	s	-	-	-	-	s	n
	Autorizar	x								
	Ativar “Controlar”				x	x	x	x	x	
	Ativar “Trocar”		x							
	Ativar “Esqueci”			x						
	Impossível									x
	Msg: “Usuário desconhecido”				x	x	x	x	x	

A tabela está correta?

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
36

Laboratório de Engenharia de Software

Nível 0 – Final Autoriza uso

LES

Estado	Final: Autoriza uso
Assertivas de saída	1. retornou < Autorizado, direitos de acesso de <idUsuario, Senha>>
Fluxo	1. Controle de acesso oblitera os registros decriptados do cadastro 2. Controle de acesso fecha a janela 3. Controle de acesso retorna < Autorizado, direitos: <idUsuario, Senha>>

O valor retornado pode ser um objeto da classe "Autorizacao". Exemplos de coisas que essa classe pode definir :

- atributo idUnicaUsuario – um inteiro identificador interno gerado ao cadastrar o par <idUsuario, senha>
- atributo condicao uma enumeração tpCondicao :: {AUTORIZA, CANCELA, NOVA_SENHA, NÃO_AUTORIZA}
- atributo lista de idDireito's – criptografada
- bool TemDireitos(char idDireito) :: retorna true sse a condição é AUTORIZA e a lista de idDireitos contém idDireito
- tpCondicao RevalidaUsuario() :: abre uma janela similar a login e que contém somente os botões Login e Cancelar. Retorna AUTORIZA sse os dados fornecidos correspondem à idUnicaUsuario

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
37

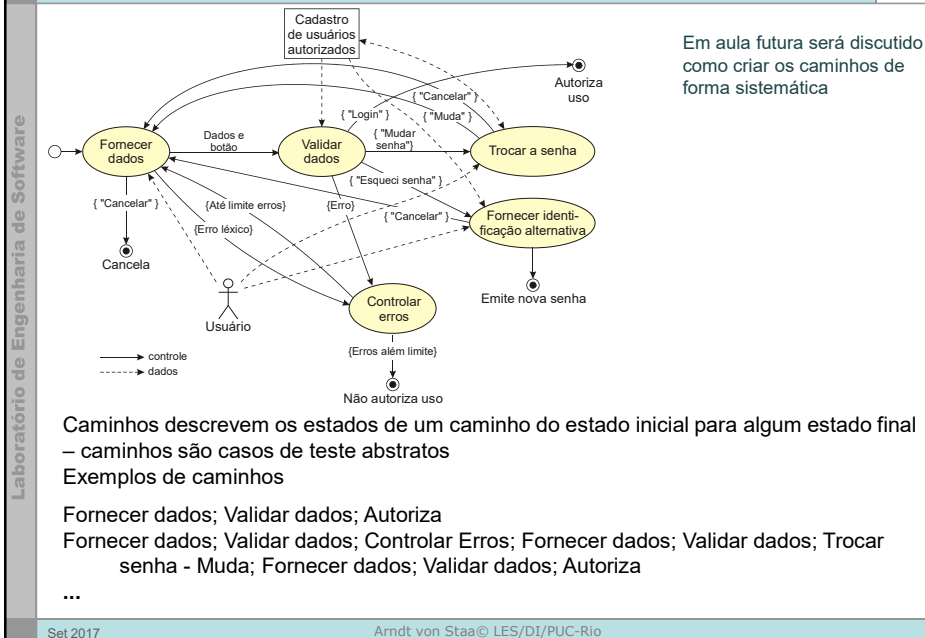
Laboratório de Engenharia de Software

LES

O resto fica para exercício 😊

Set 2017
Arndt von Staa© LES/DI/PUC-Rio
38

Máquina de estados, caminhos



Set 2017

Arndt von Staa© LES/DI/PUC-Rio

39

Máquina de estados, casos teste semânticos



- Dado o caminho, um caso de teste abstrato
 - Fornecer dados; Validar dados; Autoriza
- Converter para caso de teste semântico
 - precisa analisar o caminho,
 - usualmente faz-se de trás para diante examinando
 - a condição associada à aresta
 - as assertivas de saída (origem) e entrada (destino) dos estados
 - processamento no estado origem da aresta
- Resultado
 - Autoriza: par <Usuário , senha> existe, direitos par definidos
 - Validar dados -> Autoriza → usuário existe, par <usuário, senha> existe, botão = Login
 - Fornecer dados -> Validar dados → usuário correto, senha digitada, captcha correto, botão = Login

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

40

Qual é a ideia?



- Criar uma máquina de estados em que cada estado
 - ou é "simples"
 - ou é decomposto em outra máquina de estados
- Para cada estado
 - criar um caso de uso do estado
 - fluxo principal
 - zero ou mais fluxos alternativos
 - o término de um fluxo é uma transição de estado ou o término da máquina raiz
- O término de um fluxo corresponde
 - ou a uma mudança de estado
 - ou ao término da máquina raiz
 - ou ao término da máquina de decomposição
 - neste caso a transição (oráculo da tabela de decisão) é uma das transições de saída do estado "pai"

Qual é a ideia?



- Para cada estado cria-se uma tabela de decisão
 - a seleção de valores deve obedecer aos critérios de valoração
 - espera-se que o número de condições por estado seja pequeno, de modo que se controle o número de colunas das tabelas de decisão
- Cria-se a lista completa de caminhos da máquina de estados
 - repetições são resolvidas com base no arrasto e no limite de iterações caso exista
 - cada caminho na máquina de estados corresponde a uma cena
 - espera-se que cada máquina contenha poucos estados, de modo que o número de cenas seja pequeno
 - cada caminho determina as condições saída da tabela de decisão de cada um dos estados
- O teste cobre a combinação de condições em cada estado, mas não realiza as combinações entre estados

Laboratório de Engenharia de Software

Referências bibliográficas




- Holcombe, M.; Bogdanov, K.; Gheorghe, M.; "Functional Test Generation for Extreme Programming"; *Proceedings of the XP2001 Second International Conference on Extreme Programming and Flexible Processes in Software Engineering*; 2001; pages 109-113 Buscado em: 19/08/2004; URL: <http://www.dcs.shef.ac.uk/~wmlh/XPtest.pdf>

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

43

Laboratório de Engenharia de Software



FIM

Set 2017

Arndt von Staa© LES/DI/PUC-Rio

44