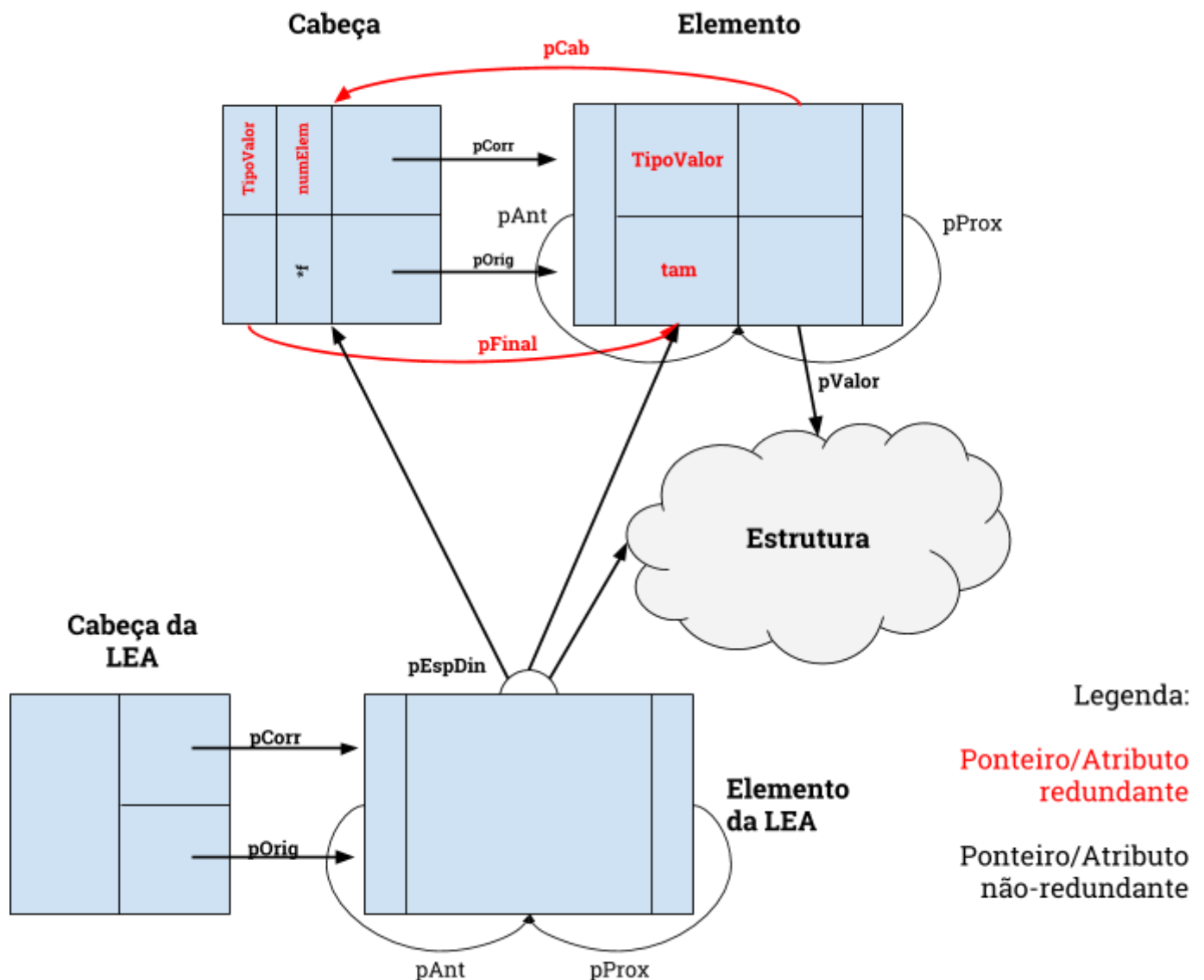


Modelo da Lista Duplamente Encadeada Auto-verificável



Legenda

Nome no modelo	Significado
TipoValor (Cabeça)	Tipo da estrutura apontada por todos os elementos da lista (char)
numElem	Número de elementos da lista (int)
*f	Ponteiro para função de exclusão da estrutura apontada por todos os elementos, que será acionada quando um elemento for excluído da lista
pFinal	Ponteiro para último elemento da lista
pOrig	Ponteiro para primeiro elemento da lista
pCorr	Ponteiro para elemento corrente da lista
pCab	Ponteiro para cabeça da lista a qual o elemento pertence
TipoValor (Elemento)	Tipo da estrutura apontada pelo elemento (char)
tam	Tamanho da estrutura apontada pelo elemento (size_t)
pValor	Ponteiro para estrutura genérica
pProx	Ponteiro para elemento sucessor
pAnt	Ponteiro para elemento antecessor
pEspDin	Ponteiro para todo espaço dinamicamente alocado que não na LEA

Assertivas Estruturais

01. Para um **elemento** $pElem$...

- a. $pElem \rightarrow pCab \neq NULL$
- b. $pElem \rightarrow pValor = NULL \Leftrightarrow pElem \rightarrow tam = 0$
- c. $pElem \rightarrow pValor \neq NULL \Rightarrow pElem \rightarrow tam = CED_ObterTamanhoValor(pElem \rightarrow pValor)$
- d. se $pElem \rightarrow pAnt = NULL$ (for o primeiro da lista) ...
 - i. $pElem \rightarrow pCab \rightarrow pOrig = pElem$
- e. senão...
 - i. $pElem \rightarrow pAnt \rightarrow pProx = pElem$
- f. se $pElem \rightarrow pProx = NULL$ (for o último da lista) ...
 - i. $pElem \rightarrow pCab \rightarrow pFinal = pElem$
- g. senão...
 - i. $pElem \rightarrow pProx \rightarrow pAnt = pElem$

OBS: $CED_ObterTamanhoValor$ (módulo CESPDI) indica o tamanho do espaço alocado.

02. Para uma **cabeça** $pCab$...

Os ponteiros $O = \{pOrig, pOrig \rightarrow pProx, pOrig \rightarrow pProx \rightarrow pProx, \dots\}$ até o primeiro ponteiro desta sequência que tenha $pProx = NULL$, apontam para elementos da lista apontada por $pCab$.

Os ponteiros $F = \{pFinal, pFinal \rightarrow pAnt, pFinal \rightarrow pAnt \rightarrow pAnt, \dots\}$ até o primeiro ponteiro desta sequência que tenha $pAnt = NULL$, apontam para elementos da lista apontada por $pCab$.

- a. $O = F$

$$numElem = |O| = |F|$$

$$E_{pCab} = \{e_1, e_2, e_3, \dots, e_{numElem}\} := O$$

e_i é sucessor de e_{i-1} e antecessor de e_{i+1} , caso existam.

$$pOrig = e_1, pFinal = e_{numElem}$$

- b. $\forall x (x \in E_{pCab} \Leftrightarrow x \rightarrow pCab = pCab)$
- c. $\forall x (x \in E_{pCab} \Rightarrow x \rightarrow TipoValor = pCab \rightarrow TipoValor)$
- d. $numElem = 0 \Leftrightarrow pCorr = pOrig = pFinal = NULL$
- e. $numElem = 1 \Leftrightarrow pCorr = pOrig = pFinal = e_1 \neq NULL$
- f. se $numElem > 1$...
 - i. $pOrig \neq pFinal$
 - ii. $pOrig \neq NULL$
 - iii. $pFinal \neq NULL$

iv. $pCorr \in E_{pCab}$

Comandos do Módulo de Teste

Comandos inéditos

Comando	Função
=exibirespacos	Exibe todos os espaços na LEA. Usado meramente para verificar as características dos espaços vazados (como tipo, arquivo e linha em que foram alocados, e tamanho), de forma a ajudar a diagnosticar e ajudar a solucionar erros no código da instrumentação.
=deturpar <idxLista> <modoDeturpacao>	Deturpa a lista de índice idxLista segundo o modo de deturpação indicado pelo identificador modoDeturpacao (listados neste mesmo documento). Torna a estrutura instável e o programa suscetível a “voar” caso chame imprudentemente funções que operem sobre ponteiros “loucos” ou nulos.
=liberaresplista <qtFalhasEsp>	Libera todos os espaços dinamicamente alocados e administrados pela LEA que estão marcados como inativos e com tipos de espaços de dados referentes ao módulo LISTA. Compara a quantidade de espaços liberados com qtFalhasEsp .
=tornarinativos	Torna inativos todos os espaços dinamicamente alocados e administrados pela LEA. Junto com o comando liberarinativos, é capaz de corrigir vazamento de memória causados por deturpações no módulo lista, preparando o programa para futuros casos de testes, sem precisar particionar em múltiplos scripts.
=verificar <idxLista> <qtFalhasEsp>	Verifica a estrutura da lista de índice idxLista do vetor de listas, segundo as assertivas estruturais do modelo de lista auto-verificável listadas neste mesmo documento. Compara o número de falhas acusado pela função LIS_VerificarLista com qtFalhasEsp fornecido.

Comandos importados do módulo INTRCNT

Comando	Descrição
=contagemcontador <número esperado>	---
=contagemtotal <somatório das contagens esperado>	---
=exibircontagem	---
=gravarcontadores <nome arquivo contadores acumulado>	grava o arquivo de contadores acumulado registrado. A gravação ocorre também ao terminar contadores.
=inicializarcontadores <nome arquivo contadores acumulado>	O nome do arquivo pode ser nulo. Se o nome do arquivo for ".", será também tratado como nulo.
=iniciarcontagem	Ativa a contagem. Este comando precisa ser dado. =inicializarcontadores não ativa a contagem, somente ativa o módulo.
=numcontadores <número esperado>	---
=lercontadores <nome arquivo contadores>	lê o arquivo e atribui os valores iniciais somente se o contador tiver sido lido de um arquivo acumulado.
=pararcontagem	Desativa a contagem.
=registraracumulador <nome arquivo contadores acumulado>	Substitui o nome registrado. O nome do arquivo pode ser nulo. Se o nome do arquivo for ".", será também tratado como nulo.
=terminarcontadores <número contadores iguais a zero esperado>	Grava o arquivo de contadores acumulados, caso esteja definido (nome não vazio).
=verificarcontagens <número contadores zero iguais a zero esperado>	---
=zerarcontador <string nome do contador>	---
=zerartodoscontadores	---

Modos de Deturpação

Para testar o modelo de estrutura auto-verificável de Lista Duplamente Encadeada, foram criados os seguintes modos de deturpação:

ID	Nome do modo	Função
1	LIS_ModoDeturpacaoEliminaCorr	Elimina o elemento corrente da lista
2	LIS_ModoDeturpacaoProxNULL	Atribui NULL ao ponteiro para o próximo nó
3	LIS_ModoDeturpacaoAntNULL	Atribui NULL ao ponteiro para o nó anterior
4	LIS_ModoDeturpacaoProxLixo	Atribui lixo ao ponteiro para o próximo nó
5	LIS_ModoDeturpacaoAntLixo	Atribui lixo ao ponteiro para o nó anterior
6	LIS_ModoDeturpacaoConteudoNULL	Atribui NULL ao ponteiro para o conteúdo do nó
7	LIS_ModoDeturpacaoAlteraTipoNo	Altera o tipo de estrutura apontado no nó
8	LIS_ModoDeturpacaoDesencadeiaNo	Desencadeia nó sem liberá-lo com free
9	LIS_ModoDeturpacaoPontCorrNULL	Atribui NULL ao ponteiro corrente
10	LIS_ModoDeturpacaoPontOrigNULL	Atribui NULL ao ponteiro de origem
11	LIS_ModoDeturpacaoPontFimNULL	Atribui NULL ao ponteiro de final
12	LIS_ModoDeturpacaoEliminaLista	Elimina lista
13	LIS_ModoDeturpacaoCorrLixo	Atribui lixo ao ponteiro para nó corrente
14	LIS_ModoDeturpacaoOrigLixo	Atribui lixo ao ponteiro para nó origem
15	LIS_ModoDeturpacaoFinalLixo	Atribui lixo ao ponteiro para nó final

16	LIS_ModoDeturpacaoNumElemNegativo	Atribui valor negativo ao contador de elementos
17	LIS_ModoDeturpacaoCorrElemNovo	Fazer ponteiro corrente apontar para novo elemento
18	LIS_ModoDeturpacaoCorrTipoErrado	Fazer ponteiro corrente apontar para lista
19	LIS_ModoDeturpacaoOrigTipoErrado	Fazer ponteiro origem apontar para lista
20	LIS_ModoDeturpacaoFinalTipoErrado	Fazer ponteiro final apontar para lista
21	LIS_ModoDeturpacaoOrigElemNovo	Fazer ponteiro origem apontar para novo elemento
22	LIS_ModoDeturpacaoElemCorrCabNULL	Fazer pCab do elemento corrente NULL
23	LIS_ModoDeturpacaoDuplicarElem	Duplica elemento de modo que se possa continuar a percorrer em ambos os sentidos por caminhos diferentes
24	LIS_ModoDeturpacaoTamNegativo	Faz tam = -1
25	LIS_ModoDeturpacaoTamUm	Faz tam = 1
26	LIS_ModoDeturpacaoConteudoLixo	Faz pValor apontar para lixo
27	LIS_ModoDeturpacaoConteudoApontaElem	Faz pValor apontar para elemento
28	LIS_ModoDeturpacaoAntApontaCab	Faz ponteiro para elemento anterior do elemento corrente apontar para cabeça
29	LIS_ModoDeturpacaoProxApontaCab	Faz ponteiro para elemento posterior do elemento corrente apontar para cabeça