TDA Garph

CLASS Graph

IMPORTS

EFFECTIVE

TYPES Graph

BASIC CONSTRUCTORS Graph

OPERATIONS

Graph: natural → Graph;

addEdge: Graph\*Natural\*Natural → Graph;

AXIOMS

…

END\_CLASS.

TDA Punto

Explicacion de porque no lo implementamos asi

CLASS Punto

IMPORTS Type

EFFECTIVE

TYPES Punto

BASIC CONSTRUCTORS

OPERATIONS

Punto : Real\*Real\*natural\*Type\*Booleano\*Natural\*Natural→ Punto;

getId: Punto → Natural;

getX: Punto → Real;

getY: Punto → Real;

getTipo: Punto → Type;

getIzq: Punto → Booleano;

getColor: Punto → Natural;

getGuardia: Punto → Natural

setX: Punto\*Real → Punto;

setY: Punto\*Real → Punto;

setTipo: Punto\*Type → Punto;

setId: Punto\*Natural → Punto;

setIzq: Punto\*Booleano → Punto;

setColor: Punto\*Natural → Punto;

setGuardia: Punto\*Natural → Punto;

AXIOMS

…

END\_CLASS.

TDA Lado

Explicacion de porque no lo implementamos asi y del helper

CLASS Lado

IMPORTS Punto, Type

EFFECTIVE

TYPES Lado

BASIC CONSTRUCTORS Lado

OPERATIONS

Lado: Punto\*Punto\*Type\*Natural\*Real → Lado;

getExtremo1: Lado → Punto;

getExtremo2: Lado → Punto;

getId: Lado → Natural;

getClave: Lado → Real

invertirSegmento: Lado → Lado;

setExtremo1: Lado\*Punto → Lado;

setExtremo2: Lado\*Punto → Lado;

setTipo: Lado\*Type → Lado;

setId: Lado\* Natural → Lado;

setClave: Lado\*Real → Lado;

incrementarvalorClave: Lado\*Real → Lado;

AXIOMS

…

END\_CLASS.

TDA BDMFile

Este TDA especifica un tipo de dato archivo que sirve leer(extraer), operar y guardar polígonos en un archivo de extensión bdm creado por nosotros. Ya que en el TDA Polygon representaría al constructor de esta clase Polygon donde mediante el nombre del archivo que contiene el polígono nos devuelve el polígono. Ademas aclaramos que tanto métodos como leerBDM y guardarBDM son métodos propios de archivos que se pueden reflejar en una esoecificacion algebraica como es NEREUS. Igualmente, brindamos una definición de este tipo de dato a modo ilustrativo a continuación:

CLASS BDMFile

IMPORTS

EFFECTIVE

TYPES BDMFile

BASIC CONSTRUCTORS BDMFile

OPERATIONS

BDMFile: Cadena → BDMFile;

leerlinea: BDMFile → Booleano;

contarCampos: BDMFile\*Caracter → Natural;

encontrarCampo: BDMFile → Cadena;

existe: BDMFile\*Real\*Real → Booleano;

AXIOMS

…

END\_CLASS.

TDA Polygon

CLASS Polygon

IMPORTS BDMFile, Lista[Natural], Vector[Natural], Punto, Lista[Vector[Natural]], Lado

EFFECTIVE

TYPES Polygon

BASIC CONSTRUCTORS Polygon

OPERATIONS

Polygon: BDMFile → Polygon;

particion2Monotono: Polygon → Polygon;

buscarMonotonos: Polygon → Polygon;

triangulacion Polygon → Polygon;

vecinos: Polygon\* Lista[Natural] \* Lista[Natural] → Booleano;

colorear: Polygon → Polygon;

encontrarColorMenosRepetido: Polygon →Natural;

colocarGuardias: Polygon\*Natural → Polygon;

colorearTriangulo:Polygon\*Vector[Natural] → Polygon;

centro: Polygon\*Vector[Natural] →Punto;

triangulos: Polygon → Lista[Vector[Natural]];

inicializar: Polygon → Polygon;

agregarDiagonal: Polygon\*Natural\*Natural → Polygon;

seleccionarSiguienteLado:Polygon\*Lado → Natural;

triangularMonotono: Polygon → Polygon;

saveAsShowme();

anterior(unsigned int i);

siguiente(unsigned int i);

manejarStartVertex(unsigned int);

manejarEndVertex(unsigned int);

manejarSplitVertex(unsigned int);

manejarMergeVertex(unsigned int);

manejarRegularVertexUp(unsigned int);

manejarRegularVertexDown(unsigned int);

cosenoAngulo(double \*A, double \*B, double \*C);

AXIOMS

…

END\_CLASS.

TDA Arbin

CLASS Arbin[elem]

IMPORTS

EFFECTIVE

TYPES Arbin

BASIC CONSTRUCTORS Arbin

OPERATIONS

Arbin: → Arbin;

Vaciar: Arbin → Arbin;

estaVacio: Arbin → Booleano;

Size: Arbin → Natural;

Root: Arbin(a) → elem

pre: not estaVacio(a);

hallar: Arbin(a)\*elem → Arbin

pre: not estaVacio(a);

hallarMin: Arbin(a) → Arbin

pre: not estaVacio(a);

hallarMax: Arbin(a) → Arbin;

pre: not estaVacio(a);

hallarMaxMenorQue: Arbin(a)\*elem → Arbin

pre: not estaVacio(a);

Insert: Arbin\*elem → Arbin;

Delete: Arbin(a)\*elem(e) → Arbin

Pre: hallar(a,e);

borrarMin: Arbin(a) → Arbin

pre: not estaVacio(a);

borrarMan: Arbin(a) → Arbin

pre: not estaVacio(a);

altura: Arbin → Natural;

altura Arbin\*arbin → Natural;

AXIOMS

…

END\_CLASS.

Para concluir con la sección de especificación algebraica de tipos de datos abstractos, cabe incluir una anotación donde expondremos que como estructura interna del árbol binario se utilizo nodo que posee un puntero a hijo izquierdo uno a hijo derecho, con dato, un atributo booleano visitado y un valorClave.